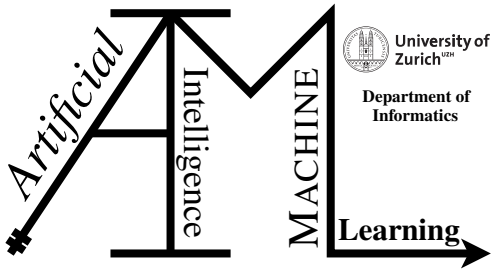
Portability of Targeted Adversarial Attacks

Master Thesis

Rohit Kaushik
17-747-669

Submitted on
April 25 2022

Thesis Supervisor
Prof. Dr. Manuel Gunther



Master Thesis

Author: Rohit Kaushik, rohit.kaushik@uzh.ch

Project period: 25.10.2021 - 25.04.2022

Artificial Intelligence and Machine Learning Group
Department of Informatics, University of Zurich

Acknowledgements

Throughout the writing of this thesis, I have received a great deal of support and guidance. I would like to thank Dr. Manuel Günther for giving me the opportunity to work on this project. His expertise was invaluable and his insightful feedback enabled me to complete the experiments successfully. Without his constant support and guidance this work could not have been possible. I would also like to acknowledge my friends and family who kept me motivated through the project. A special thanks to my friend for proof-reading the paper and pointing out mistakes in my grammar.

Abstract

In recent years, image classification with neural networks have shown promising results and hence garnered interest of researchers. These networks have been able to perform classification with near human efficiency. However it was later discovered that it is easy to fool neural networks and they are not robust to small non random perturbations. Addition of small non random pertubations which is imperceptible to human eye can cause neural network to misclassify an image which was previously correctly classified. Such a perturbed image is called adversarial image and an approach to generate such image is called adversarial attack. Morever these perturbations are portable to other neural network architecture which means an adversarial image that can fool one network can also be used to fool other networks. This poses a huge security risk for system that rely on image classification such medical analysis and autonomous vehicle. An attacker can modify the input image which could lead to failure of the system. This suggests that there is a need to build systems which are robust against such small perturbations. Portability is another import aspect, since if an adversarial example is only portable to few of the neural networks, an ensemble of network can be used to prevent against such attacks. In this work we propose experiments to evaluate portability of adversarial images across popular neural networks. We perform an in depth portability study by looking into how far the predictions are from true class. Since there has been very few research for targeted attack and it's portability we include them in our study. We generate adversarial images on 13 neural network model with 3 adversarial attack and different value of perturbation constant ϵ on the subset of ImageNet dataset (ILSVRC 2012). A distance metric was proposed to calculate the distance between any two classes of the dataset. The experiments shows that with untargeted attack and a larger ϵ the portability is high between networks of similar architecture but it drops significantly as we reduce the ϵ. With targeted attack, 3 different approach to choosing target was proposed but we observe low portability with all the targeted attacks and different values of target and perturbation constant.

Zusammenfassung

In den letzten Jahren hat die Bildklassifizierung mit neuronalen Netzwerke vielversprechende Ergebnisse gezeigt und damit das Interesse der Forscher geweckt. Diese Netze waren in der Lage, die Klassifizierung mit nahezu menschlicher Effizienz durchzuführen. Später wurde jedoch festgestellt, dass neuronale Netze leicht zu täuschen sind und dass sie gegenüber kleinen, nicht zufälligen Störungen nicht robust sind. Das Hinzufügen kleiner, nicht zufälliger Störungen, die für das menschliche Auge nicht wahrnehmbar sind, kann dazu führen, dass ein neuronales Netz ein Bild falsch klassifiziert, das zuvor korrekt klassifiziert wurde. Ein solches gestörtes Bild wird als Störungsbild bezeichnet, und ein Ansatz zur Erzeugung eines solchen Bildes wird als Störungsangriff bezeichnet. Darüber hinaus sind diese Störungen auf andere neuronale Netzarchitekturen übertragbar, d. h. ein Störbild, das ein Netz täuschen kann, kann auch zur Täuschung anderer Netze verwendet werden. Dies stellt ein großes Sicherheitsrisiko für Systeme dar, die auf Bildklassifizierung angewiesen sind, wie z. B. medizinische Analysen und autonome Fahrzeuge. Ein Angreifer kann das Eingabebild verändern, was zu einem Ausfall des Systems führen könnte. Dies legt nahe, dass es notwendig ist, Systeme zu entwickeln, die gegen solche kleinen Störungen robust sind. Ein weiterer wichtiger Aspekt ist die Übertragbarkeit, denn wenn ein negatives Beispiel nur auf wenige neuronale Netze übertragbar ist, kann ein Ensemble von Netzen verwendet werden, um solche Angriffe zu verhindern. In dieser Arbeit schlagen wir Experimente vor, um die Übertragbarkeit von Schadbildern auf gängige neuronale Netze zu bewerten. Wir führen eine eingehende Portabilitätsstudie durch, indem wir untersuchen, wie weit die Vorhersagen von der wahren Klasse entfernt sind. Da es nur sehr wenige Untersuchungen zu gezielten Angriffen und deren Übertragbarkeit gibt, beziehen wir diese in unsere Studie ein. Wir generieren gegnerische Bilder auf 13 neuronalen Netzwerkmodellen mit 3 gegnerischen Angriffen und verschiedenen Werten der Störungskonstante ϵ auf einer Teilmenge des ImageNet-Datensatzes (ILSVRC 2012). Es wurde eine Distanzmetrik vorgeschlagen, um die Distanz zwischen zwei beliebigen Klassen des Datensatzes zu berechnen. Die Experimente zeigen, dass bei einem ungezielten Angriff und einem größeren ϵ die Übertragbarkeit zwischen Netzwerken mit ähnlicher Architektur hoch ist, aber sie sinkt deutlich, wenn wir den ϵ reduzieren. Bei gezielten Angriffen wurden 3 verschiedene Ansätze zur Auswahl des Ziels vorgeschlagen, aber wir beobachten eine geringe Übertragbarkeit bei allen gezielten Angriffen und unterschiedlichen Werten für Ziel und Störungskonstante.

Contents

Chapter 1

Introduction

Over the past decade, neural network have been gaining popularity for the tasks of predictions especially image classification. A lot of research has focused on developing networks with better test accuracy. Large datasets such as Imagnet Deng et al. (2009) have been proposed to further aid the process of training better neural network. Due to increasing research, state of the art neural network models have managed to achieve classification accuracies comparable to humans.

Although it has been observed that neural networks are not robust against small perturbation in the input image. Szegedy et al. (2014b) was the first to demonstrate this property. These perturbations are small and non random and often imperceptible to human eye. An adversarial example is an Image, which was originally classified correctly by the network, but is misclassified after addition of small non-random perturbation. In the paper Goodfellow et al. (2014) suggests that the adversarial examples are also transferable across neural networks, which means that a perturbation produced on one network can work as an adversarial example on another network.

In general, an algorithm to generate adversarial example for a model f can be formulated as an optimization problem:

$$\underset{\|\hat{x}-x\| \leq \epsilon}{argmin}\ l(\hat{y}, f(\hat{x})) \tag{1.1}$$

where $\hat{x}$ denotes the adversarial example, $\hat{y}$ denotes a label that is different from the true label y and ϵ denotes the perturbation bound.

An algorithm used to create such adversarial examples is called adversarial attacks. Adversarial attacks can be classified on different measure. Based on the selection of output, it is classified as untargeted or targeted attack. The goal of targeted attack is to misclassify an image I of class c as class t, where t is the fixed target different from c whereas for untargeted attack the goal is to misclassify an image I of class c as class $c\prime$ where $c\prime$ is not same as c. Based on the model knowledge adversarial attacks can be classified as white box attack and black box attack. A white box attack is an adversarial attack when the architecture of the neural network being attacked and the learned parameters such as weight is know whereas in a black box attack we donot know the internal architecture or learned parameters of the neural network.

Any adversarial attack follows the given steps,

- change the input image I and add small pertubation to get our perturbed image $I\prime$ which is given by $I + \epsilon$
- the objective function is given by $l(f(I), c\prime)$ where $c\prime$ is different from true output label

In the figure below, 3.9 we can see perturbation added to an image of panda, the resulting image still remains visibly similar but the neural network predicts a different class with a high confidence.

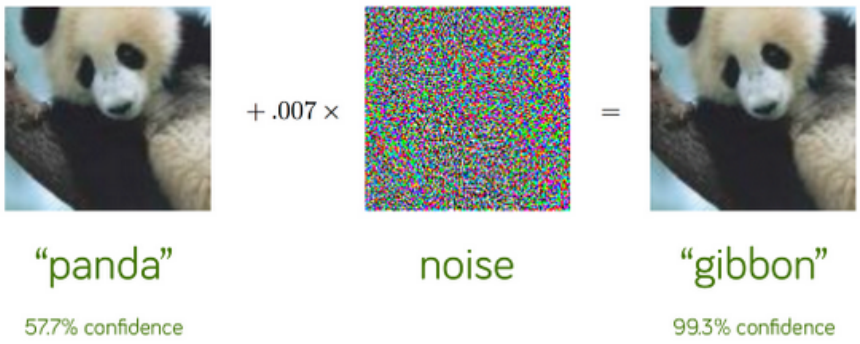


Figure 1.1: an example of adversarial attack

The intriguing property of neural networks discovered by Szegedy et al. (2014b) points to certain problem in current image classification task by neural network. The first is that small imperceptible perturbation can be confidently classified as class different from true class. Second, that presence of adversarial examples mean that with current training strategy of neural network, it doesnot generalize well enough or they are not robust to non-random perturbation in input image. In a real word image, input dataset can have noise coming from several factors. One of them can be resizing, cropping of images. The vulnerbaility of neural network against such noise or small perturbation can pose as a potential security threat for application of neural network models, especially in high risk areas such as medical treatment, or guiding a autonomous vehicle. Hence there is a need to study the robustness and develop approaches that make neural network more robust.

The discovery of adversarial property in neural network by Szegedy et al. (2014b) has intriguied researchers. Studies since then have focussed on developing attacks and defense strategies. Some of the work in this area has been listed down in the Chapter 2 on Related Works. Other work has focussed on developing approaches to make neural network robust against small perturbation or noise. The general idea behind such approaches to improve robustness is to further train the neural network with the generated adversarial example so that neural network learns to also model the noise within data. Other approach is to perform adversarial training, where a generator and discriminator model is used. Bai et al. (2021) present a survey on various adverarial training approach and its effect on robustness. But there is a trade off between making networks more robust. Yang et al. (2020) studies the effect of accuracy and robustness and how they are related. It was seen that with improve in robustness the accuracy of the neural network decreases.

Another aspect of the adversarial property is the transferability of adversarial examples. It was suggested by Goodfellow et al. (2014) that adversarial examples generated on one neural network are transferable to other neural network. Some previous research has studied the transferability of adversarial example but there are only limited work. Most of such research have only focussed on untargeted attacks. The transferability or portability of adversarial examples suggests even a bigger problem. It implies that an adversarial example can be created on a smaller or faster neural network and it will also work as an adversarial example against more dense or accurate neural network. This makes attacking deep neural network even easier.

In this work, we want to study the transferability of adversarial attacks for both targeted and

untargeted attacks. We perform an in depth study of transferability for different configurations of perturbations or ϵ. We select three different values of epsilon $0.01, 0.1\&0.3$ and three prominent adversarial attack apporach $PGD, FGSM, FGV$. Our study of portability is performed on 13 different state of the art neural network and we create our adversarial examples on the ImageNet dataset Deng et al. (2009) since this is the most popular image dataset for vision tasks.

To understand these problems described above, we formulate following research questions,

- RQ1. Are adversarial examples created using untargeted attacks portable to other neural networks?
- RQ2. How far are predictions on adversarial examples created using untargeted attacks from the ground truth?
- RQ3. Are the adversarial examples generated using a targeted attack portable on different neural networks?
- RQ4. Whether portability between networks which have similar topology is more compare to other networks?

Corresponding to each of the research questions, we perform an experiment and define metrics to quantitavely explain the results. We define a metric for calculating the distance of prediction on adversarial example from ground truth in case of untargeted attack and chosen target in case of targeted attack. The metric makes use of the ImageNet heirarchy. Since the generation of targeted adversarial example and its portability can be affected by the chosen target, we use three different ways to select target, closest to true label, farthest to true label and median class.

In the chapter Approach 3, we first discuss the neural networks and adversarial attacks we selected for our study. Then the distance metric is proposed. Before discussing our apporach, we first look at some of the previous work in the next section.

Chapter 2

Related Work

2.1 Adversarial Examples

Adversarial examples are produced by adding small perturbations to image. These perturbations are not visible to human eye and cause neural network to misclassify the image. This property of neural network was first discovered by Szegedy et al. Szegedy et al. (2014b). In the work they also show that same perturbation can cause a different network to misclassify the same input. In a similar work Goodfellow et al. (2014) proposes an approach to train neural network adversarially so they are robust against small perturbation. They also observe similar portability of adversarial examples across different neural network as shown in Szegedy et al. (2014b). Images were shown to be classified as a recognizable class with a very high confidence even though they were completely imperceptible to humans in the work Nguyen et al. (2014).

Since the discovery of adversarial property of neural network, several approaches for attacks and defenses have been proposed. Adversarial attacks are different strategies to fool the network by producing examples which network can misclassify with small perturbation whereas defense deals with making networks robust against such attacks. Liu et al. (2020) provide a study of different types of adversarial attacks and defenses. They also study how the perturbation can be interpreted at a feature or pixel level. Moosavi-Dezfooli et al. (2015) develops a new algorithm called Deepfool to generate adversarial examples that can fool state of the art neural network for images. Ma et al. in his work Ma et al. (2021) proposes an approach to generate adversarial examples in medical image system. Adversarial examples can be created for any type of data for classification and not just images. In the work Kereliuk et al. (2015) the authors adapts the approach to create adversarial examples for music dataset.

For our work, we select some efficient adversarial attacks which we use to create adversarial examples on our selected neural networks. In this section we present some previous work different type of adversarial attacks. Adversarial attack is basically an optimization problem where the input is perturbed by small epsilon and our goal is find such perturbation that the output label is different from true ground label. In the paper Madry et al. (2017), the author uses Projected Gradient Descent as a reliable first order adversary to create adversarial examples for MNIST and CIFAR10 dataset. Another approach for adversarial attack is using Fast Gradient Sign Method (FGSM). While FGSM is faster than PGD, it is not as effictive as PGD. In the work, Huang et al. (2020) the author proposes curvature regularization to bridge the performance gap between PGD and FGSM.

2.1.1 Robustness and Adversarial Portability

Adversarial examples have also raised questions regarding the robustness and security of deep neural networks. Many works have focussed on making neural network robust against small perturbations by training the network with these generated adversarial examples. Zheng et al. (2016) develops a stability training method to stabilize deep networks against small distortions that result from various types of common image processing such as compression, rescaling and cropping. Jin et al. (2015) proposes a feed forward neural network that improves robustness in the presence of adversarial noise. The proposed model works together with the CNN trained using standard objective function.

Some previous work have focussed on studying the robustness of neural network. Robustness of a neural network means how well the network perform when the input image is modified by small perturbation. A robust network would perform good for small perturbations. Dong et al. (2019) performs benchmarking of robustness of various neural network against different attacks. Wang and Wang (2022) proposes an approach called Self-Ensemble Adversarial Training that increases the robustness of model against well known adversarial attacks.

Another interesting aspect apart from robustness is portability of adversarial examples that is whether adversarial examples created on one neural network are transferable to other neural networks. Rozsa et al. (2016a) performs such a study to test robustness as well as cross model adversarial portability on image dataset from ImageNet Deng et al. (2009). Study on transferability of neural network when the target network or data is different than what the adversarial example was generated on was done by kanth Nakka and Salzmann (2021).

2.2 Similarity Metrics

Given that we also want to study how different are the predictions for adversarial example from the true label in untargeted attack and from target in case of targeted attack, we look in this section some previous work on distance metric for different classes. Some of the previous work make use of the ImageNet heirarchy while other makes use of the deep feature and it's semantic distance. Fezza et al. (2019) proposes a new metric to evaluate similarity between adversarial image and original image. In the paper Roads and Love (2020), the author includes human judgment into imagenet dataset... Deselaers and Ferrari (2011) makes use of the imagenet heirarchy to propose a distance metric. They argue the visual similarity is related to semantic similarity of categories based on wordnet and depth in the ImageNet tree.

Chapter 3

Approach

In this chapter, we describe our approach for studying the portability of adversarial examples across different network architectures. Adversarial examples are generated to fool the network, an image that was previously classifed correctly with an addition of non random perturbation which are indistinguishable to human eye gets classified to a wrong class. Given a source network S our aim is to study if an adversarial example created on S using an attack A also works as an adversarial example on target network T. Given that most current work is based on untargeted attacks, in this study we also include targeted attack. The target can be selected in various ways. We use three different criteria for selection of target which is described in the later section.

As described in the introduction, neural networks are susceptible to small perturbation in the input image. Changes in the pixel by small noise can cause networks to predict output largely different from the true label. We want to see whether these perturbations are transferable across network. Since the configuration of the adversarial attack can have an effect on the transferability we perform the study for different values of epsilon or perturbation and also for different adversarial attack. For both targeted and untargeted attack, we develop a distance metric that computes the distance between prediction and true label in case of untargeted attack or the defined target in case of targeted attack.

3.1 Neural Network Architecture

Karnala and Campbell (2021) show that certain model types and architecture are more susceptible to adversarial attacks and hence model architecture impacts the efficacy of adversarial example. They study portability of adversarial examples generated against machine learning system used in self-driving aaplications. The study uses adversarial examples generated using FGSM on three different architectures, Dave-2, VGG and ResNet. Other study also suggests that adversarial example are more portable across similar neural network architecture.

In this study, we select 13 different neural network architecture which are commonly used for vision tasks such as image classification. Each of these networks acts as a source and also as a target. In terms of portability or transferability, source model would mean the neural network on which the adversarial example was created, whereas the target model is the network on which we are evaluating the adversarial example. The input layer of all our selected network takes an image of dimension $224 * 224$. We generate adversarial example on each of the 13 networks and test whether it also is adversarial example on the other remaining network.

The network architecture we selected to perform our portability study on are listed below in the table 3.1. In the following section, we also describe the architecture of selected neural network models.

Network	type	top1	top5
ResNet	ResNet18	69.76%	89.08%
	ResNet34	73.30%	91.42%
	ResNet50	76.15%	92.87%
	ResNet101	77.37%	93.56%
	ResNet152	78.31%	94.06%
	WideResNet	78.48%	94.08%
VGG	Vgg16	71.59%	90.39%
	Vgg19	72.39%	90.88%
MobileNet	MobileNet v2	71.87%	90.29%
	MobileNet v3 small	67.67%	87.41%
	MobileNet v3 large	74.05%	91.34%
DenseNet	DenseNet	77.11%	93.56%
GoogleNet	GoogleNet	69.77%	89.53%

Table 3.1: Network architectures used for the study along with top1 & top5 accuracy

3.1.1 ResNet

ResNet stands for Residual Network. The main idea of ResNet is introduction of identity connection that skips one or more layers. It was introducted by He et al. (2015). The authors proposes that residual learning ease the training of networks that are substantially deeper. This allows for considerably increased depth neural networks which can gain better accuracy. 3.1 shows residual connections. As seen the input from a previous layer is passed to a deeper layer and is added to the output of the deep layer.

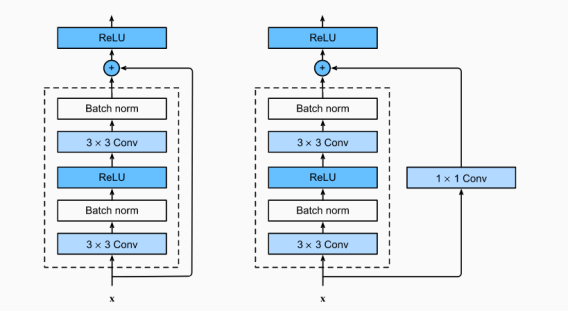


Figure 3.1: regular block (left) vs a residual block (right)

ResNet has a 3*3 convolutional layer design which is similar to VGG's layer. The residual block has two 3 * 3 convolutional layer which is followed by batch normalization and a ReLU activation. The input is skipped and added to the output from batch normalization layer. A convolutional layer of 1*1 dimension can be used to get the desired shape before performing the addition. The ResNet uses four modules stacked together which are made up of residual blocks. The input layer is 7*7 convolutional layer followed by batch normalization and 3*3 max pool layers. The residual blocks are then stakced sequentially followed by a fully connected layer.

There are different variant of ResNet architecture, in this work we use ResNet18, ResNet34, ResNet50, ResNet101, ResNet152 and WideResNet. The number after ResNet denotes the total

number of layers in the network. WideResNet was introduced to tackle the problem of training time. With increase in the number of layers, deep residual networks has a problem of diminishing feature reuse, which makes the networks slow to train. WideResNet solves this by reducing the number of layers and increasing the width of residual networks.

3.1.2 VGG

VGG was proposed in the paper "Very Deep Convolutional Networks for Large Scale Image Recognition" Simonyan and Zisserman (2014). There are currently two versions of VGG, VGG16 which has 16 layers and VGG19 which has 19 layers. The VGG block consists of sequential convolutional layers which are followed by a maximum pooling layer. The pooling layer is used for downsampling the feature space. The convolutional layer uses 3*3 kernels and a padding of 1. The pooling layer has a dimension of 2*2 and stride of 2. The VGG network uses sequentially stacked vgg blocks followed by fully connected layers. The final layers of VGG has 3 fully connected layers and the last layer has 1000 neurons, one for each class.

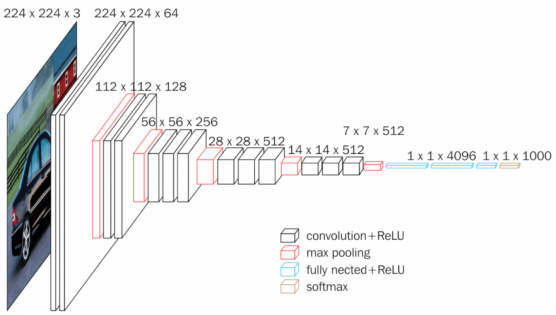


Figure 3.2: VGG16 architecture Simonyan and Zisserman (2014)

3.1.3 MobileNet

MobileNet is neural network architecture introduced by Google. It uses Depthwise Separable Convolution to reduce model size and complexity and hence can be used for mobile and embedded vision applications. A Depthwise separable convolution applies two convolution, depthwise convolution and then pointwise convolution. The two operations are explained later below.

Depthwise convolution is similar to spatial convolution but with as many filters as number of channels. The pointwise convolution is 1*1 convolution to change the dimension.

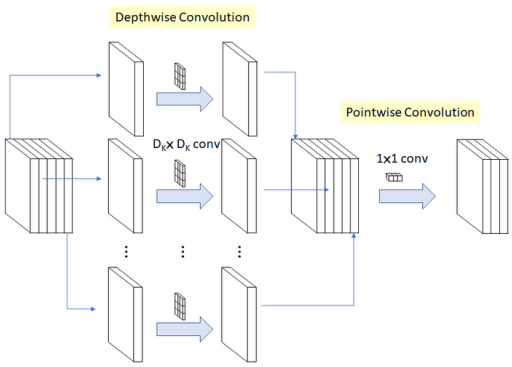


Figure 3.3: Convolution opertaion in MobileNet

We selected three different variants of the MobileNet architecture, namely mobilenet v2, mobilenet v3 small and mobilenet v3 large. MobileNet v3 adds squeeze and excitation layers in the initial building block taken from V3. The v3 large variant is targeted at high resource tasks and v3 small variant for low resource task. The architecture is optmized by a neural architecture search process.

In the figures 3.4 3.5 below , you can see the different building blocks of v2 and v3 variant.

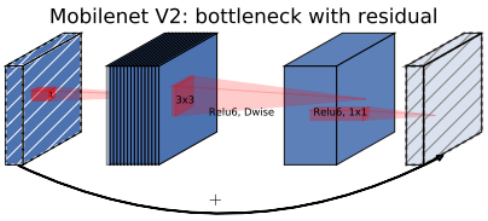


Figure 3.4: MobileNet v2 building block

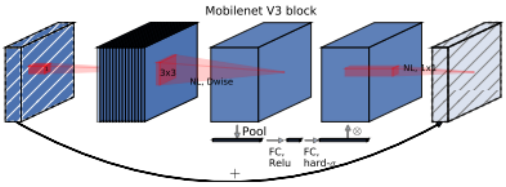


Figure 3.5: MobileNet v3 building block with squeeze and excitation layers

3.1.4 DenseNet

DensetNet stands for Densely Connected Convolutional Networks. Huang et al. (2016). It uses similar idea from ResNet of adding skip layers. In DenseNet, each layer is connected to every other layer in a feed forward fashion. Hence each layer uses it's own feature map along with features from all preceding layers. The difference between the ResNet skip connection and DenseNet skip connection is that instead of adding the input from previous connection, DenseNet concats this input as can be seen from the figure 3.6.

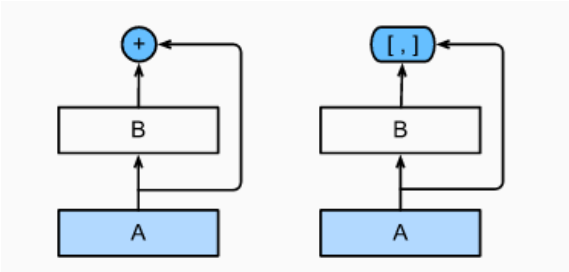


Figure 3.6: ResNet (left) vs DenseNet(right) blocks

It has several advantages over other Convolutional networks and solves the vanishing gradient problem, strengthen the feature propagation. It also substantially reduce the number of parameters since every layer using the output from all previous layer.

We use the DenseNet161 variant which has total of 161 layers. The DenseNet network is built by stacking dense blocks in a sequential manner. The DenseNet block uses a modified structure of batch normalization, activation and then convolution. Each dense block recieves input from all the previous dense blocks. Since each dense block will increase the number of channels, as we are concatenating the input from previous blocks, a transition layer is used to control the complexity. It uses a 1*1 convolutional layer with a average pooling of stride 2.

The figure 3.7 shows the dense block and the connections that it receives from previous blocks.

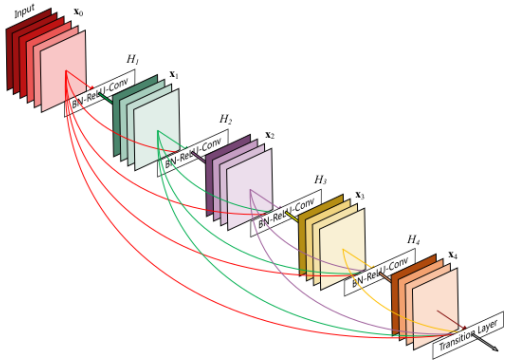


Figure 3.7: Dense block with layers receiving preceding feature maps as input Huang et al. (2016)

3.1.5 GoogleNet

GoogleNet is a 22-layer deep convolutional network that's uses a similar architecture as Inception Network developed by Google. GoogleNet uses Inception block as the basic building block. The structure of Inception block is shown below 3.8

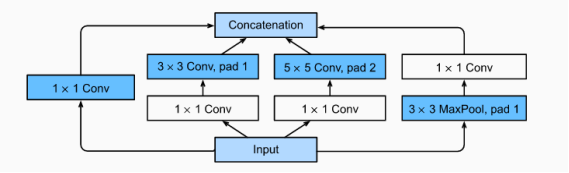


Figure 3.8: Inception Block Szegedy et al. (2014a)

The Inception block has 4 different parallel paths from the input to the output. The first three paths uses convolutional layer with dimensions of 1*1, 3*3 and 5*5. The fourth path in addition to convolutional layer it first uses a 3*3 max pool layer. The output of all the paths is then concatenated to form the output from the Inception block.

GoogleNet model stacks these inception blocks sequentially. Between the inception blocks, max pool layers of size 3*3 are used.

3.2 Attacks

In the previous section, we discussed the neural network architecture that we use to create adversarial examples and also evaulate it. In this section, we describe different adversarial attack we use. Given that some adversarial attacks are stronger than others and are able to find better adversarial example, it is necessary to test different attacks since this would also affect the study of transferability. Better adversarial example would be have a higher portability across the selected neural network model. For this reason we select 3 different attacks namely, PGD,FGSM and FGV. All the three attacks are used with three different perturbation constant of 0.01, 0.1 and 0.3.

3.2.1 FGSM

FGSM also know as Fast Gradient Sign Method, is a type of white box attack that uses gradients of the neural network to generate adversarial example. It uses the gradients of loss function with respect to image and then uses the sign of gradients to create image. The image is perturbated in the direction such that it increases the loss between predicted and true value.

The formula to express FGSM is,

$$\hat{x} = x + \epsilon * sign(\nabla_x L(\theta, x, y)) \tag{3.1}$$

where x is the input image, y is the ground truth, ϵ is the strength of the perturbation, L is the loss function and θ is the parameters of neural network.

The loss function computes the error between predicted value and the true value. When we are the training a network for classification task, the loss function drives the weights of the network such that the average loss over training data is reduced. In case of untargeted attacks, we want to maximize this loss since our goal is to make the network predict a class different fromt true class whereas for targeted attacks, since we have a specific target which we want the network to

predict, our aim is to reduce the loss between the target and prediction. We use cross entropy loss for our work since it is the most common used loss for classification tasks.

3.2.2 PGD

PGD or Projected Gradient Descent is another white box attack. It follows iterative approach and maximizes the loss of model on an input by adding small perturbation smaller that ϵ. The distance metric used between perturbed image and original image is L^2 or L∞ norm. It starts with a random perturbation around the input. It then takes a step in the direction of greatest loss using gradient descent approach. The update rule for PGD is same as that of FGSM as described here 3.2.1. The gradients of the loss are taken with respect to input image. It then moves a step of ϵ in the direction of gradient that increases the loss. Finally if the perturbation is more than allowed ϵ then it projects back to the boundary. The steps are repeated until convergence is reached.

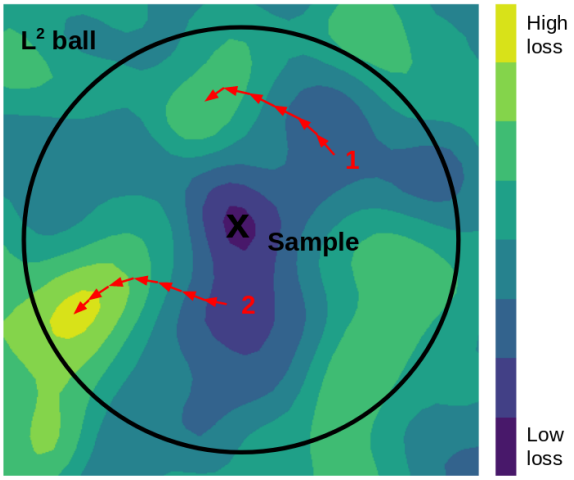


Figure 3.9: Illustration of PGD attack Madry et al. (2017)

3.2.3 FGV

FGV or Fast Gradient Value Rozsa et al. (2016b) uses a scaled version of the raw gradient of loss, this step is different than both PGD and FGSM which uses the gradient of loss with respect to input image. FGV produces significantly different adversarial perturbations than FGSM or PGD. Using the scaled verson of raw gradient rather than sign of gradient allows FGV to take into consideration the magnitude difference between corresponding pixels. The idea behind using scaled gradients is to increase the loss with minimal distortion, pixel with large gradients are changed more than others.

3.3 Choosing Target

For Targeted Attack, we need to define the target class for our adversarial attack. Based on the distance of target from the ground truth class, it will be easier or difficult to generate a good adversarial example. Our hypothesis is that it is easier to find an adversarial image using targeted

attack when the target is closer to the original image than when it is farther away from the image. We use three different ways of choosing the target, namely **1. closest 2. median 3. farthest**. The distance metric to compute these target is described in the next section in detail.

3.3.1 Distance Metric

As mentioned in the previous sections, in addition to seeing if the adversarial images are portable across neural networks, we also want to measure how good the adversarial examples are. We want to see if an adversarial example is predicted as the same class across the neural network and if not how far is the prediction from the true label. In case of targeted attack, we want to see if the prediction on the adversarial example was close to the target class if target was not reached. To study this, we introduce a distance metric that gives us the distance between any two classes from the 1000 classes of the ILSVRC 2012 dataset which is a subset of ImageNet dataset.

In this work, we define a distance metric, leveraging the information stored in the ImageNet Heirarchy. As we know the ImageNet classes are in form of a tree and are organized according to WordNet hierarchy and each node has thousand of images. In the figure 3.10, we show example of how ImageNet hierarchy looks.

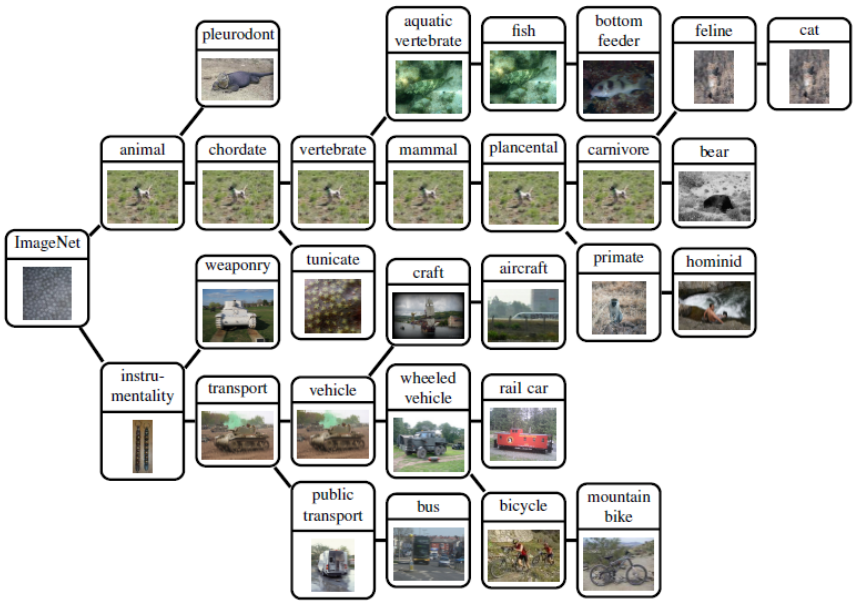


Figure 3.10: ImageNet Heirarchy Deng et al. (2009)

Based on this we define our distance or cost metric as described below,

$$d(c1, c2) = h(c1) - h(c) + h(c2) - h(c) \tag{3.2}$$

where, c1 and c2 are two different class in the ImageNet c is the lowest common ancestor of c1, c2 d(c1, c2) is the cost of misclassifying c1 as c2 h(c1) is the height of node c1 in the tree h(c2) is the height of node c2 in the tree and h(c) is the height of lowest common ancestor of c1, c2.

Based on this formula, we pre-compute the distance matrix for each pair of 1000 classes in the ImageNet heirarchy. We store the 1000*1000 matrix and later use it to get the cost so that we don't need to calculate the cost at every step. The minimum distance in the computed matrix is 2 and maximum distance is 26. The average distance or average cost of misclassifying a class c1 as class c2 is 12.83. The figure below 3.11 shows a bar plot with x axis being the possible values of distance and y scale is the number of class pair with this distance (in the logarithmic scale).

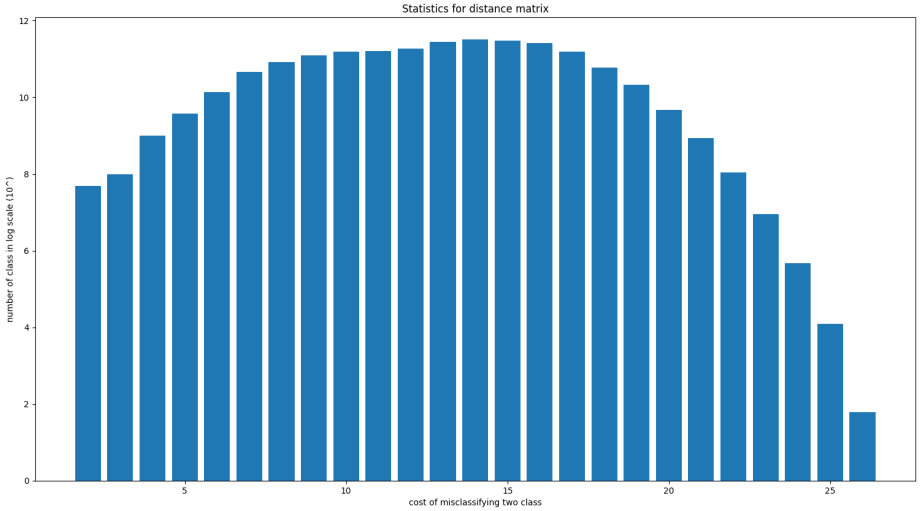


Figure 3.11: ImageNet Heirarchy

Based on the distance metric introduce, using the precomputed cost matrix, we select the target for our adversarial attack using three strategies,

closest, the closest target is the one which has the least distance to the true class. c is the closest target if $d(c, trueclass)$ is minimum and c is different from true class.

median, the median target to the true class if it is distance is the median out of all the distances from other classes to the true class.

farthest is the class which is farthest from the true class. c is the farthest target if $d(c, trueclass)$ is maximum.

Chapter 4

Experiment

In this chapter, we describe our experiments and the results. The results are further analyzed in Chapter 5. As described in Chapter 3, corresponding to each of our research problems we perform an experiment. The first two experiments study the portability of untargeted attack and last experiment studies portability of targeted attack. The first experiment is to study what fraction of adversarial examples generated on one network is portable or transferable to other selected neural networks. In the second experiment, we study how far are the prediction of adversarial example from the ground truth label. This will help us answer the question, whether the prediction of untargeted adversarial examples are same across different neural network. We use the cost matrix described in Chapter 3 to compute the distance between any two image class.

4.1 Dataset

We use ImageNet dataset Deng et al. (2009) for our experiments. It is an image database organized to the WordNet heirarchy in which each noun node of the heirarachy is depicted by hundred and thousands of images. It is an effort to make traininig on computer vision tasks easier. We chose the dataset released under ImageNet Large Scale Visual Recognition Challenge 2012 which contains a subset of ImageNet data released for the challenge. This subset contains 1000 classes and 1.2 million images in the training set. Below, in figure 4.1 you can see the images in the hierarchy of ImageNet. The leaves in the hierarchy are used as classes.

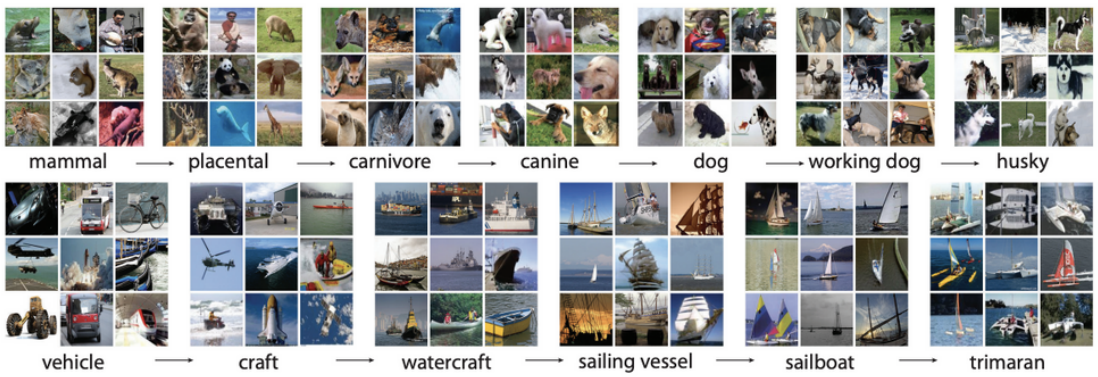


Figure 4.1: hierarchy in ImageNet dataset, citeimagenet

To study the portability of the adversarial examples, we want a subset of the data such that each of the image in the subset is correctly classified by all of our chosen neural network. We select 10000 images from ILSVRC 2012 dataset, containing 10 images per class which are correctly classified on the 13 chosen neural network.

4.2 Implementation

The implementation was done using pytorch and uses the library advertorch Ding et al. (2019) to generate adversarial examples. We use pretrained models from torchvision in eval model during testing for portability of adversarial examples.

The implementation task was split into two parts. One was to generate adversarial examples for the selected 10000 images from ILSVRC 2012 data. The adversarial examples were created on all the 13 selected neural networks which act as source. The other part of implementation was evaluating these adversarial examples on all the target networks.

Since each of the network act as source and target and we use 3 different ϵ values with 3 adversarial attacks, the number of times (13*3*3) we are evaluating and using the target network prediction is large. The running time of the experiments initially was very high. To solve this issue, we used torch multiprocessing to run the evaluation in parallel. This enables us to run interference on data with different models at the same time. The evaluation on each of the target network runs in parallel in its own process. Using multiprocessing we were able to reduce the run time to one tenth of original runtime.

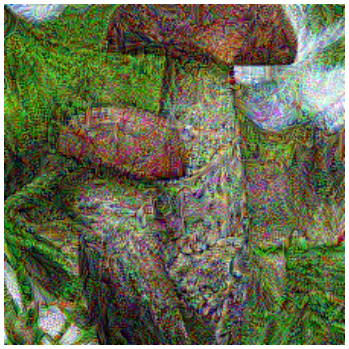
We also run the adversarial attack or generate adversarial examples in parallel. In each of the process, we load the ImageNet dataset and apply adversarial attack with different ϵ. Each process corresponds to one neural network as the source. When testing the models for portability, since each of the model acts as source and also as a target, given that we select 13 neural network models, we have 169 pairs of source and target model. We parallelize the evaluation of portability across the target model, that is for each 13 target model we have one process running in parallel. In each of these processes, we load adversarial examples created on 13 source models and then evaluate the prediction. The process at the end returns the prediction by target for each of the 10000 adversarial example created on every 13 source. Later it is aggregated to get prediction for each adversarial example for every source-target pair. The result is stored as a csv file and we calculate the distance metric and other statistic in an offline manner.

The created adversarial examples are stored as tensor, this preserves the small perturbation and allows us to have better precision between loading and saving the adversarial examples. In the figure below 4.2, we show an example of adversarial attack created using different attacks and epsilon value. As can be seen with the examples, the perturbations with 0.3 are quite visible whereas the one with ϵ 0.01 are not. Also FGV produces examples with much less perturbation values.

4.3 Portability of Untargeted Adversarial Examples

In this section we present our results from the first experiment. This experiement measure number of adversarials examples created on a network, say m1, that works as an adversarial example on other networks m2, m3....m13. It answers our first research question, 1 which to study if adversarial examples are portable across different network architecture. Our hypothesis is that examples are portable across network architectures which are similar.

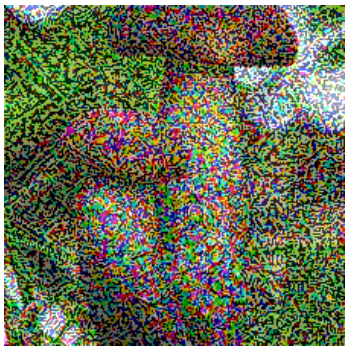
To formulate the problem, we define a source network S on which the adversarial example was created, and a target network T on which we are evaluating the adversarial example created



(a) PGD 0.3

(b) PGD 0.1

(c) PGD 0.01



(d) FGSM 0.3

(e) FGSM 0.1

(f) FGSM 0.01

(g) FGV 0.3

(h) FGV 0.1

(i) FGV 0.01

Figure 4.2: adversarial examples created on VGG16

on S. If an adversarial example, say I created on S also works as adversarial example on T, it would mean that the adversarial example I is portable to network T.

The metric we use here to define the portability is simple fraction of 10000 images that are portable to network T. So, P(S, T) can be defined as portability of adversarial example created on source network S to target network T as,

$$P(S,T) = n/N \tag{4.1}$$

where n, is the number of adversarial examples from S that are portable to T and N is the total number of adversarial examples, which in our case is 10000 since our dataset consists of 10 images for each of the 1000 ImageNet class.

We have 13 source network and each of the source network also work as a target network. Hence in total we have 169 (13*13) portability value corresponding to each pair of source and target network. In the table below 4.1, these 169 portability values are show. The portability values in the table are for PGD attack and ϵ of 0.3. Since it is easier to observe the trend with visualization, we include the portability tables in A for other attacks and values of ϵ. The cells marked with red are lowest value and green cells are highest value in the row.

source	resnet18	resnet34	resnet50	resnet101	resnet152	vgg16	vgg19	googlenet	densenet	mobilenet v2	v3 small	v3 large	wide resnet
resnet18	10000	9485	8787	8362	7589	9700	9476	8656	7784	9312	7986	8359	7699
resnet34	9685	10000	8900	8435	7843	9540	9324	8498	8021	9156	7861	8128	7973
resnet50	9466	9098	10000	8800	8157	9438	9218	8367	8005	9064	7671	7735	8403
resnet101	9409	9127	9274	10000	9085	9378	9120	8429	7974	9070	7818	8032	8340
resnet152	9290	9074	9138	9349	10000	9324	9099	8323	8216	9019	7763	7939	8560
vgg16	9540	9101	8798	8077	7677	10000	9991	8844	8087	9549	8129	8796	8205
vgg19	9468	9002	8667	8025	7608	9986	10000	8804	7917	9428	7910	8612	7954
googlenet	9087	8346	8031	7249	6649	9375	9151	10000	6523	8927	7751	7685	6763
densenet	9473	9149	8990	8592	8193	9436	9241	8577	10000	9101	7810	8170	8463
mobilenet v2	8911	8151	7527	6570	5876	9434	9146	7058	5541	10000	8148	8183	5930
v3 small	8837	8088	7401	6494	5909	9333	9175	7032	5529	9240	10000	8815	6141
v3 large	9168	8517	7841	7211	6582	9457	9218	8022	6704	9372	9045	10000	6640
wide resnet	9334	8973	9074	8620	8450	9386	9185	8354	8469	8895	7702	7803	10000

Table 4.1: Untargeted attack with PGD and alpha=0.3

Rather than including all the tables with raw portability values, we visualize portability of all the pairs of source and target along with different attacks and ϵ in the figure below 4.3 as 9 heatmaps. Each of the heatmap corresponds to one attack PGD, FGSM, FGV and one ϵ 0.01, 0.1, 0.3. The diagonal value is always 10000 since the source and target are the same.

From the visualization, we can clearly see that with low ϵ the portability is very poor. This can be attributed to the fact that it is difficult to find a strong adversarial example with low perturbation. The diagonal in the heatmaps are always dark green since it corresponds to same target network as source on which adversarial example was generated. We also see that portability with a higher perturbation works better with FGSM as compared to PGD attack, but as we decrease the perturbation PGD works slightly better than FGSM. This could be because PGD was able to produce adversarial examples with less perturbation in the image, as seen from the figure 4.2. FGV on the other hand works poorly for all the three pertubations because it produces less perturbations compared to FGSM and PGD since it uses raw gradients. Another interesting thing we observe is VGG has greater portability compared to other networks. Also the examples created on vgg16 has a very high portability to vgg19 and vice versa, which can be seen from a square green block in center of heatmap.

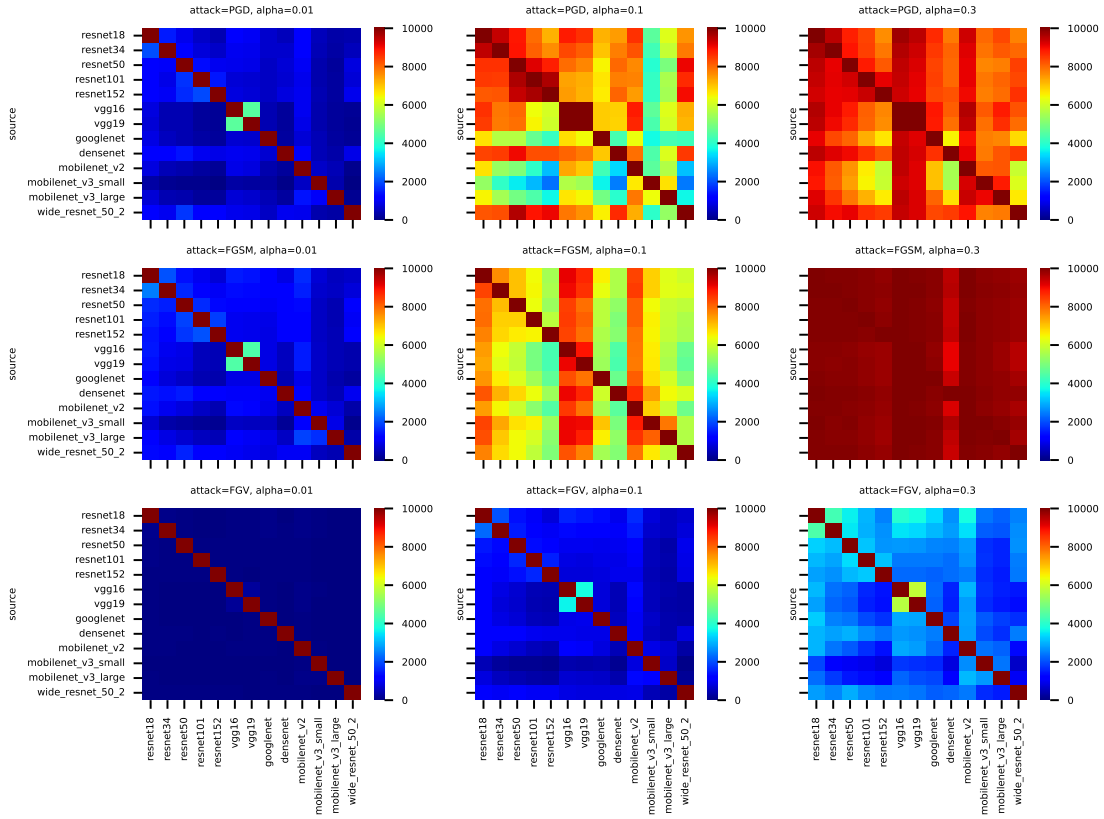


Figure 4.3: untargeted attack with fixed epsilon

As we saw from the heatmap visualization, the portability of adversarial examples decreases significantly as we reduce ϵ and also PGD and FGSM produces examples that are more portable. To better understand what pair of source and target network gives better portability, in the table 4.2 below we list down maximum and minimum portability achieved for each of the source network and also the target network for which this portability was achieved.

source	maximum	minimum
resnet18	vgg16 (9700)	resnet152 (7589)
resnet34	resnet18 (9685)	resnet152 (7843)
resnet50	resnet18 (9466)	mobilenet_v3_small (7671)
resnet101	resnet18 (9409)	mobilenet_v3_small (7818)
resnet152	resnet101 (9349)	mobilenet_v3_small (7763)
vgg16	vgg19 (9991)	resnet152 (7677)
vgg19	vgg16 (9986)	resnet152 (7608)
googlenet	vgg16 (9375)	densenet (6523)
densenet	resnet18 (9473)	mobilenet_v3_small (7810)
mobilenet_v2	vgg16 (9434)	densenet (5541)
mobilenet_v3_small	vgg16 (9333)	densenet (5529)
mobilenet_v3_large	vgg16 (9457)	resnet152 (6582)
wide_resnet_50_2	vgg16 (9386)	mobilenet_v3_small (7702)

Table 4.2: best and worst portability for every source network with PGD and alpha=0.3

We can infer from the table that maximum portability is always between architectures which are similar as is the case with different variants of resnet. This supports our hypothesis that adversarial examples have a greater portability if source and target have similar structure. We further include the above table for other adversarial attacks and ϵ values in the Appendix A.

An interesting observation made from the first experiment is that portability is affected by the ϵ. To visualize the result further we show plots for 5 different network and their corresponding average portability across every other network. We can see that with decrease in epsilon, the portability is significantly affected. 4.4. Further to that we also see how FGV performs poorly when it comes to generating portable adversarial examples.

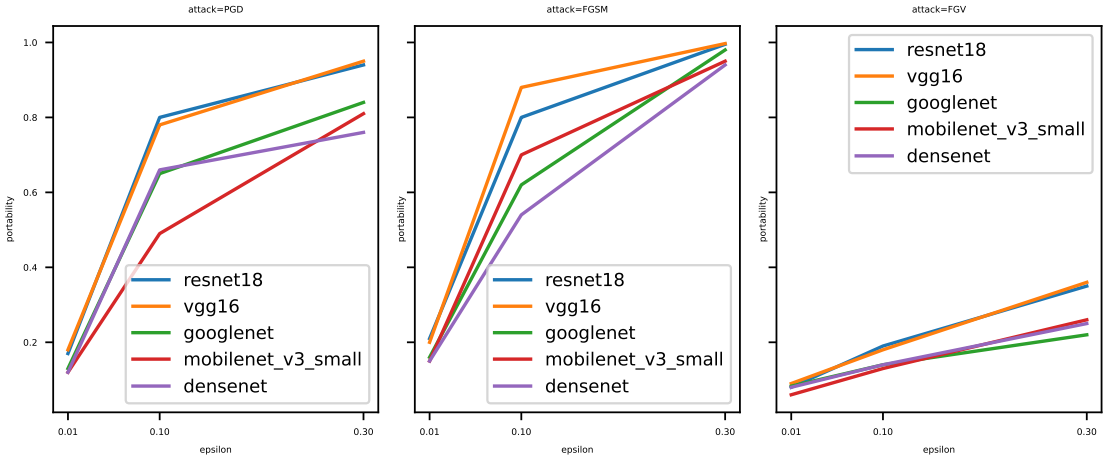


Figure 4.4: epsilon ϵ vs portability

4.4 Predictions on Untargeted Adversarial Examples

In the last section, we saw if adversarial examples are portable across different neural network architecture and how the portability is affected by type of attack we perform and pertubation constant ϵ we use. While this does give an insight into portability of adversarial examples, we still do not know how the predictions on these adversarial examples look. To look into the portability with more in depth analysis, we now look at the predictions of these adversarial example and whether the predictions are also same on different networks and how far the predictions are from the true class. To quantitatively measure this, we define mean and standard deviation of the distance of the predicted class to ground truth class. The distance is calculated based on the cost matrix as described in Approach section.

To formulate the equation, we define avg(S, T) which is the mean distance between true class and the predicted class for an adversarial example generated on S and evaluated on T and std(S, T) is the standard deviation from true class to the predicted class,

$$avg(S,T) = 1/n \times \sum_{I=1}^{I=n} d(c, T(I)) \tag{4.2}$$

$$std(S,T) = \sqrt{\frac{\sum_{I=1}^{I=n}(d(c-T(I)) - avg(S,T))^2}{n-1}} \tag{4.3}$$

where c is the true class, T(I) is the predicted class for image I on network T, d(c, T(I)) is the distance between c and T(I) from the pre computed distance matrix.

A larger mean would indicate that the predicted class was on average further away from the true class. It would also indicate that the adversarial examples work better since it is not only wrongly classified but also that the predicted class is far from true class. A higher standard deviation on the other hand means that predicted class varied across different adversarial examples. On networks with lower portability, standard deviation would be larger since more adversarial examples would be predicted with the true class. Here we include the mean and std table for PGD attack with ϵ 0.3. The rest of the tables for different values of ϵ and FGSM and FGV are included in the appendix. A.

source	resnet18	resnet34	resnet50	resnet101	resnet152	vgg16	vgg19	googlenet	densenet	mobilenet v2	v3 small	v3 large	wide resnet
resnet18	11.26, 3.72	10.29, 4.57	8.99, 5.07	8.59, 5.45	7.65, 5.73	10.68, 4.34	10.07, 4.62	8.96, 5.13	7.96, 5.67	9.90, 4.83	8.01, 5.40	8.62, 5.35	7.77, 5.68
resnet34	10.44, 4.37	10.76, 3.85	8.92, 5.00	8.45, 5.34	7.73, 5.55	10.14, 4.53	9.63, 4.75	8.49, 5.17	7.93, 5.46	9.55, 4.98	7.73, 5.43	8.12, 5.39	7.83, 5.47
resnet50	9.99, 4.60	9.31, 4.87	10.55, 3.81	8.71, 5.09	8.01, 5.41	9.87, 4.58	9.36, 4.78	8.22, 5.19	7.76, 5.39	9.41, 5.05	7.53, 5.48	7.55, 5.50	8.26, 5.27
resnet101	9.75, 4.67	9.26, 4.89	9.18, 4.72	10.20, 3.95	8.96, 4.86	9.79, 4.69	9.27, 4.88	8.18, 5.12	7.64, 5.40	9.30, 5.03	7.60, 5.40	7.82, 5.34	8.05, 5.31
resnet152	9.55, 4.75	9.11, 4.91	8.85, 4.81	9.21, 4.76	10.10, 3.95	9.63, 4.70	9.16, 4.91	8.04, 5.18	7.78, 5.25	9.28, 5.09	7.54, 5.42	7.63, 5.38	8.19, 5.16
vgg16	10.56, 4.47	9.79, 4.91	9.24, 5.15	8.28, 5.53	7.83, 5.71	11.34, 3.51	11.39, 3.59	9.38, 5.03	8.54, 5.61	10.54, 4.44	8.16, 5.29	9.19, 4.99	8.64, 5.54
vgg19	10.42, 4.58	9.65, 4.96	9.09, 5.22	8.29, 5.61	7.89, 5.78	11.62, 3.69	11.54, 3.59	9.32, 5.05	8.42, 5.73	10.47, 4.67	7.87, 5.42	8.96, 5.14	8.35, 5.64
googlenet	9.40, 4.92	8.34, 5.31	7.71, 5.33	6.94, 5.65	6.39, 5.79	9.85, 4.69	9.31, 4.85	10.69, 3.89	6.10, 5.65	9.15, 5.11	7.59, 5.46	7.55, 5.51	6.42, 5.70
densenet	9.54, 4.68	8.83, 4.92	8.38, 4.91	7.95, 5.18	7.58, 5.35	9.68, 4.65	9.20, 4.82	7.95, 5.05	9.44, 4.26	9.31, 5.05	7.54, 5.45	7.78, 5.30	7.79, 5.22
mobilenet v2	9.27, 5.00	8.27, 5.44	7.26, 5.52	6.31, 5.78	5.64, 5.85	10.05, 4.60	9.53, 4.88	6.75, 5.58	5.13, 5.64	11.41, 3.68	8.18, 5.34	8.32, 5.38	5.63, 5.76
v3 small	9.24, 5.11	8.11, 5.41	7.16, 5.58	6.23, 5.80	5.73, 5.89	9.85, 4.69	9.49, 4.84	6.85, 5.66	5.16, 5.67	9.60, 4.82	11.56, 3.61	9.26, 5.07	5.93, 5.85
v3 large	9.70, 4.84	8.76, 5.29	7.65, 5.46	7.05, 5.75	6.42, 5.84	10.11, 4.61	9.65, 4.86	8.05, 5.41	6.49, 5.74	9.95, 4.76	9.40, 4.84	11.13, 3.75	6.42, 5.80
wide resnet	9.66, 4.75	9.04, 5.00	8.77, 4.81	8.38, 5.18	8.19, 5.24	9.80, 4.71	9.27, 4.86	8.11, 5.19	8.18, 5.18	9.17, 5.20	7.46, 5.43	7.55, 5.46	10.05, 3.99

Table 4.3: Mean and Std for PGD and alpha=0.3

source	maximum	minimum
resnet18	resnet34 (10.29)	resnet152 (7.65)
resnet34	resnet18 (10.44)	mobilenet_v3_small (7.73)
resnet50	resnet18 (9.99)	mobilenet_v3_small (7.53)
resnet101	vgg16 (9.79)	mobilenet_v3_small (7.6)
resnet152	vgg16 (9.63)	mobilenet_v2 (7.54)
vgg16	vgg19 (11.54)	resnet152 (7.83)
vgg19	vgg16 (11.62)	mobilenet_v3_small (7.87)
googlenet	vgg16 (9.52)	densenet (6.1)
densenet	vgg16 (9.68)	mobilenet_v3_small (7.54)
mobilenet_v2	vgg16 (10.05)	densenet (5.13)
mobilenet_v3_small	vgg16 (9.85)	resnet152 (5.73)
mobilenet_v3_large	vgg16 (10.11)	wideresnet (6.42)
wide_resnet_50_2	vgg16 (9.8)	mobilenet_v3_small (7.46)

Table 4.4: highest and lowest mean for every source network with PGD and alpha=0.3

As expected, we find the mean in the diagonals to be largest since the source and target network are same. From the last experiment we saw that all the variants of mobilenet had lower portability compared to other networks. In the table above we also note that they have a smaller mean which suggest that predicted class on this network was closer to the true class. The mean corresponds to the portability from the last experiment and this can be seen in the table 4.4 which shows similar pair of source and target model to have higher mean although it is not always the case that pair of network with highest portability also had the highest mean. It indicates that even though portability on a network can be higher it might still predict the class closer to the true class. We find VGG network to have higest mean among all the network and adversarial example between the two variants of VGG have a higher means suggesting better portability among these

two network.

We further visualized the mean and standard deviation shown in the table above. Each of the circle correcspoding to a pair of source and target network. The radius of the circle is proportional to the std(S, T) and the color of the circle is dependednt on the avg(S, T). We use the jet colormap which means red represent a circle with higher avg and blue means the circle had low mean value.

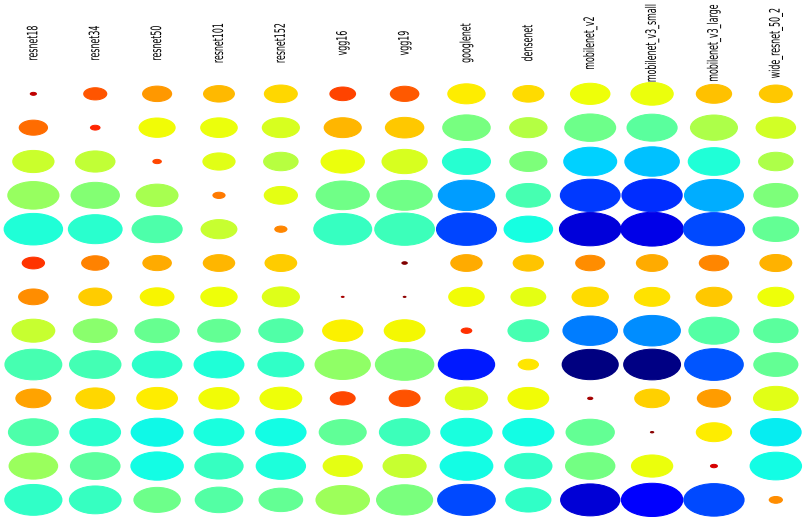


Figure 4.5: mean and std of predicted class distance from true class for PGD and epsilon 0.3

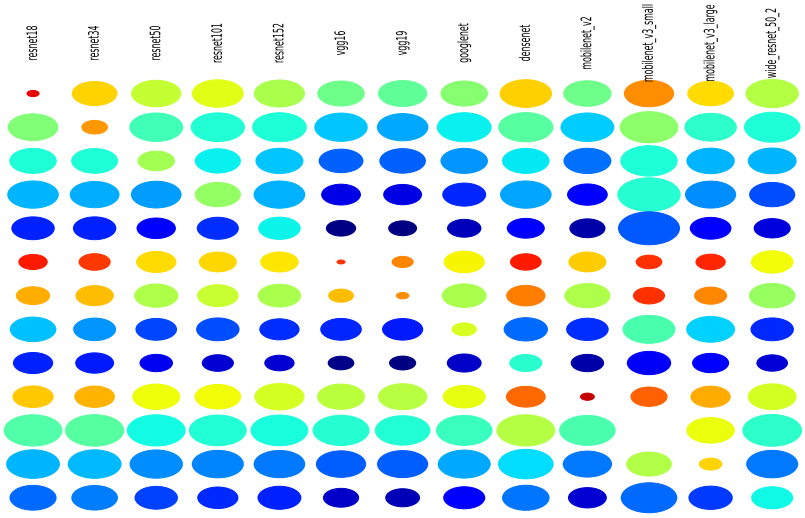


Figure 4.6: mean and std of predicted class distance from true class for FGSM and epsilon 0.1

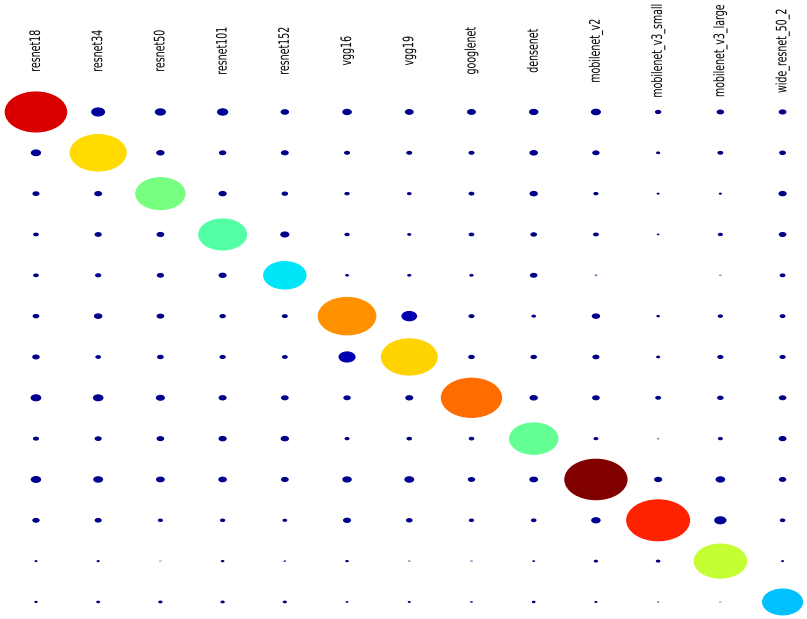


Figure 4.7: mean and std of predicted class distance from true class for FGV and epsilon 0.01

Above we show the visualization of std and avg for PGD with epsilon 0.3, FGSM with epsilon 0.1 and FGV with epsilon 0.01. The visualizations for other values of epsilon are added to the Appendix A. From the figures we observe that with FGV the std was very low expect for the diagonals which represent source and target network being same. The color of the these circles are blue indicating avg to be smaller. We intepret this as examples created on FGV when predicted on other networks, were always closer to the true class. For the other attacks, PGD and FGSM we see that standard deviation is more and varies between pair of source and target network. It means that predictions with these networks varied and were not always close to the true class. We observe blue colored circle for mobilenet variants for both FGSM and PGD. This is consistent with our previous result where we find that mobilenet showed least portability. As we reduce the epsilon we see that portability of networks reduces significantly and the predictions gets closer to the true class. This is indicated by small blue colored circles in the figure.

4.5 Portability of Targeted Adversarial Examples

In the previous experiments we studied the portability of untargeted attacks. There has been very few research for portability of targeted attacks. In our work we include the study for targeted attacks for the ϵ values of 0.01, 0.1 and 0.3 with attacks PGD, FGSM and FGV similar to our other experiments. To do so first we need to define the target for the adversarial attack which can be any of the class label different from the true label. As mentioned earlier in 3.3, we introduce a distance metric for cost of misclassification and based on this define three ways of selecting target i,e 1. closest 2. farthest 3. median. We evaluate the targeted attacks in two ways, first one by defining portability as fraction of adversarial examples that were able to reach the target and other by defining portability as fraction of adversarial examples that were misclassified even if the predicted class was not same as the target. The second evaluation is necessary because it is quite difficult to reach the target especially when the target is far from the original class and as we will see from the results below the targeted attacks do not work at all when the target is farthest from the true label.

source	resnet18	resenet34	resnet50	resnet101	resnet152	vgg16	vgg19	googlenet	densenet	mobilenet v2	v3 small	v3 large	wide resnet
resnet18	9997	1139	822	748	696	410	497	754	840	351	269	453	614
resenet34	1677	9988	1321	1206	1154	480	569	1031	1397	424	272	518	1075
resnet50	1200	1416	9996	2256	1978	468	616	1162	1692	406	290	518	2038
resnet101	1173	1416	2438	9982	3443	395	507	1220	1773	410	307	552	1917
resnet152	1129	1455	2388	3685	9985	417	536	1296	2122	395	306	618	2432
vgg16	485	422	407	357	387	9994	3115	504	420	324	221	328	373
vgg19	542	469	474	404	397	3161	9989	529	440	310	226	373	416
googlenet	612	594	647	598	586	363	416	9984	689	291	249	419	538
densenet	1033	1215	1494	1392	1461	436	551	1295	9999	371	319	670	1577
mobilenet v2	426	362	353	354	326	303	342	395	342	9990	255	390	289
v3 small	335	336	321	332	339	203	249	411	398	308	9999	860	322
v3 large	435	405	382	408	397	248	319	476	464	363	492	10000	336
wide resnet	1018	1218	2290	1943	2197	466	603	1168	2073	354	299	523	10000

Table 4.5: closest target with PGD and epsilon 0.3

The above table 4.5 shows the number of examples that were portable when we select target closest to true class and use a value of 0.3 for epsilon with PGD attack. The highest portability is colored green while red represents the pair of source and target network with lowest portability. We can see that while the source is able to generate adversarial examples, the portability of these examples to other network is low. The diagnoal values represents the number of adversarial examples where target was reached and is close to 10000 where as other values in the same row are very small. We add the other tables for different attack and epsilon values in the Appendix 3. We donot include the tables for lower value of epsilon since networks were not able to generate adversarial examples especially when target class was farthest or median.

In figures 4.8 , we present the heatmap which represent number of adversarial examples that could reach the target when closest target was selected. The other heatmaps for farthest A.8 and median A.7 target class in Appendix A. As can be seen from the heatmaps, with a value of 0.3 for epsilon and target being the closest class, while source network was able to produce adversarial example which reaches the target class but portability was insignificant and almost close to 0. We also observe that reducing the epsilon value the networks are no longer able to generate adversarial examples that reaches the target. Similarly when we select a target that is at median or farthest distance from true class, the networks are no longer able to generate adversarial examples that reaches the target. The portability for targeted adversarial examples is close to 0 and there are even cases where the adversarial example couldn't reach the target class on source network but was able to reach target on other network. These examples are very few and insignificant which can be treated as an outlier. The block structure we see for VGG variants in untargeted attack can also be seen for targeted attacks but the numbers are only significant

when we consider closest target.

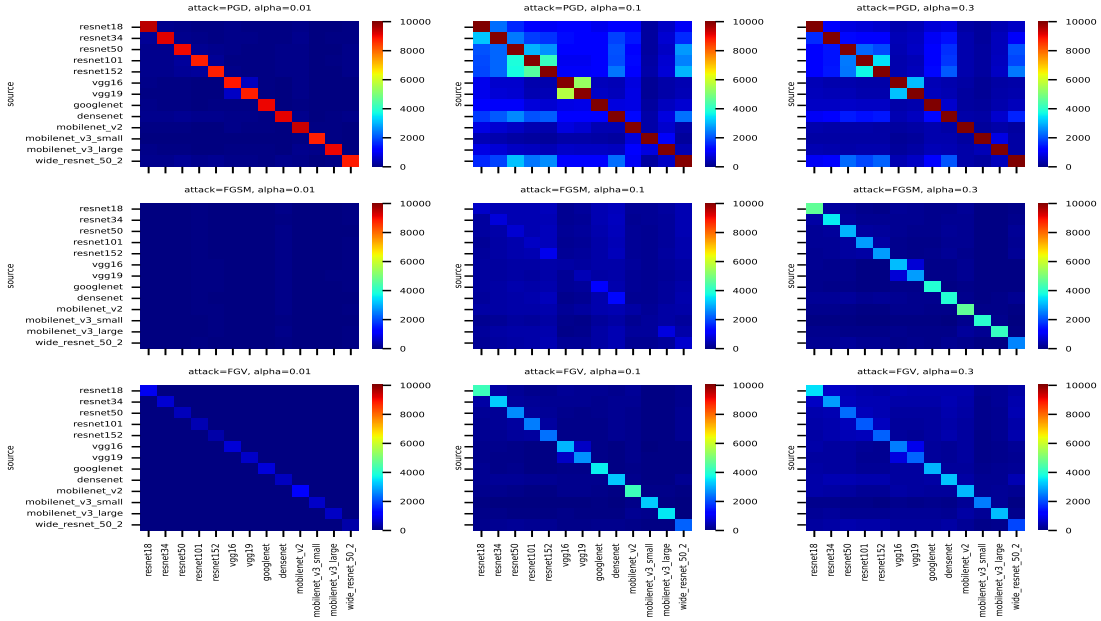


Figure 4.8: targeted attack with fixed alpha and closest target class

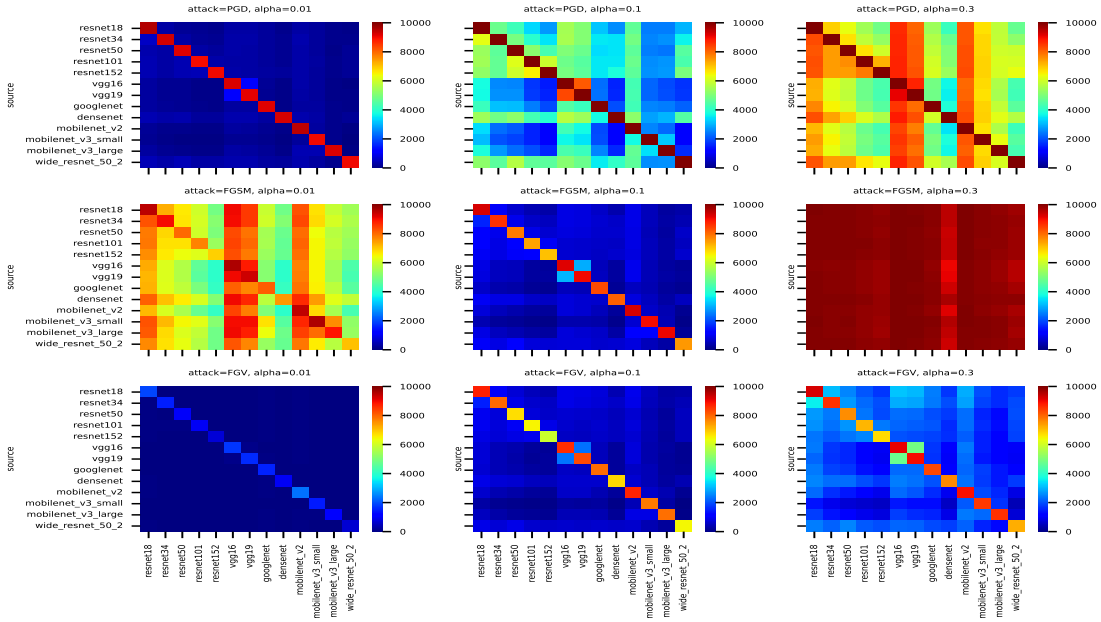


Figure 4.9: adversarial examples (including ones that doesn't reach target) with closest target class

Above we discussed portability in terms of whether the target was reached or not and we see that the selected networks were not able to produce adversarial examples that reach the specified target especially when the target was farthest or median to the true class. An interesting thing to study is whether the examples generated act as an adversarial example even if they do not reach specified target.

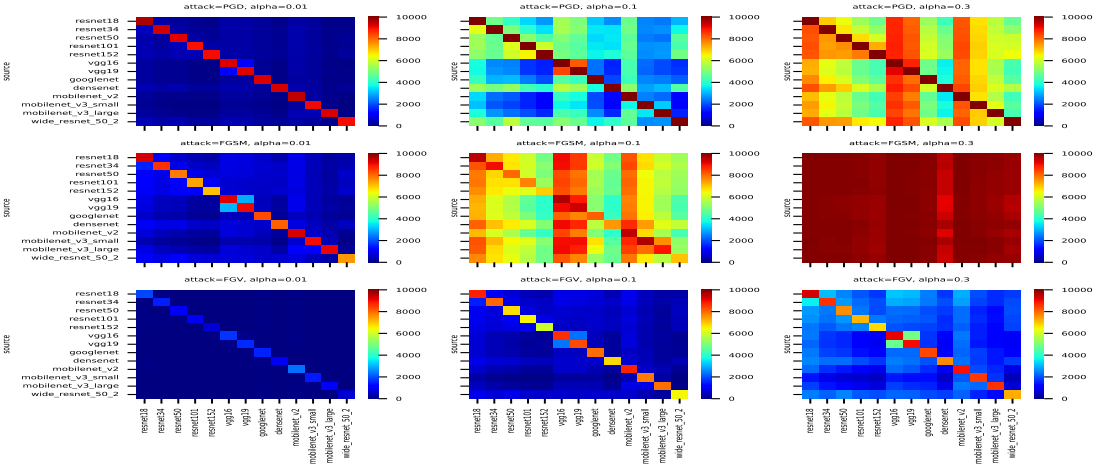


Figure 4.10: adversarial examples (including ones that doesn't reach target) with farthest target class

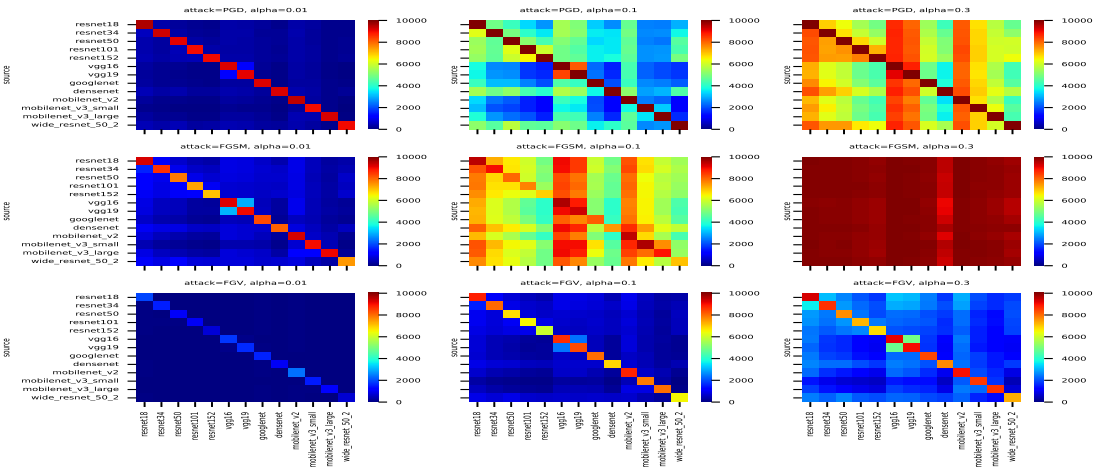


Figure 4.11: adversarial examples (including ones that doesn't reach target) with median target class

We can observe from the heatmaps 4.9, 4.10, 4.11 that even though with targeted attacks the network cannot reach the specified class, it can still produce adversarial examples which are portable. The results are similar to the untargeted attacks, and with decrease in epsilon the porta-

bility decreases significantly. Similarly, PGD and FGSM works much better compared to FGV which is not able to generate adversarial examples and the portability is significantly less.

From the above two evaluations of targeted attacks and the adversarial examples generated using these attacks, we see that while networks were not able to produce enough adversarial examples that reach selected target class these examples still get misclassified on both source and target network. Hence it would be interesting to see how far these predictions are from the target class. To study the distance of predictions from the target class we consider those images where network was not able to misclassify the image as target class but the prediction was still different that true class. We represent the distance between the prediction and target using the same distance metric we used for untargeted attacks 4.4. Although here, a low average distance is desirable since we want the predictions to be closer to the target. This is opposite to the untargeted attacks where we want the distance of predictions to be far from the true class.

source	resnet18	resnet34	resnet50	resnet101	resnet152	vgg16	vgg19	googlenet	densenet	mobilenet_v2	mobilenet_v3_small	mobilenet_v3_large	wide_resnet_50_2
resnet18	0.00, 0.28	6.09, 4.93	5.85, 4.67	5.31, 4.55	5.15, 4.56	8.42, 4.83	7.69, 4.92	5.26, 4.38	4.36, 3.98	8.15, 4.99	6.95, 4.73	5.92, 4.57	4.99, 4.34
resnet34	6.75, 5.19	0.02, 0.45	5.64, 4.76	5.12, 4.62	4.98, 4.60	8.37, 4.86	7.68, 4.93	5.20, 4.43	4.20, 4.07	8.08, 5.02	6.96, 4.72	5.84, 4.55	4.93, 4.51
resnet50	7.22, 5.09	6.30, 5.07	0.01, 0.27	4.90, 4.87	4.70, 4.68	8.43, 4.86	7.72, 4.97	5.19, 4.50	4.16, 4.14	8.19, 4.99	6.96, 4.73	5.95, 4.58	4.58, 4.64
resnet101	7.24, 5.06	6.26, 5.05	5.26, 4.96	0.02, 0.58	4.18, 4.86	8.54, 4.84	7.85, 4.94	5.25, 4.51	4.11, 4.15	8.23, 5.03	6.99, 4.75	6.05, 4.65	4.69, 4.63
resnet152	7.24, 5.06	6.23, 5.03	5.11, 4.84	4.34, 5.00	0.02, 0.44	8.51, 4.81	7.84, 4.95	5.25, 4.58	3.94, 4.13	8.20, 4.99	6.97, 4.73	5.97, 4.65	4.42, 4.63
vgg16	7.07, 4.84	6.32, 4.75	5.88, 4.55	5.30, 4.46	5.10, 4.42	0.01, 0.32	5.72, 5.44	5.16, 4.29	4.43, 3.91	8.04, 5.00	6.99, 4.71	5.92, 4.54	4.94, 4.23
vgg19	6.98, 4.87	6.22, 4.75	5.75, 4.50	5.21, 4.40	4.99, 4.34	5.96, 5.54	0.01, 0.43	5.09, 4.23	4.31, 3.79	7.99, 4.98	6.97, 4.72	5.80, 4.51	4.83, 4.20
googlenet	7.20, 4.83	6.27, 4.76	5.75, 4.52	5.23, 4.50	4.98, 4.43	8.33, 4.78	7.68, 4.84	0.02, 0.52	4.21, 3.78	8.10, 4.89	6.99, 4.69	5.92, 4.52	4.93, 4.29
densenet	7.43, 5.09	6.42, 5.02	5.67, 4.83	5.17, 4.70	4.87, 4.61	8.55, 4.82	7.81, 4.91	5.17, 4.51	0.00, 0.15	8.21, 4.97	7.00, 4.75	6.04, 4.67	4.76, 4.58
mobilenet_v2	7.04, 4.80	6.41, 4.76	5.90, 4.52	5.30, 4.47	5.11, 4.39	8.25, 4.78	7.71, 4.90	5.18, 4.23	4.36, 3.78	0.01, 0.41	6.92, 4.70	5.82, 4.49	4.96, 4.24
mobilenet_v3_small	7.31, 4.77	6.60, 4.77	6.05, 4.59	5.39, 4.44	5.27, 4.51	8.53, 4.71	8.06, 4.78	5.27, 4.27	4.56, 4.00	7.99, 4.87	0.00, 0.16	5.84, 4.66	5.18, 4.38
mobilenet_v3_large	7.09, 4.82	6.33, 4.74	5.88, 4.53	5.23, 4.42	5.08, 4.41	8.31, 4.75	7.75, 4.81	5.13, 4.21	4.37, 3.88	7.93, 4.96	6.68, 4.77	0.00, 0.00	5.06, 4.36
wide_resnet_50_2	7.39, 5.05	6.46, 5.03	5.26, 4.92	5.09, 4.84	4.69, 4.72	8.43, 4.84	7.77, 4.95	5.31, 4.55	4.11, 4.22	8.25, 4.94	7.02, 4.75	6.10, 4.65	0.00, 0.00

Table 4.6: std and mean for PGD and epsilon 0.3 with target closest from true class

The table 4.6 shows the avg and std for pgd with epsilon 0.3 and target class being closest from the true class. We present the tables for other values of epsilon and target choice in the Appendix A Below we show visualization of the avg and std with circle radius as std and color based on avg.

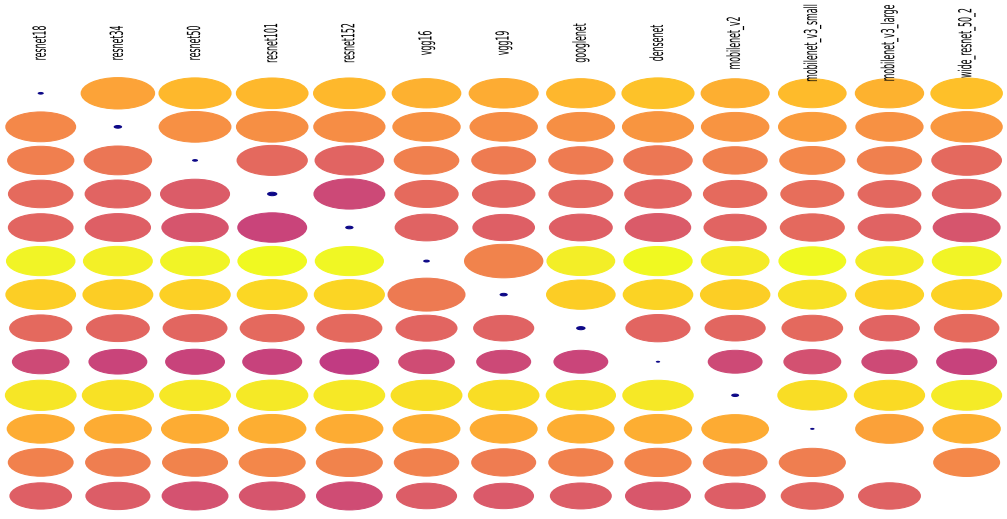


Figure 4.12: std and avg for closest target and PGD with epsilon 0.3

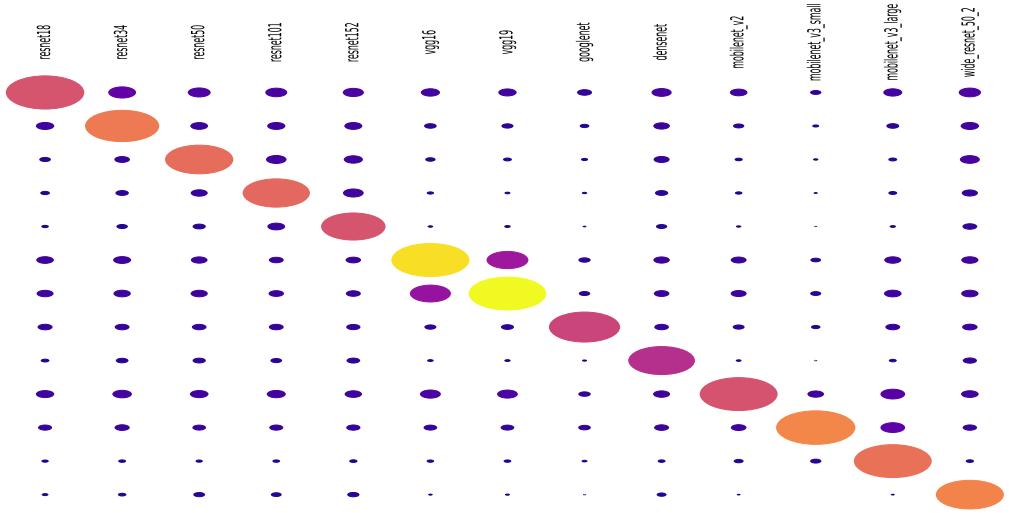


Figure 4.13: std and avg for closest target and FGSM with epsilon 0.1

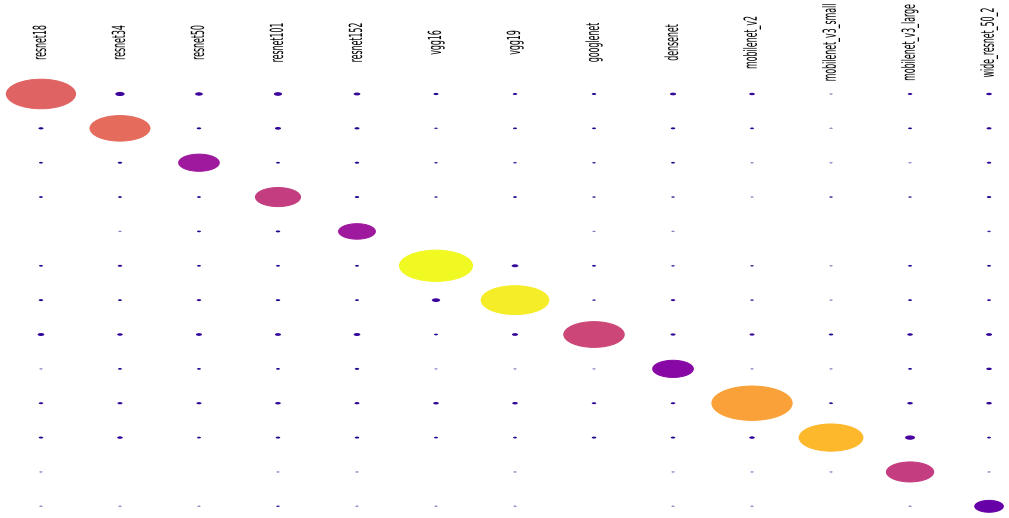


Figure 4.14: std and avg for closest target and FGV with epsilon 0.01

The other visualization are added to Appendix A. From the visualizations above we see that with FGV and ϵ 0.01, the protability is insignificant as the circles are quite small and always blue. This suggests the prediction was always close to the true class. Even for the diagonals, we observe the source networks were not able to produce adversarial example since the color of the circle maps to the lower value of jet color scheme. For PGD we see that while network was able to generate adversarial image they were not portable and the predictions on other network varied between images, as the std represented by radius of circle is more. The color of the circle is maps to lower value in jet scale and hence the predictions were on average closer to true class.

Chapter 5

Discussion

Our experiments shows some interesting results. In this section we would like further discuss and consolidate our results as well as give some ideas on what could be changed in the experiments.

Before discussing the result, we would like to discuss another approach for computing the distance between two classes. In our work we introduce a distance metric that uses the information from ImageNet hierarchy. We realize that while this uses already know information, it doesn't make use of the learned representation for each of the classes by the networks we select. An approach to do so is measuring similarity between feature space representation of two classes. Mean Activation Vector which is one of the way to do it is discussed in the next section.

5.1 Mean Activation Vector

In the previous section we describe our proposed distance metric and how we use this distance metric to select our target for adversarial attack. In this section we describe another approach that can be also used to compute distance or cost of misclassification. This approach uses mean activation vector representation for each of the output class, which is 1000 in our case.

The mean activation vector is calculating by taking the feature space from the last layer. The feature space is of dimension 1000, since it contains nodes corresponding to each class. The feature space is computed and average across all the images of same class. At the end, we get MAV(c) which is the mean activation vector for class c. The distance can then be described as,

$$d(c1, c2) = D(MAV(c1), MAV(c2)) \tag{5.1}$$

where c1, c2 are two different class d(c1, c2) is the distance or cost of misclassifying c1 as c2 MAV(c1) and MAV(c2) is the mean activation vector of c1 and c2 D is another distance function to calculate distance between 1 dimensional vectors

Any distance function that can compute distance between vectors can be chosen as D. One of the most common used function is the cosine similarity.

5.2 Results

In this section we would like to highlight some of the interesting results we observe.

- We observe that perturbations made by FGV for generating adversarial examples as compared to PGD and FGSM 4.2. This is because FGV uses raw gradients for updating the image. Given that FGV produces much less perturbation, it was expected that portability

of examples created by FGV would not be high and from our experiments we see that FGV performs the worst when it comes to portability both for untargeted and targeted attacks.

- We also see the portability was quite high close to 90% in case of untargeted attacks, which verifies the claim made by previous works that adversarial examples are portable to other networks. But we also observe that it is depended on the setting of attack. As we start reducing our perturbation constant, the portability decrease significantly. 4.4. The reason behind this could be that since the perturbation are not strong enough with low ϵ, the predictions are not far away from the true class. And a stronger network is able to predict the example correctly with small perturbation. This is supported by experiments where we computed the mean distance between predicted class and true class and see that with reducing ϵ the mean distance also reduces.

- We expected networks with similar topology to have more portability and this is supported by our experiment results. We present a table which consists of source and target network pair for which we observe maximum and minimum portability. For the variants of resnet, maximum portability is with another resnet variant network. Similarly we see the vgg variants have a higher portability among themselves. This can also be seen in the heatmap 4.3 as a block structure. One suprising observation is that variants of mobilenet do not show a higher portability among themselves and this needs to further explored.

- As expected for targeted attacks, we see a very poor portability among different network pairs. In case of the farthest and median target, the source network was not able to produce adversarial example which reaches this target. This should be further looked into with more advanced targeted attacks. By evaluating the targeted attacks in an untargeted manner, that is instead of looking at examples which can reach the specified target, we look at all the examples which are predicted different from the true class, we observe that while the target class was not reached, the network was able to generated adversarial image which are portable. We need to further look into the learned parameters from different networks to understand why was portability so low for targeted attacks.

Chapter 6

Conclusion

In this thesis, I present a study on portability of adversarial examples created on 13 different popular neural network models for image classifcation. The adversarial examples were generated with different perturbation constant ϵ 0.01, 0.1 and 0.3 with three adversarial attacks, FGSM, PGD and FGV in both targeted and untargeted setting. FGV produces lesser perturbations compared to FGSM and PGD but the adversarial examples created on FGV were not portable to other networks even with a pertubation constant of 0.3. We propose a distance metric that leverages the heirarchy of ImageNet. The cost of misclassifying is stored as a matrix which can be used for further work in the area. We represent our results of portability both statistically and with visualization which helps us understand the trend easily. For untargeted attack we observe that with and ϵ of 0.3 the portability was quite high ranging from 70-95%. But we observe that reducing the ϵ significantly affects our results and it leads to decrease in portability of the adversarial examples. To get a further insight into portability we also propose two metrics namely the avg of distance and standard deviation which tells us how far the predictions made were from true label or target label. We observe a similar trend as in the case with portability. Pair of source and target network which has higher portability also predicts a class with was farther from the true class which can be seen through a higher mean distance. Before the work we proposed a hypothesis that portability should be higher on pair of networks which are structurally similar. We observe VGG16 and VGG19 to have best portability, also variants of resnet show higher portability among themselves as compared to other network.

Furthermore, we included targeted attacks in our work which has very less previous research. We propose three approach to choose the target for generating targeted adversarial examples. We select target which is farthest, closest and at median distance to original class. We observe a very poor portability for all the specification of targeted attacks. With a closer target is easier to generate adversarial example, we don't see any significant result with the targeted attack. With epsilon of 0.1 and 0.01, the networks were not able to even produce adversarial example which can reach the farthest and median target. We evaluated the targeted attack study, by studying if the image doesn't read the target, does it still gets predicted as a class different from the original class. We observe that it was indeed the case. While the target class was not reached the predictions were still different from the original class. To understand how far the prediction were from the supposed target, we calculate the average and standard deviation of the distance of predicted to target class. We observe that for farthest and median target class, the prediction was not close to the target at all.

6.1 Future Work

We study portability of adversarial attacks for both targeted and untargeted attack but there are certain gaps in the work that can be further researched.

- We only use a tree based distance metric for computing the cost of miss classification, a better approach like using mean activation vector and cosine similarity can be used. It would be interesting to see if a different distance metric provides consistent report. Mean activation vector would take into account the learned representation for the classes rather than just considering known tree. We add a brief discussion on mean activation vector in Chapter 5
- More neural networks can be included to our study, we had a limitation of number of tasks we can run on the gpu and some of the neural networks in our study are no longer state of the art model. This work can be further extended to include the state of the art models.
- Another possible area of improvement is to consider a different subset of dataset. In our work we use the training images for generation of adversarial examples which is not the best approach since we also use pretrained models that were trained on this subset. This is a possible flaw of our work. We plan as a next step to use validate set from ImageNet data to generate adversarial examples and evaluate portability on these images.
- We observe some interesting trends in the portability, but an indepth study of why the portability of certain networks are higher needs to be done.

Appendix A

Attachements

source	resnet18	resnet34	resnet50	resnet101	resnet152	vgg16	vgg19	googlenet	densenet	mobilenet v2	v3 small	v3 large	wide resnet
resnet18	10000	9485	8787	8362	7589	9700	9476	8656	7784	9312	7986	8359	7699
resnet34	9685	10000	8900	8435	7843	9540	9324	8498	8021	9156	7861	8128	7973
resnet50	9466	9098	10000	8800	8157	9438	9218	8367	8005	9064	7671	7735	8403
resnet101	9409	9127	9274	10000	9085	9378	9120	8429	7974	9070	7818	8032	8340
resnet152	9290	9074	9138	9349	10000	9324	9099	8323	8216	9019	7763	7939	8560
vgg16	9540	9101	8798	8077	7677	10000	9991	8844	8087	9549	8129	8796	8205
vgg19	9468	9002	8667	8025	7608	9986	10000	8804	7917	9428	7910	8612	7954
googlenet	9087	8346	8031	7249	6649	9375	9151	10000	6523	8927	7751	7685	6763
densenet	9473	9149	8990	8592	8193	9436	9241	8577	10000	9101	7810	8170	8463
mobilenet v2	8911	8151	7527	6570	5876	9434	9146	7058	5541	10000	8148	8183	5930
v3 small	8837	8088	7401	6494	5909	9333	9175	7032	5529	9240	10000	8815	6141
v3 large	9168	8517	7841	7211	6582	9457	9218	8022	6704	9372	9045	10000	6640
wide resnet	9334	8973	9074	8620	8450	9386	9185	8354	8469	8895	7702	7803	10000

Table A.1: Untargeted attack with PGD and alpha=0.3

source	resnet18	resenet34	resnet50	resnet101	resnet152	vgg16	vgg19	googlenet	densenet	mobilenet v2	v3 small	v3 large	wide resnet
resnet18	10000	9382	8839	7954	7189	8828	8571	7218	7455	8567	4594	6208	7514
resenet34	9430	10000	8875	8081	7591	8327	8094	7012	7740	8214	4266	5864	7777
resnet50	8586	8568	10000	9179	8753	7968	7666	6513	8000	7622	3967	4980	9036
resnet101	8506	8469	9588	10000	9594	7367	7227	6664	7748	7729	4122	5570	8647
resnet152	8136	8189	9401	9641	10000	7430	7195	6563	8022	7443	4112	5485	8966
vgg16	8587	7832	8004	6384	6084	10000	9990	6939	6893	8811	4631	6205	7248
vgg19	8452	7803	7916	6493	6006	9978	10000	6946	6868	8705	4378	6091	7186
googlenet	6686	5683	5701	4730	4148	6754	6425	10000	4353	6347	3727	4311	4235
densenet	8388	8464	9008	8250	8244	8036	7792	6969	10000	7917	4449	6139	8510
mobilenet v2	6692	5335	5054	4041	3086	7336	6990	4274	3370	10000	4588	5636	3149
v3 small	5312	4099	3579	3065	2364	5634	5439	3671	2485	6633	10000	6715	2452
v3 large	6488	5470	4909	4451	3676	6675	6460	4912	3994	7822	6387	10000	3695
wide resnet	8176	8262	9537	8906	9041	7842	7578	6557	8634	7239	4049	5297	10000

Table A.2: Untargeted attack with PGD and alpha=0.1

source	resnet18	resenet34	resnet50	resnet101	resnet152	vgg16	vgg19	googlenet	densenet	mobilenet v2	v3 small	v3 large	wide resnet
resnet18	10000	1513	961	653	465	1200	1053	667	522	1195	539	360	404
resenet34	2024	10000	1204	808	664	1010	969	673	701	1103	478	345	545
resnet50	1264	1075	10000	1351	978	955	900	594	772	1033	404	259	849
resnet101	1168	928	1728	10000	1476	747	706	601	709	943	416	288	743
resnet152	1174	1027	1642	1980	10000	826	777	600	844	991	431	300	961
vgg16	842	463	463	248	233	10000	4556	457	264	1015	421	241	218
vgg19	765	446	453	244	213	4594	10000	452	264	921	410	239	181
googlenet	898	553	454	327	267	708	680	10000	290	747	471	307	210
densenet	1284	1217	1427	998	915	990	939	713	10000	1063	499	369	924
mobilenet v2	826	490	419	303	219	922	822	423	258	10000	655	476	153
v3 small	318	164	148	121	73	341	309	293	108	619	10000	413	56
v3 large	774	463	404	271	247	765	680	506	311	1408	1081	10000	173
wide resnet	1211	1001	1792	1237	1216	972	933	621	978	971	409	323	10000

Table A.3: Untargeted attack with PGD and alpha=0.01

source	resnet18	resenet34	resnet50	resnet101	resnet152	vgg16	vgg19	googlenet	densenet	mobilenet v2	v3 small	v3 large	wide resnet
resnet18	10000	9936	9929	9896	9830	9977	9959	9887	9478	9962	9876	9836	9741
resenet34	9961	10000	9933	9910	9859	9978	9964	9909	9526	9961	9905	9864	9785
resnet50	9949	9945	10000	9905	9834	9966	9949	9881	9391	9959	9873	9840	9746
resnet101	9949	9940	9929	10000	9855	9973	9951	9901	9407	9964	9880	9843	9754
resnet152	9950	9935	9925	9917	10000	9973	9955	9901	9416	9962	9876	9830	9733
vgg16	9918	9883	9882	9793	9731	10000	9964	9830	9072	9956	9851	9761	9526
vgg19	9918	9878	9886	9821	9744	9976	10000	9857	9072	9951	9862	9741	9499
googlenet	9930	9902	9905	9836	9758	9965	9964	10000	9358	9957	9873	9815	9685
densenet	9963	9929	9922	9905	9813	9967	9969	9890	10000	9964	9904	9841	9836
mobilenet v2	9944	9925	9913	9845	9777	9985	9952	9861	9195	10000	9879	9789	9615
v3 small	9962	9913	9922	9830	9743	9974	9981	9929	9598	9975	10000	9906	9771
v3 large	9947	9905	9901	9822	9698	9983	9969	9913	9430	9962	9935	10000	9676
wide resnet	9943	9928	9928	9879	9796	9970	9946	9891	9353	9956	9885	9822	10000

Table A.4: Untargeted attack with FGSM and alpha=0.3

source	resnet18	resenet34	resnet50	resnet101	resnet152	vgg16	vgg19	googlenet	densenet	mobilenet v2	v3 small	v3 large	wide resnet
resnet18	10000	7590	7047	6407	5506	9088	8703	6539	5615	8394	6893	6205	5982
resenet34	8387	10000	7054	6448	5556	8996	8583	6316	5607	8473	6891	6242	6144
resnet50	7897	7104	10000	6351	5390	8406	7917	5777	5268	8061	6435	5907	5793
resnet101	7963	6878	6907	10000	5744	8440	8053	5761	5064	8085	6557	5891	5621
resnet152	7725	6816	6600	6506	10000	8366	7900	5620	5120	7888	6523	5849	5559
vgg16	7454	6388	5889	4946	4358	10000	8890	5480	4471	7835	6582	5653	4796
vgg19	7320	6226	5857	4939	4341	9062	10000	5419	4397	7788	6610	5585	4672
googlenet	7501	6592	6156	5435	4682	8273	7867	10000	4772	8093	6733	6093	5107
densenet	8189	7192	6819	6364	5422	9024	8695	5984	10000	8648	7452	6424	6120
mobilenet v2	7451	6471	5959	5191	4531	8489	7883	5512	4584	10000	6901	5899	4827
v3 small	8376	7341	6696	6543	5409	9058	8998	6993	4975	8798	10000	7851	5669
v3 large	8257	6992	6439	6077	5189	9034	8787	6543	5243	8558	8054	10000	5653
wide resnet	7703	6792	6440	5804	5051	8230	7725	5488	5079	7889	6620	5795	10000

Table A.5: Untargeted attack with FGSM and alpha=0.1

source	resnet18	resenet34	resnet50	resnet101	resnet152	vgg16	vgg19	googlenet	densenet	mobilenet v2	v3 small	v3 large	wide resnet
resnet18	10000	2069	1379	996	760	1495	1416	1201	880	1651	775	553	628
resenet34	2484	10000	1623	1185	1021	1343	1306	1205	1165	1529	701	589	809
resnet50	1679	1552	10000	1675	1323	1221	1155	1021	1188	1301	606	448	1186
resnet101	1545	1348	1907	10000	1840	983	950	926	1099	1201	588	462	1020
resnet152	1424	1262	1790	2051	10000	981	947	876	1130	1164	589	473	1152
vgg16	1455	984	910	542	477	10000	4451	876	568	1518	661	472	406
vgg19	1372	927	868	539	473	4458	10000	871	581	1498	627	471	415
googlenet	1155	786	659	450	398	880	873	10000	465	963	658	391	281
densenet	1466	1368	1506	1184	1106	1092	1059	995	10000	1213	661	507	1008
mobilenet v2	1373	984	839	595	467	1315	1245	908	571	10000	1004	781	363
v3 small	742	387	312	230	185	609	567	585	233	1175	10000	920	140
v3 large	1240	865	740	566	493	1109	1027	912	598	1909	1680	10000	355
wide resnet	1312	1127	1565	1238	1161	1020	960	847	1049	1053	585	431	10000

Table A.6: Untargeted attack with FGSM and alpha=0.01

source	resnet18	resenet34	resnet50	resnet101	resnet152	vgg16	vgg19	googlenet	densenet	mobilenet v2	v3 small	v3 large	wide resnet
resnet18	10000	4311	3617	3022	2531	3982	3782	3394	2681	3789	2411	2221	2513
resenet34	4431	10000	3541	3006	2695	3481	3388	3081	2775	3441	2158	2084	2648
resnet50	3311	3097	10000	3165	2721	2764	2696	2541	2513	2776	1716	1580	2695
resnet101	3074	2872	3399	10000	3146	2450	2426	2423	2332	2706	1750	1580	2467
resnet152	2829	2694	3076	3238	10000	2304	2272	2218	2315	2493	1677	1492	2445
vgg16	3077	2355	2218	1640	1430	10000	5858	2274	1592	3012	1742	1556	1465
vgg19	2838	2243	2097	1541	1391	5729	10000	2214	1548	2786	1654	1483	1357
googlenet	3245	2520	2399	1986	1750	2910	2855	10000	1903	2822	2148	1857	1709
densenet	3037	2848	2970	2618	2400	2759	2677	2465	10000	2794	1957	1714	2472
mobilenet v2	3016	2408	2138	1701	1401	2904	2780	2304	1567	10000	2378	2188	1319
v3 small	1921	1256	1030	941	798	1786	1697	1635	821	2653	10000	2406	638
v3 large	2335	1818	1466	1335	1130	2231	2023	1923	1246	3049	2773	10000	1024
wide resnet	2789	2573	2949	2594	2433	2468	2386	2179	2248	2427	1676	1487	10000

Table A.7: Untargeted attack with FGV and alpha=0.3

source	resnet18	resenet34	resnet50	resnet101	resnet152	vgg16	vgg19	googlenet	densenet	mobilenet v2	v3 small	v3 large	wide resnet
resnet18	10000	2038	1394	1078	814	1563	1450	1346	943	1649	800	577	697
resenet34	2299	10000	1496	1205	1030	1323	1276	1256	1091	1488	671	547	817
resnet50	1471	1290	10000	1364	1127	1014	996	946	1003	1112	528	380	975
resnet101	1361	1180	1630	10000	1554	861	841	931	934	1068	519	408	916
resnet152	1169	1073	1370	1617	10000	798	803	857	941	971	483	367	886
vgg16	1247	866	736	475	431	10000	3761	810	494	1292	537	419	340
vgg19	1141	791	686	450	407	3662	10000	808	507	1198	525	373	319
googlenet	1274	890	753	545	490	933	946	10000	553	1027	625	393	375
densenet	1288	1166	1242	1010	964	962	937	928	10000	1021	575	427	826
mobilenet v2	1251	877	710	533	437	1136	1071	832	493	10000	921	713	315
v3 small	494	262	190	178	136	419	377	429	159	821	10000	662	85
v3 large	847	571	438	355	320	705	672	661	402	1326	1209	10000	204
wide resnet	1140	1020	1288	1051	1024	881	837	822	893	921	477	350	10000

Table A.8: Untargeted attack with FGV and alpha=0.1

source	resnet18	resenet34	resnet50	resnet101	resnet152	vgg16	vgg19	googlenet	densenet	mobilenet v2	v3 small	v3 large	wide resnet
resnet18	10000	95	50	27	29	49	55	112	35	94	54	16	13
resenet34	154	10000	55	31	38	63	49	99	48	88	46	17	16
resnet50	101	65	10000	41	47	56	51	73	54	74	28	11	21
resnet101	88	62	68	10000	63	41	40	70	53	76	29	16	21
resnet152	66	61	52	53	10000	42	36	61	54	60	32	12	19
vgg16	70	33	30	20	18	10000	235	65	21	82	40	14	8
vgg19	62	37	28	15	15	231	10000	70	23	93	35	12	10
googlenet	65	36	27	19	19	34	43	10000	28	60	32	8	9
densenet	74	58	57	29	45	38	45	70	10000	72	34	12	15
mobilenet v2	78	46	27	22	13	65	43	57	22	10000	67	25	9
v3 small	33	13	9	6	4	14	16	26	6	51	10000	20	4
v3 large	52	28	14	18	11	29	37	46	23	87	116	10000	4
wide resnet	56	56	64	41	41	44	38	63	57	61	35	14	10000

Table A.9: Untargeted attack with FGV and alpha=0.01

source	resnet18	resnet34	resnet50	resnet101	resnet152	vgg16	vgg19	googlenet	densenet	mobilenet_v2	mobilenet_v3_small	mobilenet_v3_large	wide_resnet_50_2
resnet18	11.26, 3.72	10.29, 4.57	8.99, 5.07	8.59, 5.45	7.65, 5.73	10.68, 4.34	10.07, 4.62	8.96, 5.13	7.96, 5.67	9.90, 4.83	8.01, 5.40	8.62, 5.35	7.77, 5.68
resnet34	10.44, 4.37	10.76, 3.85	8.92, 5.00	8.45, 5.34	7.73, 5.55	10.14, 4.53	9.63, 4.75	8.49, 5.17	7.93, 5.46	9.55, 4.98	7.73, 5.43	8.12, 5.39	7.83, 5.47
resnet50	9.99, 4.60	9.31, 4.87	10.55, 3.81	8.71, 5.09	8.01, 5.41	9.87, 4.58	9.36, 4.78	8.22, 5.19	7.76, 5.39	9.41, 5.05	7.53, 5.48	7.55, 5.50	8.26, 5.27
resnet101	9.75, 4.67	9.26, 4.89	9.18, 4.72	10.20, 3.95	8.96, 4.86	9.79, 4.69	9.27, 4.88	8.18, 5.12	7.64, 5.40	9.30, 5.03	7.60, 5.40	7.82, 5.34	8.05, 5.31
resnet152	9.55, 4.75	9.11, 4.91	8.85, 4.81	9.21, 4.76	10.10, 3.95	9.63, 4.70	9.16, 4.91	8.04, 5.18	7.78, 5.25	9.28, 5.09	7.54, 5.42	7.63, 5.38	8.19, 5.16
vgg16	10.56, 4.47	9.79, 4.91	9.24, 5.15	8.28, 5.53	7.83, 5.71	11.34, 3.51	11.39, 3.59	9.38, 5.03	8.54, 5.61	10.54, 4.44	8.16, 5.29	9.19, 4.99	8.64, 5.54
vgg19	10.42, 4.58	9.65, 4.96	9.09, 5.22	8.29, 5.61	7.89, 5.78	11.62, 3.69	11.54, 3.59	9.32, 5.05	8.42, 5.73	10.47, 4.67	7.87, 5.42	8.96, 5.14	8.35, 5.64
googlenet	9.40, 4.92	8.34, 5.31	7.71, 5.33	6.94, 5.65	6.39, 5.79	9.85, 4.69	9.31, 4.85	10.69, 3.89	6.10, 5.65	9.15, 5.11	7.59, 5.46	7.55, 5.51	6.42, 5.70
densenet	9.54, 4.68	8.83, 4.92	8.38, 4.91	7.95, 5.18	7.58, 5.35	9.68, 4.65	9.20, 4.82	7.95, 5.05	9.44, 4.26	9.31, 5.05	7.54, 5.45	7.78, 5.30	7.79, 5.22
mobilenet_v2	9.27, 5.00	8.27, 5.44	7.26, 5.52	6.31, 5.78	5.64, 5.85	10.05, 4.60	9.53, 4.88	6.75, 5.58	5.13, 5.64	11.41, 3.68	8.18, 5.34	8.32, 5.38	5.63, 5.76
mobilenet_v3_small	9.24, 5.11	8.11, 5.41	7.16, 5.58	6.23, 5.80	5.73, 5.89	9.85, 4.69	9.49, 4.84	6.85, 5.66	5.16, 5.67	9.60, 4.82	11.56, 3.61	9.26, 5.07	5.93, 5.85
mobilenet_v3_large	9.70, 4.84	8.76, 5.29	7.65, 5.46	7.05, 5.75	6.42, 5.84	10.11, 4.61	9.65, 4.86	8.05, 5.41	6.49, 5.74	9.95, 4.76	9.40, 4.84	11.13, 3.75	6.42, 5.80
wide_resnet_50_2	9.66, 4.75	9.04, 5.00	8.77, 4.81	8.38, 5.18	8.19, 5.24	9.80, 4.71	9.27, 4.86	8.11, 5.19	8.18, 5.18	9.17, 5.20	7.46, 5.43	7.55, 5.46	10.05, 3.99

Table A.10: Mean and Std for PGD and alpha=0.3

source	resnet18	resnet34	resnet50	resnet101	resnet152	vgg16	vgg19	googlenet	densenet	mobilenet_v2	mobilenet_v3_small	mobilenet_v3_large	wide_resnet_50_2
resnet18	9.50, 4.27	8.78, 4.76	8.14, 5.02	7.21, 5.32	6.43, 5.47	8.23, 5.01	7.88, 5.11	6.44, 5.42	6.75, 5.44	7.92, 5.12	3.98, 5.20	5.50, 5.46	6.82, 5.44
resnet34	8.44, 4.66	9.07, 4.29	7.84, 4.92	7.02, 5.18	6.54, 5.26	7.40, 5.09	7.12, 5.14	5.98, 5.30	6.70, 5.26	7.29, 5.18	3.61, 5.03	4.98, 5.30	6.81, 5.28
resnet50	7.31, 4.96	7.27, 4.96	8.74, 4.30	7.86, 4.75	7.43, 4.92	6.82, 5.13	6.52, 5.19	5.35, 5.20	6.69, 5.08	6.47, 5.20	3.28, 4.85	4.07, 5.08	7.75, 4.82
resnet101	7.17, 4.99	7.03, 4.98	8.11, 4.59	8.57, 4.39	8.14, 4.60	6.20, 5.20	6.11, 5.26	5.46, 5.18	6.38, 5.15	6.51, 5.21	3.45, 4.96	4.57, 5.18	7.22, 4.97
resnet152	6.73, 5.07	6.69, 5.02	7.78, 4.64	8.02, 4.56	8.43, 4.40	6.19, 5.17	5.98, 5.21	5.26, 5.16	6.48, 5.04	6.18, 5.21	3.41, 4.93	4.40, 5.08	7.43, 4.86
vgg16	8.12, 5.17	7.29, 5.39	7.49, 5.36	5.75, 5.55	5.46, 5.54	9.75, 4.21	9.77, 4.21	6.28, 5.50	6.35, 5.56	8.40, 5.01	4.03, 5.21	5.61, 5.54	6.79, 5.58
vgg19	8.01, 5.23	7.34, 5.45	7.46, 5.44	5.89, 5.57	5.40, 5.55	9.81, 4.24	9.79, 4.24	6.30, 5.52	6.37, 5.61	8.36, 5.10	3.82, 5.17	5.51, 5.53	6.80, 5.64
googlenet	5.94, 5.40	4.89, 5.31	4.89, 5.28	4.00, 5.11	3.44, 4.92	5.92, 5.34	5.62, 5.34	9.79, 4.06	3.65, 5.00	5.64, 5.44	3.23, 4.91	3.71, 5.06	3.57, 4.99
densenet	6.90, 5.02	6.82, 4.95	7.31, 4.82	6.56, 4.99	6.53, 4.97	6.67, 5.07	6.45, 5.10	5.52, 5.10	8.36, 4.49	6.53, 5.13	3.62, 4.95	4.87, 5.10	6.89, 5.00
mobilenet_v2	5.97, 5.46	4.62, 5.33	4.26, 5.20	3.39, 4.95	2.49, 4.43	6.61, 5.39	6.24, 5.44	3.58, 4.99	2.72, 4.54	9.86, 4.24	3.97, 5.18	4.93, 5.38	2.59, 4.52
mobilenet_v3_small	4.70, 5.39	3.48, 4.99	3.00, 4.75	2.53, 4.47	1.92, 4.07	4.95, 5.39	4.75, 5.35	3.07, 4.78	2.00, 4.07	5.99, 5.49	10.15, 4.15	6.07, 5.48	1.99, 4.10
mobilenet_v3_large	5.64, 5.41	4.62, 5.26	4.04, 5.06	3.62, 4.94	2.95, 4.67	5.84, 5.39	5.64, 5.40	4.09, 5.11	3.23, 4.79	6.94, 5.28	5.51, 5.36	9.30, 4.30	2.99, 4.71
wide_resnet_50_2	6.81, 5.07	6.85, 5.07	8.06, 4.65	7.44, 4.89	7.55, 4.84	6.57, 5.15	6.31, 5.17	5.30, 5.17	7.17, 4.97	6.02, 5.25	3.32, 4.84	4.24, 5.06	8.55, 4.41

Table A.11: Mean and Std for PGD and alpha=0.1

source	resnet18	resnet34	resnet50	resnet101	resnet152	vgg16	vgg19	googlenet	densenet	mobilenet_v2	mobilenet_v3_small	mobilenet_v3_large	wide_resnet_50_2
resnet18	8.39, 4.33	1.14, 3.15	0.69, 2.44	0.47, 2.06	0.33, 1.73	0.88, 2.77	0.79, 2.67	0.47, 2.02	0.37, 1.81	0.90, 2.82	0.42, 1.99	0.26, 1.54	0.28, 1.57
resnet34	1.53, 3.56	8.18, 4.34	0.87, 2.75	0.59, 2.32	0.47, 2.08	0.73, 2.52	0.72, 2.55	0.49, 2.10	0.50, 2.12	0.83, 2.72	0.38, 1.91	0.26, 1.58	0.39, 1.88
resnet50	0.92, 2.82	0.78, 2.63	7.92, 4.25	0.98, 2.93	0.69, 2.48	0.68, 2.44	0.65, 2.40	0.43, 1.98	0.54, 2.16	0.76, 2.57	0.31, 1.69	0.18, 1.29	0.59, 2.27
resnet101	0.87, 2.78	0.67, 2.43	1.24, 3.21	7.89, 4.32	1.03, 2.93	0.53, 2.14	0.50, 2.09	0.41, 1.92	0.50, 2.08	0.67, 2.41	0.32, 1.77	0.20, 1.36	0.50, 2.09
resnet152	0.84, 2.72	0.70, 2.45	1.15, 3.08	1.39, 3.35	7.73, 4.30	0.59, 2.27	0.57, 2.28	0.40, 1.85	0.58, 2.22	0.72, 2.51	0.33, 1.77	0.21, 1.37	0.64, 2.34
vgg16	0.64, 2.45	0.32, 1.72	0.33, 1.71	0.18, 1.27	0.15, 1.10	8.53, 4.27	3.69, 4.96	0.33, 1.72	0.19, 1.32	0.78, 2.68	0.33, 1.79	0.18, 1.30	0.15, 1.16
vgg19	0.58, 2.34	0.32, 1.71	0.31, 1.65	0.17, 1.25	0.15, 1.15	3.74, 4.97	8.64, 4.32	0.32, 1.71	0.19, 1.30	0.70, 2.54	0.33, 1.81	0.17, 1.27	0.12, 1.02
googlenet	0.67, 2.48	0.39, 1.90	0.32, 1.70	0.24, 1.50	0.19, 1.32	0.54, 2.21	0.51, 2.18	9.03, 4.15	0.22, 1.44	0.58, 2.32	0.39, 1.93	0.23, 1.48	0.15, 1.15
densenet	0.90, 2.78	0.84, 2.68	0.98, 2.86	0.68, 2.42	0.61, 2.29	0.68, 2.38	0.66, 2.38	0.48, 2.05	7.63, 4.34	0.73, 2.49	0.38, 1.89	0.25, 1.51	0.62, 2.32
mobilenet_v2	0.61, 2.32	0.34, 1.75	0.31, 1.68	0.22, 1.44	0.15, 1.12	0.69, 2.49	0.63, 2.43	0.31, 1.70	0.18, 1.28	8.38, 4.26	0.51, 2.17	0.35, 1.80	0.11, 0.98
mobilenet_v3_small	0.23, 1.46	0.11, 1.00	0.10, 0.96	0.08, 0.85	0.05, 0.72	0.25, 1.48	0.23, 1.47	0.21, 1.37	0.08, 0.88	0.47, 2.06	8.98, 4.31	0.32, 1.74	0.04, 0.55
mobilenet_v3_large	0.56, 2.25	0.33, 1.77	0.28, 1.58	0.19, 1.31	0.18, 1.31	0.55, 2.20	0.51, 2.18	0.37, 1.83	0.22, 1.40	1.08, 3.08	0.82, 2.71	8.09, 4.29	0.11, 0.96
wide_resnet_50_2	0.89, 2.81	0.72, 2.53	1.28, 3.24	0.87, 2.75	0.85, 2.70	0.68, 2.41	0.69, 2.49	0.43, 1.97	0.68, 2.43	0.70, 2.50	0.32, 1.76	0.22, 1.39	8.17, 4.46

Table A.12: Mean and Std for PGD and alpha=0.01

source	resnet18	resnet34	resnet50	resnet101	resnet152	vgg16	vgg19	googlenet	densenet	mobilenet_v2	mobilenet_v3_small	mobilenet_v3_large	wide_resnet_50_2
resnet18	16.33, 4.73	11.84, 4.60	14.95, 5.17	12.36, 4.60	10.96, 4.13	12.69, 3.67	11.76, 3.65	11.03, 3.93	10.44, 4.78	13.66, 4.52	10.55, 3.44	10.58, 3.66	10.74, 4.13
resnet34	15.90, 5.04	11.98, 4.56	15.34, 5.09	12.37, 4.54	11.02, 4.18	12.54, 3.67	11.61, 3.63	11.10, 3.89	10.60, 4.78	14.28, 4.60	10.54, 3.42	10.52, 3.62	10.90, 4.07
resnet50	15.70, 5.19	11.90, 4.66	16.34, 4.64	12.38, 4.49	11.07, 4.32	12.39, 3.68	11.50, 3.73	10.97, 3.96	10.20, 4.89	14.88, 4.69	10.45, 3.42	10.45, 3.58	10.83, 4.13
resnet101	16.14, 4.94	11.77, 4.56	15.57, 5.00	12.87, 4.59	11.45, 4.62	12.32, 3.65	11.47, 3.70	11.02, 3.90	10.19, 4.88	15.01, 4.66	10.44, 3.42	10.54, 3.66	10.77, 4.14
resnet152	15.60, 5.23	12.16, 4.79	15.23, 5.14	12.30, 4.42	11.82, 4.72	12.35, 3.69	11.46, 3.70	11.05, 3.91	10.28, 4.85	14.69, 4.79	10.45, 3.41	10.48, 3.64	10.79, 4.15
vgg16	13.47, 5.65	11.86, 4.83	13.13, 5.25	11.73, 4.65	10.90, 4.30	11.95, 3.73	11.74, 3.67	10.77, 3.99	9.58, 5.13	12.65, 4.45	10.46, 3.53	10.39, 3.78	10.18, 4.41
vgg19	13.38, 5.62	11.98, 4.87	13.14, 5.24	11.60, 4.50	10.90, 4.32	11.99, 3.73	11.70, 3.66	10.80, 3.94	9.55, 5.11	12.84, 4.48	10.50, 3.48	10.33, 3.82	10.09, 4.41
googlenet	13.36, 5.56	12.05, 4.85	12.64, 4.85	11.35, 4.15	10.50, 4.11	12.43, 3.65	11.36, 3.63	11.14, 3.87	9.98, 4.78	13.67, 4.81	10.56, 3.49	10.59, 3.65	10.33, 4.13
densenet	15.38, 5.22	12.68, 5.09	14.74, 5.21	13.26, 4.81	10.65, 4.26	12.20, 3.64	11.17, 3.58	11.18, 4.01	11.00, 4.88	15.67, 4.56	10.82, 3.53	10.67, 3.70	11.08, 4.17
mobilenet_v2	12.71, 5.42	12.15, 4.94	13.90, 5.25	11.74, 4.56	11.11, 4.19	12.57, 3.68	11.94, 3.63	10.76, 3.91	9.82, 5.05	12.74, 4.18	10.57, 3.56	10.42, 3.76	10.41, 4.27
mobilenet_v3_small	10.79, 3.56	11.15, 4.23	11.16, 3.69	11.87, 4.83	10.77, 4.27	11.66, 3.60	12.11, 3.75	10.82, 3.65	11.10, 4.67	12.59, 4.48	11.79, 3.74	10.45, 3.59	10.76, 4.04
mobilenet_v3_large	13.22, 5.45	11.53, 4.54	12.97, 5.07	12.67, 5.00	10.88, 4.27	11.88, 3.73	11.79, 3.65	11.19, 3.84	10.65, 4.91	13.37, 4.53	10.45, 3.37	10.51, 3.51	10.52, 4.31
wide_resnet_50_2	15.29, 5.34	12.59, 5.00	15.05, 5.15	12.20, 4.48	11.05, 4.46	12.44, 3.65	11.63, 3.71	10.92, 3.91	10.14, 4.86	14.95, 4.76	10.53, 3.47	10.57, 3.72	10.89, 4.09

Table A.13: Mean and Std for FGSM and alpha=0.3

source	resnet18	resnet34	resnet50	resnet101	resnet152	vgg16	vgg19	googlenet	densenet	mobilenet_v2	mobilenet_v3_small	mobilenet_v3_large	wide_resnet_50_2
resnet18	9.67, 4.60	6.94, 5.43	6.14, 5.36	5.56, 5.45	4.58, 5.27	9.49, 4.96	8.43, 5.07	5.64, 5.34	4.59, 5.20	8.23, 5.22	6.54, 5.61	5.56, 5.50	5.06, 5.35
resnet34	8.17, 5.32	8.58, 4.90	6.15, 5.35	5.51, 5.41	4.59, 5.27	9.28, 5.02	8.32, 5.16	5.36, 5.26	4.54, 5.17	8.41, 5.21	6.59, 5.62	5.59, 5.50	5.19, 5.34
resnet50	7.50, 5.42	6.41, 5.50	7.23, 5.14	5.41, 5.43	4.38, 5.18	8.10, 5.20	7.31, 5.29	4.84, 5.23	4.19, 5.05	7.84, 5.37	6.04, 5.61	5.30, 5.49	4.81, 5.27
resnet101	7.74, 5.46	6.18, 5.51	5.96, 5.33	7.10, 5.33	4.67, 5.24	8.14, 5.15	7.52, 5.23	4.87, 5.27	4.01, 5.02	7.86, 5.34	6.17, 5.59	5.25, 5.46	4.64, 5.22
resnet152	7.30, 5.44	6.14, 5.52	5.68, 5.37	5.55, 5.45	5.99, 5.25	8.03, 5.16	7.28, 5.27	4.68, 5.20	4.01, 4.98	7.62, 5.42	6.10, 5.59	5.16, 5.45	4.60, 5.28
vgg16	6.79, 5.36	5.68, 5.49	5.02, 5.30	4.14, 5.19	3.56, 4.98	9.32, 4.51	8.32, 4.88	4.61, 5.23	3.58, 4.90	7.41, 5.37	6.20, 5.57	4.98, 5.42	3.96, 5.11
vgg19	6.67, 5.40	5.50, 5.45	5.01, 5.34	4.12, 5.17	3.52, 4.95	8.71, 4.80	8.67, 4.61	4.55, 5.22	3.54, 4.91	7.40, 5.40	6.17, 5.54	4.98, 5.44	3.86, 5.08
googlenet	6.99, 5.36	5.96, 5.52	5.36, 5.36	4.62, 5.28	3.87, 5.07	7.91, 5.22	7.29, 5.30	7.66, 4.87	3.96, 5.08	7.80, 5.26	6.37, 5.56	5.49, 5.48	4.33, 5.24
densenet	8.18, 5.47	6.61, 5.53	5.92, 5.36	5.47, 5.45	4.38, 5.15	9.49, 5.01	8.78, 5.18	5.07, 5.29	6.24, 5.04	8.92, 5.19	7.37, 5.61	5.83, 5.54	5.15, 5.36
mobilenet_v2	6.78, 5.38	5.74, 5.50	5.13, 5.37	4.32, 5.21	3.78, 5.11	8.21, 5.15	7.32, 5.33	4.66, 5.25	3.76, 5.05	9.87, 4.64	6.49, 5.56	5.16, 5.40	4.02, 5.17
mobilenet_v3_small	8.66, 5.42	7.04, 5.61	6.14, 5.58	6.21, 5.71	4.97, 5.68	9.33, 4.90	9.33, 5.02	6.49, 5.48	4.26, 5.29	8.97, 5.13	10.29, 4.35	7.33, 5.32	5.08, 5.56
mobilenet_v3_large	8.12, 5.34	6.25, 5.47	5.57, 5.38	5.30, 5.44	4.33, 5.23	9.40, 4.98	8.72, 5.04	5.76, 5.39	4.27, 5.13	8.45, 5.20	7.83, 5.38	8.15, 4.83	4.76, 5.29
wide_resnet_50_2	7.38, 5.49	6.16, 5.56	5.57, 5.39	4.87, 5.33	4.09, 5.13	7.87, 5.25	7.12, 5.34	4.65, 5.27	4.03, 5.01	7.63, 5.39	6.25, 5.63	5.16, 5.46	6.03, 5.24

Table A.14: Mean and Std for FGSM and alpha=0.1

source	resnet18	resnet34	resnet50	resnet101	resnet152	vgg16	vgg19	googlenet	densenet	mobilenet_v2	mobilenet_v3_small	mobilenet_v3_large	wide_resnet_50_2
resnet18	7.32, 4.28	1.46, 3.42	0.93, 2.76	0.69, 2.44	0.51, 2.09	1.03, 2.88	0.99, 2.88	0.84, 2.69	0.59, 2.24	1.17, 3.10	0.59, 2.31	0.38, 1.81	0.41, 1.86
resnet34	1.74, 3.67	6.83, 4.40	1.09, 2.96	0.79, 2.58	0.68, 2.39	0.90, 2.70	0.90, 2.74	0.83, 2.65	0.78, 2.56	1.07, 2.99	0.53, 2.21	0.41, 1.89	0.52, 2.09
resnet50	1.18, 3.11	1.04, 2.90	6.20, 4.41	1.11, 2.99	0.85, 2.64	0.81, 2.57	0.78, 2.57	0.69, 2.44	0.79, 2.59	0.90, 2.75	0.47, 2.09	0.30, 1.63	0.75, 2.47
resnet101	1.08, 3.00	0.92, 2.77	1.26, 3.13	5.89, 4.54	1.20, 3.07	0.65, 2.30	0.64, 2.33	0.63, 2.35	0.72, 2.43	0.83, 2.66	0.44, 2.01	0.31, 1.62	0.65, 2.30
resnet152	0.99, 2.89	0.85, 2.66	1.17, 3.02	1.35, 3.23	5.68, 4.50	0.65, 2.31	0.62, 2.26	0.59, 2.23	0.74, 2.47	0.79, 2.59	0.45, 2.03	0.32, 1.65	0.71, 2.37
vgg16	1.04, 2.97	0.67, 2.40	0.60, 2.21	0.37, 1.80	0.31, 1.63	7.18, 4.40	3.14, 4.43	0.61, 2.32	0.39, 1.86	1.09, 3.03	0.52, 2.20	0.33, 1.72	0.26, 1.46
vgg19	0.97, 2.88	0.63, 2.32	0.58, 2.19	0.37, 1.82	0.31, 1.61	3.15, 4.45	7.04, 4.42	0.59, 2.25	0.39, 1.82	1.07, 2.98	0.49, 2.13	0.33, 1.71	0.27, 1.51
googlenet	0.86, 2.75	0.56, 2.25	0.46, 2.01	0.32, 1.72	0.27, 1.56	0.63, 2.33	0.63, 2.39	7.11, 4.58	0.32, 1.70	0.71, 2.49	0.52, 2.22	0.28, 1.61	0.18, 1.26
densenet	1.00, 2.87	0.91, 2.73	0.97, 2.75	0.75, 2.47	0.70, 2.40	0.72, 2.41	0.71, 2.42	0.66, 2.38	6.12, 4.31	0.82, 2.60	0.51, 2.16	0.33, 1.68	0.62, 2.23
mobilenet_v2	0.98, 2.88	0.68, 2.44	0.57, 2.19	0.41, 1.91	0.32, 1.68	0.93, 2.79	0.89, 2.78	0.65, 2.40	0.38, 1.83	7.32, 4.30	0.76, 2.61	0.56, 2.22	0.23, 1.36
mobilenet_v3_small	0.54, 2.19	0.27, 1.60	0.22, 1.42	0.16, 1.21	0.13, 1.08	0.45, 2.02	0.43, 1.99	0.43, 2.00	0.16, 1.21	0.87, 2.76	7.67, 4.42	0.66, 2.41	0.09, 0.88
mobilenet_v3_large	0.87, 2.73	0.59, 2.25	0.49, 2.05	0.37, 1.76	0.33, 1.70	0.77, 2.55	0.75, 2.61	0.65, 2.40	0.40, 1.88	1.35, 3.29	1.26, 3.28	6.87, 4.29	0.23, 1.38
wide_resnet_50_2	0.92, 2.79	0.75, 2.50	1.03, 2.88	0.81, 2.59	0.74, 2.45	0.68, 2.37	0.65, 2.36	0.57, 2.21	0.69, 2.42	0.73, 2.49	0.44, 2.02	0.30, 1.64	5.82, 4.53

Table A.15: Mean and Std for FGSM and alpha=0.01

source	resnet18	resnet34	resnet50	resnet101	resnet152	vgg16	vgg19	googlenet	densenet	mobilenet_v2	mobilenet_v3_small	mobilenet_v3_large	wide_resnet_50_2
resnet18	7.67, 4.44	3.25, 4.65	2.68, 4.36	2.24, 4.13	1.84, 3.81	3.05, 4.60	2.88, 4.49	2.57, 4.33	1.94, 3.87	2.95, 4.60	1.91, 3.96	1.66, 3.66	1.83, 3.81
resnet34	3.38, 4.72	7.02, 4.54	2.59, 4.29	2.17, 4.01	1.92, 3.85	2.62, 4.35	2.54, 4.29	2.29, 4.14	1.98, 3.87	2.66, 4.42	1.70, 3.79	1.57, 3.59	1.89, 3.81
resnet50	2.44, 4.22	2.21, 4.03	6.25, 4.61	2.25, 4.06	1.85, 3.70	2.00, 3.88	1.94, 3.82	1.83, 3.78	1.73, 3.64	2.06, 4.00	1.33, 3.40	1.17, 3.15	1.87, 3.75
resnet101	2.27, 4.15	2.03, 3.90	2.41, 4.15	5.87, 4.71	2.17, 3.97	1.79, 3.72	1.78, 3.76	1.74, 3.69	1.59, 3.51	2.02, 3.97	1.36, 3.42	1.13, 3.06	1.71, 3.64
resnet152	2.09, 4.02	1.90, 3.80	2.15, 3.96	2.27, 4.07	5.58, 4.67	1.66, 3.63	1.63, 3.59	1.57, 3.53	1.55, 3.44	1.82, 3.77	1.28, 3.32	1.07, 2.99	1.65, 3.54
vgg16	2.29, 4.14	1.70, 3.65	1.57, 3.52	1.16, 3.14	1.00, 2.90	7.29, 4.45	4.34, 4.83	1.65, 3.62	1.11, 3.05	2.28, 4.17	1.37, 3.43	1.17, 3.17	1.00, 2.88
vgg19	2.12, 4.07	1.60, 3.57	1.49, 3.45	1.08, 3.01	0.95, 2.82	4.25, 4.83	7.15, 4.50	1.62, 3.61	1.08, 3.02	2.11, 4.05	1.31, 3.40	1.12, 3.13	0.94, 2.82
googlenet	2.55, 4.40	1.91, 3.90	1.81, 3.81	1.47, 3.49	1.28, 3.29	2.23, 4.14	2.19, 4.10	7.38, 4.67	1.41, 3.44	2.24, 4.22	1.78, 3.92	1.45, 3.51	1.25, 3.24
densenet	2.22, 4.09	1.99, 3.86	2.08, 3.91	1.81, 3.70	1.63, 3.54	2.05, 4.00	1.97, 3.92	1.75, 3.67	5.97, 4.47	2.08, 4.00	1.55, 3.67	1.24, 3.19	1.65, 3.51
mobilenet_v2	2.25, 4.13	1.74, 3.70	1.55, 3.55	1.20, 3.18	0.98, 2.89	2.17, 4.06	2.04, 3.93	1.71, 3.71	1.08, 3.02	7.51, 4.52	1.89, 3.94	1.65, 3.67	0.91, 2.78
mobilenet_v3_small	1.48, 3.55	0.93, 2.86	0.76, 2.61	0.70, 2.56	0.60, 2.37	1.40, 3.47	1.31, 3.34	1.29, 3.37	0.61, 2.36	2.10, 4.12	7.74, 4.69	1.87, 3.92	0.47, 2.06
mobilenet_v3_large	1.75, 3.77	1.28, 3.22	1.01, 2.90	0.94, 2.83	0.76, 2.53	1.63, 3.59	1.49, 3.48	1.45, 3.50	0.85, 2.71	2.27, 4.14	2.18, 4.17	6.84, 4.51	0.70, 2.43
wide_resnet_50_2	2.06, 3.99	1.82, 3.73	2.09, 3.95	1.81, 3.71	1.68, 3.62	1.80, 3.75	1.74, 3.70	1.58, 3.57	1.53, 3.46	1.83, 3.84	1.33, 3.43	1.11, 3.09	5.74, 4.74

Table A.16: Mean and Std for FGV and alpha=0.3

source	resnet18	resnet34	resnet50	resnet101	resnet152	vgg16	vgg19	googlenet	densenet	mobilenet_v2	mobilenet_v3_small	mobilenet_v3_large	wide_resnet_50_2
resnet18	7.15, 4.33	1.41, 3.33	0.94, 2.77	0.74, 2.51	0.55, 2.16	1.07, 2.93	0.99, 2.84	0.93, 2.80	0.63, 2.30	1.15, 3.07	0.61, 2.35	0.39, 1.83	0.44, 1.92
resnet34	1.59, 3.52	6.53, 4.48	1.00, 2.83	0.78, 2.55	0.68, 2.42	0.88, 2.68	0.86, 2.66	0.86, 2.70	0.71, 2.43	1.05, 2.97	0.50, 2.14	0.37, 1.82	0.52, 2.05
resnet50	1.01, 2.90	0.85, 2.64	5.67, 4.53	0.87, 2.67	0.71, 2.41	0.66, 2.32	0.65, 2.30	0.64, 2.38	0.67, 2.40	0.75, 2.51	0.40, 1.93	0.26, 1.51	0.59, 2.16
resnet101	0.93, 2.80	0.77, 2.54	1.07, 2.93	5.46, 4.59	0.98, 2.77	0.55, 2.12	0.54, 2.12	0.62, 2.28	0.61, 2.24	0.74, 2.52	0.39, 1.91	0.27, 1.54	0.55, 2.09
resnet152	0.80, 2.58	0.68, 2.35	0.88, 2.64	1.04, 2.87	5.05, 4.54	0.50, 2.03	0.51, 2.03	0.58, 2.20	0.60, 2.22	0.65, 2.35	0.36, 1.83	0.24, 1.42	0.54, 2.06
vgg16	0.85, 2.66	0.57, 2.21	0.47, 1.98	0.31, 1.67	0.28, 1.57	6.79, 4.40	2.62, 4.21	0.56, 2.20	0.34, 1.74	0.92, 2.79	0.42, 1.99	0.30, 1.64	0.21, 1.30
vgg19	0.82, 2.68	0.52, 2.12	0.44, 1.90	0.29, 1.58	0.26, 1.52	2.54, 4.15	6.60, 4.46	0.55, 2.18	0.34, 1.73	0.84, 2.66	0.41, 1.99	0.26, 1.49	0.21, 1.33
googlenet	0.93, 2.83	0.62, 2.32	0.52, 2.10	0.37, 1.80	0.32, 1.67	0.66, 2.39	0.68, 2.46	6.97, 4.63	0.39, 1.85	0.75, 2.57	0.50, 2.17	0.28, 1.62	0.24, 1.43
densenet	0.87, 2.69	0.76, 2.52	0.79, 2.51	0.63, 2.27	0.59, 2.18	0.63, 2.26	0.59, 2.18	0.60, 2.24	5.54, 4.42	0.70, 2.44	0.43, 2.00	0.27, 1.51	0.51, 2.04
mobilenet_v2	0.88, 2.74	0.60, 2.28	0.47, 2.01	0.35, 1.77	0.29, 1.61	0.78, 2.54	0.74, 2.50	0.60, 2.34	0.32, 1.67	7.09, 4.33	0.71, 2.54	0.50, 2.09	0.20, 1.26
mobilenet_v3_small	0.35, 1.79	0.18, 1.33	0.13, 1.07	0.13, 1.07	0.10, 0.97	0.30, 1.65	0.27, 1.60	0.31, 1.68	0.11, 0.98	0.60, 2.32	7.12, 4.58	0.48, 2.07	0.06, 0.71
mobilenet_v3_large	0.61, 2.34	0.39, 1.84	0.29, 1.58	0.24, 1.45	0.21, 1.35	0.49, 2.05	0.48, 2.09	0.47, 2.08	0.28, 1.57	0.92, 2.78	0.91, 2.84	6.38, 4.35	0.13, 1.01
wide_resnet_50_2	0.77, 2.55	0.65, 2.32	0.83, 2.58	0.65, 2.31	0.63, 2.25	0.57, 2.17	0.55, 2.14	0.54, 2.14	0.58, 2.21	0.63, 2.31	0.35, 1.80	0.24, 1.44	5.19, 4.59

Table A.17: Mean and Std for FGV and alpha=0.1

source	resnet18	resnet34	resnet50	resnet101	resnet152	vgg16	vgg19	googlenet	densenet	mobilenet_v2	mobilenet_v3_small	mobilenet_v3_large	wide_resnet_50_2
resnet18	3.37, 4.48	0.06, 0.74	0.03, 0.51	0.02, 0.41	0.02, 0.41	0.03, 0.48	0.03, 0.54	0.07, 0.77	0.02, 0.44	0.06, 0.75	0.03, 0.53	0.01, 0.22	0.01, 0.25
resnet34	0.11, 1.00	2.49, 4.09	0.04, 0.57	0.03, 0.52	0.02, 0.45	0.04, 0.61	0.03, 0.41	0.06, 0.76	0.03, 0.51	0.06, 0.71	0.03, 0.51	0.01, 0.23	0.01, 0.28
resnet50	0.07, 0.80	0.04, 0.60	1.81, 3.60	0.03, 0.56	0.03, 0.52	0.03, 0.55	0.03, 0.48	0.05, 0.65	0.03, 0.55	0.05, 0.63	0.02, 0.37	0.00, 0.15	0.01, 0.31
resnet101	0.07, 0.81	0.03, 0.53	0.04, 0.59	1.66, 3.50	0.04, 0.58	0.03, 0.46	0.02, 0.45	0.04, 0.58	0.04, 0.59	0.05, 0.62	0.02, 0.38	0.01, 0.26	0.01, 0.31
resnet152	0.04, 0.61	0.04, 0.56	0.03, 0.47	0.04, 0.66	1.29, 3.11	0.02, 0.43	0.02, 0.42	0.04, 0.55	0.04, 0.60	0.04, 0.56	0.02, 0.34	0.01, 0.18	0.01, 0.30
vgg16	0.05, 0.69	0.02, 0.43	0.02, 0.38	0.01, 0.38	0.01, 0.28	2.77, 4.20	0.16, 1.22	0.04, 0.53	0.01, 0.36	0.05, 0.68	0.03, 0.58	0.01, 0.25	0.00, 0.20
vgg19	0.04, 0.63	0.02, 0.43	0.02, 0.34	0.01, 0.30	0.01, 0.30	0.15, 1.13	2.51, 4.07	0.04, 0.58	0.02, 0.40	0.06, 0.71	0.02, 0.48	0.01, 0.16	0.01, 0.22
googlenet	0.04, 0.64	0.02, 0.43	0.02, 0.42	0.02, 0.41	0.01, 0.31	0.02, 0.43	0.03, 0.47	2.92, 4.39	0.02, 0.40	0.04, 0.53	0.02, 0.38	0.00, 0.16	0.01, 0.19
densenet	0.05, 0.68	0.04, 0.61	0.04, 0.60	0.02, 0.48	0.03, 0.54	0.02, 0.33	0.03, 0.47	0.04, 0.60	1.73, 3.54	0.05, 0.63	0.02, 0.40	0.01, 0.20	0.01, 0.27
mobilenet_v2	0.05, 0.72	0.03, 0.53	0.02, 0.37	0.02, 0.42	0.01, 0.17	0.04, 0.60	0.03, 0.51	0.03, 0.55	0.01, 0.35	3.67, 4.53	0.05, 0.68	0.01, 0.31	0.01, 0.22
mobilenet_v3_small	0.02, 0.46	0.01, 0.30	0.01, 0.22	0.00, 0.20	0.00, 0.13	0.01, 0.26	0.01, 0.30	0.02, 0.42	0.00, 0.16	0.04, 0.58	3.20, 4.58	0.01, 0.33	0.00, 0.16
mobilenet_v3_large	0.03, 0.54	0.02, 0.42	0.01, 0.23	0.01, 0.37	0.00, 0.16	0.02, 0.38	0.02, 0.44	0.03, 0.47	0.01, 0.36	0.06, 0.69	0.08, 0.88	2.16, 3.83	0.00, 0.15
wide_resnet_50_2	0.04, 0.56	0.03, 0.49	0.04, 0.59	0.03, 0.55	0.02, 0.42	0.02, 0.42	0.02, 0.44	0.04, 0.54	0.04, 0.57	0.04, 0.53	0.02, 0.40	0.01, 0.22	1.15, 2.94

Table A.18: Mean and Std for FGV and alpha=0.01

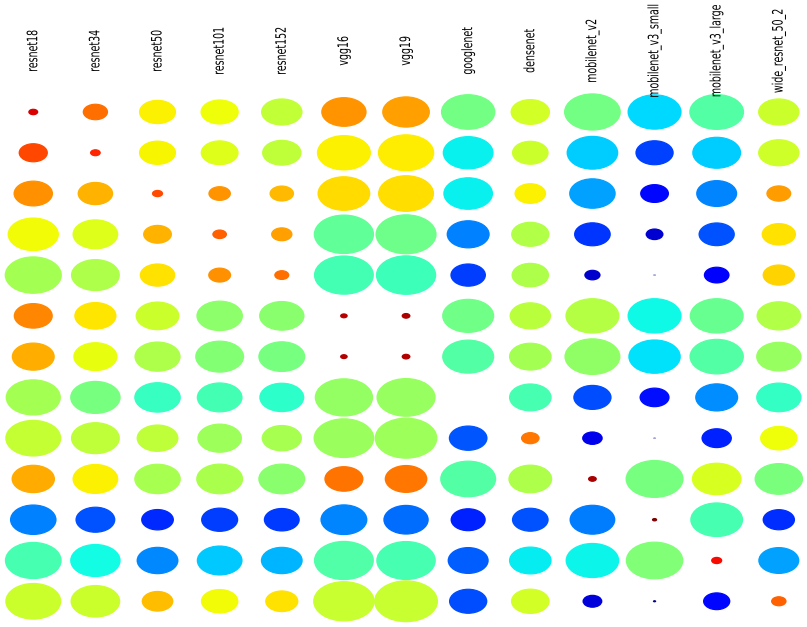


Figure A.1: mean and std of predicted class distance from true class for PGD and epsilon 0.1

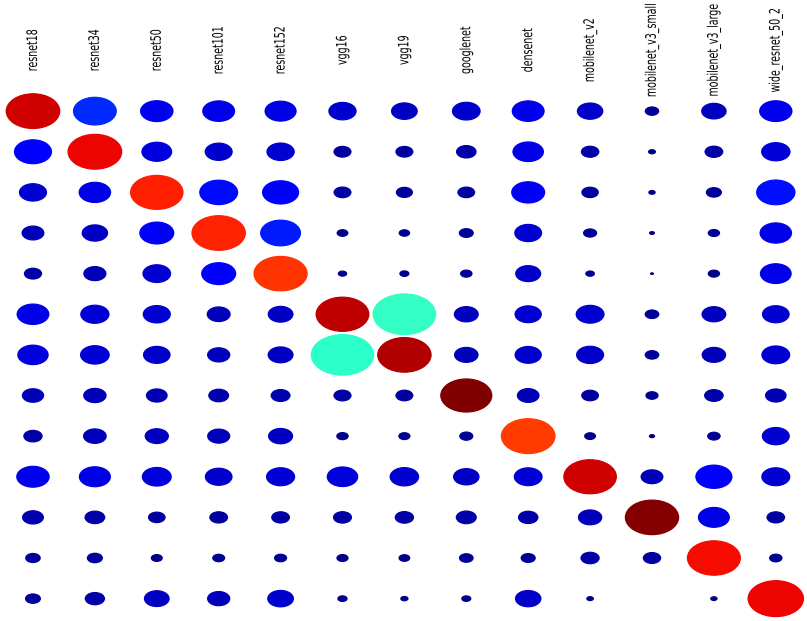


Figure A.2: mean and std of predicted class distance from true class for PGD and epsilon 0.01

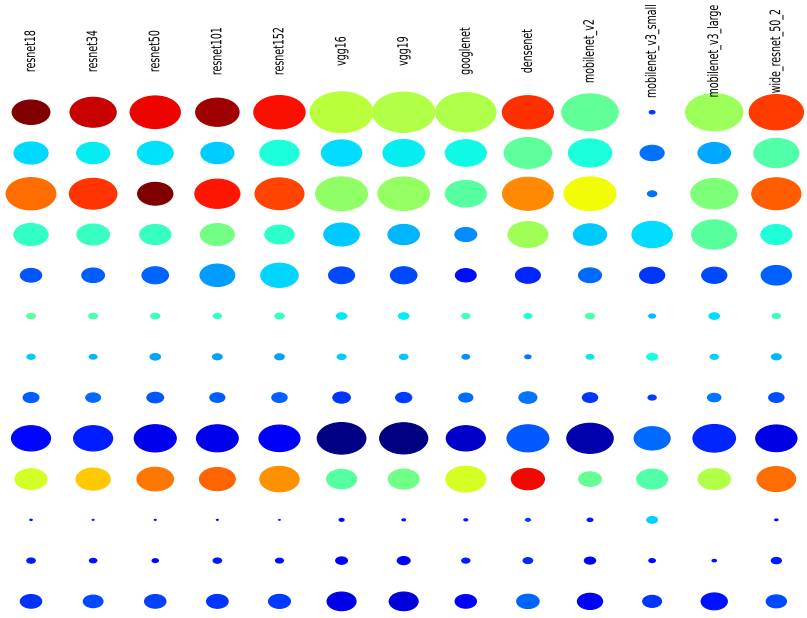


Figure A.3: mean and std of predicted class distance from true class for FGSM and epsilon 0.3

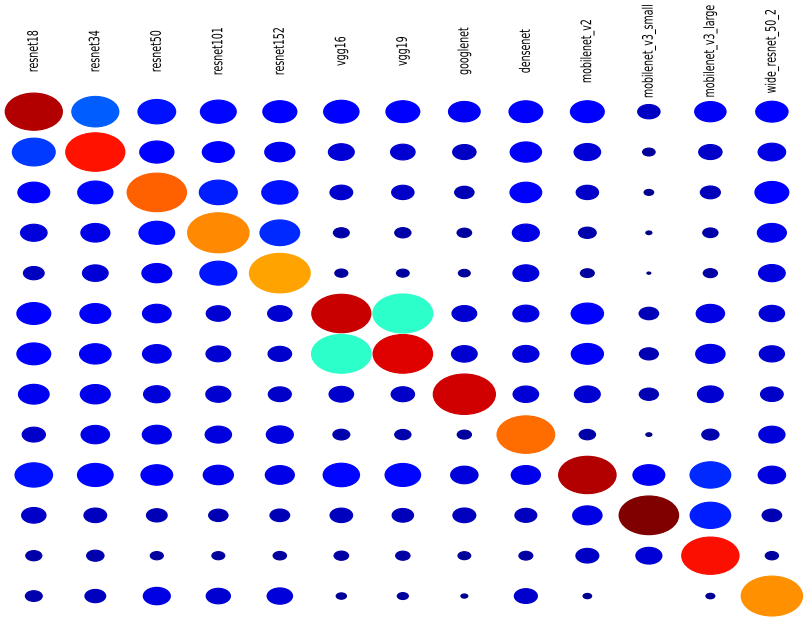


Figure A.4: mean and std of predicted class distance from true class for FGSM and epsilon 0.01

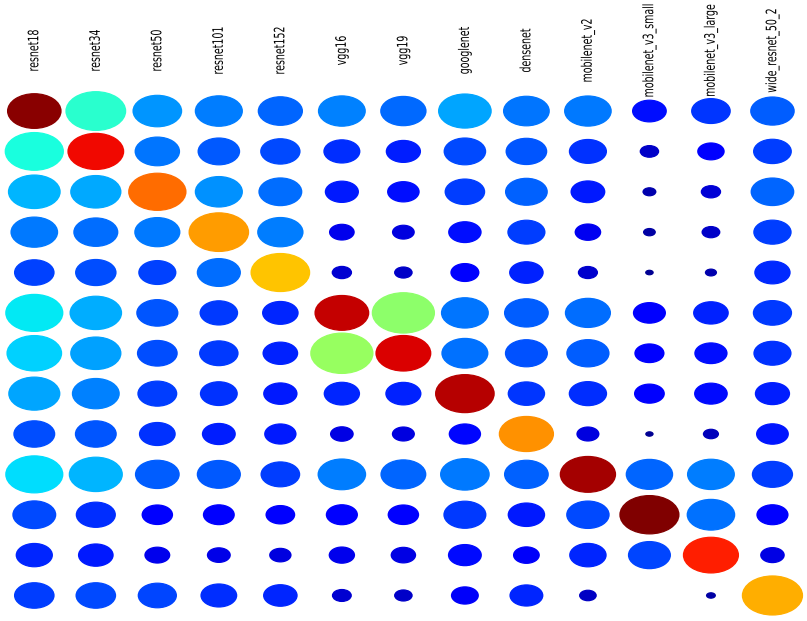


Figure A.5: mean and std of predicted class distance from true class for FGV and epsilon 0.3

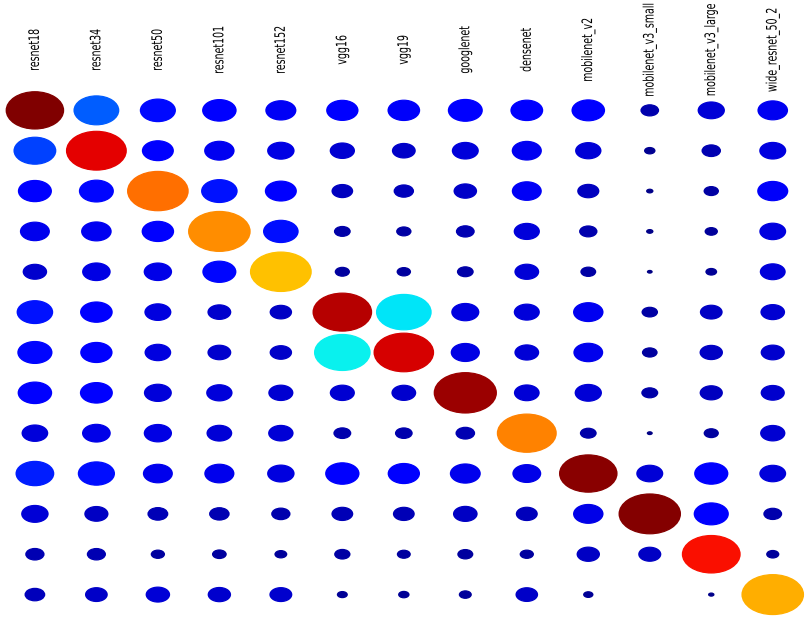


Figure A.6: mean and std of predicted class distance from true class for FGV and epsilon 0.1

source	resnet18	resenet34	resnet50	resnet101	resnet152	vgg16	vgg19	googlenet	densenet	mobilenet v2	v3 small	v3 large	wide resnet
resnet18	9997	1139	822	748	696	410	497	754	840	351	269	453	614
resenet34	1677	9988	1321	1206	1154	480	569	1031	1397	424	272	518	1075
resnet50	1200	1416	9996	2256	1978	468	616	1162	1692	406	290	518	2038
resnet101	1173	1416	2438	9982	3443	395	507	1220	1773	410	307	552	1917
resnet152	1129	1455	2388	3685	9985	417	536	1296	2122	395	306	618	2432
vgg16	485	422	407	357	387	9994	3115	504	420	324	221	328	373
vgg19	542	469	474	404	397	3161	9989	529	440	310	226	373	416
googlenet	612	594	647	598	586	363	416	9984	689	291	249	419	538
densenet	1033	1215	1494	1392	1461	436	551	1295	9999	371	319	670	1577
mobilenet v2	426	362	353	354	326	303	342	395	342	9990	255	390	289
v3 small	335	336	321	332	339	203	249	411	398	308	9999	860	322
v3 large	435	405	382	408	397	248	319	476	464	363	492	10000	336
wide resnet	1018	1218	2290	1943	2197	466	603	1168	2073	354	299	523	10000

Table A.19: closest target with PGD and epsilon 0.3

source	resnet18	resenet34	resnet50	resnet101	resnet152	vgg16	vgg19	googlenet	densenet	mobilenet v2	v3 small	v3 large	wide resnet
resnet18	9981	2333	1633	1182	1053	1255	1164	974	1133	1182	282	605	1038
resenet34	3194	9957	2444	1861	1738	1344	1270	1223	1875	1482	312	729	1663
resnet50	2054	2256	9937	3024	2668	1222	1240	1156	2058	1265	288	624	2749
resnet101	1990	2257	3835	9946	4208	987	999	1204	2029	1309	330	721	2515
resnet152	1916	2265	3782	4550	9926	1064	1084	1248	2413	1228	356	771	3028
vgg16	943	690	687	424	425	9949	5329	532	505	847	199	385	459
vgg19	944	765	732	446	470	5674	9919	588	531	892	214	372	498
googlenet	1200	1056	1080	816	741	877	903	9971	831	885	276	556	710
densenet	1805	2126	2616	2065	2150	1257	1240	1281	9994	1300	384	828	2352
mobilenet v2	883	722	622	455	407	831	721	463	445	9958	385	677	352
v3 small	484	394	327	282	260	381	373	338	289	799	9985	1063	245
v3 large	868	770	598	567	499	633	596	609	552	1284	839	9991	444
wide resnet	1611	1878	3149	2403	2645	1129	1174	1118	2317	1032	292	611	9960

Table A.20: closest target with PGD and epsilon 0.1

source	resnet18	resenet34	resnet50	resnet101	resnet152	vgg16	vgg19	googlenet	densenet	mobilenet v2	v3 small	v3 large	wide resnet
resnet18	9476	156	92	70	54	87	75	67	65	102	33	37	41
resenet34	215	9168	156	93	73	98	84	67	76	123	24	40	58
resnet50	125	124	9003	166	123	97	72	63	94	100	23	29	103
resnet101	132	123	240	8772	206	78	60	74	88	84	32	41	101
resnet152	124	120	221	268	8781	87	73	70	114	93	28	36	117
vgg16	85	56	55	26	33	8834	571	45	31	86	25	19	25
vgg19	71	50	50	25	32	624	8752	42	31	82	25	24	21
googlenet	81	59	53	35	41	59	51	9030	33	61	26	35	28
densenet	140	140	184	138	121	108	88	76	9106	118	30	42	113
mobilenet v2	64	45	53	28	27	81	50	41	32	9369	45	42	16
v3 small	25	16	13	11	12	25	16	22	12	41	8773	42	3
v3 large	79	58	49	29	29	74	50	45	36	127	83	9029	19
wide resnet	121	132	229	151	166	102	84	70	136	91	29	40	8800

Table A.21: closest target with PGD and epsilon 0.01

source	resnet18	resenet34	resnet50	resnet101	resnet152	vgg16	vgg19	googlenet	densenet	mobilenet v2	v3 small	v3 large	wide resnet
resnet18	654	430	444	452	516	162	239	458	531	211	217	281	458
resenet34	290	802	445	459	504	177	227	449	571	188	206	271	430
resnet50	275	360	714	460	514	212	264	395	528	224	174	261	446
resnet101	225	323	399	608	525	174	221	354	495	201	163	262	421
resnet152	246	322	398	464	942	215	242	366	525	217	185	287	425
vgg16	301	307	327	323	363	331	296	370	373	243	180	260	322
vgg19	283	292	331	308	371	234	480	342	359	213	166	230	310
googlenet	284	320	403	371	422	227	263	1160	404	212	189	241	359
densenet	237	321	388	414	533	144	173	394	1307	158	180	257	424
mobilenet v2	328	340	365	370	392	231	285	400	443	320	225	348	353
v3 small	179	234	278	255	314	134	117	252	327	181	273	318	262
v3 large	248	345	361	378	419	117	144	350	443	251	247	834	379
wide resnet	227	289	343	364	445	199	233	335	443	181	182	245	681

Table A.22: closest target with FGSM and epsilon 0.3

source	resnet18	resenet34	resnet50	resnet101	resnet152	vgg16	vgg19	googlenet	densenet	mobilenet v2	v3 small	v3 large	wide resnet
resnet18	4651	304	200	143	112	213	173	140	123	198	65	69	110
resenet34	351	3588	249	193	176	194	164	141	181	194	52	67	146
resnet50	234	256	3028	289	247	161	137	139	207	168	51	66	213
resnet101	218	210	305	2794	324	134	108	113	168	160	55	60	173
resnet152	187	213	288	351	2787	139	109	118	201	147	52	65	212
vgg16	167	127	134	68	69	3098	761	100	79	171	50	61	63
vgg19	150	130	134	72	74	747	2810	109	81	171	55	61	65
googlenet	125	105	99	63	60	95	94	3918	62	110	47	50	42
densenet	201	223	257	194	196	157	138	131	3930	151	51	81	188
mobilenet v2	153	137	117	75	72	153	135	115	82	4671	85	109	49
v3 small	63	53	46	25	24	52	48	58	23	131	3993	121	14
v3 large	150	127	107	79	81	121	102	109	87	255	171	4175	55
wide resnet	157	173	250	195	192	115	123	110	161	128	53	53	2545

Table A.23: closest target with FGSM and epsilon 0.1

source	resnet18	resenet34	resnet50	resnet101	resnet152	vgg16	vgg19	googlenet	densenet	mobilenet v2	v3 small	v3 large	wide resnet
resnet18	5	17	14	51	35	10	8	19	87	7	18	26	48
resenet34	4	15	11	51	33	18	7	24	77	3	19	24	48
resnet50	5	15	12	53	23	17	9	25	105	5	19	22	39
resnet101	2	11	13	53	29	20	6	21	98	3	21	22	46
resnet152	3	12	14	59	31	16	8	27	84	7	17	30	48
vgg16	7	14	17	60	44	16	9	26	104	9	24	38	58
vgg19	7	15	18	56	42	13	7	26	113	5	26	47	76
googlenet	3	14	22	51	44	9	8	25	91	4	19	22	41
densenet	3	9	14	46	32	15	7	31	84	5	19	24	31
mobilenet v2	7	15	11	53	42	9	10	24	115	6	18	37	59
v3 small	9	17	17	38	66	16	7	9	52	4	10	21	29
v3 large	9	14	18	35	58	16	12	22	88	8	14	28	49
wide resnet	6	10	15	50	27	15	5	19	69	6	20	28	43

Table A.24: closest target with FGSM and epsilon 0.01

source	resnet18	resenet34	resnet50	resnet101	resnet152	vgg16	vgg19	googlenet	densenet	mobilenet v2	v3 small	v3 large	wide resnet
resnet18	3460	646	498	418	380	438	424	409	412	426	188	221	350
resenet34	592	2812	540	445	439	368	377	356	457	375	151	212	396
resnet50	398	438	2332	513	452	301	281	290	420	321	139	164	451
resnet101	361	422	509	2144	563	249	245	283	362	284	118	167	386
resnet152	317	399	474	536	2221	247	228	275	395	286	115	159	414
vgg16	363	311	298	211	192	2462	942	247	212	312	117	146	204
vgg19	299	307	283	193	195	842	2254	252	218	297	117	134	194
googlenet	350	306	312	226	226	277	282	3021	265	247	145	157	212
densenet	377	435	453	427	413	301	295	315	3175	322	164	191	432
mobilenet v2	360	330	282	219	203	347	305	246	241	3082	179	227	193
v3 small	147	124	103	88	76	129	111	136	85	231	2495	221	64
v3 large	234	250	190	159	160	213	196	202	195	351	228	3114	147
wide resnet	281	332	419	357	385	245	245	241	348	230	115	129	1943

Table A.25: closest target with FGV and epsilon 0.3

source	resnet18	resenet34	resnet50	resnet101	resnet152	vgg16	vgg19	googlenet	densenet	mobilenet v2	v3 small	v3 large	wide resnet
resnet18	4253	316	225	153	123	196	176	173	150	202	68	70	127
resenet34	326	3254	252	202	175	184	178	144	192	189	54	72	149
resnet50	202	202	2624	240	203	156	116	131	167	152	46	60	179
resnet101	188	196	276	2463	278	113	86	110	163	140	52	50	171
resnet152	159	182	243	278	2353	117	98	113	175	126	48	50	161
vgg16	137	118	112	62	59	3025	619	97	74	151	37	51	50
vgg19	125	114	98	53	65	589	2659	96	69	149	42	43	54
googlenet	158	116	120	67	74	112	109	3612	71	115	54	40	59
densenet	174	205	221	175	190	129	127	127	3208	141	45	72	162
mobilenet v2	146	126	97	72	67	138	129	93	76	4252	87	91	46
v3 small	43	29	24	21	15	36	34	40	17	82	3253	66	10
v3 large	84	86	63	52	47	84	70	76	55	169	126	3536	30
wide resnet	148	159	216	169	166	109	114	120	148	104	40	50	2205

Table A.26: closest target with FGV and epsilon 0.1

source	resnet18	resenet34	resnet50	resnet101	resnet152	vgg16	vgg19	googlenet	densenet	mobilenet v2	v3 small	v3 large	wide resnet
resnet18	1010	14	6	4	4	7	5	16	4	7	6	4	2
resenet34	15	736	10	0	3	4	5	13	5	3	6	4	2
resnet50	13	12	512	4	8	4	4	11	5	6	6	4	3
resnet101	13	12	12	457	10	6	6	11	6	5	4	5	3
resnet152	10	15	9	4	352	3	3	9	4	5	6	6	2
vgg16	10	9	5	1	3	754	22	10	2	7	4	4	2
vgg19	9	9	5	0	2	25	664	11	1	7	5	5	2
googlenet	9	8	6	0	4	3	3	800	1	2	4	3	1
densenet	11	12	9	3	7	4	2	11	549	6	4	4	4
mobilenet v2	10	9	6	1	3	5	5	9	1	1206	5	6	3
v3 small	3	1	2	1	1	0	1	6	2	2	640	4	0
v3 large	8	5	2	0	1	1	4	12	1	5	9	591	1
wide resnet	9	8	8	2	8	6	2	12	4	5	5	3	365

Table A.27: closest target with FGV and epsilon 0.01

source	resnet18	resenet34	resnet50	resnet101	resnet152	vgg16	vgg19	googlenet	densenet	mobilenet v2	v3 small	v3 large	wide resnet
resnet18	0	0	8	17	16	0	0	2	1	1	0	0	12
resenet34	0	0	11	16	17	0	0	0	2	4	1	1	5
resnet50	1	0	0	23	17	0	0	0	1	1	0	0	12
resnet101	1	0	5	0	9	0	0	1	0	3	0	1	6
resnet152	0	0	8	8	0	0	0	1	0	0	0	0	5
vgg16	1	0	7	11	20	0	0	0	1	3	0	1	8
vgg19	0	0	5	15	14	2	0	0	0	1	0	0	7
googlenet	0	0	2	18	14	0	2	0	0	1	0	0	6
densenet	0	0	2	10	15	0	0	0	0	1	0	1	6
mobilenet v2	0	0	8	17	16	2	3	2	2	0	0	0	9
v3 small	0	0	8	13	21	0	1	2	0	3	0	1	13
v3 large	1	1	11	20	20	0	1	1	0	2	0	0	10
wide resnet	1	0	4	16	12	0	1	1	2	1	0	0	0

Table A.28: farthest target with PGD and epsilon 0.3

source	resnet18	resenet34	resnet50	resnet101	resnet152	vgg16	vgg19	googlenet	densenet	mobilenet v2	v3 small	v3 large	wide resnet
resnet18	0	0	4	4	9	0	0	0	0	0	0	0	8
resenet34	0	1	5	7	7	0	0	0	0	0	0	1	10
resnet50	0	0	2	6	3	0	0	0	0	0	0	0	6
resnet101	0	0	4	4	8	1	0	1	0	1	0	1	5
resnet152	0	2	6	5	8	0	0	0	0	0	0	1	6
vgg16	0	0	2	4	7	0	0	0	0	1	0	1	3
vgg19	0	0	4	7	5	0	0	0	1	0	0	1	3
googlenet	0	0	2	8	6	0	0	0	1	0	0	0	4
densenet	0	0	1	4	2	0	0	0	0	1	0	0	4
mobilenet v2	0	0	3	5	9	0	0	1	0	0	0	1	4
v3 small	0	0	5	4	10	0	0	0	1	0	0	2	3
v3 large	0	0	1	5	7	0	0	1	0	1	0	0	4
wide resnet	0	0	2	8	9	0	0	0	0	1	0	0	7

Table A.29: farthest target with FGSM and epsilon 0.3

source	resnet18	resenet34	resnet50	resnet101	resnet152	vgg16	vgg19	googlenet	densenet	mobilenet v2	v3 small	v3 large	wide resnet
resnet18	2	1	1	0	0	0	1	0	0	1	2	0	0
resenet34	0	1	1	2	0	0	0	0	0	1	1	0	2
resnet50	0	1	2	0	0	1	0	0	0	1	1	0	1
resnet101	1	1	1	1	1	0	0	1	0	1	1	0	0
resnet152	0	0	1	1	1	1	0	0	0	1	1	0	1
vgg16	0	1	1	0	0	1	0	0	0	0	0	0	0
vgg19	0	1	1	0	0	1	1	0	0	1	1	0	2
googlenet	1	1	1	0	0	0	0	0	0	1	1	0	0
densenet	1	1	1	0	1	0	0	0	0	2	1	0	1
mobilenet v2	0	1	1	0	0	0	0	0	0	1	1	0	0
v3 small	0	1	1	1	1	0	0	0	0	1	0	0	0
v3 large	0	1	1	0	0	0	0	0	0	1	1	0	0
wide resnet	0	1	1	0	0	0	0	1	1	1	1	0	1

Table A.30: farthest target with FGV and epsilon 0.3

source	resnet18	resenet34	resnet50	resnet101	resnet152	vgg16	vgg19	googlenet	densenet	mobilenet v2	v3 small	v3 large	wide resnet
resnet18	0	2	3	2	1	0	1	1	3	3	7	7	1
resenet34	4	0	1	1	2	0	1	3	1	2	7	7	1
resnet50	3	0	0	2	0	3	0	5	4	2	8	8	1
resnet101	8	1	3	0	2	1	1	2	3	4	5	6	2
resnet152	3	2	5	1	0	1	1	5	1	3	12	6	0
vgg16	3	2	1	1	1	0	1	0	1	5	11	9	0
vgg19	6	2	0	1	1	0	0	3	2	4	8	9	0
googlenet	8	1	1	0	1	3	2	0	2	2	11	8	1
densenet	3	0	0	2	3	0	3	5	0	2	9	8	1
mobilenet v2	7	3	2	1	3	1	1	4	4	0	9	11	4
v3 small	6	1	0	1	4	0	2	7	2	0	0	6	1
v3 large	7	1	1	5	2	2	1	1	2	0	6	0	0
wide resnet	3	2	4	1	1	1	3	3	4	3	14	6	0

Table A.31: median target with PGD and epsilon 0.3

source	resnet18	resenet34	resnet50	resnet101	resnet152	vgg16	vgg19	googlenet	densenet	mobilenet v2	v3 small	v3 large	wide resnet
resnet18	1	0	0	857	1	0	0	3	19	0	6	4	3
resenet34	1	0	0	837	2	1	0	1	23	0	6	3	4
resnet50	3	0	0	867	2	0	2	4	11	0	11	5	1
resnet101	1	0	0	872	0	0	0	3	8	0	6	5	3
resnet152	2	0	0	987	0	0	2	4	9	0	10	7	1
vgg16	7	0	0	904	5	6	0	11	19	0	15	4	0
vgg19	11	1	0	876	4	2	3	11	26	0	14	5	0
googlenet	20	2	0	832	11	0	2	6	33	0	22	2	0
densenet	1	0	0	604	4	2	0	4	18	0	4	2	2
mobilenet v2	13	0	0	730	2	0	2	8	17	0	3	3	0
v3 small	1	0	0	257	2	0	0	6	3	0	1	3	0
v3 large	2	0	0	270	3	15	0	2	4	0	4	2	0
wide resnet	1	0	0	923	3	1	2	5	19	0	11	6	0

Table A.32: median target with FGSM and epsilon 0.3

source	resnet18	resenet34	resnet50	resnet101	resnet152	vgg16	vgg19	googlenet	densenet	mobilenet v2	v3 small	v3 large	wide resnet
resnet18	2	1	1	2	1	1	2	1	3	1	0	0	1
resenet34	2	1	2	1	0	1	2	1	1	0	0	2	0
resnet50	2	0	1	0	0	1	0	1	1	1	0	0	0
resnet101	0	0	0	0	1	0	1	1	3	0	0	0	0
resnet152	1	0	0	0	1	2	2	1	2	1	0	0	0
vgg16	1	1	1	0	0	3	1	2	2	0	0	1	0
vgg19	1	1	1	1	1	2	1	0	1	0	1	0	0
googlenet	2	1	3	0	0	3	1	5	1	0	0	0	0
densenet	1	0	1	0	0	2	3	1	1	0	0	1	0
mobilenet v2	1	0	0	0	0	1	1	0	2	0	0	0	0
v3 small	1	0	1	0	1	0	1	0	1	1	1	0	0
v3 large	1	1	1	0	0	0	1	1	1	1	0	1	0
wide resnet	1	0	1	0	0	1	0	1	2	0	0	0	1

Table A.33: median target with FGV and epsilon 0.3

source	resnet18	resnet34	resnet50	resnet101	resnet152	vgg16	vgg19	googlenet	densenet	mobilenet_v2	mobilenet_v3_small	mobilenet_v3_large	wide_resnet_50_2
resnet18	20.44, 1.79	19.95, 2.15	20.23, 2.36	20.23, 2.50	20.31, 2.52	19.48, 2.53	19.65, 2.56	20.27, 2.16	20.50, 2.08	19.45, 2.74	19.91, 2.46	20.20, 2.31	20.26, 2.61
resnet34	20.09, 2.18	20.44, 1.79	20.22, 2.40	20.22, 2.48	20.31, 2.56	19.50, 2.56	19.67, 2.60	20.29, 2.14	20.48, 2.12	19.45, 2.83	19.92, 2.48	20.22, 2.36	20.29, 2.57
resnet50	20.00, 2.26	19.83, 2.22	20.44, 1.79	20.19, 2.55	20.26, 2.57	19.40, 2.64	19.60, 2.66	20.24, 2.16	20.46, 2.10	19.38, 2.78	19.90, 2.47	20.20, 2.35	20.24, 2.64
resnet101	20.00, 2.26	19.84, 2.18	20.14, 2.38	20.44, 1.79	20.27, 2.40	19.37, 2.67	19.60, 2.65	20.21, 2.19	20.48, 2.11	19.43, 2.83	19.89, 2.44	20.13, 2.40	20.21, 2.58
resnet152	20.01, 2.23	19.83, 2.19	20.21, 2.30	20.20, 2.32	20.44, 1.79	19.35, 2.69	19.60, 2.63	20.21, 2.20	20.47, 2.07	19.37, 2.81	19.89, 2.50	20.14, 2.36	20.25, 2.49
vgg16	20.13, 2.17	19.94, 2.15	20.30, 2.26	20.25, 2.48	20.37, 2.49	20.44, 1.79	19.94, 2.32	20.36, 2.10	20.53, 2.05	19.53, 2.72	19.87, 2.46	20.22, 2.30	20.31, 2.48
vgg19	20.12, 2.19	19.98, 2.11	20.35, 2.24	20.24, 2.51	20.36, 2.45	19.79, 2.41	20.44, 1.79	20.37, 2.10	20.55, 2.00	19.60, 2.65	19.91, 2.44	20.25, 2.28	20.36, 2.44
googlenet	20.08, 2.21	19.91, 2.19	20.28, 2.21	20.25, 2.52	20.35, 2.47	19.43, 2.56	19.67, 2.58	20.43, 1.79	20.55, 1.98	19.45, 2.69	19.92, 2.44	20.19, 2.33	20.33, 2.51
densenet	20.01, 2.17	19.83, 2.20	20.18, 2.34	20.18, 2.42	20.30, 2.51	19.44, 2.53	19.62, 2.54	20.20, 2.21	20.44, 1.79	19.42, 2.75	19.89, 2.48	20.13, 2.37	20.25, 2.57
mobilenet_v2	20.10, 2.24	19.90, 2.18	20.29, 2.30	20.25, 2.50	20.33, 2.52	19.43, 2.69	19.66, 2.65	20.35, 2.13	20.53, 2.03	20.44, 1.79	19.92, 2.41	20.23, 2.31	20.32, 2.51
mobilenet_v3_small	20.07, 2.20	19.85, 2.20	20.27, 2.33	20.21, 2.51	20.31, 2.58	19.36, 2.71	19.52, 2.68	20.32, 2.20	20.51, 2.03	19.45, 2.73	20.44, 1.79	20.24, 2.34	20.24, 2.68
mobilenet_v3_large	20.12, 2.25	19.94, 2.21	20.29, 2.35	20.26, 2.51	20.33, 2.57	19.47, 2.61	19.65, 2.66	20.36, 2.16	20.53, 2.05	19.49, 2.74	19.96, 2.45	20.44, 1.79	20.31, 2.54
wide_resnet_50_2	20.00, 2.20	19.81, 2.18	20.20, 2.31	20.18, 2.46	20.29, 2.40	19.43, 2.59	19.63, 2.59	20.21, 2.14	20.45, 2.09	19.41, 2.76	19.88, 2.45	20.15, 2.42	20.44, 1.79

Table A.34: std and mean for PGD and epsilon 0.3 with target farthest from true class

source	resnet18	resnet34	resnet50	resnet101	resnet152	vgg16	vgg19	googlenet	densenet	mobilenet_v2	mobilenet_v3_small	mobilenet_v3_large	wide_resnet_50_2
resnet18	20.43, 1.81	20.46, 1.96	20.47, 1.94	20.53, 1.92	20.58, 1.88	20.21, 2.10	20.25, 2.13	20.53, 1.93	20.62, 1.83	20.35, 2.14	20.55, 2.00	20.55, 1.96	20.56, 1.88
resnet34	20.34, 2.09	20.43, 1.80	20.45, 1.96	20.52, 1.93	20.57, 1.88	20.23, 2.08	20.27, 2.14	20.51, 1.93	20.58, 1.89	20.32, 2.18	20.53, 2.00	20.54, 1.91	20.55, 1.89
resnet50	20.34, 2.11	20.48, 1.89	20.43, 1.80	20.53, 1.90	20.57, 1.85	20.30, 2.08	20.33, 2.09	20.53, 1.95	20.60, 1.84	20.36, 2.13	20.56, 1.97	20.57, 1.90	20.55, 1.88
resnet101	20.31, 2.12	20.45, 1.97	20.41, 1.95	20.43, 1.81	20.50, 1.89	20.27, 2.11	20.27, 2.12	20.50, 1.97	20.59, 1.88	20.32, 2.19	20.52, 2.04	20.53, 1.96	20.50, 1.93
resnet152	20.37, 2.08	20.48, 1.95	20.45, 1.93	20.47, 1.85	20.43, 1.80	20.26, 2.08	20.31, 2.06	20.53, 1.92	20.58, 1.85	20.36, 2.14	20.53, 2.05	20.55, 1.96	20.54, 1.90
vgg16	20.45, 2.00	20.57, 1.87	20.58, 1.86	20.64, 1.83	20.64, 1.83	20.42, 1.82	20.26, 2.03	20.62, 1.85	20.67, 1.79	20.41, 1.99	20.57, 1.98	20.60, 1.87	20.64, 1.83
vgg19	20.46, 1.98	20.57, 1.88	20.58, 1.85	20.62, 1.84	20.63, 1.83	20.27, 2.03	20.42, 1.83	20.61, 1.85	20.66, 1.81	20.40, 1.99	20.60, 1.92	20.61, 1.87	20.63, 1.84
googlenet	20.43, 1.98	20.57, 1.85	20.54, 1.87	20.60, 1.84	20.63, 1.80	20.38, 1.96	20.40, 1.95	20.43, 1.79	20.64, 1.80	20.41, 2.01	20.56, 1.99	20.57, 1.89	20.62, 1.83
densenet	20.33, 2.05	20.43, 1.92	20.43, 1.92	20.53, 1.86	20.56, 1.84	20.19, 2.13	20.25, 2.11	20.50, 1.92	20.44, 1.79	20.28, 2.14	20.52, 2.02	20.51, 1.92	20.53, 1.86
mobilenet_v2	20.47, 2.00	20.59, 1.88	20.60, 1.87	20.64, 1.82	20.66, 1.80	20.36, 2.04	20.39, 2.02	20.62, 1.83	20.68, 1.80	20.43, 1.80	20.55, 2.01	20.58, 1.89	20.65, 1.82
mobilenet_v3_small	20.48, 2.04	20.58, 1.91	20.61, 1.90	20.64, 1.85	20.65, 1.84	20.37, 2.04	20.39, 2.00	20.62, 1.84	20.67, 1.80	20.48, 2.06	20.43, 1.79	20.54, 1.94	20.65, 1.84
mobilenet_v3_large	20.42, 2.04	20.56, 1.88	20.59, 1.90	20.61, 1.86	20.65, 1.82	20.29, 2.12	20.32, 2.11	20.58, 1.91	20.67, 1.82	20.35, 2.18	20.51, 2.05	20.44, 1.79	20.64, 1.85
wide_resnet_50_2	20.31, 2.09	20.46, 1.91	20.42, 1.93	20.52, 1.87	20.52, 1.87	20.25, 2.09	20.30, 2.08	20.50, 1.95	20.55, 1.90	20.32, 2.17	20.54, 2.04	20.53, 1.95	20.43, 1.81

Table A.35: std and mean for PGD and epsilon 0.1 with target farthest from true class

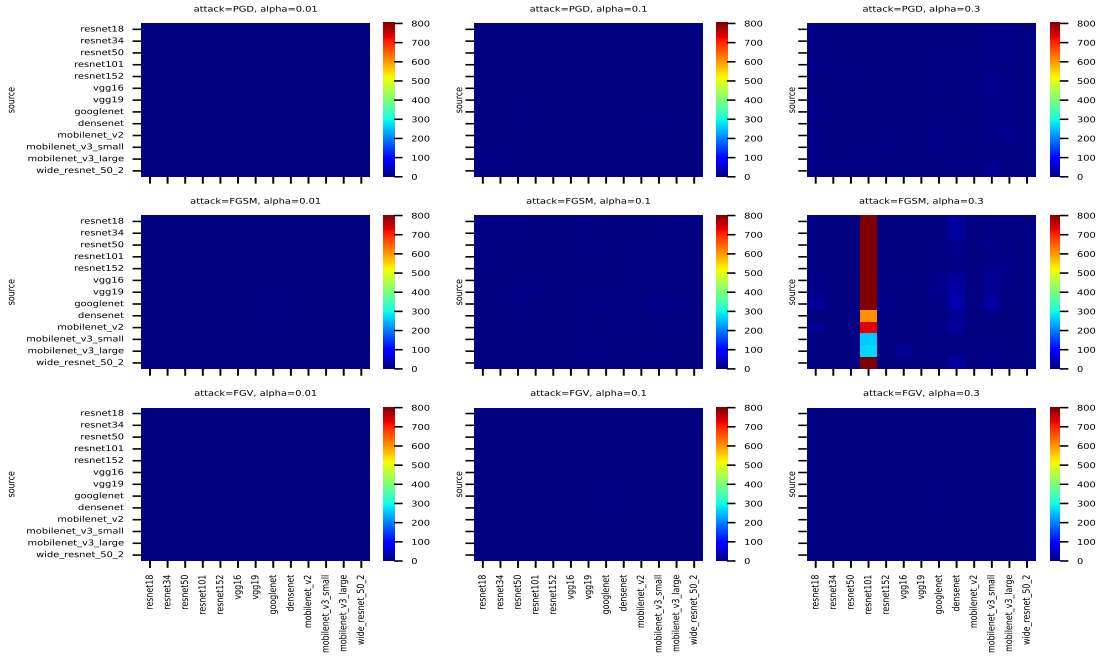


Figure A.7: targeted attack with fixed alpha and median target class

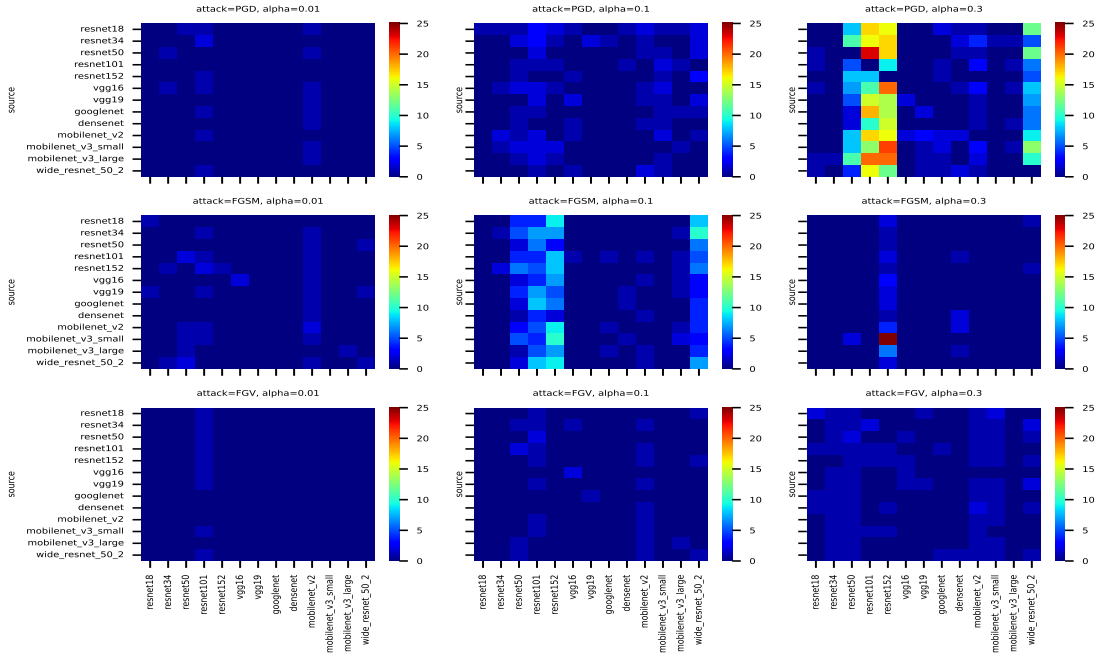


Figure A.8: targeted attack with fixed alpha and farthest target class

source	resnet18	resnet34	resnet50	resnet101	resnet152	vgg16	vgg19	googlenet	densenet	mobilenet_v2	mobilenet_v3_small	mobilenet_v3_large	wide_resnet_50_2
resnet18	20.44, 1.80	20.70, 1.75	20.69, 1.76	20.70, 1.76	20.70, 1.75	20.70, 1.75	20.70, 1.76	20.70, 1.75	20.71, 1.74	20.70, 1.75	20.70, 1.76	20.71, 1.74	20.70, 1.74
resnet34	20.69, 1.78	20.46, 1.80	20.70, 1.75	20.70, 1.77	20.70, 1.74	20.70, 1.75	20.69, 1.77	20.70, 1.75	20.71, 1.75	20.69, 1.75	20.70, 1.77	20.70, 1.74	20.70, 1.75
resnet50	20.70, 1.75	20.70, 1.76	20.46, 1.80	20.71, 1.74	20.70, 1.74	20.70, 1.75	20.69, 1.76	20.70, 1.75	20.70, 1.74	20.70, 1.76	20.70, 1.76	20.71, 1.74	20.70, 1.74
resnet101	20.69, 1.75	20.70, 1.75	20.70, 1.75	20.47, 1.79	20.70, 1.74	20.70, 1.75	20.69, 1.76	20.70, 1.75	20.70, 1.75	20.70, 1.76	20.70, 1.75	20.70, 1.75	20.70, 1.75
resnet152	20.70, 1.76	20.70, 1.76	20.70, 1.75	20.70, 1.75	20.47, 1.79	20.69, 1.76	20.69, 1.77	20.70, 1.75	20.70, 1.74	20.70, 1.75	20.70, 1.76	20.71, 1.74	20.70, 1.75
vgg16	20.70, 1.75	20.70, 1.76	20.70, 1.74	20.70, 1.75	20.71, 1.74	20.45, 1.82	20.68, 1.79	20.70, 1.75	20.70, 1.74	20.69, 1.77	20.69, 1.76	20.71, 1.74	20.70, 1.74
vgg19	20.70, 1.75	20.70, 1.75	20.70, 1.75	20.70, 1.75	20.71, 1.74	20.69, 1.75	20.45, 1.82	20.70, 1.75	20.71, 1.74	20.69, 1.76	20.69, 1.76	20.70, 1.74	20.71, 1.74
googlenet	20.70, 1.75	20.70, 1.74	20.71, 1.74	20.70, 1.75	20.70, 1.74	20.70, 1.75	20.69, 1.77	20.47, 1.78	20.71, 1.74	20.70, 1.76	20.70, 1.76	20.71, 1.74	20.70, 1.75
densenet	20.69, 1.77	20.70, 1.74	20.69, 1.75	20.70, 1.75	20.70, 1.74	20.70, 1.75	20.69, 1.77	20.70, 1.75	20.46, 1.79	20.70, 1.77	20.70, 1.76	20.70, 1.75	20.70, 1.74
mobilenet_v2	20.70, 1.75	20.70, 1.74	20.70, 1.75	20.70, 1.75	20.70, 1.74	20.70, 1.74	20.70, 1.75	20.70, 1.75	20.70, 1.74	20.45, 1.81	20.70, 1.76	20.71, 1.74	20.70, 1.74
mobilenet_v3_small	20.70, 1.75	20.70, 1.74	20.70, 1.75	20.71, 1.74	20.70, 1.74	20.70, 1.74	20.70, 1.75	20.70, 1.75	20.71, 1.74	20.70, 1.76	20.48, 1.78	20.70, 1.75	20.71, 1.74
mobilenet_v3_large	20.70, 1.75	20.70, 1.75	20.70, 1.75	20.70, 1.75	20.70, 1.74	20.70, 1.75	20.70, 1.76	20.70, 1.75	20.70, 1.74	20.70, 1.77	20.69, 1.77	20.46, 1.81	20.71, 1.74
wide_resnet_50_2	20.69, 1.76	20.70, 1.75	20.70, 1.76	20.70, 1.77	20.70, 1.75	20.69, 1.77	20.69, 1.77	20.70, 1.76	20.71, 1.74	20.70, 1.76	20.70, 1.77	20.71, 1.74	20.46, 1.80

Table A.36: std and mean for PGD and epsilon 0.01 with target farthest from true class

source	resnet18	resnet34	resnet50	resnet101	resnet152	vgg16	vgg19	googlenet	densenet	mobilenet_v2	mobilenet_v3_small	mobilenet_v3_large	wide_resnet_50_2
resnet18	21.29, 3.20	18.49, 2.49	20.50, 2.94	19.04, 2.47	18.03, 2.88	19.75, 1.17	18.79, 1.53	18.53, 1.50	19.11, 2.44	19.13, 3.50	17.76, 2.46	17.62, 2.80	18.26, 2.21
resnet34	20.94, 3.27	18.63, 2.41	20.74, 3.00	19.02, 2.49	18.11, 2.90	19.59, 1.25	18.70, 1.64	18.51, 1.48	18.96, 2.51	19.52, 3.60	17.83, 2.35	17.67, 2.68	18.20, 2.18
resnet50	20.83, 3.31	18.44, 2.65	21.33, 3.03	19.02, 2.41	17.98, 2.93	19.45, 1.28	18.46, 1.79	18.52, 1.49	19.10, 2.45	20.08, 3.51	17.75, 2.51	17.67, 2.65	18.27, 2.18
resnet101	21.11, 3.27	18.36, 2.61	20.90, 3.00	19.25, 2.55	18.30, 2.96	19.37, 1.31	18.55, 1.75	18.53, 1.43	19.11, 2.42	20.14, 3.54	17.84, 2.53	17.68, 2.63	18.33, 2.25
resnet152	20.69, 3.34	18.62, 2.74	20.67, 2.98	18.88, 2.37	18.54, 3.08	19.40, 1.32	18.42, 1.81	18.51, 1.52	19.04, 2.48	19.87, 3.52	17.78, 2.48	17.70, 2.63	18.32, 2.22
vgg16	19.41, 3.11	18.54, 2.34	19.56, 2.53	18.93, 2.36	18.25, 3.06	19.04, 1.32	18.81, 1.56	18.49, 1.61	19.35, 2.51	18.52, 3.16	18.18, 2.29	18.07, 2.59	18.63, 2.43
vgg19	19.36, 3.10	18.63, 2.31	19.55, 2.55	18.88, 2.24	18.27, 3.04	19.06, 1.30	18.76, 1.58	18.48, 1.60	19.36, 2.46	18.78, 3.22	18.18, 2.31	18.10, 2.55	18.67, 2.43
googlenet	19.25, 3.22	18.57, 2.51	19.23, 2.41	18.62, 2.12	18.07, 2.82	19.42, 1.44	18.24, 1.88	18.56, 1.46	19.08, 2.44	19.17, 3.29	17.92, 2.35	17.80, 2.64	18.41, 2.37
densenet	20.63, 3.23	18.86, 3.04	20.43, 2.86	19.52, 2.88	17.96, 2.81	19.29, 1.39	18.44, 1.98	18.69, 1.53	19.07, 2.55	20.70, 3.48	17.98, 2.29	17.81, 2.79	18.33, 2.25
mobilenet_v2	18.93, 3.00	18.66, 2.49	19.91, 2.83	18.90, 2.33	18.15, 3.08	19.60, 1.18	18.59, 1.77	18.46, 1.52	19.27, 2.60	19.11, 2.49	18.31, 2.23	17.98, 2.65	18.64, 2.32
mobilenet_v3_small	17.75, 1.85	18.18, 1.89	18.50, 1.48	18.98, 2.55	18.36, 2.91	19.00, 1.96	19.64, 1.40	18.11, 1.38	18.99, 2.53	19.02, 2.24	18.81, 1.86	17.76, 2.34	18.50, 2.38
mobilenet_v3_large	19.16, 2.92	18.27, 2.39	19.51, 2.39	19.30, 2.77	18.36, 3.01	18.97, 1.44	19.00, 1.73	18.55, 1.61	19.26, 2.67	18.74, 3.65	17.90, 2.19	17.76, 2.61	18.51, 2.34
wide_resnet_50_2	20.56, 3.25	18.87, 2.82	20.56, 2.95	18.93, 2.37	18.15, 3.00	19.50, 1.33	18.52, 1.79	18.50, 1.43	19.09, 2.32	20.17, 3.46	17.85, 2.37	17.85, 2.60	18.26, 2.16

Table A.37: std and mean for FGSM and epsilon 0.3 with target farthest from true class

source	resnet18	resnet34	resnet50	resnet101	resnet152	vgg16	vgg19	googlenet	densenet	mobilenet_v2	mobilenet_v3_small	mobilenet_v3_large	wide_resnet_50_2
resnet18	19.60, 2.02	19.86, 2.09	20.12, 2.20	20.16, 2.19	20.33, 2.27	19.36, 1.90	19.65, 2.11	20.14, 2.16	20.42, 2.02	19.54, 2.33	19.95, 2.37	20.11, 2.11	20.29, 2.28
resnet34	19.83, 2.07	19.68, 2.11	20.10, 2.24	20.14, 2.26	20.31, 2.24	19.39, 1.89	19.66, 2.09	20.16, 2.15	20.41, 2.05	19.50, 2.27	19.97, 2.30	20.06, 2.17	20.27, 2.24
resnet50	19.94, 2.12	19.94, 2.09	20.01, 2.16	20.21, 2.17	20.36, 2.23	19.59, 2.06	19.79, 2.16	20.21, 2.14	20.45, 2.03	19.59, 2.34	20.03, 2.31	20.14, 2.10	20.33, 2.20
resnet101	19.90, 2.10	19.94, 2.04	20.12, 2.18	20.00, 2.22	20.29, 2.28	19.54, 2.07	19.77, 2.14	20.24, 2.11	20.47, 1.98	19.59, 2.37	19.98, 2.34	20.13, 2.17	20.31, 2.24
resnet152	19.93, 2.14	19.97, 2.06	20.16, 2.22	20.18, 2.17	20.20, 2.29	19.58, 2.05	19.81, 2.13	20.29, 2.05	20.47, 2.00	19.60, 2.40	20.03, 2.34	20.16, 2.14	20.30, 2.27
vgg16	20.03, 2.05	20.04, 2.03	20.25, 2.14	20.33, 2.12	20.44, 2.16	19.45, 1.93	19.69, 2.01	20.31, 2.04	20.52, 1.95	19.72, 2.37	20.02, 2.23	20.18, 2.10	20.40, 2.15
vgg19	20.04, 2.10	20.05, 2.02	20.23, 2.19	20.32, 2.13	20.41, 2.16	19.53, 1.97	19.62, 2.09	20.30, 2.06	20.50, 1.98	19.74, 2.35	20.01, 2.22	20.19, 2.08	20.38, 2.18
googlenet	19.98, 2.16	20.02, 2.09	20.21, 2.18	20.26, 2.16	20.41, 2.15	19.65, 2.16	19.80, 2.17	19.93, 2.28	20.47, 1.99	19.56, 2.40	19.98, 2.36	20.13, 2.15	20.33, 2.23
densenet	19.89, 2.01	19.88, 2.05	20.15, 2.14	20.19, 2.16	20.36, 2.10	19.47, 1.88	19.62, 2.10	20.22, 2.09	20.31, 2.10	19.46, 2.20	19.76, 2.30	20.08, 2.12	20.28, 2.13
mobilenet_v2	20.06, 2.12	20.04, 2.10	20.30, 2.19	20.32, 2.16	20.43, 2.24	19.61, 2.11	19.84, 2.16	20.30, 2.10	20.49, 2.03	19.44, 2.14	19.99, 2.25	20.18, 2.14	20.38, 2.24
mobilenet_v3_small	20.02, 1.90	19.93, 2.03	20.06, 2.53	20.40, 1.94	20.10, 2.72	19.80, 1.90	19.80, 1.90	19.94, 2.12	20.38, 2.14	19.76, 2.33	19.35, 2.09	19.85, 2.34	20.22, 2.59
mobilenet_v3_large	19.91, 2.05	19.92, 2.06	20.21, 2.16	20.29, 2.08	20.31, 2.33	19.43, 1.87	19.65, 2.09	20.11, 2.13	20.45, 2.03	19.66, 2.43	19.78, 2.33	19.78, 2.23	20.32, 2.25
wide_resnet_50_2	19.97, 2.08	19.98, 2.06	20.17, 2.20	20.21, 2.22	20.35, 2.24	19.66, 2.06	19.83, 2.18	20.25, 2.08	20.42, 2.02	19.61, 2.33	19.96, 2.33	20.16, 2.11	20.19, 2.27

Table A.38: std and mean for FGSM and epsilon 0.1 with target farthest from true class

source	resnet18	resnet34	resnet50	resnet101	resnet152	vgg16	vgg19	googlenet	densenet	mobilenet_v2	mobilenet_v3_small	mobilenet_v3_large	wide_resnet_50_2
resnet18	20.29, 2.20	20.67, 1.79	20.69, 1.76	20.69, 1.76	20.70, 1.75	20.67, 1.79	20.68, 1.79	20.69, 1.79	20.70, 1.76	20.68, 1.77	20.68, 1.79	20.69, 1.76	20.70, 1.75
resnet34	20.64, 1.85	20.38, 2.11	20.68, 1.77	20.69, 1.77	20.69, 1.77	20.68, 1.78	20.67, 1.81	20.68, 1.78	20.69, 1.78	20.67, 1.80	20.68, 1.79	20.69, 1.77	20.69, 1.75
resnet50	20.65, 1.83	20.67, 1.79	20.43, 1.97	20.68, 1.78	20.69, 1.77	20.68, 1.77	20.67, 1.80	20.68, 1.79	20.69, 1.77	20.68, 1.79	20.69, 1.77	20.69, 1.76	20.68, 1.77
resnet101	20.66, 1.81	20.68, 1.78	20.67, 1.80	20.43, 2.01	20.69, 1.76	20.69, 1.77	20.68, 1.79	20.69, 1.77	20.69, 1.78	20.67, 1.80	20.69, 1.77	20.70, 1.75	20.69, 1.76
resnet152	20.65, 1.81	20.67, 1.80	20.67, 1.79	20.67, 1.81	20.44, 2.04	20.69, 1.76	20.68, 1.78	20.68, 1.78	20.69, 1.77	20.67, 1.79	20.69, 1.77	20.69, 1.77	20.68, 1.78
vgg16	20.67, 1.80	20.69, 1.76	20.69, 1.76	20.70, 1.77	20.70, 1.75	20.28, 2.21	20.60, 1.90	20.69, 1.77	20.70, 1.75	20.66, 1.80	20.69, 1.76	20.70, 1.75	20.70, 1.75
vgg19	20.66, 1.81	20.69, 1.77	20.69, 1.76	20.70, 1.76	20.70, 1.75	20.59, 1.91	20.27, 2.19	20.69, 1.78	20.70, 1.75	20.67, 1.80	20.69, 1.78	20.70, 1.76	20.70, 1.76
googlenet	20.67, 1.81	20.69, 1.78	20.69, 1.76	20.70, 1.76	20.70, 1.75	20.68, 1.77	20.68, 1.78	20.41, 2.12	20.70, 1.75	20.68, 1.79	20.69, 1.79	20.70, 1.75	20.70, 1.74
densenet	20.66, 1.81	20.69, 1.79	20.68, 1.78	20.69, 1.77	20.69, 1.76	20.68, 1.78	20.68, 1.78	20.68, 1.79	20.49, 1.97	20.68, 1.80	20.68, 1.78	20.69, 1.76	20.69, 1.75
mobilenet_v2	20.66, 1.81	20.69, 1.77	20.69, 1.77	20.70, 1.77	20.70, 1.75	20.68, 1.77	20.68, 1.79	20.69, 1.78	20.70, 1.75	20.28, 2.12	20.69, 1.78	20.69, 1.78	20.70, 1.75
mobilenet_v3_small	20.69, 1.77	20.70, 1.75	20.70, 1.76	20.70, 1.76	20.70, 1.75	20.69, 1.75	20.69, 1.76	20.69, 1.76	20.70, 1.75	20.68, 1.78	20.28, 2.10	20.69, 1.77	20.70, 1.74
mobilenet_v3_large	20.66, 1.79	20.69, 1.77	20.69, 1.77	20.69, 1.76	20.69, 1.75	20.68, 1.77	20.67, 1.78	20.68, 1.79	20.70, 1.75	20.66, 1.81	20.66, 1.79	20.35, 2.05	20.70, 1.75
wide_resnet_50_2	20.66, 1.83	20.68, 1.81	20.67, 1.82	20.69, 1.79	20.68, 1.77	20.68, 1.78	20.67, 1.80	20.69, 1.78	20.70, 1.78	20.68, 1.78	20.69, 1.78	20.69, 1.76	20.42, 2.08

Table A.39: std and mean for FGSM and epsilon 0.01 with target farthest from true class

source	resnet18	resnet34	resnet50	resnet101	resnet152	vgg16	vgg19	googlenet	densenet	mobilenet_v2	mobilenet_v3_small	mobilenet_v3_large	wide_resnet_50_2
resnet18	20.11, 2.28	20.54, 1.92	20.57, 1.87	20.59, 1.86	20.62, 1.82	20.50, 1.98	20.53, 1.95	20.58, 1.92	20.62, 1.85	20.53, 1.94	20.60, 1.89	20.62, 1.83	20.60, 1.83
resnet34	20.49, 2.03	20.21, 2.18	20.54, 1.89	20.58, 1.90	20.61, 1.84	20.53, 1.95	20.55, 1.94	20.59, 1.90	20.63, 1.83	20.53, 1.93	20.60, 1.89	20.62, 1.84	20.59, 1.87
resnet50	20.54, 1.97	20.57, 1.89	20.32, 2.01	20.60, 1.83	20.63, 1.81	20.59, 1.88	20.60, 1.88	20.62, 1.85	20.64, 1.83	20.57, 1.88	20.63, 1.84	20.64, 1.82	20.62, 1.82
resnet101	20.56, 1.94	20.59, 1.88	20.55, 1.88	20.36, 1.98	20.61, 1.84	20.60, 1.87	20.60, 1.86	20.60, 1.88	20.64, 1.84	20.57, 1.90	20.62, 1.87	20.65, 1.81	20.62, 1.82
resnet152	20.57, 1.91	20.58, 1.88	20.58, 1.85	20.59, 1.87	20.38, 2.05	20.59, 1.88	20.60, 1.84	20.63, 1.86	20.64, 1.85	20.59, 1.89	20.62, 1.86	20.64, 1.82	20.61, 1.84
vgg16	20.58, 1.91	20.61, 1.86	20.62, 1.82	20.65, 1.80	20.66, 1.78	20.17, 2.23	20.40, 2.04	20.61, 1.87	20.65, 1.79	20.56, 1.88	20.61, 1.86	20.63, 1.81	20.65, 1.78
vgg19	20.57, 1.94	20.61, 1.88	20.62, 1.81	20.65, 1.79	20.66, 1.79	20.43, 2.02	20.16, 2.23	20.64, 1.84	20.66, 1.79	20.58, 1.87	20.63, 1.85	20.64, 1.82	20.65, 1.81
googlenet	20.56, 1.97	20.61, 1.89	20.63, 1.83	20.64, 1.81	20.65, 1.80	20.58, 1.93	20.59, 1.89	20.31, 2.19	20.66, 1.79	20.58, 1.92	20.60, 1.90	20.63, 1.85	20.64, 1.83
densenet	20.57, 1.91	20.58, 1.89	20.60, 1.86	20.61, 1.83	20.63, 1.83	20.60, 1.87	20.59, 1.88	20.62, 1.86	20.43, 1.98	20.57, 1.89	20.62, 1.87	20.62, 1.81	20.62, 1.82
mobilenet_v2	20.57, 1.92	20.64, 1.83	20.65, 1.81	20.65, 1.79	20.67, 1.79	20.59, 1.85	20.61, 1.86	20.63, 1.84	20.67, 1.78	20.09, 2.14	20.58, 1.90	20.62, 1.83	20.67, 1.78
mobilenet_v3_small	20.62, 1.84	20.64, 1.80	20.66, 1.80	20.67, 1.79	20.66, 1.78	20.62, 1.82	20.62, 1.82	20.62, 1.81	20.67, 1.78	20.54, 1.89	20.14, 2.05	20.60, 1.84	20.68, 1.77
mobilenet_v3_large	20.61, 1.89	20.64, 1.83	20.66, 1.81	20.67, 1.78	20.67, 1.79	20.61, 1.85	20.61, 1.88	20.63, 1.85	20.66, 1.79	20.56, 1.91	20.57, 1.90	20.26, 2.10	20.68, 1.78
wide_resnet_50_2	20.56, 1.94	20.60, 1.87	20.57, 1.86	20.63, 1.83	20.63, 1.80	20.60, 1.88	20.60, 1.87	20.62, 1.87	20.64, 1.84	20.59, 1.87	20.63, 1.85	20.63, 1.82	20.36, 2.05

Table A.40: std and mean for FGV and epsilon 0.3 with target farthest from true class

source	resnet18	resnet34	resnet50	resnet101	resnet152	vgg16	vgg19	googlenet	densenet	mobilenet_v2	mobilenet_v3_small	mobilenet_v3_large	wide_resnet_50_2
resnet18	20.35, 2.14	20.68, 1.78	20.68, 1.76	20.69, 1.78	20.70, 1.76	20.67, 1.79	20.68, 1.79	20.68, 1.80	20.70, 1.76	20.67, 1.79	20.69, 1.78	20.70, 1.76	20.70, 1.76
resnet34	20.64, 1.87	20.39, 2.10	20.68, 1.78	20.69, 1.78	20.69, 1.76	20.68, 1.79	20.68, 1.80	20.68, 1.81	20.69, 1.79	20.67, 1.79	20.69, 1.76	20.70, 1.75	20.70, 1.74
resnet50	20.66, 1.84	20.68, 1.78	20.51, 1.93	20.68, 1.79	20.69, 1.77	20.69, 1.76	20.69, 1.79	20.69, 1.78	20.69, 1.78	20.68, 1.78	20.69, 1.76	20.70, 1.76	20.69, 1.76
resnet101	20.68, 1.80	20.68, 1.78	20.68, 1.80	20.49, 1.95	20.69, 1.76	20.69, 1.77	20.69, 1.77	20.69, 1.77	20.69, 1.78	20.68, 1.80	20.69, 1.77	20.70, 1.76	20.70, 1.75
resnet152	20.67, 1.79	20.69, 1.77	20.68, 1.78	20.68, 1.78	20.51, 1.97	20.69, 1.76	20.69, 1.77	20.69, 1.78	20.68, 1.78	20.68, 1.77	20.70, 1.77	20.70, 1.76	20.70, 1.77
vgg16	20.67, 1.80	20.69, 1.76	20.69, 1.76	20.70, 1.75	20.69, 1.75	20.36, 2.17	20.62, 1.86	20.69, 1.77	20.70, 1.75	20.67, 1.79	20.69, 1.77	20.70, 1.75	20.70, 1.75
vgg19	20.68, 1.79	20.69, 1.76	20.69, 1.76	20.70, 1.76	20.70, 1.75	20.63, 1.87	20.36, 2.12	20.70, 1.77	20.70, 1.76	20.67, 1.79	20.70, 1.76	20.70, 1.75	20.70, 1.74
googlenet	20.67, 1.81	20.69, 1.79	20.69, 1.76	20.70, 1.76	20.70, 1.75	20.69, 1.78	20.68, 1.81	20.43, 2.11	20.70, 1.75	20.67, 1.78	20.69, 1.78	20.71, 1.74	20.70, 1.75
densenet	20.68, 1.78	20.69, 1.78	20.68, 1.77	20.69, 1.76	20.69, 1.76	20.69, 1.77	20.69, 1.80	20.69, 1.77	20.54, 1.91	20.68, 1.80	20.69, 1.76	20.70, 1.75	20.70, 1.75
mobilenet_v2	20.68, 1.78	20.69, 1.76	20.69, 1.76	20.70, 1.76	20.69, 1.75	20.68, 1.76	20.69, 1.77	20.69, 1.77	20.70, 1.75	20.34, 2.11	20.69, 1.78	20.70, 1.76	20.70, 1.74
mobilenet_v3_small	20.68, 1.77	20.70, 1.74	20.70, 1.75	20.70, 1.76	20.70, 1.74	20.70, 1.75	20.69, 1.75	20.70, 1.75	20.70, 1.74	20.69, 1.77	20.37, 1.99	20.70, 1.76	20.70, 1.74
mobilenet_v3_large	20.68, 1.79	20.70, 1.75	20.69, 1.77	20.70, 1.75	20.70, 1.75	20.69, 1.76	20.69, 1.77	20.69, 1.76	20.70, 1.75	20.67, 1.79	20.68, 1.77	20.44, 2.03	20.70, 1.75
wide_resnet_50_2	20.66, 1.83	20.69, 1.78	20.68, 1.79	20.69, 1.78	20.69, 1.77	20.70, 1.75	20.69, 1.78	20.69, 1.77	20.70, 1.76	20.68, 1.80	20.70, 1.76	20.70, 1.75	20.49, 1.98

Table A.41: std and mean for FGV and epsilon 0.1 with target farthest from true class

source	resnet18	resnet34	resnet50	resnet101	resnet152	vgg16	vgg19	googlenet	densenet	mobilenet_v2	mobilenet_v3_small	mobilenet_v3_large	wide_resnet_50_2
resnet18	20.67, 1.80	20.70, 1.74	20.71, 1.75	20.70, 1.75	20.71, 1.74	20.71, 1.74	20.71, 1.74	20.71, 1.74	20.71, 1.74	20.70, 1.74	20.71, 1.74	20.71, 1.74	20.71, 1.74
resnet34	20.71, 1.74	20.68, 1.78	20.71, 1.75	20.70, 1.75	20.71, 1.74	20.71, 1.74	20.71, 1.74	20.70, 1.75	20.71, 1.74	20.70, 1.74	20.70, 1.74	20.71, 1.74	20.71, 1.74
resnet50	20.71, 1.74	20.71, 1.74	20.68, 1.77	20.70, 1.75	20.70, 1.74	20.71, 1.74	20.71, 1.74	20.71, 1.75	20.71, 1.74	20.70, 1.74	20.71, 1.74	20.71, 1.74	20.71, 1.74
resnet101	20.71, 1.74	20.70, 1.74	20.71, 1.74	20.69, 1.77	20.71, 1.74	20.71, 1.74	20.70, 1.75	20.71, 1.74	20.71, 1.74	20.70, 1.74	20.71, 1.74	20.71, 1.74	20.71, 1.74
resnet152	20.71, 1.74	20.71, 1.74	20.70, 1.75	20.70, 1.75	20.69, 1.77	20.71, 1.74	20.71, 1.74	20.71, 1.75	20.70, 1.74	20.70, 1.74	20.71, 1.74	20.71, 1.74	20.71, 1.74
vgg16	20.71, 1.74	20.70, 1.74	20.71, 1.74	20.70, 1.75	20.71, 1.74	20.67, 1.81	20.71, 1.75	20.71, 1.74	20.71, 1.74	20.70, 1.74	20.71, 1.74	20.71, 1.74	20.71, 1.74
vgg19	20.71, 1.74	20.71, 1.74	20.71, 1.74	20.70, 1.75	20.71, 1.74	20.71, 1.74	20.68, 1.80	20.71, 1.74	20.71, 1.74	20.70, 1.74	20.71, 1.74	20.71, 1.74	20.71, 1.74
googlenet	20.71, 1.74	20.70, 1.74	20.71, 1.74	20.71, 1.74	20.71, 1.74	20.71, 1.74	20.70, 1.74	20.67, 1.81	20.71, 1.74	20.70, 1.74	20.71, 1.74	20.71, 1.74	20.71, 1.74
densenet	20.71, 1.74	20.70, 1.74	20.71, 1.74	20.71, 1.74	20.71, 1.74	20.70, 1.74	20.71, 1.75	20.71, 1.74	20.69, 1.77	20.70, 1.74	20.71, 1.74	20.71, 1.74	20.71, 1.74
mobilenet_v2	20.71, 1.74	20.70, 1.74	20.71, 1.74	20.71, 1.74	20.71, 1.74	20.71, 1.74	20.71, 1.74	20.71, 1.74	20.71, 1.74	20.66, 1.82	20.71, 1.74	20.71, 1.74	20.71, 1.74
mobilenet_v3_small	20.71, 1.74	20.71, 1.74	20.71, 1.74	20.70, 1.75	20.71, 1.74	20.71, 1.74	20.71, 1.74	20.71, 1.74	20.71, 1.74	20.70, 1.74	20.67, 1.79	20.71, 1.74	20.71, 1.74
mobilenet_v3_large	20.71, 1.74	20.70, 1.74	20.71, 1.74	20.71, 1.74	20.71, 1.74	20.71, 1.74	20.71, 1.74	20.71, 1.74	20.71, 1.74	20.71, 1.74	20.71, 1.74	20.70, 1.77	20.71, 1.74
wide_resnet_50_2	20.71, 1.74	20.71, 1.74	20.71, 1.74	20.70, 1.75	20.71, 1.74	20.71, 1.74	20.71, 1.74	20.71, 1.74	20.70, 1.74	20.71, 1.74	20.71, 1.74	20.71, 1.74	20.70, 1.75

Table A.42: std and mean for FGV and epsilon 0.01 with target farthest from true class

source	resnet18	resnet34	resnet50	resnet101	resnet152	vgg16	vgg19	googlenet	densenet	mobilenet_v2	mobilenet_v3_small	mobilenet_v3_large	wide_resnet_50_2
resnet18	12.57, 2.20	12.47, 2.54	12.66, 2.44	12.73, 2.53	12.84, 2.45	12.47, 2.95	12.43, 2.88	12.70, 2.48	12.80, 2.33	12.49, 3.07	12.51, 2.66	12.70, 2.58	12.80, 2.42
resnet34	12.70, 2.55	12.57, 2.20	12.64, 2.46	12.76, 2.54	12.86, 2.49	12.54, 2.94	12.45, 2.88	12.68, 2.46	12.80, 2.34	12.50, 3.12	12.53, 2.69	12.68, 2.62	12.78, 2.41
resnet50	12.61, 2.60	12.45, 2.59	12.58, 2.21	12.76, 2.53	12.87, 2.48	12.49, 2.96	12.45, 2.93	12.67, 2.50	12.80, 2.39	12.43, 3.16	12.50, 2.69	12.69, 2.60	12.80, 2.44
resnet101	12.63, 2.62	12.44, 2.60	12.64, 2.45	12.58, 2.21	12.80, 2.43	12.50, 2.95	12.45, 2.91	12.68, 2.46	12.80, 2.34	12.49, 3.18	12.51, 2.69	12.65, 2.60	12.76, 2.46
resnet152	12.62, 2.60	12.42, 2.60	12.62, 2.43	12.72, 2.49	12.58, 2.21	12.49, 2.97	12.40, 2.91	12.71, 2.48	12.79, 2.34	12.47, 3.10	12.53, 2.68	12.67, 2.59	12.74, 2.39
vgg16	12.60, 2.57	12.44, 2.56	12.68, 2.44	12.73, 2.52	12.86, 2.45	12.57, 2.20	12.45, 2.63	12.69, 2.42	12.82, 2.31	12.48, 3.03	12.48, 2.72	12.64, 2.59	12.74, 2.42
vgg19	12.64, 2.58	12.48, 2.53	12.68, 2.44	12.73, 2.48	12.86, 2.45	12.57, 2.67	12.57, 2.20	12.70, 2.43	12.81, 2.32	12.47, 3.02	12.48, 2.68	12.70, 2.57	12.77, 2.41
googlenet	12.56, 2.62	12.47, 2.55	12.65, 2.41	12.74, 2.50	12.85, 2.48	12.49, 2.92	12.38, 2.85	12.57, 2.21	12.82, 2.32	12.38, 3.07	12.51, 2.68	12.64, 2.60	12.76, 2.43
densenet	12.58, 2.58	12.44, 2.58	12.65, 2.44	12.74, 2.52	12.80, 2.44	12.46, 2.90	12.41, 2.93	12.67, 2.44	12.57, 2.20	12.44, 3.13	12.49, 2.71	12.66, 2.62	12.76, 2.43
mobilenet_v2	12.62, 2.59	12.44, 2.58	12.68, 2.45	12.75, 2.51	12.83, 2.49	12.51, 2.89	12.41, 2.90	12.70, 2.44	12.82, 2.33	12.57, 2.20	12.49, 2.68	12.67, 2.58	12.77, 2.42
mobilenet_v3_small	12.55, 2.61	12.38, 2.61	12.68, 2.45	12.72, 2.57	12.83, 2.50	12.41, 2.91	12.34, 2.91	12.70, 2.47	12.80, 2.35	12.35, 3.07	12.57, 2.20	12.69, 2.55	12.74, 2.46
mobilenet_v3_large	12.67, 2.61	12.46, 2.56	12.69, 2.41	12.72, 2.51	12.84, 2.49	12.48, 2.94	12.43, 2.85	12.74, 2.41	12.81, 2.33	12.49, 3.03	12.57, 2.65	12.57, 2.20	12.76, 2.44
wide_resnet_50_2	12.61, 2.62	12.39, 2.59	12.61, 2.43	12.72, 2.49	12.80, 2.43	12.49, 2.96	12.43, 2.87	12.64, 2.47	12.77, 2.34	12.48, 3.09	12.47, 2.69	12.68, 2.60	12.57, 2.20

Table A.43: std and mean for PGD and epsilon 0.3 with target median from true class

source	resnet18	resnet34	resnet50	resnet101	resnet152	vgg16	vgg19	googlenet	densenet	mobilenet_v2	mobilenet_v3_small	mobilenet_v3_large	wide_resnet_50_2
resnet18	12.58, 2.21	12.78, 2.24	12.74, 2.23	12.78, 2.20	12.80, 2.18	12.72, 2.32	12.67, 2.32	12.81, 2.23	12.82, 2.20	12.75, 2.34	12.82, 2.25	12.79, 2.23	12.81, 2.21
resnet34	12.77, 2.31	12.58, 2.21	12.76, 2.22	12.79, 2.22	12.78, 2.19	12.68, 2.32	12.69, 2.31	12.80, 2.24	12.83, 2.20	12.80, 2.36	12.82, 2.26	12.80, 2.21	12.81, 2.20
resnet50	12.79, 2.30	12.77, 2.22	12.58, 2.21	12.77, 2.25	12.80, 2.19	12.70, 2.27	12.71, 2.29	12.81, 2.24	12.82, 2.19	12.79, 2.30	12.83, 2.26	12.81, 2.21	12.81, 2.20
resnet101	12.76, 2.29	12.75, 2.24	12.72, 2.23	12.58, 2.22	12.74, 2.20	12.70, 2.29	12.69, 2.29	12.81, 2.24	12.82, 2.20	12.76, 2.33	12.83, 2.27	12.79, 2.21	12.78, 2.21
resnet152	12.74, 2.30	12.76, 2.25	12.71, 2.24	12.73, 2.23	12.58, 2.21	12.71, 2.31	12.71, 2.29	12.80, 2.24	12.82, 2.18	12.78, 2.33	12.80, 2.26	12.80, 2.21	12.78, 2.20
vgg16	12.75, 2.23	12.80, 2.20	12.80, 2.19	12.81, 2.18	12.81, 2.18	12.58, 2.22	12.64, 2.27	12.85, 2.20	12.85, 2.18	12.73, 2.29	12.82, 2.24	12.82, 2.20	12.81, 2.17
vgg19	12.79, 2.24	12.80, 2.19	12.80, 2.20	12.80, 2.20	12.83, 2.19	12.64, 2.27	12.58, 2.22	12.81, 2.19	12.83, 2.18	12.72, 2.27	12.81, 2.24	12.83, 2.20	12.83, 2.17
googlenet	12.77, 2.23	12.80, 2.19	12.79, 2.21	12.79, 2.19	12.81, 2.17	12.72, 2.24	12.71, 2.24	12.57, 2.21	12.82, 2.18	12.77, 2.29	12.82, 2.26	12.83, 2.21	12.83, 2.19
densenet	12.75, 2.31	12.75, 2.24	12.72, 2.24	12.77, 2.21	12.78, 2.18	12.65, 2.33	12.65, 2.31	12.79, 2.24	12.57, 2.20	12.72, 2.32	12.80, 2.30	12.79, 2.22	12.79, 2.20
mobilenet_v2	12.81, 2.24	12.80, 2.21	12.82, 2.19	12.82, 2.19	12.82, 2.18	12.73, 2.24	12.72, 2.24	12.84, 2.19	12.85, 2.16	12.57, 2.21	12.82, 2.24	12.82, 2.21	12.83, 2.15
mobilenet_v3_small	12.79, 2.24	12.81, 2.20	12.81, 2.22	12.81, 2.20	12.83, 2.19	12.68, 2.27	12.67, 2.27	12.82, 2.20	12.82, 2.19	12.79, 2.29	12.57, 2.21	12.80, 2.23	12.83, 2.18
mobilenet_v3_large	12.79, 2.27	12.80, 2.21	12.81, 2.20	12.81, 2.19	12.82, 2.19	12.73, 2.32	12.70, 2.29	12.82, 2.21	12.84, 2.19	12.78, 2.33	12.81, 2.27	12.57, 2.20	12.82, 2.18
wide_resnet_50_2	12.75, 2.30	12.74, 2.24	12.71, 2.22	12.75, 2.23	12.76, 2.19	12.66, 2.30	12.70, 2.28	12.79, 2.24	12.81, 2.23	12.75, 2.34	12.81, 2.27	12.79, 2.22	12.58, 2.21

Table A.44: std and mean for PGD and epsilon 0.1 with target median from true class

source	resnet18	resnet34	resnet50	resnet101	resnet152	vgg16	vgg19	googlenet	densenet	mobilenet_v2	mobilenet_v3_small	mobilenet_v3_large	wide_resnet_50_2
resnet18	12.59, 2.20	12.84, 2.15	12.84, 2.15	12.84, 2.15	12.84, 2.15	12.85, 2.16	12.85, 2.16	12.84, 2.16	12.85, 2.14	12.85, 2.15	12.84, 2.15	12.84, 2.14	12.84, 2.15
resnet34	12.84, 2.15	12.60, 2.21	12.84, 2.15	12.84, 2.15	12.84, 2.15	12.84, 2.15	12.84, 2.15	12.84, 2.15	12.84, 2.15	12.85, 2.15	12.85, 2.15	12.85, 2.15	12.84, 2.15
resnet50	12.83, 2.14	12.85, 2.15	12.60, 2.20	12.84, 2.15	12.84, 2.15	12.84, 2.15	12.84, 2.15	12.84, 2.15	12.85, 2.15	12.84, 2.15	12.85, 2.15	12.84, 2.15	12.84, 2.15
resnet101	12.84, 2.15	12.85, 2.15	12.84, 2.15	12.62, 2.20	12.84, 2.14	12.84, 2.15	12.85, 2.16	12.84, 2.15	12.85, 2.14	12.84, 2.16	12.84, 2.15	12.85, 2.15	12.84, 2.14
resnet152	12.84, 2.15	12.85, 2.15	12.84, 2.15	12.84, 2.15	12.61, 2.20	12.84, 2.15	12.84, 2.16	12.84, 2.15	12.84, 2.15	12.85, 2.16	12.84, 2.15	12.85, 2.15	12.84, 2.14
vgg16	12.84, 2.15	12.84, 2.15	12.84, 2.15	12.84, 2.15	12.84, 2.14	12.59, 2.22	12.84, 2.16	12.84, 2.15	12.84, 2.14	12.84, 2.15	12.84, 2.15	12.84, 2.15	12.84, 2.14
vgg19	12.84, 2.15	12.84, 2.15	12.84, 2.15	12.84, 2.15	12.84, 2.14	12.84, 2.16	12.61, 2.22	12.84, 2.15	12.84, 2.14	12.84, 2.15	12.84, 2.15	12.84, 2.15	12.84, 2.14
googlenet	12.84, 2.15	12.84, 2.15	12.84, 2.14	12.84, 2.15	12.84, 2.14	12.84, 2.14	12.84, 2.16	12.61, 2.18	12.84, 2.14	12.84, 2.15	12.84, 2.15	12.85, 2.14	12.84, 2.15
densenet	12.84, 2.15	12.85, 2.15	12.84, 2.15	12.84, 2.15	12.84, 2.15	12.84, 2.15	12.84, 2.16	12.84, 2.15	12.61, 2.19	12.84, 2.16	12.84, 2.15	12.85, 2.15	12.84, 2.15
mobilenet_v2	12.84, 2.15	12.84, 2.15	12.84, 2.15	12.84, 2.15	12.84, 2.15	12.84, 2.15	12.85, 2.15	12.85, 2.15	12.84, 2.14	12.59, 2.21	12.84, 2.16	12.85, 2.14	12.84, 2.15
mobilenet_v3_small	12.84, 2.14	12.84, 2.15	12.84, 2.14	12.84, 2.15	12.84, 2.14	12.84, 2.15	12.84, 2.15	12.84, 2.15	12.84, 2.14	12.84, 2.15	12.61, 2.19	12.85, 2.15	12.84, 2.14
mobilenet_v3_large	12.84, 2.15	12.84, 2.15	12.84, 2.15	12.84, 2.15	12.84, 2.14	12.85, 2.15	12.85, 2.15	12.84, 2.15	12.84, 2.15	12.84, 2.16	12.84, 2.15	12.61, 2.22	12.84, 2.14
wide_resnet_50_2	12.84, 2.15	12.85, 2.15	12.84, 2.15	12.84, 2.15	12.84, 2.15	12.84, 2.15	12.85, 2.16	12.84, 2.15	12.85, 2.15	12.85, 2.15	12.84, 2.16	12.85, 2.15	12.61, 2.20

Table A.45: std and mean for PGD and epsilon 0.01 with target median from true class

source	resnet18	resnet34	resnet50	resnet101	resnet152	vgg16	vgg19	googlenet	densenet	mobilenet_v2	mobilenet_v3_small	mobilenet_v3_large	wide_resnet_50_2
resnet18	16.99, 4.06	12.34, 4.09	15.72, 4.33	12.33, 5.23	11.89, 3.56	12.54, 3.87	11.91, 3.56	11.40, 3.41	12.23, 3.40	14.24, 4.47	11.39, 3.19	11.51, 3.44	11.76, 3.18
resnet34	16.60, 4.30	12.51, 4.08	16.14, 4.21	12.37, 5.19	11.92, 3.62	12.45, 3.78	11.82, 3.45	11.45, 3.43	12.24, 3.56	14.83, 4.54	11.33, 3.20	11.43, 3.35	11.78, 3.27
resnet50	16.38, 4.42	12.34, 4.24	17.01, 3.88	12.35, 5.18	11.92, 3.69	12.37, 3.70	11.77, 3.44	11.38, 3.39	12.13, 3.35	15.45, 4.43	11.37, 3.14	11.36, 3.34	11.78, 3.22
resnet101	16.87, 4.17	12.26, 4.21	16.28, 4.19	12.74, 5.37	12.34, 3.94	12.38, 3.59	11.73, 3.46	11.37, 3.39	12.18, 3.28	15.58, 4.38	11.39, 3.15	11.44, 3.34	11.78, 3.23
resnet152	16.27, 4.47	12.66, 4.36	15.94, 4.30	12.14, 5.30	12.71, 4.15	12.33, 3.69	11.70, 3.53	11.42, 3.44	12.19, 3.46	15.24, 4.46	11.43, 3.19	11.40, 3.32	11.87, 3.16
vgg16	14.20, 4.78	12.47, 4.04	13.94, 4.29	11.91, 5.04	12.10, 3.47	11.87, 3.71	11.80, 3.69	11.28, 3.42	12.27, 3.30	13.20, 4.28	11.51, 3.04	11.53, 3.21	11.89, 3.01
vgg19	14.15, 4.79	12.63, 4.09	13.95, 4.26	11.81, 4.91	12.10, 3.53	11.88, 3.74	11.80, 3.65	11.32, 3.40	12.26, 3.28	13.36, 4.36	11.54, 3.05	11.55, 3.17	11.90, 3.02
googlenet	14.05, 4.86	12.63, 4.22	13.36, 4.07	11.59, 4.64	11.62, 3.46	12.42, 3.71	11.53, 3.54	11.36, 3.55	12.06, 3.36	14.27, 4.49	11.46, 3.16	11.57, 3.29	11.67, 3.01
densenet	16.09, 4.46	13.23, 4.63	15.52, 4.35	13.43, 5.24	11.64, 3.59	12.34, 3.43	11.56, 3.16	11.65, 3.33	12.43, 3.67	16.24, 4.29	11.53, 3.24	11.59, 3.43	11.82, 3.41
mobilenet_v2	13.38, 4.70	12.77, 4.21	14.61, 4.47	12.01, 4.84	12.19, 3.42	12.35, 3.92	12.23, 3.45	11.25, 3.36	12.38, 3.28	12.96, 4.10	11.54, 3.01	11.52, 3.19	11.87, 3.02
mobilenet_v3_small	11.07, 3.15	11.76, 3.42	11.59, 3.15	12.51, 4.40	11.99, 3.24	12.15, 2.99	12.52, 2.86	11.14, 3.11	12.71, 3.22	13.03, 3.98	12.39, 3.22	11.27, 3.12	11.83, 3.06
mobilenet_v3_large	13.85, 4.67	12.10, 3.93	13.71, 4.20	13.29, 4.62	12.23, 3.25	11.83, 3.66	11.95, 3.57	11.62, 3.33	12.66, 3.39	13.94, 4.54	11.23, 3.03	11.51, 3.28	11.83, 3.18
wide_resnet_50_2	16.02, 4.54	13.12, 4.48	15.85, 4.31	12.15, 5.21	12.05, 3.79	12.41, 3.75	11.85, 3.54	11.33, 3.41	12.09, 3.32	15.52, 4.42	11.41, 3.21	11.51, 3.35	11.86, 3.19

Table A.46: std and mean for FGSM and epsilon 0.3 with target median from true class

source	resnet18	resnet34	resnet50	resnet101	resnet152	vgg16	vgg19	googlenet	densenet	mobilenet_v2	mobilenet_v3_small	mobilenet_v3_large	wide_resnet_50_2
resnet18	12.40, 2.80	12.42, 2.52	12.51, 2.42	12.61, 2.44	12.73, 2.38	12.11, 3.08	12.30, 2.74	12.60, 2.43	12.72, 2.36	12.36, 2.73	12.55, 2.61	12.53, 2.46	12.72, 2.36
resnet34	12.46, 2.67	12.40, 2.60	12.55, 2.43	12.62, 2.47	12.72, 2.35	12.12, 2.99	12.32, 2.70	12.62, 2.43	12.74, 2.36	12.27, 2.74	12.55, 2.58	12.52, 2.52	12.70, 2.39
resnet50	12.55, 2.63	12.46, 2.52	12.49, 2.45	12.62, 2.46	12.74, 2.37	12.31, 2.73	12.34, 2.62	12.65, 2.41	12.76, 2.34	12.33, 2.73	12.56, 2.56	12.57, 2.48	12.73, 2.36
resnet101	12.51, 2.70	12.43, 2.49	12.55, 2.43	12.54, 2.60	12.71, 2.40	12.28, 2.76	12.34, 2.63	12.65, 2.41	12.75, 2.31	12.36, 2.70	12.58, 2.53	12.57, 2.47	12.71, 2.38
resnet152	12.54, 2.63	12.47, 2.47	12.58, 2.41	12.61, 2.48	12.71, 2.43	12.28, 2.73	12.36, 2.61	12.65, 2.40	12.79, 2.33	12.40, 2.72	12.64, 2.55	12.58, 2.46	12.70, 2.36
vgg16	12.54, 2.51	12.48, 2.43	12.60, 2.37	12.68, 2.34	12.77, 2.34	12.13, 2.78	12.23, 2.66	12.66, 2.37	12.79, 2.30	12.43, 2.69	12.52, 2.55	12.57, 2.41	12.72, 2.32
vgg19	12.53, 2.51	12.47, 2.44	12.62, 2.39	12.67, 2.36	12.76, 2.35	12.18, 2.77	12.25, 2.66	12.66, 2.36	12.78, 2.30	12.42, 2.67	12.51, 2.54	12.58, 2.40	12.72, 2.31
googlenet	12.50, 2.56	12.50, 2.45	12.60, 2.39	12.63, 2.41	12.72, 2.36	12.35, 2.68	12.33, 2.63	12.51, 2.56	12.76, 2.32	12.28, 2.73	12.52, 2.59	12.56, 2.44	12.71, 2.36
densenet	12.53, 2.70	12.39, 2.54	12.53, 2.40	12.59, 2.47	12.71, 2.34	12.17, 2.92	12.35, 2.76	12.62, 2.44	12.71, 2.44	12.26, 2.77	12.44, 2.68	12.53, 2.53	12.63, 2.42
mobilenet_v2	12.57, 2.50	12.50, 2.47	12.65, 2.35	12.69, 2.38	12.78, 2.37	12.35, 2.80	12.40, 2.64	12.66, 2.38	12.77, 2.31	12.43, 2.89	12.49, 2.54	12.56, 2.45	12.72, 2.34
mobilenet_v3_small	12.65, 2.55	12.45, 2.46	12.61, 2.44	12.78, 2.27	12.80, 2.49	12.44, 2.55	12.43, 2.64	12.40, 2.56	12.75, 2.34	12.60, 2.64	12.26, 2.88	12.37, 2.60	12.73, 2.43
mobilenet_v3_large	12.57, 2.68	12.40, 2.49	12.62, 2.35	12.70, 2.36	12.74, 2.37	12.08, 3.06	12.30, 2.77	12.59, 2.44	12.76, 2.33	12.51, 2.70	12.45, 2.73	12.37, 2.61	12.73, 2.34
wide_resnet_50_2	12.54, 2.64	12.48, 2.49	12.56, 2.40	12.63, 2.45	12.73, 2.36	12.36, 2.68	12.40, 2.59	12.64, 2.42	12.71, 2.33	12.35, 2.73	12.56, 2.61	12.57, 2.47	12.66, 2.45

Table A.47: std and mean for FGSM and epsilon 0.1 with target median from true class

source	resnet18	resnet34	resnet50	resnet101	resnet152	vgg16	vgg19	googlenet	densenet	mobilenet_v2	mobilenet_v3_small	mobilenet_v3_large	wide_resnet_50_2
resnet18	12.82, 2.37	12.84, 2.16	12.84, 2.16	12.84, 2.16	12.85, 2.15	12.84, 2.19	12.84, 2.17	12.84, 2.17	12.85, 2.16	12.84, 2.17	12.85, 2.16	12.84, 2.15	12.84, 2.15
resnet34	12.83, 2.20	12.81, 2.36	12.84, 2.16	12.85, 2.16	12.84, 2.16	12.84, 2.17	12.84, 2.17	12.84, 2.17	12.84, 2.17	12.85, 2.17	12.84, 2.17	12.85, 2.15	12.84, 2.15
resnet50	12.83, 2.18	12.85, 2.16	12.79, 2.31	12.85, 2.17	12.85, 2.16	12.84, 2.17	12.84, 2.17	12.84, 2.17	12.84, 2.16	12.85, 2.17	12.84, 2.16	12.84, 2.14	12.84, 2.15
resnet101	12.83, 2.17	12.84, 2.17	12.83, 2.17	12.79, 2.32	12.84, 2.16	12.84, 2.17	12.84, 2.16	12.84, 2.16	12.85, 2.17	12.84, 2.17	12.84, 2.16	12.85, 2.15	12.83, 2.16
resnet152	12.83, 2.17	12.84, 2.17	12.83, 2.17	12.85, 2.17	12.80, 2.33	12.84, 2.16	12.84, 2.17	12.83, 2.17	12.84, 2.17	12.84, 2.15	12.85, 2.15	12.85, 2.15	12.84, 2.16
vgg16	12.84, 2.17	12.84, 2.16	12.84, 2.16	12.84, 2.15	12.84, 2.15	12.72, 2.40	12.82, 2.22	12.84, 2.16	12.84, 2.15	12.84, 2.17	12.85, 2.15	12.85, 2.15	12.84, 2.15
vgg19	12.83, 2.17	12.84, 2.16	12.84, 2.15	12.84, 2.15	12.84, 2.15	12.84, 2.21	12.78, 2.40	12.85, 2.16	12.84, 2.15	12.85, 2.17	12.85, 2.15	12.85, 2.15	12.84, 2.15
googlenet	12.84, 2.17	12.85, 2.16	12.84, 2.16	12.84, 2.16	12.84, 2.15	12.83, 2.16	12.85, 2.16	12.83, 2.38	12.84, 2.15	12.84, 2.17	12.85, 2.15	12.85, 2.15	12.84, 2.15
densenet	12.83, 2.17	12.85, 2.17	12.84, 2.16	12.85, 2.16	12.84, 2.15	12.84, 2.17	12.84, 2.16	12.84, 2.17	12.84, 2.31	12.84, 2.18	12.84, 2.16	12.84, 2.15	12.84, 2.15
mobilenet_v2	12.84, 2.17	12.85, 2.16	12.84, 2.16	12.85, 2.16	12.84, 2.15	12.84, 2.17	12.84, 2.17	12.85, 2.16	12.84, 2.15	12.76, 2.36	12.84, 2.16	12.85, 2.16	12.84, 2.15
mobilenet_v3_small	12.84, 2.15	12.84, 2.15	12.84, 2.15	12.84, 2.15	12.84, 2.15	12.83, 2.15	12.83, 2.15	12.84, 2.15	12.84, 2.15	12.83, 2.17	12.67, 2.36	12.84, 2.16	12.84, 2.14
mobilenet_v3_large	12.83, 2.16	12.85, 2.16	12.84, 2.15	12.84, 2.15	12.84, 2.14	12.84, 2.16	12.84, 2.16	12.84, 2.17	12.84, 2.15	12.84, 2.18	12.84, 2.16	12.75, 2.35	12.84, 2.15
wide_resnet_50_2	12.84, 2.18	12.85, 2.16	12.84, 2.17	12.85, 2.16	12.84, 2.17	12.84, 2.16	12.84, 2.18	12.84, 2.17	12.86, 2.17	12.85, 2.17	12.84, 2.16	12.85, 2.15	12.81, 2.32

Table A.48: std and mean for FGSM and epsilon 0.01 with target median from true class

source	resnet18	resnet34	resnet50	resnet101	resnet152	vgg16	vgg19	googlenet	densenet	mobilenet_v2	mobilenet_v3_small	mobilenet_v3_large	wide_resnet_50_2
resnet18	12.72, 2.45	12.79, 2.24	12.80, 2.21	12.80, 2.21	12.82, 2.20	12.78, 2.26	12.78, 2.25	12.83, 2.24	12.82, 2.18	12.80, 2.25	12.83, 2.20	12.81, 2.19	12.80, 2.20
resnet34	12.82, 2.29	12.71, 2.43	12.78, 2.22	12.80, 2.20	12.81, 2.19	12.80, 2.24	12.78, 2.24	12.83, 2.23	12.82, 2.18	12.79, 2.24	12.82, 2.20	12.82, 2.20	12.79, 2.19
resnet50	12.79, 2.24	12.81, 2.22	12.71, 2.33	12.80, 2.20	12.83, 2.17	12.79, 2.21	12.81, 2.21	12.82, 2.20	12.82, 2.19	12.80, 2.22	12.83, 2.18	12.83, 2.18	12.81, 2.19
resnet101	12.79, 2.24	12.80, 2.20	12.76, 2.19	12.71, 2.32	12.82, 2.20	12.80, 2.21	12.79, 2.20	12.81, 2.21	12.83, 2.19	12.79, 2.22	12.83, 2.20	12.81, 2.19	12.81, 2.18
resnet152	12.80, 2.23	12.79, 2.21	12.78, 2.20	12.81, 2.21	12.78, 2.33	12.80, 2.22	12.79, 2.20	12.83, 2.21	12.83, 2.20	12.80, 2.21	12.82, 2.18	12.83, 2.18	12.79, 2.20
vgg16	12.81, 2.22	12.81, 2.19	12.80, 2.18	12.82, 2.17	12.82, 2.18	12.65, 2.45	12.73, 2.30	12.81, 2.20	12.82, 2.17	12.80, 2.21	12.82, 2.19	12.82, 2.18	12.81, 2.16
vgg19	12.81, 2.23	12.82, 2.19	12.80, 2.18	12.82, 2.17	12.82, 2.17	12.77, 2.28	12.67, 2.42	12.84, 2.21	12.83, 2.17	12.81, 2.21	12.82, 2.19	12.83, 2.17	12.81, 2.16
googlenet	12.82, 2.26	12.83, 2.21	12.81, 2.19	12.82, 2.17	12.83, 2.17	12.80, 2.22	12.80, 2.20	12.76, 2.43	12.83, 2.17	12.81, 2.24	12.83, 2.21	12.83, 2.19	12.82, 2.17
densenet	12.80, 2.23	12.80, 2.21	12.80, 2.20	12.81, 2.18	12.82, 2.18	12.79, 2.22	12.79, 2.22	12.82, 2.22	12.80, 2.33	12.79, 2.23	12.83, 2.20	12.81, 2.19	12.81, 2.17
mobilenet_v2	12.80, 2.22	12.82, 2.19	12.82, 2.18	12.82, 2.17	12.82, 2.16	12.81, 2.20	12.81, 2.20	12.83, 2.20	12.83, 2.16	12.61, 2.42	12.81, 2.22	12.82, 2.19	12.82, 2.16
mobilenet_v3_small	12.80, 2.18	12.81, 2.16	12.82, 2.17	12.82, 2.18	12.82, 2.16	12.80, 2.18	12.79, 2.19	12.80, 2.19	12.83, 2.16	12.76, 2.21	12.54, 2.43	12.81, 2.19	12.84, 2.16
mobilenet_v3_large	12.81, 2.20	12.81, 2.18	12.81, 2.18	12.82, 2.17	12.82, 2.17	12.81, 2.19	12.81, 2.20	12.82, 2.20	12.83, 2.17	12.79, 2.22	12.79, 2.19	12.67, 2.40	12.82, 2.17
wide_resnet_50_2	12.80, 2.24	12.82, 2.21	12.78, 2.23	12.81, 2.19	12.82, 2.20	12.80, 2.20	12.79, 2.20	12.83, 2.20	12.84, 2.19	12.81, 2.21	12.83, 2.20	12.82, 2.19	12.76, 2.32

Table A.49: std and mean for FGV and epsilon 0.3 with target median from true class

source	resnet18	resnet34	resnet50	resnet101	resnet152	vgg16	vgg19	googlenet	densenet	mobilenet_v2	mobilenet_v3_small	mobilenet_v3_large	wide_resnet_50_2
resnet18	12.81, 2.35	12.84, 2.17	12.83, 2.16	12.84, 2.16	12.84, 2.15	12.84, 2.18	12.84, 2.18	12.84, 2.17	12.84, 2.16	12.83, 2.18	12.84, 2.16	12.84, 2.15	12.84, 2.15
resnet34	12.83, 2.20	12.80, 2.33	12.83, 2.17	12.85, 2.16	12.84, 2.16	12.84, 2.17	12.84, 2.16	12.84, 2.18	12.84, 2.17	12.84, 2.18	12.84, 2.15	12.85, 2.15	12.84, 2.15
resnet50	12.83, 2.18	12.85, 2.16	12.80, 2.29	12.84, 2.16	12.85, 2.16	12.84, 2.16	12.85, 2.16	12.84, 2.16	12.84, 2.15	12.84, 2.15	12.84, 2.16	12.84, 2.15	12.84, 2.15
resnet101	12.85, 2.17	12.85, 2.17	12.84, 2.16	12.80, 2.29	12.84, 2.16	12.84, 2.16	12.84, 2.16	12.84, 2.16	12.85, 2.16	12.84, 2.16	12.84, 2.15	12.85, 2.15	12.84, 2.15
resnet152	12.83, 2.16	12.84, 2.15	12.83, 2.17	12.83, 2.16	12.83, 2.28	12.84, 2.16	12.85, 2.16	12.84, 2.16	12.84, 2.16	12.83, 2.16	12.84, 2.15	12.85, 2.15	12.84, 2.15
vgg16	12.83, 2.17	12.84, 2.15	12.84, 2.15	12.84, 2.15	12.84, 2.14	12.78, 2.39	12.84, 2.20	12.84, 2.16	12.84, 2.15	12.83, 2.17	12.84, 2.15	12.85, 2.15	12.84, 2.14
vgg19	12.84, 2.16	12.84, 2.16	12.84, 2.15	12.84, 2.15	12.84, 2.14	12.83, 2.20	12.80, 2.35	12.85, 2.16	12.84, 2.15	12.83, 2.18	12.84, 2.15	12.85, 2.15	12.84, 2.14
googlenet	12.84, 2.17	12.85, 2.16	12.84, 2.15	12.84, 2.15	12.84, 2.15	12.84, 2.16	12.85, 2.17	12.84, 2.38	12.85, 2.15	12.83, 2.17	12.85, 2.15	12.85, 2.14	12.84, 2.15
densenet	12.83, 2.16	12.84, 2.15	12.84, 2.15	12.84, 2.16	12.83, 2.15	12.84, 2.17	12.84, 2.17	12.83, 2.16	12.84, 2.27	12.84, 2.17	12.84, 2.15	12.84, 2.15	12.84, 2.16
mobilenet_v2	12.84, 2.16	12.85, 2.15	12.83, 2.15	12.84, 2.15	12.84, 2.15	12.84, 2.16	12.84, 2.16	12.84, 2.16	12.84, 2.15	12.78, 2.35	12.84, 2.15	12.85, 2.15	12.84, 2.15
mobilenet_v3_small	12.83, 2.15	12.84, 2.15	12.84, 2.15	12.84, 2.15	12.84, 2.15	12.83, 2.15	12.84, 2.15	12.84, 2.16	12.84, 2.14	12.84, 2.15	12.71, 2.32	12.85, 2.15	12.84, 2.14
mobilenet_v3_large	12.84, 2.16	12.85, 2.15	12.84, 2.15	12.84, 2.15	12.84, 2.14	12.83, 2.15	12.84, 2.15	12.84, 2.16	12.84, 2.15	12.83, 2.17	12.84, 2.16	12.79, 2.33	12.84, 2.15
wide_resnet_50_2	12.83, 2.18	12.85, 2.15	12.84, 2.17	12.84, 2.16	12.84, 2.17	12.84, 2.16	12.84, 2.16	12.85, 2.16	12.85, 2.16	12.84, 2.17	12.84, 2.15	12.85, 2.15	12.82, 2.27

Table A.50: std and mean for FGV and epsilon 0.1 with target median from true class

source	resnet18	resnet34	resnet50	resnet101	resnet152	vgg16	vgg19	googlenet	densenet	mobilenet_v2	mobilenet_v3_small	mobilenet_v3_large	wide_resnet_50_2
resnet18	12.84, 2.17	12.84, 2.14	12.84, 2.15	12.84, 2.14	12.84, 2.14	12.84, 2.14	12.84, 2.14	12.84, 2.14	12.84, 2.14	12.84, 2.14	12.84, 2.14	12.84, 2.14	12.84, 2.14
resnet34	12.84, 2.14	12.85, 2.16	12.84, 2.15	12.84, 2.14	12.84, 2.14	12.84, 2.14	12.84, 2.14	12.84, 2.14	12.84, 2.14	12.84, 2.14	12.84, 2.14	12.84, 2.14	12.84, 2.14
resnet50	12.84, 2.14	12.84, 2.14	12.84, 2.15	12.84, 2.14	12.84, 2.14	12.84, 2.14	12.84, 2.14	12.84, 2.14	12.84, 2.14	12.84, 2.14	12.84, 2.14	12.84, 2.14	12.84, 2.14
resnet101	12.84, 2.14	12.84, 2.14	12.84, 2.14	12.84, 2.16	12.84, 2.14	12.84, 2.14	12.84, 2.14	12.84, 2.15	12.84, 2.14	12.84, 2.14	12.84, 2.14	12.84, 2.14	12.84, 2.14
resnet152	12.84, 2.14	12.84, 2.14	12.84, 2.14	12.84, 2.14	12.84, 2.15	12.84, 2.14	12.84, 2.14	12.84, 2.15	12.84, 2.14	12.84, 2.14	12.84, 2.14	12.84, 2.14	12.84, 2.14
vgg16	12.84, 2.14	12.84, 2.14	12.84, 2.14	12.84, 2.14	12.84, 2.14	12.85, 2.19	12.84, 2.15	12.84, 2.14	12.84, 2.14	12.84, 2.14	12.84, 2.14	12.84, 2.14	12.84, 2.14
vgg19	12.84, 2.14	12.84, 2.14	12.84, 2.14	12.84, 2.14	12.84, 2.14	12.84, 2.14	12.85, 2.17	12.84, 2.14	12.84, 2.14	12.84, 2.14	12.84, 2.14	12.84, 2.14	12.84, 2.14
googlenet	12.84, 2.14	12.84, 2.14	12.84, 2.14	12.84, 2.14	12.84, 2.14	12.84, 2.14	12.84, 2.14	12.84, 2.18	12.84, 2.14	12.84, 2.14	12.84, 2.14	12.84, 2.14	12.84, 2.14
densenet	12.84, 2.14	12.84, 2.14	12.84, 2.14	12.84, 2.14	12.84, 2.14	12.84, 2.14	12.84, 2.14	12.84, 2.14	12.85, 2.16	12.84, 2.14	12.84, 2.14	12.84, 2.14	12.84, 2.14
mobilenet_v2	12.84, 2.14	12.84, 2.14	12.84, 2.14	12.84, 2.14	12.84, 2.14	12.84, 2.14	12.84, 2.14	12.84, 2.14	12.84, 2.14	12.84, 2.19	12.84, 2.14	12.84, 2.14	12.84, 2.14
mobilenet_v3_small	12.84, 2.14	12.84, 2.14	12.84, 2.14	12.84, 2.14	12.84, 2.14	12.84, 2.14	12.84, 2.14	12.84, 2.14	12.84, 2.14	12.84, 2.14	12.85, 2.16	12.84, 2.14	12.84, 2.14
mobilenet_v3_large	12.84, 2.14	12.84, 2.14	12.84, 2.14	12.84, 2.14	12.84, 2.14	12.84, 2.14	12.84, 2.14	12.84, 2.14	12.84, 2.14	12.84, 2.14	12.85, 2.14	12.85, 2.15	12.84, 2.14
wide_resnet_50_2	12.84, 2.14	12.84, 2.14	12.84, 2.14	12.84, 2.14	12.84, 2.14	12.84, 2.14	12.84, 2.14	12.84, 2.14	12.84, 2.14	12.84, 2.14	12.84, 2.14	12.84, 2.14	12.85, 2.15

Table A.51: std and mean for FGV and epsilon 0.01 with target median from true class

source	resnet18	resnet34	resnet50	resnet101	resnet152	vgg16	vgg19	googlenet	densenet	mobilenet_v2	mobilenet_v3_small	mobilenet_v3_large	wide_resnet_50_2
resnet18	5.29, 5.23	3.74, 3.32	3.58, 3.12	3.37, 2.87	3.20, 2.59	4.00, 3.49	3.91, 3.37	3.60, 3.09	3.17, 2.59	3.85, 3.34	3.54, 2.93	3.29, 2.62	3.26, 2.67
resnet34	4.10, 3.66	5.23, 4.94	3.60, 3.15	3.42, 2.94	3.27, 2.69	3.86, 3.30	3.82, 3.26	3.60, 3.03	3.23, 2.65	3.81, 3.25	3.50, 2.88	3.29, 2.61	3.30, 2.72
resnet50	3.74, 3.25	3.50, 2.98	4.92, 4.61	3.43, 2.94	3.21, 2.62	3.52, 2.90	3.51, 2.87	3.38, 2.75	3.17, 2.55	3.59, 3.02	3.33, 2.65	3.14, 2.41	3.25, 2.68
resnet101	3.71, 3.18	3.49, 2.95	3.60, 3.10	4.81, 4.54	3.39, 2.91	3.53, 2.89	3.48, 2.83	3.45, 2.86	3.14, 2.50	3.59, 3.00	3.39, 2.72	3.16, 2.42	3.25, 2.66
resnet152	3.60, 3.06	3.38, 2.85	3.43, 2.90	3.46, 3.00	4.48, 4.37	3.43, 2.80	3.40, 2.75	3.34, 2.70	3.13, 2.49	3.50, 2.91	3.31, 2.60	3.10, 2.33	3.19, 2.59
vgg16	3.54, 2.98	3.22, 2.56	3.17, 2.50	2.99, 2.19	2.93, 2.05	5.84, 4.96	4.49, 3.99	3.30, 2.67	2.96, 2.17	3.63, 3.06	3.33, 2.62	3.14, 2.38	2.93, 2.09
vgg19	3.52, 2.98	3.22, 2.58	3.15, 2.46	2.97, 2.12	2.91, 2.01	4.50, 3.95	5.85, 4.87	3.28, 2.63	2.95, 2.15	3.60, 3.02	3.30, 2.60	3.12, 2.35	2.91, 2.02
googlenet	3.66, 3.15	3.30, 2.70	3.24, 2.58	3.09, 2.35	2.99, 2.20	3.57, 2.96	3.54, 2.94	4.84, 4.81	2.97, 2.17	3.59, 3.01	3.52, 2.91	3.23, 2.53	3.01, 2.23
densenet	3.62, 3.08	3.44, 2.91	3.41, 2.85	3.28, 2.74	3.14, 2.49	3.64, 3.06	3.61, 3.05	3.40, 2.81	4.18, 4.48	3.61, 3.07	3.46, 2.84	3.22, 2.56	3.17, 2.55
mobilenet_v2	3.53, 3.04	3.17, 2.51	3.10, 2.41	2.94, 2.09	2.88, 1.98	3.52, 2.95	3.42, 2.82	3.26, 2.65	2.89, 2.05	5.42, 5.15	3.53, 2.95	3.25, 2.59	2.84, 1.91
mobilenet_v3_small	3.22, 2.52	2.98, 2.17	2.88, 1.91	2.86, 1.89	2.81, 1.78	3.26, 2.56	3.18, 2.44	3.18, 2.47	2.82, 1.82	3.60, 3.08	5.75, 5.12	3.33, 2.70	2.77, 1.68
mobilenet_v3_large	3.35, 2.74	3.09, 2.37	2.97, 2.11	2.90, 1.97	2.85, 1.89	3.33, 2.64	3.31, 2.64	3.25, 2.57	2.88, 2.00	3.61, 3.08	3.68, 3.12	5.07, 4.97	2.83, 1.86
wide_resnet_50_2	3.68, 3.13	3.47, 2.92	3.56, 3.04	3.37, 2.76	3.28, 2.66	3.55, 2.93	3.50, 2.85	3.39, 2.73	3.24, 2.62	3.58, 2.99	3.36, 2.70	3.17, 2.45	4.97, 4.50

Table A.52: std and mean for FGV and epsilon 0.3 with target closest from true class

source	resnet18	resnet34	resnet50	resnet101	resnet152	vgg16	vgg19	googlenet	densenet	mobilenet_v2	mobilenet_v3_small	mobilenet_v3_large	wide_resnet_50_2
resnet18	4.09, 4.82	2.84, 1.95	2.74, 1.69	2.71, 1.58	2.67, 1.44	2.85, 1.88	2.85, 1.87	2.84, 1.88	2.67, 1.50	2.87, 1.96	2.80, 1.72	2.67, 1.39	2.64, 1.39
resnet34	3.08, 2.39	4.34, 4.60	2.79, 1.79	2.74, 1.68	2.71, 1.59	2.85, 1.88	2.84, 1.86	2.85, 1.88	2.73, 1.65	2.89, 1.98	2.80, 1.72	2.68, 1.42	2.67, 1.41
resnet50	2.91, 2.04	2.79, 1.78	4.03, 4.13	2.77, 1.78	2.71, 1.61	2.78, 1.69	2.79, 1.71	2.79, 1.73	2.70, 1.57	2.81, 1.80	2.77, 1.65	2.65, 1.33	2.67, 1.48
resnet101	2.91, 2.02	2.81, 1.85	2.85, 1.96	4.06, 4.14	2.79, 1.83	2.79, 1.70	2.78, 1.64	2.81, 1.76	2.72, 1.62	2.85, 1.84	2.78, 1.66	2.66, 1.36	2.67, 1.45
resnet152	2.86, 1.92	2.78, 1.74	2.81, 1.83	2.81, 1.90	3.86, 3.97	2.76, 1.63	2.77, 1.64	2.79, 1.71	2.71, 1.61	2.81, 1.74	2.76, 1.62	2.65, 1.33	2.68, 1.47
vgg16	2.84, 1.87	2.70, 1.51	2.66, 1.41	2.63, 1.30	2.63, 1.27	4.88, 4.78	3.27, 2.77	2.73, 1.57	2.63, 1.30	2.86, 1.90	2.76, 1.62	2.66, 1.36	2.61, 1.19
vgg19	2.83, 1.83	2.70, 1.55	2.66, 1.39	2.63, 1.28	2.63, 1.27	3.31, 2.80	4.97, 4.70	2.74, 1.58	2.64, 1.31	2.86, 1.91	2.76, 1.62	2.66, 1.34	2.60, 1.18
googlenet	2.80, 1.82	2.71, 1.57	2.66, 1.39	2.65, 1.34	2.63, 1.28	2.76, 1.67	2.76, 1.66	4.03, 4.55	2.64, 1.31	2.77, 1.70	2.78, 1.64	2.64, 1.28	2.62, 1.22
densenet	2.86, 1.90	2.78, 1.76	2.75, 1.71	2.69, 1.52	2.68, 1.53	2.79, 1.70	2.78, 1.70	2.79, 1.72	3.61, 4.04	2.82, 1.81	2.80, 1.68	2.64, 1.30	2.65, 1.42
mobilenet_v2	2.80, 1.77	2.69, 1.52	2.66, 1.38	2.63, 1.34	2.62, 1.27	2.77, 1.69	2.77, 1.66	2.72, 1.57	2.62, 1.27	4.08, 4.80	2.84, 1.80	2.71, 1.50	2.60, 1.18
mobilenet_v3_small	2.70, 1.47	2.63, 1.26	2.61, 1.18	2.61, 1.19	2.60, 1.14	2.69, 1.45	2.67, 1.37	2.69, 1.43	2.60, 1.14	2.75, 1.62	4.35, 4.64	2.68, 1.44	2.58, 1.09
mobilenet_v3_large	2.78, 1.69	2.69, 1.47	2.64, 1.30	2.62, 1.27	2.62, 1.23	2.74, 1.58	2.75, 1.62	2.73, 1.57	2.63, 1.29	2.88, 1.95	2.95, 2.06	4.12, 4.57	2.59, 1.14
wide_resnet_50_2	2.91, 2.00	2.81, 1.78	2.84, 1.88	2.75, 1.65	2.76, 1.68	2.79, 1.69	2.79, 1.71	2.80, 1.72	2.74, 1.63	2.83, 1.79	2.76, 1.60	2.65, 1.31	4.22, 4.16

Table A.53: std and mean for FGV and epsilon 0.1 with target closest from true class

source	resnet18	resnet34	resnet50	resnet101	resnet152	vgg16	vgg19	googlenet	densenet	mobilenet_v2	mobilenet_v3_small	mobilenet_v3_large	wide_resnet_50_2
resnet18	2.75, 2.24	2.57, 1.05	2.57, 1.03	2.57, 1.03	2.56, 1.00	2.57, 1.03	2.57, 1.04	2.58, 1.08	2.57, 1.01	2.58, 1.04	2.57, 1.04	2.57, 1.01	2.57, 1.01
resnet34	2.59, 1.13	2.76, 2.07	2.57, 1.04	2.57, 1.03	2.57, 1.01	2.58, 1.04	2.57, 1.03	2.58, 1.06	2.57, 1.03	2.58, 1.05	2.58, 1.06	2.57, 1.00	2.57, 1.01
resnet50	2.59, 1.10	2.57, 1.04	2.67, 1.72	2.57, 1.03	2.57, 1.03	2.57, 1.03	2.58, 1.04	2.58, 1.07	2.57, 1.03	2.58, 1.05	2.57, 1.03	2.57, 1.00	2.57, 1.01
resnet101	2.59, 1.11	2.58, 1.07	2.57, 1.03	2.71, 1.80	2.57, 1.04	2.57, 1.03	2.57, 1.04	2.58, 1.07	2.57, 1.03	2.58, 1.06	2.57, 1.01	2.57, 1.01	2.57, 1.02
resnet152	2.58, 1.08	2.57, 1.05	2.57, 1.04	2.57, 1.04	2.67, 1.65	2.57, 1.03	2.57, 1.03	2.58, 1.08	2.57, 1.04	2.58, 1.05	2.57, 1.04	2.57, 1.01	2.57, 1.01
vgg16	2.57, 1.05	2.57, 1.02	2.57, 1.02	2.57, 1.02	2.57, 1.00	2.88, 2.31	2.59, 1.11	2.57, 1.03	2.57, 1.01	2.58, 1.06	2.57, 1.03	2.57, 1.00	2.57, 1.01
vgg19	2.58, 1.04	2.57, 1.03	2.57, 1.02	2.57, 1.03	2.57, 1.00	2.58, 1.08	2.87, 2.21	2.58, 1.07	2.57, 1.01	2.58, 1.06	2.57, 1.03	2.57, 1.01	2.57, 1.01
googlenet	2.57, 1.04	2.57, 1.03	2.57, 1.02	2.57, 1.02	2.57, 1.01	2.57, 1.03	2.57, 1.02	2.72, 2.08	2.57, 1.01	2.58, 1.04	2.57, 1.04	2.57, 1.00	2.57, 1.00
densenet	2.58, 1.07	2.57, 1.04	2.57, 1.03	2.57, 1.02	2.57, 1.01	2.57, 1.02	2.58, 1.04	2.58, 1.05	2.65, 1.72	2.58, 1.04	2.57, 1.04	2.57, 1.01	2.57, 1.01
mobilenet_v2	2.58, 1.06	2.57, 1.03	2.57, 1.01	2.57, 1.01	2.56, 1.00	2.57, 1.02	2.57, 1.02	2.58, 1.05	2.57, 1.01	2.81, 2.44	2.58, 1.06	2.57, 1.01	2.57, 1.01
mobilenet_v3_small	2.57, 1.01	2.57, 1.01	2.57, 1.01	2.57, 1.02	2.56, 1.00	2.57, 1.01	2.57, 1.01	2.57, 1.04	2.57, 1.01	2.57, 1.03	2.83, 2.14	2.57, 1.01	2.57, 1.00
mobilenet_v3_large	2.58, 1.04	2.57, 1.03	2.57, 1.01	2.57, 1.02	2.56, 1.00	2.57, 1.03	2.57, 1.03	2.58, 1.06	2.57, 1.03	2.58, 1.06	2.60, 1.14	2.71, 1.84	2.57, 1.01
wide_resnet_50_2	2.58, 1.06	2.58, 1.05	2.58, 1.04	2.57, 1.04	2.57, 1.02	2.57, 1.03	2.58, 1.03	2.58, 1.07	2.58, 1.06	2.58, 1.06	2.57, 1.03	2.57, 1.01	2.62, 1.50

Table A.54: std and mean for FGV and epsilon 0.01 with target closest from true class

source	resnet18	resnet34	resnet50	resnet101	resnet152	vgg16	vgg19	googlenet	densenet	mobilenet_v2	mobilenet_v3_small	mobilenet_v3_large	wide_resnet_50_2
resnet18	16.08, 4.78	11.57, 4.62	14.73, 5.19	12.13, 4.59	10.74, 4.14	12.44, 3.69	11.51, 3.70	10.76, 3.92	10.24, 4.61	13.35, 4.51	10.29, 3.36	10.37, 3.60	10.46, 4.08
resnet34	15.66, 5.09	11.75, 4.60	15.16, 5.06	12.15, 4.58	10.78, 4.18	12.27, 3.74	11.34, 3.66	10.83, 3.89	10.45, 4.61	14.04, 4.57	10.28, 3.44	10.27, 3.56	10.70, 4.06
resnet50	15.44, 5.21	11.58, 4.65	16.05, 4.67	12.11, 4.50	10.83, 4.24	12.13, 3.73	11.21, 3.77	10.71, 3.93	10.01, 4.67	14.66, 4.70	10.22, 3.33	10.27, 3.55	10.58, 4.04
resnet101	15.94, 4.92	11.51, 4.61	15.30, 5.04	12.55, 4.65	11.24, 4.58	12.05, 3.73	11.20, 3.71	10.78, 3.92	10.10, 4.66	14.75, 4.69	10.23, 3.36	10.30, 3.59	10.59, 4.10
resnet152	15.31, 5.26	11.91, 4.83	14.96, 5.15	12.02, 4.49	11.65, 4.74	12.07, 3.75	11.21, 3.74	10.81, 3.92	10.16, 4.60	14.39, 4.82	10.28, 3.36	10.26, 3.58	10.61, 4.04
vgg16	13.22, 5.64	11.59, 4.80	12.92, 5.22	11.50, 4.55	10.71, 4.20	11.70, 3.77	11.30, 3.69	10.56, 3.94	9.54, 4.76	12.32, 4.41	10.25, 3.46	10.16, 3.67	9.98, 4.22
vgg19	13.17, 5.64	11.75, 4.87	12.90, 5.20	11.35, 4.48	10.67, 4.24	11.71, 3.78	11.47, 3.69	10.59, 3.90	9.51, 4.74	12.54, 4.47	10.27, 3.41	10.13, 3.72	9.93, 4.20
googlenet	13.10, 5.58	11.80, 4.81	12.44, 4.85	11.13, 4.12	10.28, 4.07	12.18, 3.69	11.10, 3.69	10.87, 3.89	9.83, 4.50	13.49, 4.86	10.28, 3.48	10.35, 3.57	10.13, 4.00
densenet	15.15, 5.25	12.42, 5.10	14.51, 5.24	12.98, 4.81	10.39, 4.25	11.91, 3.70	10.91, 3.63	10.95, 3.99	10.82, 4.77	15.40, 4.58	10.57, 3.52	10.45, 3.67	10.87, 4.16
mobilenet_v2	12.46, 5.48	11.89, 5.01	13.63, 5.25	11.51, 4.55	10.87, 4.13	12.31, 3.73	11.68, 3.64	10.54, 3.91	9.73, 4.73	12.41, 4.21	10.33, 3.52	10.14, 3.61	10.22, 4.11
mobilenet_v3_small	10.50, 3.59	10.89, 4.27	10.87, 3.73	11.59, 4.85	10.55, 4.18	11.36, 3.68	11.84, 3.79	10.57, 3.65	10.89, 4.51	12.29, 4.55	11.59, 3.75	10.22, 3.59	10.54, 3.97
mobilenet_v3_large	12.96, 5.48	11.22, 4.51	12.76, 5.07	12.45, 4.96	10.67, 4.17	11.63, 3.80	11.54, 3.70	10.93, 3.86	10.54, 4.69	13.05, 4.51	10.20, 3.37	10.29, 3.51	10.37, 4.17
wide_resnet_50_2	15.07, 5.36	12.32, 5.02	14.85, 5.17	11.95, 4.48	10.85, 4.42	12.19, 3.70	11.37, 3.70	10.69, 3.92	9.98, 4.61	14.68, 4.79	10.31, 3.44	10.34, 3.63	10.69, 4.01

Table A.55: std and mean for FGSM and epsilon 0.3 with target closest from true class

source	resnet18	resnet34	resnet50	resnet101	resnet152	vgg16	vgg19	googlenet	densenet	mobilenet_v2	mobilenet_v3_small	mobilenet_v3_large	wide_resnet_50_2
resnet18	4.11, 4.98	2.84, 1.93	2.72, 1.59	2.68, 1.49	2.65, 1.38	2.84, 1.89	2.84, 1.86	2.78, 1.76	2.67, 1.44	2.86, 1.93	2.79, 1.71	2.66, 1.37	2.64, 1.34
resnet34	3.09, 2.41	4.45, 4.76	2.80, 1.79	2.74, 1.68	2.70, 1.58	2.86, 1.91	2.85, 1.88	2.80, 1.75	2.73, 1.64	2.89, 1.99	2.82, 1.77	2.67, 1.40	2.67, 1.43
resnet50	2.97, 2.16	2.83, 1.90	4.33, 4.46	2.80, 1.86	2.73, 1.67	2.85, 1.84	2.86, 1.87	2.79, 1.74	2.74, 1.67	2.88, 1.94	2.80, 1.69	2.66, 1.37	2.71, 1.60
resnet101	2.94, 2.11	2.85, 1.92	2.89, 2.04	4.29, 4.41	2.81, 1.90	2.81, 1.75	2.82, 1.78	2.80, 1.75	2.72, 1.59	2.89, 1.96	2.80, 1.71	2.67, 1.39	2.70, 1.55
resnet152	2.93, 2.07	2.84, 1.91	2.86, 1.97	2.87, 2.05	4.10, 4.27	2.82, 1.78	2.81, 1.76	2.78, 1.72	2.75, 1.69	2.86, 1.88	2.80, 1.72	2.67, 1.42	2.72, 1.62
vgg16	2.88, 1.97	2.74, 1.64	2.70, 1.52	2.65, 1.37	2.63, 1.28	5.20, 4.96	3.48, 3.08	2.75, 1.61	2.65, 1.33	2.93, 2.06	2.79, 1.69	2.67, 1.38	2.62, 1.23
vgg19	2.87, 1.93	2.74, 1.61	2.69, 1.45	2.64, 1.30	2.64, 1.33	3.53, 3.12	5.37, 4.94	2.76, 1.66	2.65, 1.32	2.93, 2.05	2.79, 1.69	2.67, 1.39	2.63, 1.25
googlenet	2.78, 1.76	2.69, 1.49	2.65, 1.36	2.62, 1.27	2.61, 1.19	2.75, 1.62	2.74, 1.58	3.96, 4.61	2.62, 1.25	2.75, 1.63	2.77, 1.64	2.64, 1.30	2.60, 1.15
densenet	2.91, 2.02	2.82, 1.84	2.80, 1.81	2.73, 1.67	2.70, 1.57	2.83, 1.83	2.82, 1.79	2.79, 1.74	3.75, 4.38	2.85, 1.86	2.83, 1.76	2.67, 1.40	2.68, 1.50
mobilenet_v2	2.84, 1.89	2.71, 1.57	2.66, 1.41	2.65, 1.38	2.62, 1.27	2.83, 1.80	2.83, 1.81	2.73, 1.60	2.63, 1.29	4.09, 4.95	2.84, 1.79	2.70, 1.50	2.61, 1.19
mobilenet_v3_small	2.74, 1.57	2.65, 1.35	2.63, 1.27	2.62, 1.21	2.60, 1.15	2.73, 1.54	2.73, 1.56	2.69, 1.48	2.60, 1.15	2.84, 1.84	4.56, 5.02	2.71, 1.57	2.59, 1.11
mobilenet_v3_large	2.88, 1.96	2.75, 1.65	2.69, 1.45	2.67, 1.45	2.64, 1.32	2.86, 1.86	2.86, 1.89	2.80, 1.75	2.67, 1.40	3.01, 2.25	3.06, 2.25	4.38, 4.95	2.61, 1.20
wide_resnet_50_2	2.97, 2.13	2.87, 1.93	2.90, 2.01	2.80, 1.82	2.78, 1.75	2.87, 1.87	2.86, 1.88	2.82, 1.78	2.78, 1.72	2.87, 1.89	2.81, 1.73	2.68, 1.42	4.54, 4.45

Table A.56: std and mean for FGSM and epsilon 0.1 with target closest from true class

source	resnet18	resnet34	resnet50	resnet101	resnet152	vgg16	vgg19	googlenet	densenet	mobilenet_v2	mobilenet_v3_small	mobilenet_v3_large	wide_resnet_50_2
resnet18	9.06, 4.90	6.98, 4.81	6.33, 4.63	5.88, 4.57	5.10, 4.31	9.32, 4.65	8.31, 4.64	5.86, 4.49	5.00, 4.21	8.26, 4.73	6.99, 4.77	6.12, 4.59	5.53, 4.47
resnet34	8.14, 4.84	8.05, 4.93	6.39, 4.66	5.96, 4.63	5.22, 4.33	9.15, 4.62	8.28, 4.65	5.84, 4.44	5.09, 4.24	8.41, 4.69	7.06, 4.75	6.23, 4.59	5.65, 4.47
resnet50	7.72, 4.82	6.71, 4.74	7.07, 4.81	5.91, 4.61	5.12, 4.27	8.18, 4.66	7.49, 4.63	5.57, 4.34	4.92, 4.13	7.99, 4.76	6.72, 4.68	6.00, 4.51	5.47, 4.38
resnet101	7.85, 4.80	6.65, 4.75	6.42, 4.63	7.06, 4.93	5.37, 4.41	8.31, 4.64	7.69, 4.60	5.60, 4.32	4.86, 4.08	8.00, 4.71	6.82, 4.68	6.03, 4.52	5.43, 4.37
resnet152	7.56, 4.79	6.60, 4.72	6.18, 4.57	5.98, 4.63	5.99, 4.75	8.14, 4.64	7.46, 4.61	5.50, 4.27	4.85, 4.10	7.86, 4.74	6.73, 4.67	5.92, 4.51	5.34, 4.31
vgg16	7.10, 4.66	6.22, 4.59	5.73, 4.42	5.07, 4.19	4.67, 4.01	9.06, 4.44	8.24, 4.55	5.36, 4.26	4.61, 3.91	7.62, 4.72	6.81, 4.65	5.80, 4.44	4.94, 4.13
vgg19	7.04, 4.63	6.22, 4.59	5.73, 4.43	5.06, 4.20	4.64, 3.99	8.63, 4.51	8.43, 4.55	5.35, 4.25	4.61, 3.93	7.65, 4.68	6.78, 4.62	5.81, 4.43	4.90, 4.11
googlenet	7.05, 4.61	6.27, 4.60	5.76, 4.45	5.23, 4.30	4.63, 4.00	7.87, 4.65	7.37, 4.60	6.65, 4.82	4.65, 3.96	7.83, 4.65	6.80, 4.65	6.03, 4.48	4.99, 4.18
densenet	8.30, 4.93	7.00, 4.82	6.33, 4.60	6.08, 4.66	5.16, 4.29	9.40, 4.63	8.79, 4.69	5.73, 4.40	5.90, 4.77	8.99, 4.71	7.80, 4.81	6.44, 4.69	5.79, 4.52
mobilenet_v2	6.94, 4.69	6.14, 4.61	5.67, 4.45	5.10, 4.24	4.73, 4.08	8.22, 4.69	7.47, 4.68	5.31, 4.26	4.54, 3.92	9.55, 4.64	6.97, 4.71	5.72, 4.46	4.85, 4.09
mobilenet_v3_small	8.62, 4.85	7.34, 4.87	6.56, 4.72	6.64, 4.80	5.75, 4.68	9.20, 4.51	9.33, 4.63	6.87, 4.69	5.04, 4.20	8.91, 4.68	9.96, 4.41	7.43, 4.73	5.72, 4.57
mobilenet_v3_large	8.24, 4.79	6.67, 4.72	6.07, 4.54	5.95, 4.57	5.19, 4.31	9.34, 4.57	8.70, 4.54	6.26, 4.54	5.05, 4.20	8.52, 4.78	7.97, 4.74	7.84, 4.81	5.50, 4.41
wide_resnet_50_2	7.62, 4.75	6.74, 4.76	6.18, 4.54	5.71, 4.49	5.03, 4.20	8.00, 4.62	7.39, 4.63	5.50, 4.28	4.91, 4.08	7.83, 4.71	6.90, 4.73	5.98, 4.48	6.26, 4.67

Table A.57: std and mean for FGSM and epsilon 0.1 with target closest from true class

source	resnet18	resnet34	resnet50	resnet101	resnet152	vgg16	vgg19	googlenet	densenet	mobilenet_v2	mobilenet_v3_small	mobilenet_v3_large	wide_resnet_50_2
resnet18	0.02, 0.59	3.46, 3.79	3.68, 3.75	3.53, 3.40	3.23, 3.00	4.86, 4.41	4.60, 4.24	3.54, 3.27	3.10, 2.84	4.46, 4.27	3.81, 3.32	3.49, 3.11	3.46, 3.28
resnet34	3.68, 4.32	0.05, 0.84	3.44, 3.83	3.29, 3.40	3.11, 3.15	4.76, 4.39	4.50, 4.19	3.55, 3.38	3.01, 3.07	4.28, 4.20	3.82, 3.32	3.47, 3.16	3.31, 3.38
resnet50	4.05, 4.17	3.34, 3.60	0.08, 1.03	3.06, 3.64	2.81, 3.16	4.51, 4.15	4.33, 4.07	3.50, 3.30	2.83, 2.90	4.33, 4.15	3.73, 3.21	3.37, 2.96	2.93, 3.37
resnet101	4.17, 4.25	3.40, 3.66	3.02, 3.91	0.07, 0.93	2.53, 3.49	4.57, 4.15	4.39, 4.03	3.57, 3.43	2.87, 2.95	4.38, 4.18	3.85, 3.38	3.53, 3.20	3.04, 3.42
resnet152	4.07, 4.15	3.36, 3.60	2.89, 3.73	2.59, 3.67	0.09, 1.06	4.57, 4.20	4.37, 4.04	3.52, 3.36	2.72, 2.92	4.29, 4.08	3.79, 3.28	3.45, 3.11	2.91, 3.46
vgg16	3.92, 3.65	3.39, 2.99	3.40, 3.03	3.16, 2.61	3.02, 2.39	0.07, 0.94	3.03, 4.48	3.38, 2.88	2.98, 2.36	4.44, 4.04	3.67, 3.11	3.34, 2.81	3.13, 2.60
vgg19	3.91, 3.63	3.40, 3.05	3.41, 3.04	3.15, 2.59	3.01, 2.39	2.85, 4.43	0.10, 1.18	3.37, 2.91	3.00, 2.36	4.41, 4.07	3.66, 3.13	3.36, 2.81	3.12, 2.59
googlenet	3.94, 3.74	3.40, 3.17	3.38, 3.17	3.19, 2.79	3.00, 2.48	4.40, 3.94	4.16, 3.78	0.03, 0.61	2.93, 2.42	4.30, 3.91	3.77, 3.21	3.45, 3.00	3.18, 2.78
densenet	4.31, 4.30	3.64, 3.85	3.48, 3.95	3.37, 3.58	3.08, 3.32	4.98, 4.45	4.78, 4.35	3.66, 3.56	0.01, 0.30	4.69, 4.39	3.93, 3.45	3.65, 3.36	3.25, 3.59
mobilenet_v2	3.78, 3.50	3.25, 2.86	3.22, 2.76	3.01, 2.39	2.88, 2.13	4.34, 3.94	4.10, 3.70	3.26, 2.71	2.88, 2.14	0.06, 0.90	3.71, 3.23	3.30, 2.90	2.92, 2.17
mobilenet_v3_small	3.95, 3.52	3.37, 2.83	3.33, 2.77	3.20, 2.56	3.00, 2.26	4.36, 3.79	4.28, 3.72	3.35, 2.76	2.93, 2.12	4.05, 3.72	0.02, 0.51	3.49, 3.25	3.01, 2.25
mobilenet_v3_large	4.07, 3.76	3.45, 3.10	3.33, 2.89	3.20, 2.73	3.03, 2.45	4.64, 4.09	4.53, 3.99	3.47, 3.02	2.96, 2.35	4.32, 4.14	3.88, 3.59	0.01, 0.39	3.09, 2.50
wide_resnet_50_2	4.32, 4.22	3.66, 3.76	3.29, 3.90	3.26, 3.62	3.03, 3.43	4.66, 4.24	4.32, 4.00	3.65, 3.44	2.99, 3.25	4.48, 4.12	3.86, 3.36	3.55, 3.14	0.05, 0.78

Table A.58: std and mean for PGD and epsilon 0.1 with target closest from true class

source	resnet18	resnet34	resnet50	resnet101	resnet152	vgg16	vgg19	googlenet	densenet	mobilenet_v2	mobilenet_v3_small	mobilenet_v3_large	wide_resnet_50_2
resnet18	0.20, 1.16	2.60, 1.27	2.60, 1.19	2.60, 1.21	2.59, 1.14	2.66, 1.37	2.66, 1.35	2.63, 1.29	2.59, 1.16	2.65, 1.36	2.68, 1.40	2.60, 1.15	2.58, 1.08
resnet34	2.68, 1.53	0.29, 1.24	2.60, 1.24	2.59, 1.21	2.59, 1.17	2.67, 1.41	2.68, 1.43	2.63, 1.27	2.59, 1.15	2.67, 1.44	2.69, 1.45	2.60, 1.18	2.59, 1.13
resnet50	2.67, 1.45	2.62, 1.31	0.34, 1.29	2.59, 1.25	2.59, 1.22	2.66, 1.39	2.67, 1.39	2.62, 1.25	2.58, 1.16	2.67, 1.40	2.69, 1.43	2.59, 1.14	2.58, 1.15
resnet101	2.68, 1.50	2.62, 1.29	2.60, 1.33	0.42, 1.46	2.58, 1.26	2.67, 1.39	2.69, 1.44	2.64, 1.32	2.59, 1.19	2.68, 1.44	2.70, 1.45	2.60, 1.16	2.58, 1.16
resnet152	2.69, 1.53	2.62, 1.32	2.61, 1.36	2.58, 1.32	0.44, 1.56	2.68, 1.42	2.67, 1.41	2.64, 1.33	2.59, 1.19	2.67, 1.43	2.68, 1.42	2.60, 1.16	2.58, 1.16
vgg16	2.63, 1.28	2.60, 1.17	2.59, 1.16	2.59, 1.14	2.58, 1.09	0.52, 1.95	2.68, 1.78	2.63, 1.26	2.58, 1.09	2.67, 1.41	2.67, 1.37	2.60, 1.14	2.58, 1.08
vgg19	2.64, 1.30	2.60, 1.17	2.59, 1.15	2.59, 1.11	2.58, 1.09	2.67, 1.82	0.51, 1.80	2.63, 1.27	2.58, 1.08	2.67, 1.40	2.68, 1.39	2.60, 1.13	2.58, 1.09
googlenet	2.64, 1.32	2.61, 1.21	2.60, 1.19	2.60, 1.15	2.58, 1.10	2.66, 1.36	2.66, 1.37	0.31, 1.15	2.59, 1.12	2.67, 1.38	2.69, 1.42	2.60, 1.15	2.58, 1.09
densenet	2.67, 1.43	2.63, 1.36	2.61, 1.32	2.59, 1.22	2.59, 1.22	2.68, 1.45	2.70, 1.51	2.64, 1.33	0.32, 1.32	2.67, 1.43	2.71, 1.48	2.60, 1.16	2.57, 1.14
mobilenet_v2	2.62, 1.23	2.59, 1.15	2.59, 1.14	2.58, 1.08	2.58, 1.09	2.65, 1.33	2.65, 1.33	2.62, 1.22	2.58, 1.08	0.24, 1.19	2.69, 1.43	2.60, 1.18	2.57, 1.04
mobilenet_v3_small	2.62, 1.18	2.59, 1.12	2.59, 1.09	2.58, 1.09	2.58, 1.07	2.63, 1.23	2.63, 1.22	2.62, 1.24	2.58, 1.06	2.65, 1.32	0.42, 1.43	2.60, 1.16	2.57, 1.02
mobilenet_v3_large	2.64, 1.34	2.60, 1.18	2.59, 1.15	2.58, 1.10	2.58, 1.10	2.66, 1.35	2.67, 1.38	2.63, 1.27	2.59, 1.11	2.69, 1.48	2.72, 1.56	0.37, 1.46	2.58, 1.06
wide_resnet_50_2	2.70, 1.52	2.65, 1.41	2.61, 1.37	2.60, 1.29	2.60, 1.31	2.69, 1.49	2.70, 1.48	2.64, 1.31	2.59, 1.24	2.70, 1.48	2.71, 1.47	2.60, 1.18	0.43, 1.55

Table A.59: std and mean for PGD and epsilon 0.01 with target closest from true class

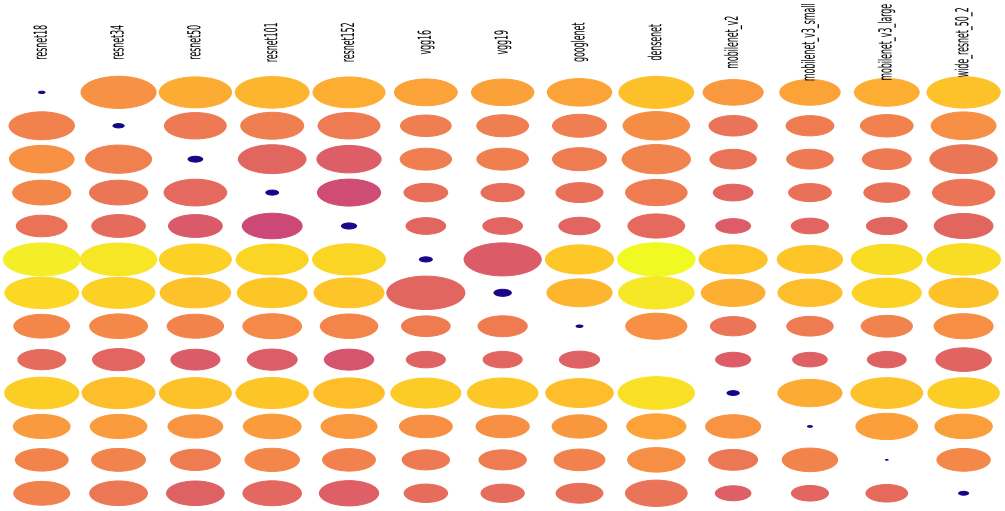


Figure A.9: std and avg for closest target and PGD with epsilon 0.1

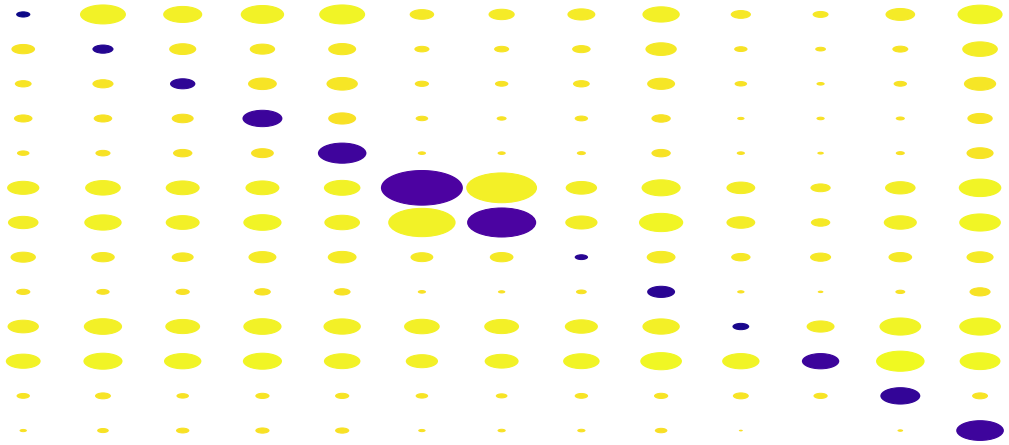


Figure A.10: std and avg for closest target and PGD with epsilon 0.01

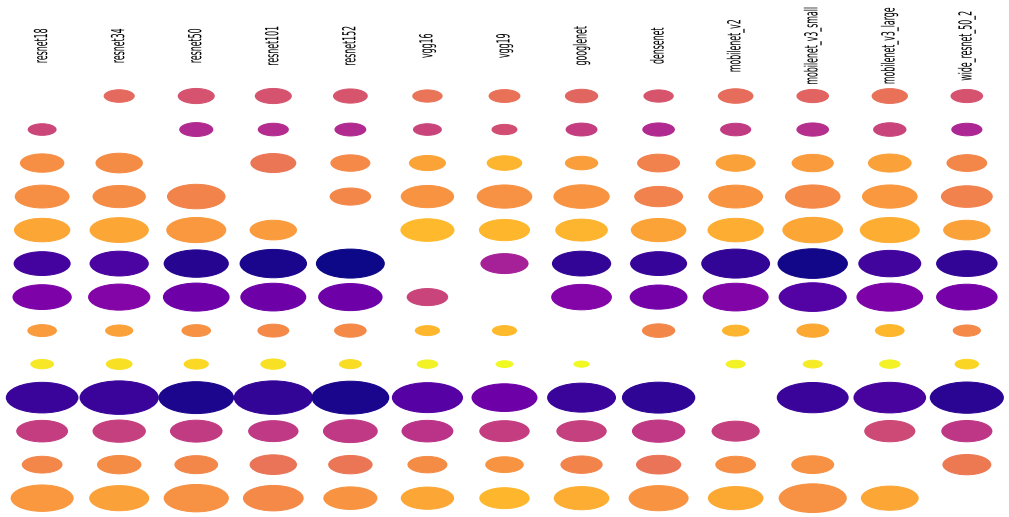


Figure A.11: std and avg for farthest target and PGD with epsilon 0.3

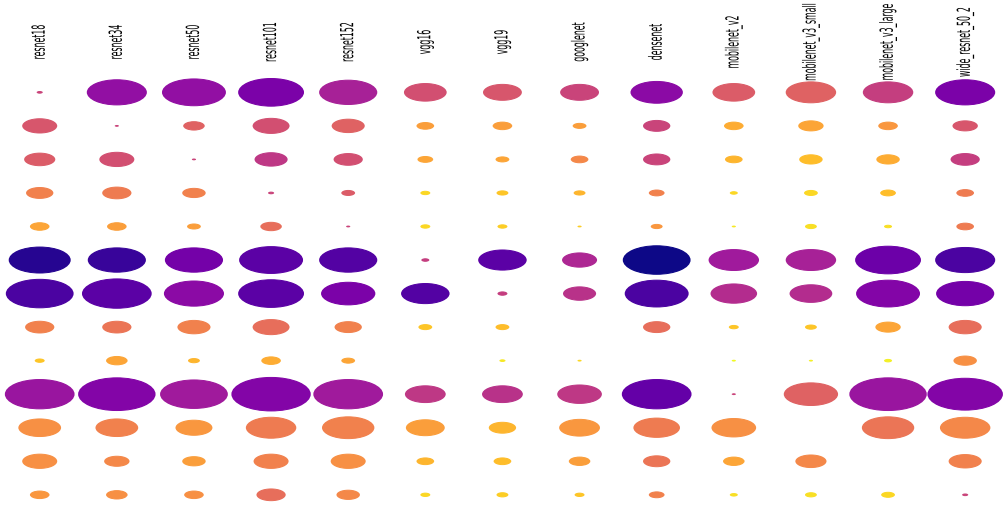


Figure A.12: std and avg for farthest target and PGD with epsilon 0.1

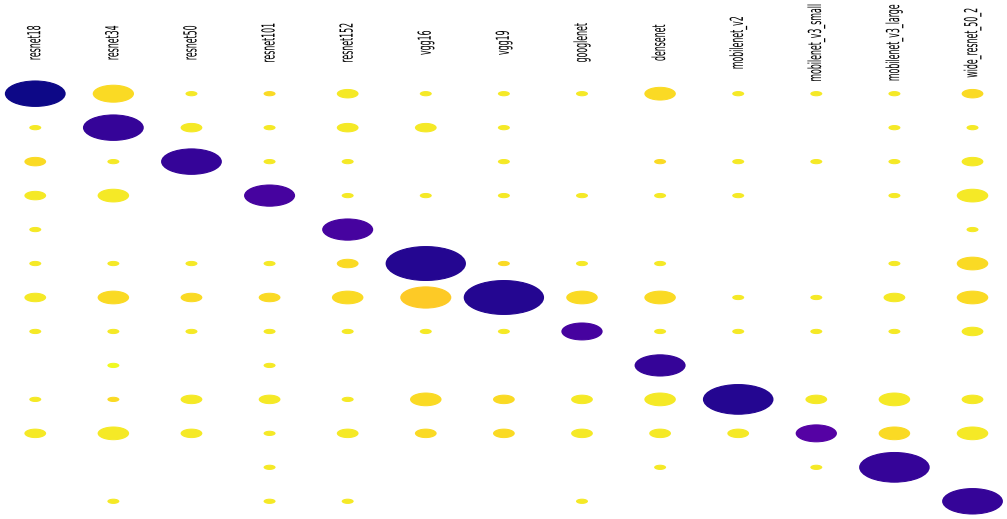


Figure A.13: std and avg for farthest target and PGD with epsilon 0.01

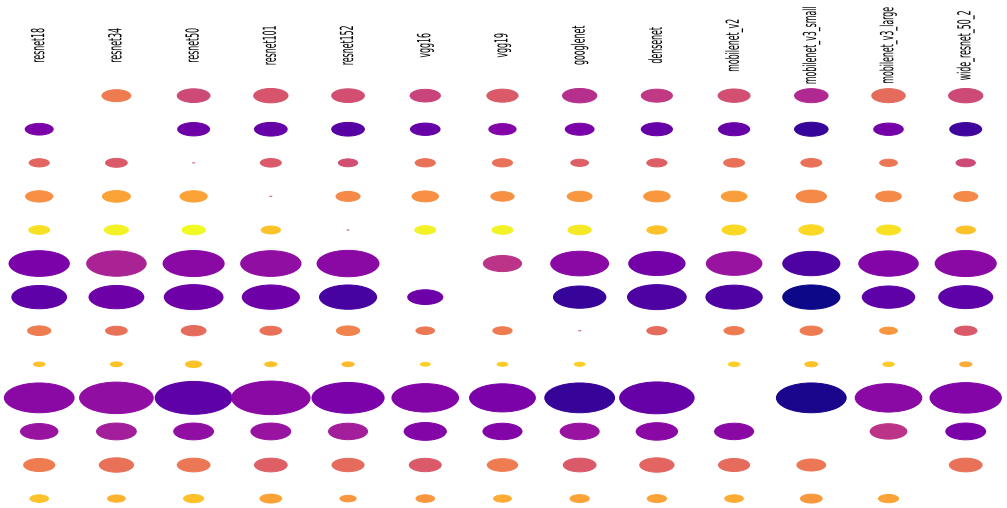


Figure A.14: std and avg for median target and PGD with epsilon 0.3

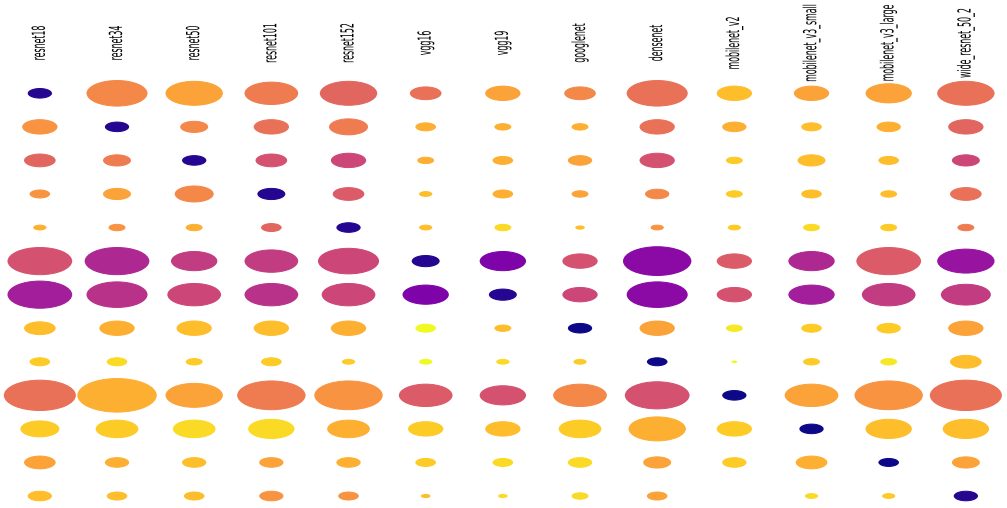


Figure A.15: std and avg for median target and PGD with epsilon 0.1

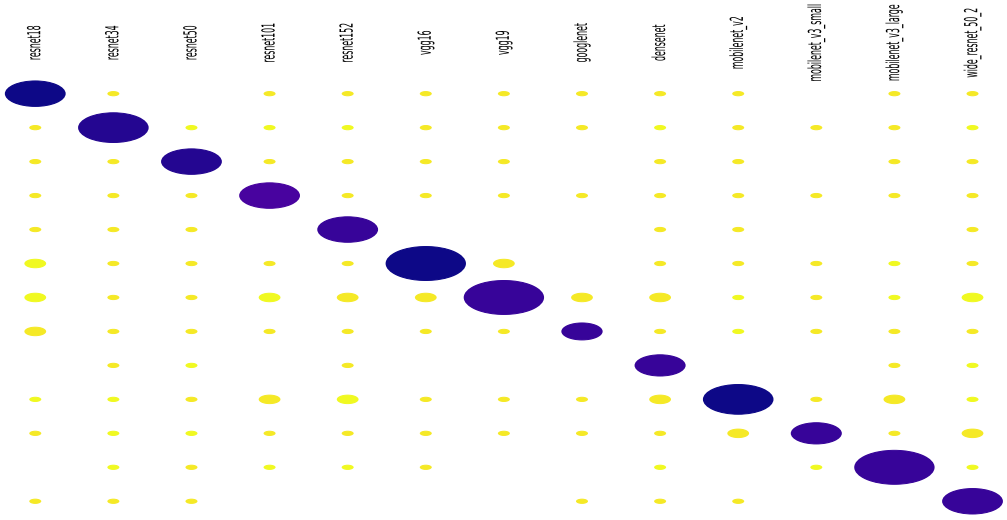


Figure A.16: std and avg for median target and PGD with epsilon 0.01

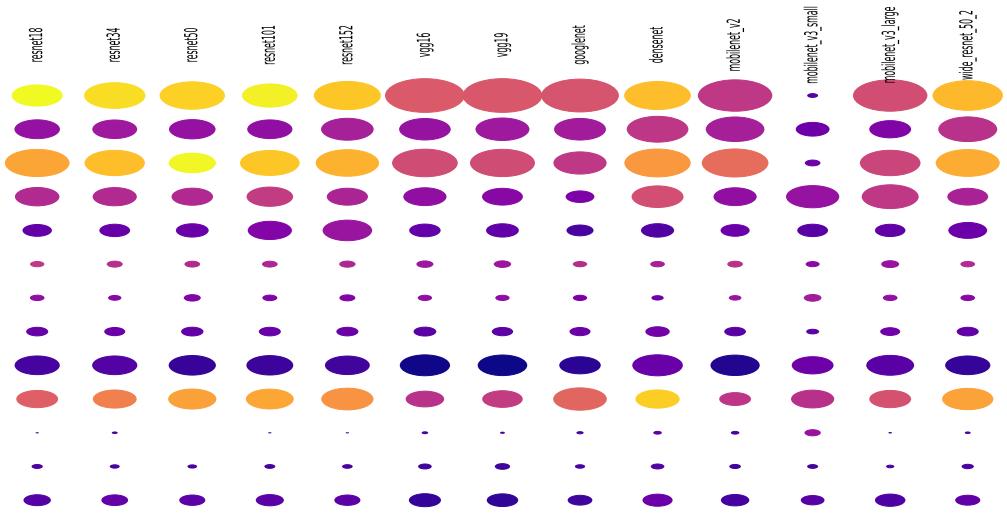


Figure A.17: std and avg for closest target and FGSM with epsilon 0.3

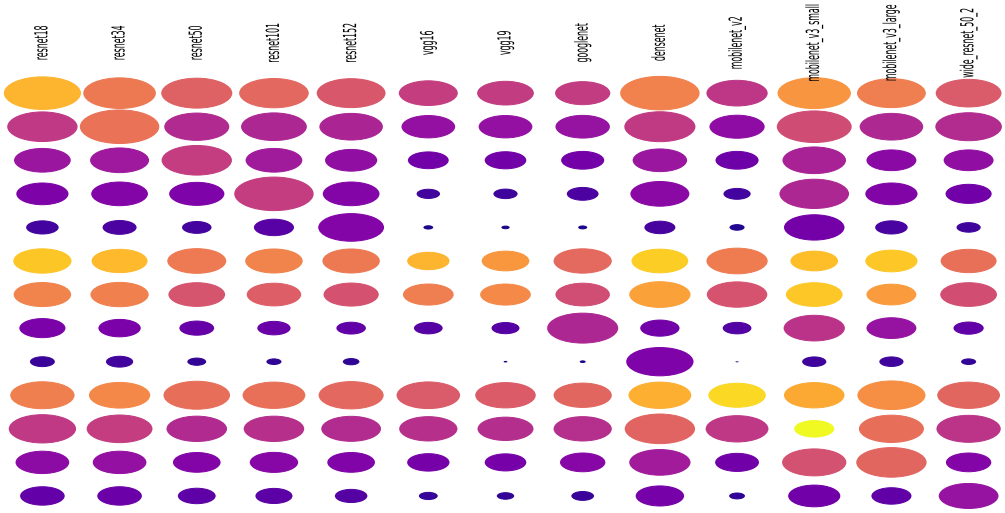


Figure A.18: std and avg for closest target and FGSM with epsilon 0.01

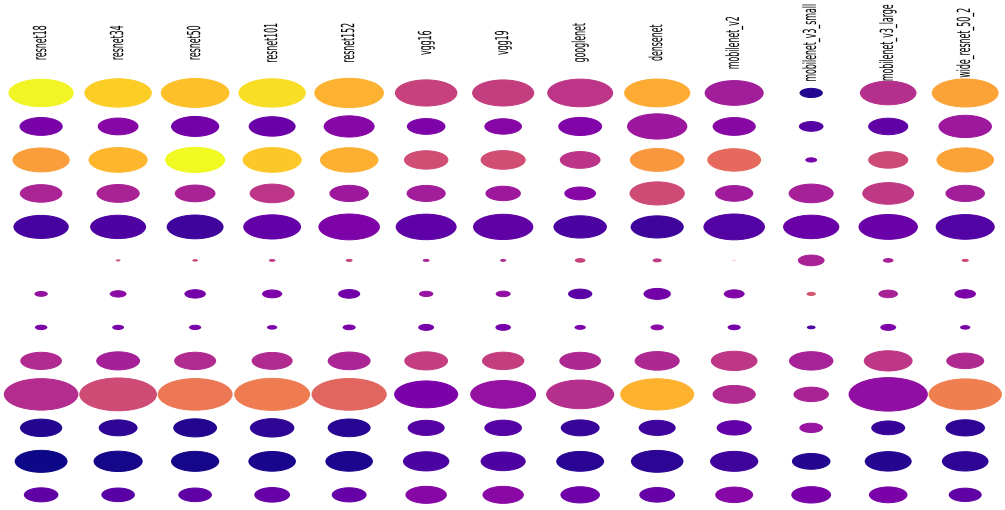


Figure A.19: std and avg for farthest target and FGSM with epsilon 0.3

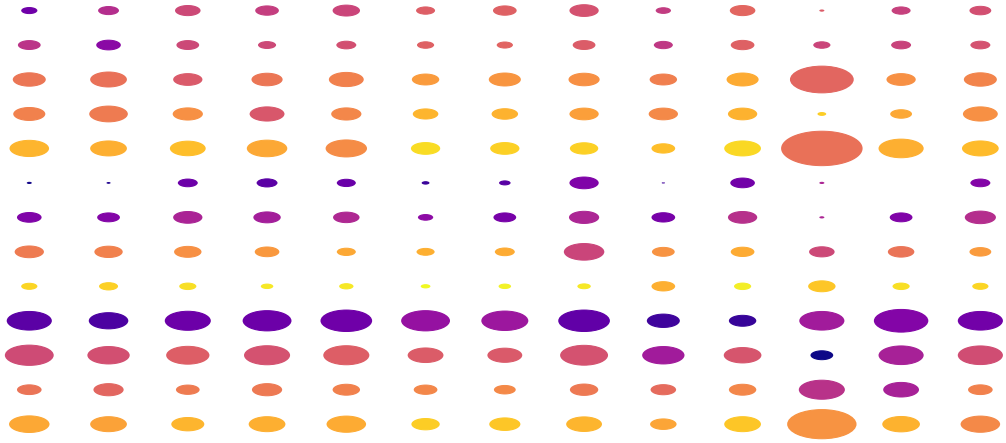


Figure A.20: std and avg for farthest target and FGSM with epsilon 0.1

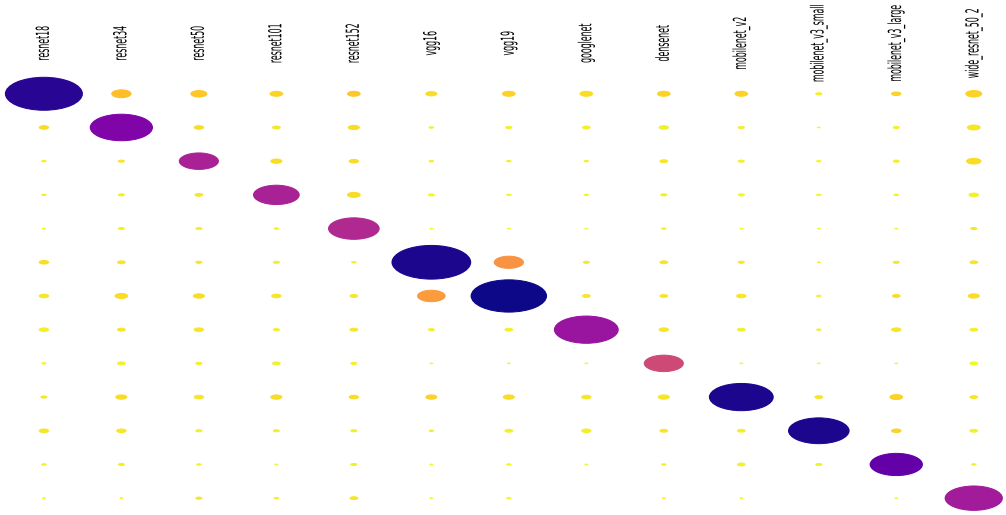


Figure A.21: std and avg for farthest target and FGSM with epsilon 0.01

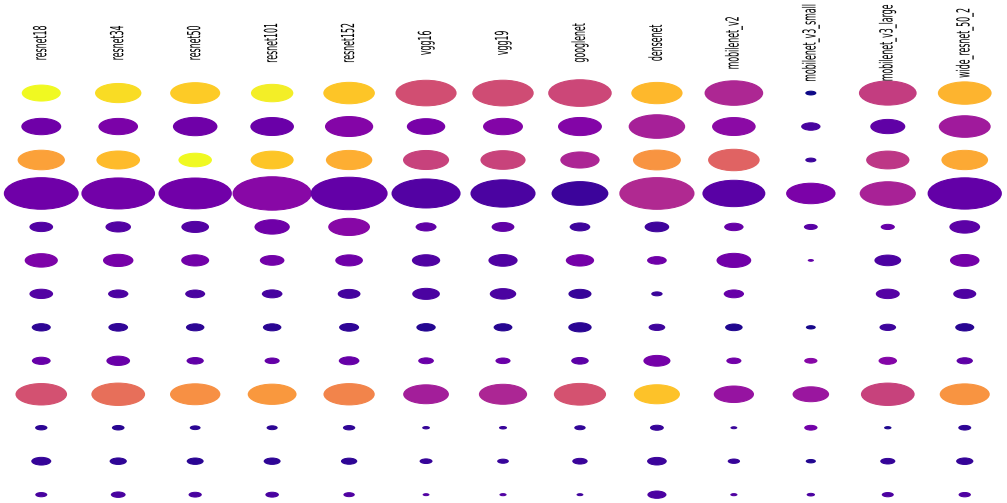


Figure A.22: std and avg for median target and FGSM with epsilon 0.3

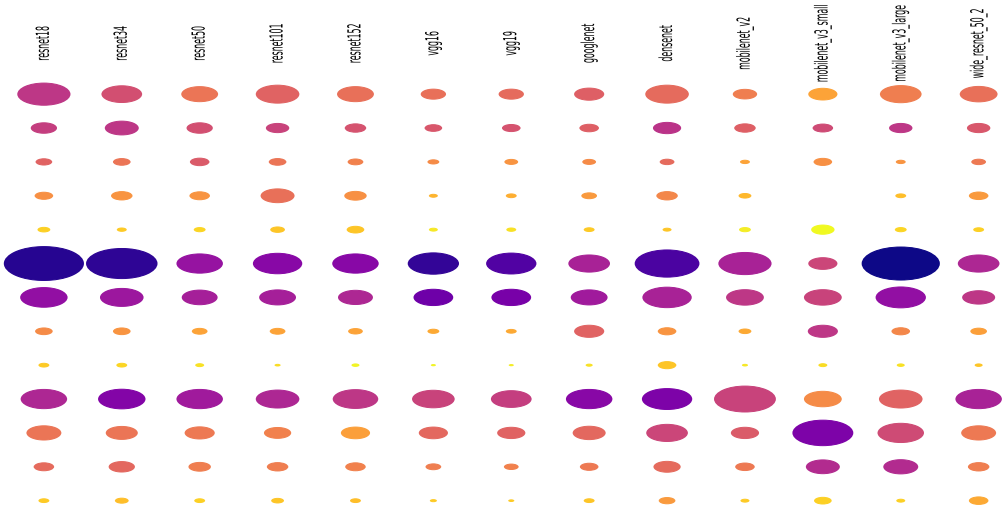


Figure A.23: std and avg for median target and FGSM with epsilon 0.1

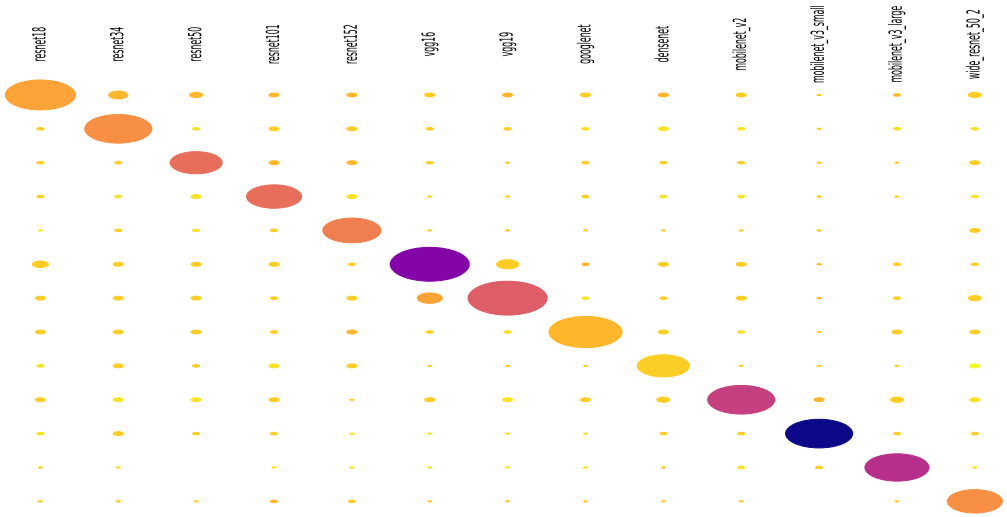


Figure A.24: std and avg for median target and FGSM with epsilon 0.01

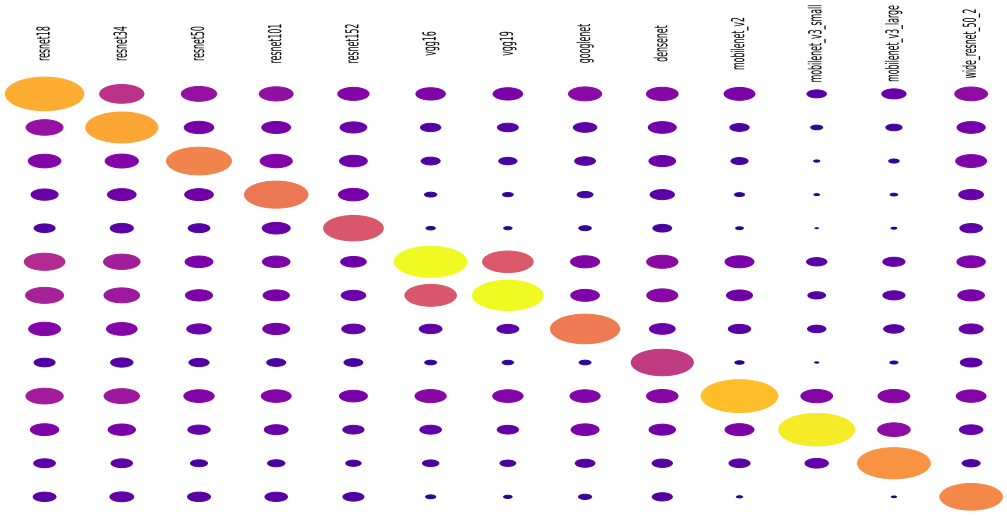


Figure A.25: std and avg for closest target and FGV with epsilon 0.3

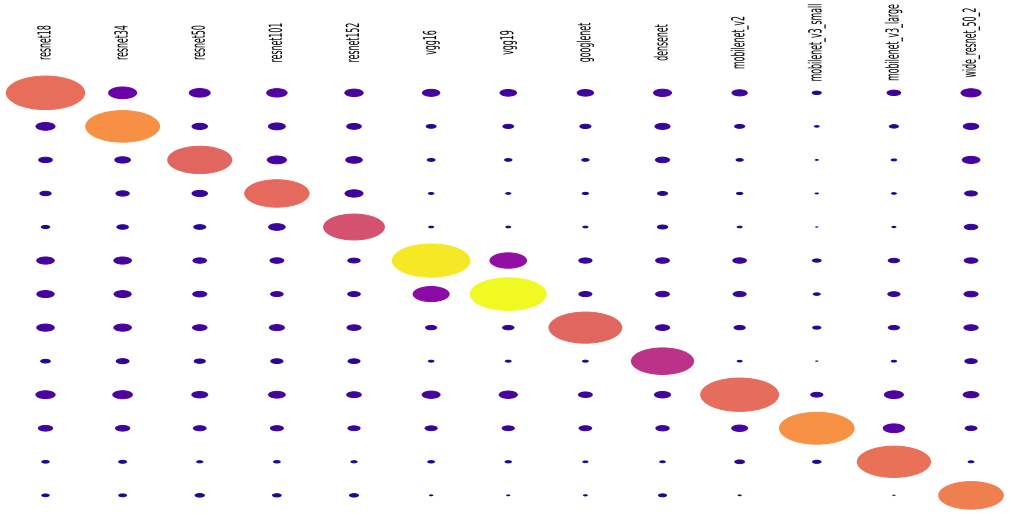


Figure A.26: std and avg for closest target and FGV with epsilon 0.1

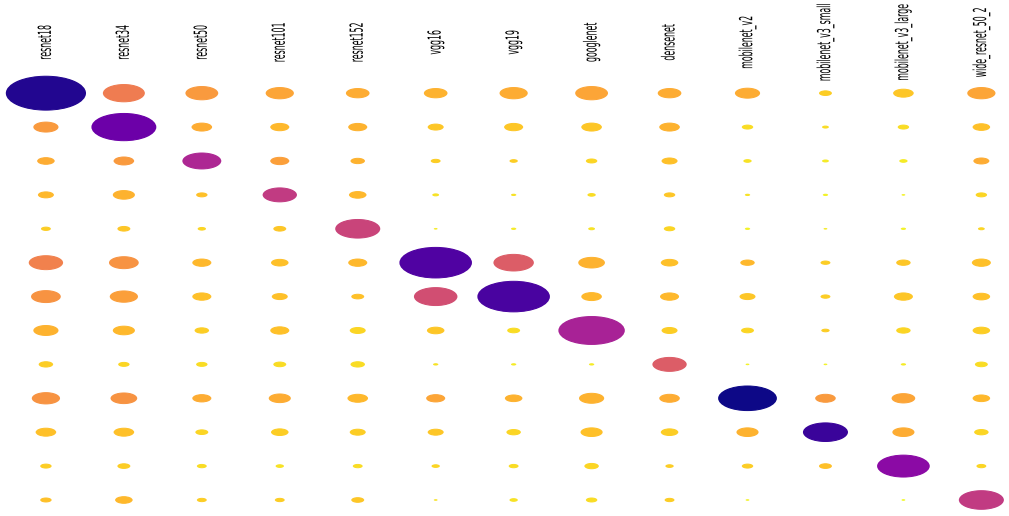


Figure A.27: std and avg for farthest target and FGV with epsilon 0.3

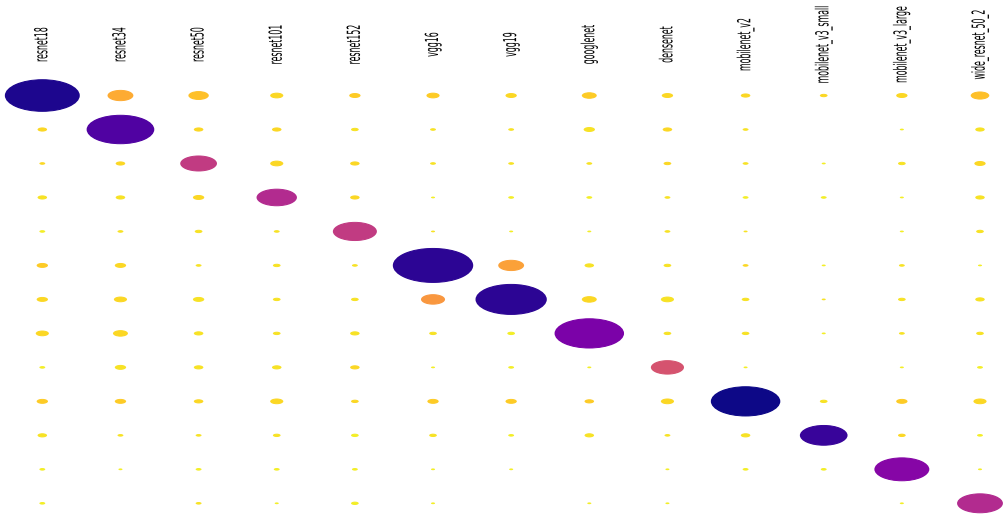


Figure A.28: std and avg for farthest target and FGV with epsilon 0.1

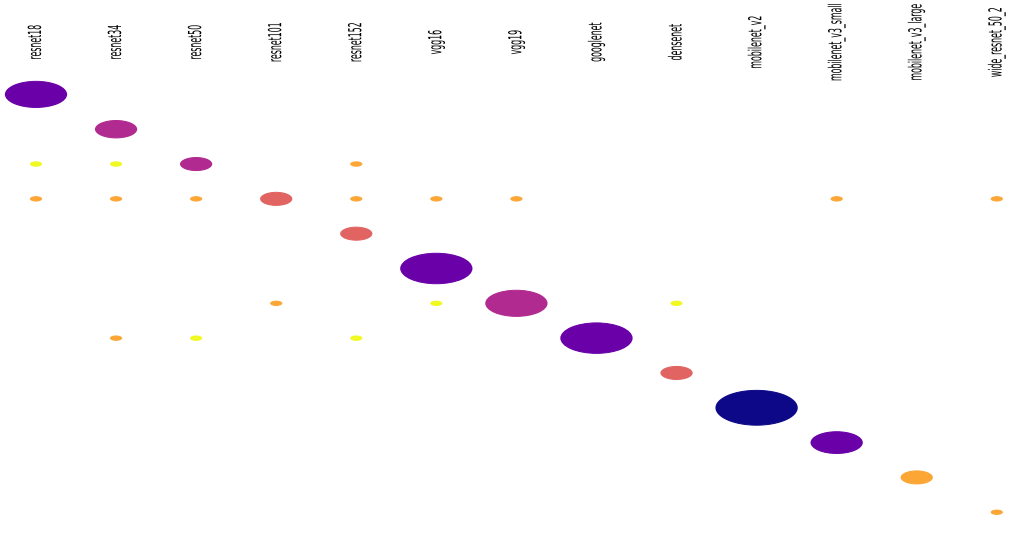


Figure A.29: std and avg for farthest target and FGV with epsilon 0.01

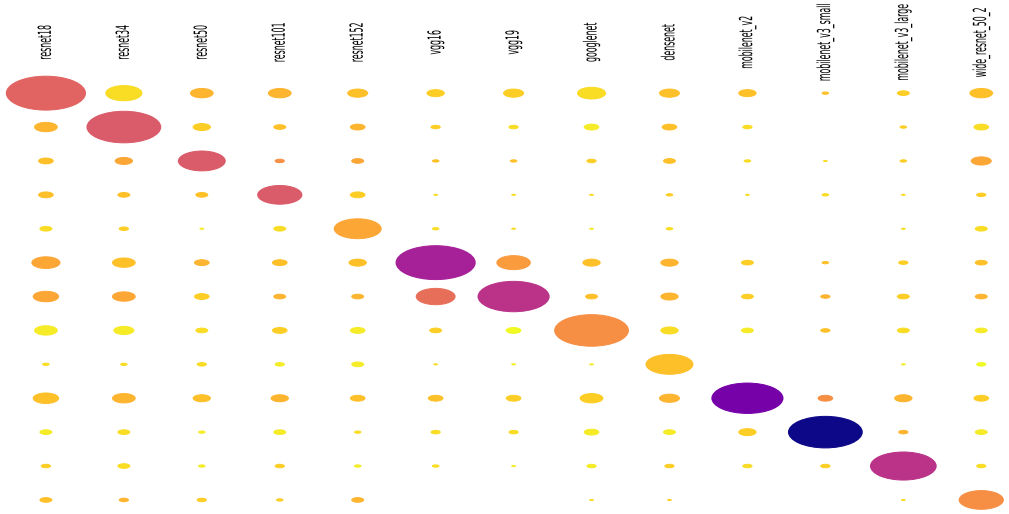


Figure A.30: std and avg for median target and FGV with epsilon 0.3

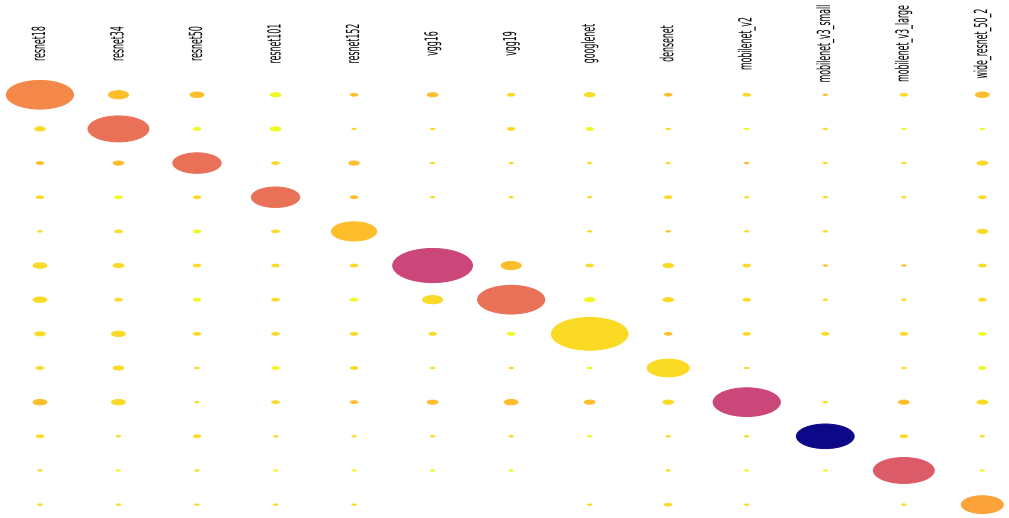


Figure A.31: std and avg for median target and FGV with epsilon 0.1

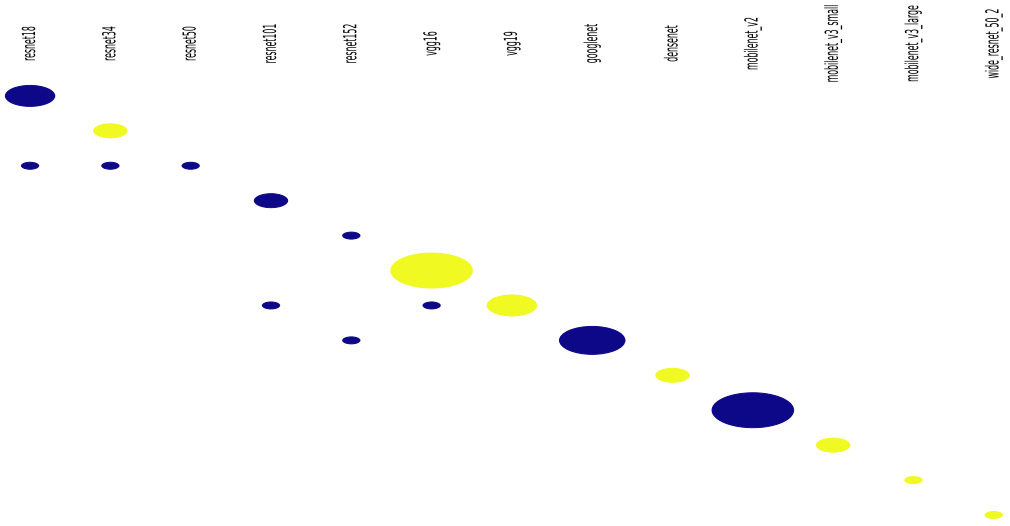


Figure A.32: std and avg for median target and FGV with epsilon 0.01

List of Figures

List of Tables

List of Listings

Bibliography

Bai, T., Luo, J., Zhao, J., Wen, B., and Wang, Q. (2021). Recent advances in adversarial training for adversarial robustness.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255.

Deselaers, T. and Ferrari, V. (2011). Visual and semantic similarity in imagenet. In *CVPR 2011*, pages 1777–1784.

Ding, G. W., Wang, L., and Jin, X. (2019). AdverTorch v0.1: An adversarial robustness toolbox based on pytorch. *arXiv preprint arXiv:1902.07623*.

Dong, Y., Fu, Q., Yang, X., Pang, T., Su, H., Xiao, Z., and Zhu, J. (2019). Benchmarking adversarial robustness. *CoRR*, abs/1912.11852.

Fezza, S. A., Bakhti, Y., Hamidouche, W., and Déforges, O. (2019). Perceptual evaluation of adversarial attacks for cnn-based image classification.

Goodfellow, I. J., Shlens, J., and Szegedy, C. (2014). Explaining and harnessing adversarial examples.

He, K., Zhang, X., Ren, S., and Sun, J. (2015). Deep residual learning for image recognition.

Huang, G., Liu, Z., van der Maaten, L., and Weinberger, K. Q. (2016). Densely connected convolutional networks.

Huang, T., Menkovski, V., Pei, Y., and Pechenizkiy, M. (2020). Bridging the performance gap between FGSM and PGD adversarial training. *CoRR*, abs/2011.05157.

Jin, J., Dundar, A., and Culurciello, E. (2015). Robust convolutional neural networks under adversarial noise. *CoRR*, abs/1511.06306.

kanth Nakka, K. and Salzmann, M. (2021). Learning transferable adversarial perturbations. In Beygelzimer, A., Dauphin, Y., Liang, P., and Vaughan, J. W., editors, *Advances in Neural Information Processing Systems*.

Karnala, V. and Campbell, M. (2021). Impact of model architecture against adversarial example's effectivity. *Journal of Student Research*, 10(2).

Kereliuk, C., Sturm, B. L., and Larsen, J. (2015). Deep learning and music adversaries. *CoRR*, abs/1507.04761.

Liu, N., Du, M., and Hu, X. (2020). Adversarial machine learning: An interpretation perspective. *CoRR*, abs/2004.11488.

Ma, X., Niu, Y., Gu, L., Wang, Y., Zhao, Y., Bailey, J., and Lu, F. (2021). Understanding adversarial attacks on deep learning based medical image analysis systems. *Pattern Recognition*, 110:107332.

Madry, A., Makelov, A., Schmidt, L., Tsipras, D., and Vladu, A. (2017). Towards deep learning models resistant to adversarial attacks.

Moosavi-Dezfooli, S., Fawzi, A., and Frossard, P. (2015). Deepfool: a simple and accurate method to fool deep neural networks. *CoRR*, abs/1511.04599.

Nguyen, A., Yosinski, J., and Clune, J. (2014). Deep neural networks are easily fooled: High confidence predictions for unrecognizable images.

Roads, B. D. and Love, B. C. (2020). Enriching imagenet with human similarity judgments and psychological embeddings.

Rozsa, A., Günther, M., and Boult, T. E. (2016a). Are accuracy and robustness correlated? *CoRR*, abs/1610.04563.

Rozsa, A., Rudd, E. M., and Boult, T. E. (2016b). Adversarial diversity and hard positive generation.

Simonyan, K. and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S. E., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. (2014a). Going deeper with convolutions. *CoRR*, abs/1409.4842.

Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I. J., and Fergus, R. (2014b). Intriguing properties of neural networks. In Bengio, Y. and LeCun, Y., editors, *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*.

Wang, H. and Wang, Y. (2022). Self-ensemble adversarial training for improved robustness.

Yang, Y.-Y., Rashtchian, C., Zhang, H., Salakhutdinov, R. R., and Chaudhuri, K. (2020). A closer look at accuracy vs. robustness. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M. F., and Lin, H., editors, *Advances in Neural Information Processing Systems*, volume 33, pages 8588–8601. Curran Associates, Inc.

Zheng, S., Song, Y., Leung, T., and Goodfellow, I. J. (2016). Improving the robustness of deep neural networks via stability training. *CoRR*, abs/1604.04326.