Black-box Adversarial Attacks using Substitute models: Effects of Data Distributions on Sample Transferability

Pietro M. Vigilanza Lorenzo
Supervisor(s): Stefanie Roos, Jiyue Huang, Chi Hong
EEMCS, Delft University of Technology, The Netherlands
June 17, 2022

A Dissertation Submitted to EEMCS faculty Delft University of Technology,
In Partial Fulfilment of the Requirements
For the Bachelor of Computer Science and Engineering

Abstract

Machine Learning (ML) models are vulnerable to **adversarial samples** — human imperceptible changes to regular input to elicit wrong output on a given model. Plenty of adversarial attacks assume an attacker has access to the underlying model or access to the data used to train the model. Instead, in this paper we focus on the effects the data distributions has on the transferability of adversarial samples under a "black-box" scenario. We assume an attacker has to train a separate model (the "substitute model") and generate adversaries using this independent model. The substitute models are trained with different data distributions: symmetric, cross-section or completely disjoint data to the one used to train the target model. The results demonstrate that an attacker only needs semantically similar data to execute an effective attack using a substitute model and well-known gradient based adversarial generation techniques. Under ideal attack scenarios, target model accuracies can drop below 50%. Furthermore, our research shows that generating adversarial images from an ensemble increases average attack success.

1 INTRODUCTION

Machine Learning (ML) techniques have gained enormous popularity within the last decade, and their presence in our everyday life decision making is increasingly common. Thus, understanding when and why ML systems can fail in their respective tasks is of utmost importance for safety and security of a digital system. Failing to acknowledge these errors can be a tangible risk for all sectors of society that heavily rely on the decisions of these algorithms. For example, one of the latest innovations in machine learning technology is to create models that can effectively drive a vehicle in public streets [1]. If a malicious actor can find a feasible vector of attack that can trick the system into making incorrect decisions, both the driver and nearby civilians can be in imminent danger.

One such area that exploits predictable vulnerabilities in how ML systems classify input is known as adversarial research. Adversarial samples are data intentionally generated by an attacker to cause a model to wrongly classify the input or make a mistake [4]. This area of research has been gaining momentum since Szegedy [26] and his colleagues uncovered that deep learning models used for computer vision tasks can make drastic mistakes in classification just by adding these small, human-imperceptible alterations to the original image as shown in Figure 1. These adversarial images are able to trick ML systems into making erroneous classifications with a high degree of success [6].

Multiple well-known approaches exist to generate adversarial data and trick a model. In this paper, we will focus primarily on a vector of attack known as *model substitution*, wherein an attacker trains a separate independent model that attempts to closely represent the model the attacker wishes to compromise i.e. the target model[8]. The attacker then generates adversarial data using the substitute model and verifies if the data is also adversarial in the target.

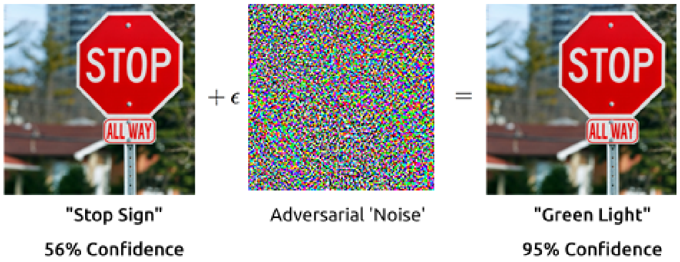

Figure 1: Example of a correctly classified road sign ("stop sign"), and later misclassified after imperceptible adversarial noise is added (turning into a "green light"). A huge risk vector for autonomous driving models

This research specifically delves into the effects of adversarial data transferability under different data distributions using multiple classifier models. The focus on the effects of data distributions is a novel extension to previous work, as the aim of this paper is to evaluate how well an attacker could fare if his data is different from the one used during the training phase of the target model. This premise also uncovers important implications in the capability of an attacker - if an attacker is able to successfully compromise a target model using different training data, it implies that an attacker simply needs data that is "similar" enough to the one used by the target to successfully conduct an attack.

The results from this research[1] shows that an attacker does not necessarily need the same data used by the target model to elicit drops in accuracy below 50% from the target when feeding adversarial samples generated by the substitute model.

2 RELATED WORK

Before delving further into the research, we will briefly look at some of the body of work related with substitute model attacks, semantic similarity and robustness of data.

2.1 Using a substitute model for black-box attacks

The underlying assumption of a black-box attack is that an attacker does not have access to the underlying model in question to generate adversarial samples. A worrying discovery in the field shows that rather than being an impediment, an attacker does not need to know the underlying model to generate successful adversarial samples. Usually, adversarial samples that are found in independent models trained to solve the same task have high chance of also being adversarial in the target model. This property is known as *adversarial trasnferability* [16] [28] [29]. The implication is that adversarial samples are *not* unique to a specific model as long as the models solve a similar problem [6].

Furthermore, Papernot et. al. [19] demonstrated that adversarial attacks using substitute models can be both successful in intra-technique and cross-technique situations. This means these attacks also succeed when the substitute model uses a

[1] all code is openly available in the following link: https://github.com/Ray-Escobar/substitute$_attack$

different ML technique to that of the target. The results show that adversarial attacks are feasible under a black-box scenario where the attacker does not have access to the architecture of the underlying target model.

2.2 Adversarial attacks using semantically similar data

To our knowledge, only one example of using semantically similar data to conduct an attack has been used. Pepernot et. al. [20] successfully generated adversarial data by training a substitute model using new MNIST [14] numbers written on a track-pad, and adapting them so that the new numbers are similar to the original MNIST dataset numbers. This small example with MNIST does hint that two models can be trained on distinct, semantically similar data to generate adversarial samples. Further research on this topic can illustrate how well adversarial samples transfer from one model to another when their training data becomes increasingly different and complex.

2.3 Robust and non-robust data features

Most of the spotlight in adversarial research has been given to uncovering new forms of attack, or training models that are resistant to modern adversarial techniques. These papers focus on the models internals while potentially ignoring the effects the dataset itself might have on the model. Ilyas et. al. [11] argues that adversarial samples exist in deep neural networks due to the features present in the dataset that modern classifiers use to reach high degree of classification. The researchers demonstrate that the existence of "non-robust" features in images (i.e. human-imperceptible features present in the data distribution of the samples) are what allow models to reach high levels of accuracy at the cost of becoming susceptible to adversarial samples.

The theories presented in this research suggest that adversarial data can be generated from a substitute model as long as the substitute model also learns similar non-robust features to the ones present in the target model. This would indicate that transferability between models could be due to the inherent non-robustness of dataset distributions.

2.4 Research Gap

From the previous section, we noticed that there seems to be a gap in research on the efficacy of successfully creating adversarial samples from a substitute model trained with data that follows a different distribution from the one used by target model. Papernot et. al. [20] did showcase that the attack vector was possible in a rather small experiment with a very simple MNIST dataset. Using different distributions on a feature-rich dataset could yield interesting results that could line up well with the previously conducted MNIST experiment, and further solidify the notion that an attacker simply needs similar enough data.

Furthermore, Papernot's experiment showcases how to conduct a substitute attack, but could be further substantiated with other practical and concrete reason that shows why the samples are transferable. Looking deeper into how both the target and substitute models form their decision boundaries could yield interesting insights into what is happening inside the model. That is why Ilyas' research on non-robustness is a fit method to explain why samples transfer from one model to another. If two separate models are using the same non-robust features to perform its classification, then we can safely conclude that transferability between independent model occurs due to the non-robust features present in the dataset. Combining both Papernot's and Ilyas' insights in adversarial research would present a new explanation for adversarial transferability.

In the next sections we will present the methods used to combine both approaches to conduct our experiment. Using these two insights, this research will primarily focus on training a substitute model by taking a dataset from a separate distribution that still preserves key features that make it *semantically similar* to the ones used to train the target model. Creating a concrete framework for measuring semantic similarity is out of the scope of this paper, but we intuitively define semantically similar datasets as a two datasets that are derived from a different distributions, yet the underlying features that both datasets contain are equal — thus making them fit to solve the same classification problem.

3 PROBLEM DESCRIPTION

The following section describes the components used to evaluate different models under a substitute attack. This experiment is based on a binary classification problem in which Convolutional Neural Networks (CNN) are tasked with correctly differentiating cat and dog images. The section starts with a brief description of CNNs, followed by defining our threat model and finally explaining the dataset splits used to create new semantically similar distributions from a single big dataset.

3.1 Convolutional Neural Networks (CNN)

In the field of deep learning a CNN is a type of deep neural network most commonly used in image classification tasks. These networks use a special architecture known as convolution layer which uses *local receptive fields* composed of kernels and pooling layers to reduce the amount of weights a model uses [17]. Following the convolution layers, the output is fed into a fully connected layer which then determines a classification.

As such, a CNN is represented by the following mapping function:

$$F(\theta, X) : R^d \to R^L$$

Where θ are the weights of the model, $X \in R^d$ is a vector of real numbers of d dimensions from a distribution D, and L is the number of classes in our classification problem.

In a given classification problem, we optimize the set of parameters $\theta_F = \{\theta_1, ..., \theta_n\}$ during a training phase so that in a test phase the parameters output a confidence $F(X, \theta) = y$ if a given image is a cat or a dog. To optimize the performance of a model, a CNN uses a *loss function* that measures how far the model is from the expected results. We can define a loss function as the following mapping:

$$J(\theta, X, Y) : R^L \to R$$

As the mapping suggests, the function takes the output vector from our CNN, known as the *logits*, and outputs a real number that measure how far the output is from the expected Y. The greater this magnitude, the farther our model is from a correct output. The optimization process works by taking the gradient of the loss function in terms of weights and biases to slowly decrease the error of the network.

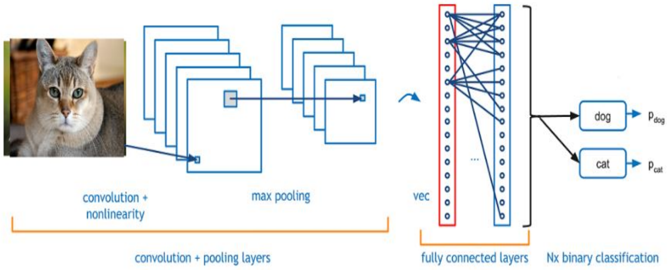

Figure 2: Architecture of the cat or dog CNN model[23].

3.2 Threat Model Taxonomy

Before delving into the specific strategies used to setup the experiment, first we will describe the components of the threat model. The given scenario assumes an attacker wishes to compromise a CNN's accuracy without having access to the model or to its hyper-parameters. This classic scenario is known as a *black-box* attack, where the attacker tries compromise a system in which they have limited knowledge or access to the internals. The attacker's goal is to produce a minimally altered image that is miscalssified by the CNN in question. The threat model is composed of the following actors:

Target Model: The *target models* are CNNs trained at high accuracy that the attacker wishes to compromise. The attacker aims to generate adversarial data such that the target outputs an incorrect classification on an otherwise correctly classified image. The attacker wishes to minimize the target model's accuracy.

Substitute Model: The *substitute model* is the model trained by the attacker to generate adversarial images. The attacker only knows the type of data required to train a substitute model, and uses gradient based attacks to generate adversarial images using the substitute model. The attacker also knows the classification problem the target classifier is solving, thus the substitute model is trained to solve the same exact problem.

Robustness: As defined by frameworks from Ilyas et. al. [11] and Tsipras et. al. [27], features of a data distribution D may have *"robust"* and *"non-robust"* features. As defined by these frameworks, non-robust features are features that correlate to the true expected label of the sample Y, but when an adversarial perturbation is applied, these non-robust features correlate to wrong label Y'. On the other hand, robust features correlate to the expected label Y regardless if an adversarial perturbation has been applied.

One can intuitively think about robust features as the human-perceivable attributes of an image — "a furry animal with a snout, ears and four legs". Non-robust features are visually imperceptible and are inherent to the overall distribution D.

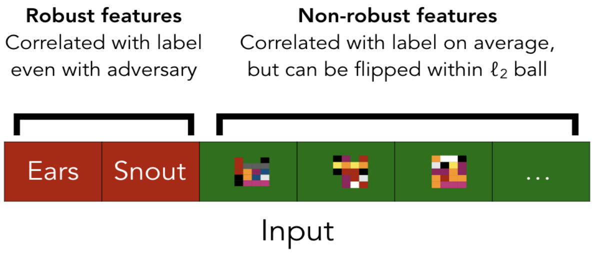

Figure 3: Robust features and non-robust features[3].

An attacker alters these non-robust features within a certain $l2$ euclidean boundary, referred to as "ball", in the hopes of causing an incorrect classification by an arbitrary classifier within the output domain L. Figure 4 shows an example of how these robust and non-robust features are perceived.

Gradient-Based Attacks: This research only considers gradient based attacks. Under normal training conditions, to increase the model accuracy the gradient is taken with respects to the weights and biases of the CNN model. Over multiple iterations, the slight changes to these parameters slowly increase the model's accuracy.

The general notion of gradient based attacks is that the gradient of the model is instead calculated with respects to the input vector [5] [6]. Since the gradient points towards the direction that increases the cost function given an input vector, an attacker hopes that by applying this gradient to the original image the target model misclassifies the image.

The two gradient based attacks explored in this experiment is the Fast Gradient Sign Method (FGSM) [6] and Projected Gradient Descent (PGD) [13]. The definition of both attacks are listed below. Note that PGD is an iterative version of FGSM:

- Fast Gradient Sign Method (FGSM) [6]

$$X^{adv} = X + \epsilon \operatorname{sign}(\nabla_X J(\theta, X, Y))$$

- Projected Gradient Descent (PGD) [13]

$$X_0^{adv} = X$$

$$X_{N+1}^{adv} = Clip_{X,\epsilon}(X_N^{adv} + \alpha \operatorname{sign}(\nabla_X J(\theta, X_N^{adv}, Y)))$$

3.3 Dataset Distributions and Semantic Similarity

The core of this experiment is to generate two datasets D_T and D_S from common dataset distribution D such that D_T and D_S are semantically similar. Classifier C_T (target model) is then trained using dataset D_T, while classifier C_S (the substitute model) is trained with D_S. The three splits used to alter the distributions of the datasets are the following:

1. **Equal**. D_T and D_S have all items in common.
2. **Cross Section**. D_T and D_S share some items.
3. **Disjoint**. D_T and D_S share no items.

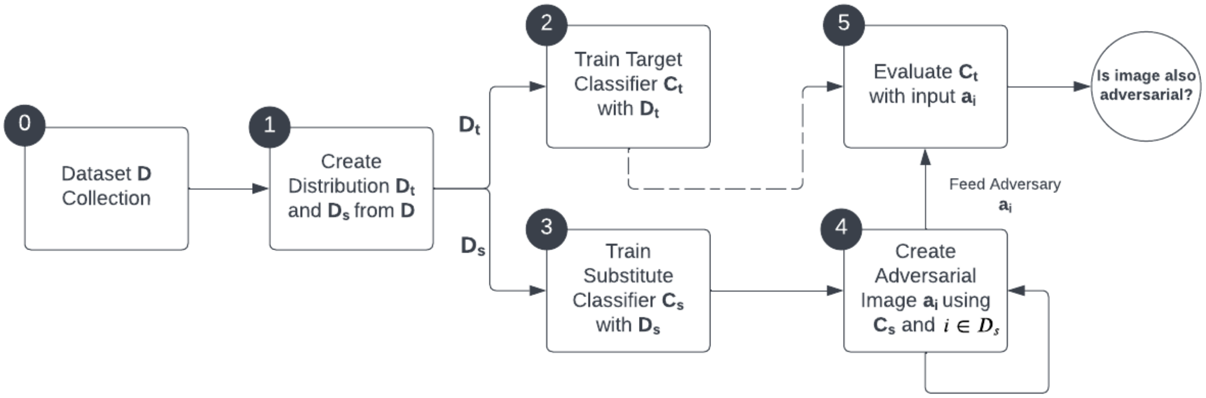
0
Dataset D Collection
1
Create Distribution D_t and D_s from D
D_t
D_s
2
Train Target Classifier C_t with D_t
3
Train Substitute Classifier C_s with D_s
4
Create Adversarial Image a_i using C_s and $i \in D_s$
Feed Adversary a_i
5
Evaluate C_t with input a_i
Is image also adversarial?

Figure 4: Anatomy of an adversarial attack via substitute model. (0) Dataset is collected and then (1) split according to a data distribution. We train a (2) target model and a (3) substitute model with their respective data. (4) Adversarial data can be generated iteratively over a dataset and then (5) fed to the target model.

Since its necessary that both our datasets D_T and D_S can be used to solve the same classification problem, both datasets need to be *semantically similar*. In this context, semantic similarity consists of two datasets that follow different data distributions, but portray similar features. This allows either dataset to be used for solving the same classification problem.

4 EXPERIMENTAL SETUP

The experimental setup is composed of a cats and dogs dataset, pre-trained CNN models and the execution of the adversarial attacks. This section explains in detail how the models were trained with the dataset, and how the attacks were conducted. Figure 4 above can be referenced for a high level diagram of the substitute attack procedure.

4.1 Dataset - OxfordIIITPets

The ideal dataset for this experiment requires having substantial data that can be easily split into our three distributions while preserving semantic similarity. In the context of a classification problem, we wish to split data items on an arbitrary feature while preserving the output label domain L of the target classification problem i.e. classifying an image as a cat or a dog.

To derive different data distributions that are semantically similar, we use the Oxford IIIT Pets image dataset [21]. This dataset contains a total of 7,349 cats and dogs split on 37 unique breeds. The original purpose of the dataset is to solve the multi-class breed classification problem.

The reason this dataset fulfils the requirements for this experiment is that it allows to make simple splits on breeds while still preserving the nature of the cats and dogs classification problem. Given the numerous breeds, splits can be made to create datasets that have symmetric sets of breeds, share a cross-section of breeds or have no breeds in common and use them to train the classifiers.

It is important to highlight that different datasets can be considered for this experiment. For example, one can take datasets with traffic signs images from around the world and create splits based on the country of origin, or take a dataset of vehicles and create splits based on old and new vehicles. The important constraint to preserve is that the datasets should be semantically similar.

4.2 CNN Models Used

The models used for this experiment consist of popular, and accessible CNNs that are proficient at solving image classification problems. All models were created and trained using the machine learning library PyTorch [22]. The following three models below are used to generate target and the substitute models for this experiment:

1. GoogLeNet [25]
2. Resnet-50 [8]
3. DenseNet-121 [9]

The reasoning behind choosing this subset of models was due to their sizes. GoogLeNet is the smallest model with 11 million parameters, followed by Resnet-50 with 23 million parameters, while DenseNet-121 is the biggest model with 60 million parameters. By choosing the models of different architectures and sizes we can make our attacks more diverse and objective.

All models in the experiment are trained using an ADAM stochastic optimizer[12] for nine epochs or until the model reaches a 99% accuracy on a given test set. After seven epochs, the learning rate is slightly decremented by a value of 0.01 to make parameter changes less drastic and more precise.

Furthermore, training is done by fine-tuning pre-trained models via transfer-learning[30] — a method by which one adjusts a pre-trained model by changing the output layer to represent the desired classification problem, and modify the already existing weights and biases of the model. In this case the the models are all pre-trained with ImageNet[2] data and the final layer is altered to only have two output neurons. Using the transfer-learning technique allowed to train models that reach classification accuracy of 95% or more in fewer epochs and using less data.

To guarantee greater entropy between models while using transfer-learning, non of the weights and biases were frozen during training phase and the training gradients were applied throughout the whole model. This creates a desired setup in which any two model's weights and biases have a greater difference between each other, thus simulating a realistic black-box attack where the attacker may be far from having the same parameter values as the target model.

4.3 Executing the adversarial attack

The experimental phase closely resembles the work of Liu et. al. [15] with the added factor that data distributions are not the same across model, and the adversarial samples are derived using the substitute models.

The ensuing process starts by preparing the datasets that represent the three data distributions described above. For the symmetric datasets the training and test data used was the one suggested by the OxfordIIIT dataset. For the other two distributions, a process of random sampling without replacement was used to create random splits on the data based on breeds. Once the datasets are prepared, each target model and substitute model are loaded and trained with the assigned data.

After the training phase, the attack phase is performed and measured under intra-model and cross-model attacks. For the attack, a random batch of 200 images are selected from the whole OxfordIIIT dataset. The selected batch is first evaluated on the target model untampered to measure a baseline accuracy. The accuracy of the model is measured by counting all correct classifications over the number of items in the whole batch. The higher this value, the better the model is at solving the classification problem.

After the baseline accuracy evaluation, each image from the batch is then used to generate two adversarial samples where one is made using FGSM and the other using PGD. The attacks were made possible using Clever Hans [18], an open source library which has a collection of popular adversarial attacks.

For all experiments we used an α value of 0.1 for each attack. This α value determines an alteration range/ball in an image — the greater this α the greater the image perturbation. The α value chosen for this experiment was arbitrarily picked, but resulted in images where perturbations are unnoticeable for the human viewer as depicted in Figure 5.

Additionally, since PGD is iterative it also requires a step size δ chosen to be 0.01. This step size allows PGD to explore the direction of highest increment in the loss function given an exploration size of α. Forty iterations were used for the PGD exploration phase.

After the 200 adversarial images are generated per attack, the target's accuracy is measured using the adversarial batch and compared with the original, untampered data accuracy.

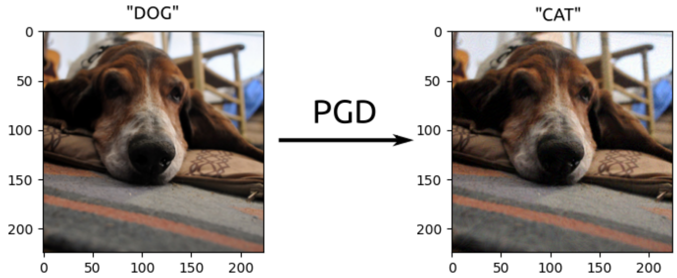

Figure 5: Example of a correctly classified dog, which is then classified as a cat after PGD is applied in one of our models. Notice that using an α value of 0.1 makes human-imperceptible changes on the image.

4.4 Verifying Models use similar Non-Robust features

The final step in the test-bed is to generate a non-robust dataset D_{NR}. Non-robust data per-definition, is data that is perturbed with adversarial methods to be classified differently than it's actual real class Y [11]. The process to generate a non-robust dataset follows closely the one by Ilyas et. al. [11] with one distinct difference. Instead of using a single model to generate non-robust data, we create an ensemble of three substitute models — one from each of the aforementioned CNNs chosen for the experiment. Each model is trained with the whole OxfordIIIT dataset until accuracy levels of each model is above 95%.

Algorithm 1 An algorithm to generate D_{NR}. Assumes PGD runs for 40 iterations.

```
Require: $C = \{C_1, C_2, ...C_n\}$    ▷ Ensemble of classifiers
Require: D    ▷ Dataset. In this case OxfordIIIT Pets
  $D_{NR} \leftarrow \{\}$
  $\alpha \leftarrow 0.1$    ▷ Perturbation ball
  $\epsilon \leftarrow 0.01$    ▷ Perturbation step
  for $x \in D$ do
      if $x$ is correctly classified in ensemble $C$ then
          for $c \in C$ do
              $x' \leftarrow PGD(c, x, \alpha, \epsilon)$
              if $x'$ is adversarial for all $c \in C$ then
                  $D_{NR} \leftarrow D_{NR} \cup x'$
                  break
              end if
          end for
      end if
  end for
```

With our newly trained ensemble, we pick images from the OxfordIIIT dataset and apply PGD using one of the substitute models, and verify that the image is also adversarial in the the other two members of the substitute ensemble. This entails making sure that cat or dog images are classified as their opposite class by all three substitute models after adversarial perturbation is applied. The parameters used for the PGD attack were the same as discussed in 4.3. Pseudo-code for this algorithm is present in Algorithm 1.

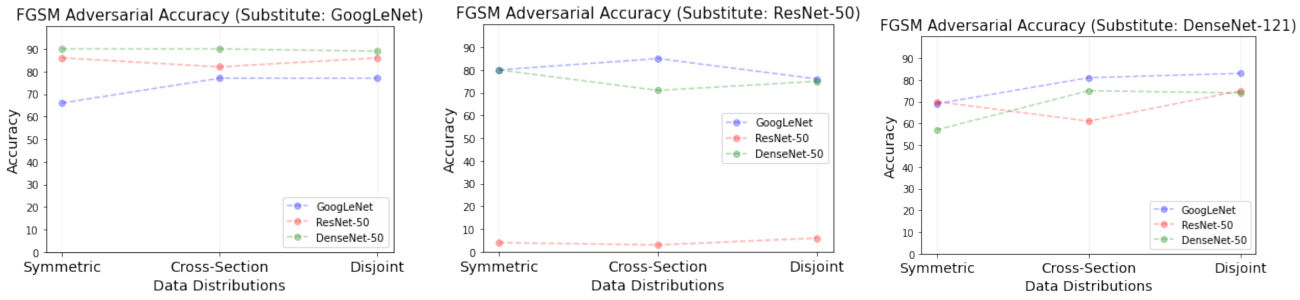

Figure 6: FGSM attack transferability. Accuracy of target models (lines in graph) under the different distributions with following substitutes: (a) GoogLeNet (b) ResNet-50 (c) DenseNet-121

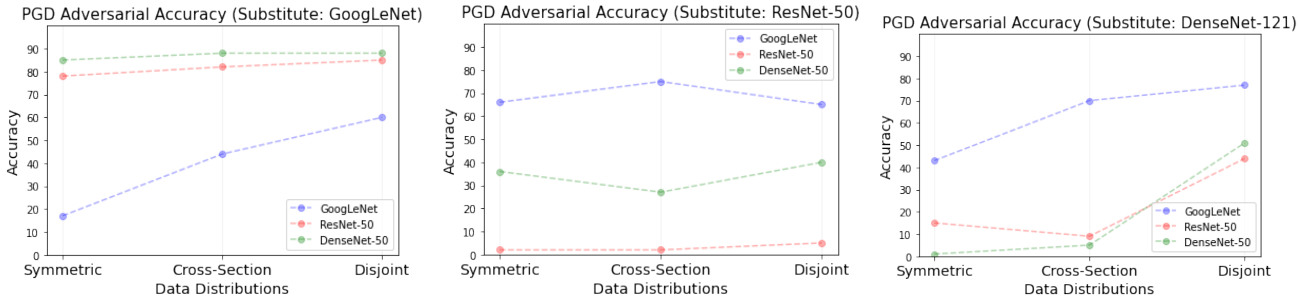

Figure 7: PGD attack transferability. Accuracy of target models (lines in graph) under the different distributions with following substitutes: (a) GoogLeNet (b) ResNet-50 (c) DenseNet-121

Using this technique we were able to generate a small dataset of 500 images that together form D_{NR}. These images were then subsequently fed into target and substitute models to measure if the images elicited a wrong classification from both the adversary and the substitute. If the image is wrongly classified in both models, then we can conclude that both models were using the same non-robust features to perform their respective classifications.

5 RESULTS

The result section is split into three parts. For all sections we measure the accuracy of models — the percentage of images classified correctly given a batch to evaluate. The higher the accuracy, the better the model performed in classifying the image batch.

We first look at the accuracies reached with all models in the classification problem without any image perturbations. Afterwards, the accuracy of these same models are measured after adversarial samples are generated using the substitute models under the different data distributions. The final part uses the the non-robust dataset D_{NR} to measure how susceptible the target and substitute models are to the same adversarial alteration of non-robust features.

5.1 Baseline Accuracies

Given the simple cat or dog binary classification problem, all target models trained have near perfect accuracy at identifying the animal type in their respective test sets as seen in Table 1. The results are the average accuracies obtained by randomly selecting 200 samples from the whole dataset.

In section 5.2, we take these same 200 samples and apply an adversarial perturbation to measure the effects in the accuracy of the models.

Target Model Accuracy on Clean Test Data			
Models	GoogLeNet	ResNet-50	DenseNet-121
Symmetric	0.95	0.98	1.00
Cross-Section	0.97	0.99	0.99
Disjoint	0.99	0.97	0.98

Table 1: Target model accuracy in cat or dog image classification with no perturbations in images. All models are very accurate with their assigned test data.

5.2 Attack Transferability

We now look at the success of transferability using our substitute models. As shown by our baseline accuracies, when given non-adversarial data these models are proficient at classifying our original 200 cat and dog images. After adversarial perturbations are applied, substantial degradation of accuracy is observed over three data splits.

Symmetric: Both target and substitute models are trained with the same exact data. For the training, we used the recommended test and train data provided by OxfordIIIT dataset. The results in Figure 5 and Figure 6 show that symmetric attacks are the most effective attack overall. A well selected model by an attacker can decrease performance of the target to worse than a guess. The results closely follow what had

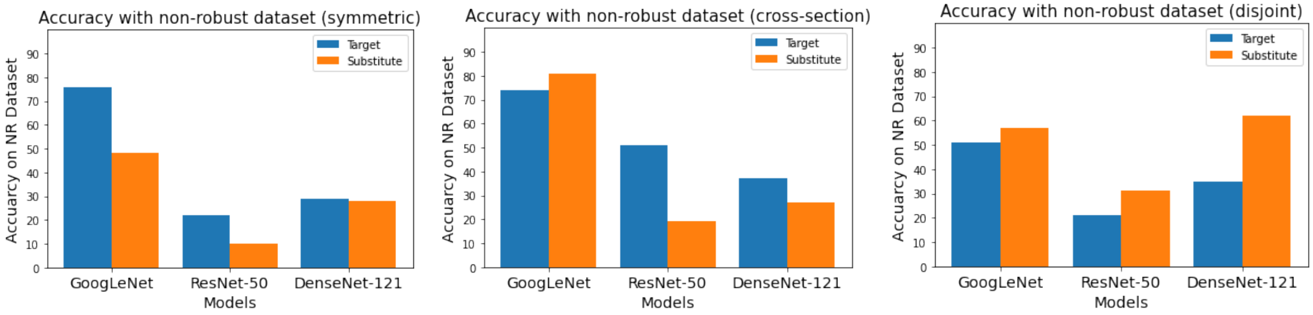

Figure 8: Performance of target and substitute model with D_{NR} dataset (a) Symmetric (b) Cross-Section (c) Disjoint

been seen in previous adversarial research [15].

Cross-Section: The following experiment assumes a cross-section of size 10. This means that the data used to train the target and substitute models had 10 breeds in common each. In the context of the OxfordIIIT, this means that slightly less than half of the samples in the the target and substitute datasets are the same. The results are similar to those using symmetric dataset except for GoogLeNet. In the cross-section attacks the data shows the the GoogLeNet target is generally more resilient than the other two targets except when the substitute is also a GoogLeNet. On the other hand, both the Resnet-50 and Densenet-121 accuracies are greatly diminished with PGD when using the Resnet-50 or DenseNet-121 as substitutes.

Disjoint: Under a disjoint distribution, the target and substitute models were trained with a total of 18 breeds each. Under the disjoint experiment no two images or breeds were equal between the datasets of the substitute and the target. To ensure this, distinct breeds were picked at random for each dataset. The results in Figure 5 and Figure 6 show that overall, the targets are the most resilient to the substitute attack compared to symmetric and cross-section attacks. This tendency can be seen with every model — the greater the difference between the datasets, transferability of adversarial samples decreases (and accuracy of model increases). It's still important to highlight that in best-case scenarios, all target models under any substitute still drop to accuracies that could be considered low by the standards of well trained models. When using the correct substitute model and PGD attack the accuracy of the target models can drop to less than 50%.

5.3 Sensing Non-Robustness

In our second experiment, we generated dataset D_{NR} composed of only non-robust samples. These non-robust samples are generated by applying PGD on given image s_{adv} using one of our three substitutes from an ensemble such that image s_{adv} is adversarial for all ensemble members. In the context of our dataset, this would mean perturbing a cat such that all three models classify it as a dog, or vice versa. The method above yielded around 500 adversarial images from a set of 2500 randomly selected images from the whole OxfordIIIT dataset.

We then measure the accuracy of the same target and substitute models used in section 5.2 under the new dataset D_{NR}. Similar accuracies between target and substitute model of same architecture would suggest that the misclassification of the image is elicited by similar non-robust features. The main aim of the experiment is to understand adversarial transferability as a consequence of models using similar non-robust features to assist in their classification task. The results of the experiment are present in Figure 8.

Compared to the results from the first experiment, the results shown in the graphs have more variance and are less conclusive. Some results point strongly towards our hypothesis of similar non-robust feature usage, such as the symmetric DenseNet-121 results, both the Resnet-50 from symmetric and disjoint experiments, and both the GoogLeNet results from the cross-section and disjoint experiment. The target and substitute models in these tests had a similar accuracies and within a 15% range in distance.

Other results were less conclusive, such as the symmetric GoogLeNet that had a difference of 27% in accuracy. Furthermore, some model pairs had very similar accuracies in one distribution test, while in others the difference could be very sizable. This phenomenon was present in the GoogLeNet, where in the symmetric distribution the accuracies between the target and substitute were very distant, while in the other two distributions the accuracies were rather similar.

Two additional observations arose from conducting the experiment. We can notice that overall, these non-robust samples s_{adv} on average can transfer to any model. Compared to the previous experiment, the overall accuracies of all models were on average lower. Rather than seeing one model drastically under performing, we can see that with the non-robust data all three models generally under perform in the classification task (with GoogLeNet still being the most resilient). Additionally, once inspecting the samples from our non-robust data, we noticed that a clear majority of the samples were from cats being miscalssified as dogs. Even more interesting is that the images of adversarial dogs had an uncanny similarity to cat features. An explanation of this observation could be due to the fact that cats were under represented in the OxfordIIIT dataset. Overall, 70% of the samples were dogs while only 30% of the samples were cats. This could suggest that our classifiers had trouble identifying cat features, thus making cat images, and cat-like images more susceptible to adversarial perturbation.

6 CONCLUSION AND DISCUSSION

This section summarizes the implications from the results found in section 5.

6.1 Attackers only need similar enough data

Results found in first part of the experiment point towards a general notion that *an attacker simply needs similar enough data to perform adversarial attacks using a substitute model under a black-box scenario*. This conclusion both aligns with the current notions of adversarial research, and expands the body of work by providing a new attack vector using semantically similar data. If an attacker knows the underlying problem that a CNN in the wild is trying to solve, he can attempt to find a dataset that is similar enough to the one used in the target model and use it to train a substitute model. The attacker does not need the exact same data used to train the target model to create effective adversarial samples that majorly degrade the target model's accuracy.

Furthermore, the more similar the data is to the one used to train the target model, the more effective the attack is. From the results, the symmetric attacks are the most effective rending some models completely unusable when adversaries are fed into the model. Regardless, the cross-section and disjoint attack also decreased the accuracy of target model below the boundaries of what is considered a high-accuracy classifier by modern standards (above 95% accuracy).

In terms of transferability, the results show that the *closer the model's architecture is to the target model's architecture, the better the adversarial samples will transfer*. A noticeable observation within these results is that more complex models seem more prone to be affected by adversarial images. To the best of our knowledge, we are the first to observe that the more complex the target model is, the more complex the substitute model should be for an effective attack. In general more complex models fared worse in all three data distributions, and samples transferred between the Resnet-50 and Densenet-121 samples had a higher chance of also being adversarial over samples derived from the GoogLeNet. This results could suggest that that models with more hyperparameters tend to over-fit more thus making them more susceptible to adversarial samples, or that the Resnet-50 and the Densenet-121 architectures are very similar hence increasing chance of transferability. Given these observations, an attacker wishing to compromise a target model could consider choosing a diverse set of substitute models based on their sizes and architectures to maximize chance of success.

This threat vector can be appealing for numerous reasons. First, if an attacker is able to find a dataset that closely follows the target dataset, it's feasible — and in some scenarios trivial — to train a high-accuracy model using transfer-learning technique discussed in 4.3. Second, is that an attack via substitute model uses conventional techniques to train models to generate adversaries without delving into more complex methods that can be found in adversarial literature.

6.2 Non-Robustness in Data

Results found in our non-robustness analysis are *not conclusive on the notion of models picking-up similar non-robust features*. Our experimental results show multiple fluctuations in the accuracy between model pairs and between different architectures.

This experiment still yielded fascinating observations in adversarial generation. First of all, if an attacker wants to increase his chances of generating an adversarial sample, using an ensemble of models could increase the likelihood of success. Our result suggest that using an ensemble produces samples that were overall adversarial regardless of the target model. This method of attack can be appealing in a black-box scenario, since an attacker with little knowledge of the underlying model could simply select a set of popular architectures and generate a highly non-robust dataset from an ensemble.

The second relevant observation was that our non-robust dataset D_{NR} turned out to be mostly composed of cat samples (that under adversarial perturbation are classified as dogs). As explained in our results section, a potential reason behind this frequency of cat samples could be due to the fact that cats are underrepresented in the dataset compared to dogs. This could mean that if an attacker has any insights in the data distribution used to train a target, he could increase his chances of success by generating adversaries from underrepresented groups in the dataset.

6.3 Future Work

The main purpose of this research was to expand our notion of adversarial transferability. The current paper addresses a lot of core concepts in regards to similarity of data, but more could be expanded on this idea.

This work uses an intuitive version of what semantic similarity between datasets entail, yet it does not present a concrete framework in which one can objectively analyze semantic similarity or measure it. Developing this similar experiment but with a measurable way to evaluate similarity between datasets could yield more precise results and further substantiate the results achieved in this paper.

An important shortcoming of this experiment was the second part on non-robust features since it could not conclusively explain why transferability occurs using non-robust data. Even though the notion of non-robust features is rather compelling, using these concepts did not yield very conclusive evidence. Further research should look into other methods to explain why transferability of adversarial samples occur based on how models form their decision boundaries.

Of course, recreation is an important aspect in research, and we believe that recreating this experiment while altering some parameters could be insightful. For example, further research could look into studying the transferability of samples while manipulating the α values of the gradient based attacks used in this experiment, or even explore new types of attacks. Expanding on the currently selected subset of models would be useful in attempting to provide more evidence on how the size of networks affect transferability.

7 RESPONSIBLE RESEARCH

7.1 Integrity of Research

When performing the experimental phase of the research, we designed the test-bed to emphasize reproducible results.

Following the current standards for reproducible research in computational science[24], we decided to use a well-known and easily available dataset, make the code base public and give a brief description of the environment used to run the code. This way its guaranteed that someone can download the code and run the experiments again.

All results presented here are also not unique instances, rather they were the average of multiple runs of the whole experiment. This meant training the specified CNN models, and running each model through numerous iterations of randomly selected adversarial images and measuring the model's accuracy.

7.2 Ethics of Adversarial Research

Adversarial samples have been an alarming discovery since they first started to appear in the early 2010's. Invariably, any research that expands on the possibilities of attacks should raise ethical concerns. In this paper all adversarial work has been performed for expository purposes that intend to raise awareness and concern for a new potential way attacking Deep Neural Networks. The more is known about what is possible with adversarial research, the more we will get to understand why it happens and how it can be addressed. It is still very concerning topic, and according to Gupta and Dasgubta [7], adversarial attacks are not yet considered a threat that is being allocated significant cyber-security resource. Both researchers still argue that as adversarial research expands further, it will soon become a threat that will cause AI-based systems vulnerable. Furthermore, as ML systems become more common in our environment, these models might need to go under scrutinous certification processes to ensure some guarantees — such as using state-of-the-art adversarial training methods to make ML systems more resistant [10].

References

[1] Mariusz Bojarski, Davide Del Testa, Daniel Dworakowski, Bernhard Firner, Beat Flepp, Prasoon Goyal, Lawrence D Jackel, Mathew Monfort, Urs Muller, Jiakai Zhang, et al. End to end learning for self-driving cars. *arXiv preprint arXiv:1604.07316*, 2016.

[2] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[3] Logan Engstrom, Andrew Ilyas, Aleksander Madry, Shibani Santurkar, Brandon Tran, and Dimitris Tsipras. Adversarial examples are not bugs, they are features, May 2019.

[4] Ian Goodfellow. Defense against the dark arts: An overview of adversarial example security research and future research directions. *arXiv preprint arXiv:1806.04169*, 2018.

[5] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

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

[7] Kishor Datta Gupta and Dipankar Dasgupta. Who is responsible for adversarial defense? *arXiv preprint arXiv:2106.14152*, 2021.

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[9] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017.

[10] Xiaowei Huang, Daniel Kroening, Wenjie Ruan, James Sharp, Youcheng Sun, Emese Thamo, Min Wu, and Xinping Yi. A survey of safety and trustworthiness of deep neural networks: Verification, testing, adversarial attack and defence, and interpretability. *Computer Science Review*, 37:100270, 2020.

[11] Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry. Adversarial examples are not bugs, they are features. *Advances in neural information processing systems*, 32, 2019.

[12] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[13] Alexey Kurakin, Ian Goodfellow, Samy Bengio, et al. Adversarial examples in the physical world. In *The International Conference on Learning Representations (ICLR) Workshops*, 2017.

[14] Yann LeCun. The mnist database of handwritten digits. *http://yann. lecun. com/exdb/mnist/*, 1998.

[15] Yanpei Liu, Xinyun Chen, Chang Liu, and Dawn Song. Delving into transferable adversarial examples and black-box attacks. *arXiv preprint arXiv:1611.02770*, 2016.

[16] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.

[17] Michael A Nielsen. *Neural networks and deep learning*, volume 25. Determination press San Francisco, CA, USA, 2015.

[18] Nicolas Papernot, Fartash Faghri, Nicholas Carlini, Ian Goodfellow, Reuben Feinman, Alexey Kurakin, Cihang Xie, Yash Sharma, Tom Brown, Aurko Roy, Alexander Matyasko, Vahid Behzadan, Karen Hambardzumyan, Zhishuai Zhang, Yi-Lin Juang, Zhi Li, Ryan Sheatsley, Abhibhav Garg, Jonathan Uesato, Willi Gierke, Yinpeng Dong, David Berthelot, Paul Hendricks, Jonas

Rauber, and Rujun Long. Technical report on the cleverhans v2.1.0 adversarial examples library. *arXiv preprint arXiv:1610.00768*, 2018.

[19] Nicolas Papernot, Patrick McDaniel, and Ian Goodfellow. Transferability in machine learning: from phenomena to black-box attacks using adversarial samples. *arXiv preprint arXiv:1605.07277*, 2016.

[20] Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z Berkay Celik, and Ananthram Swami. Practical black-box attacks against machine learning. In *Proceedings of the 2017 ACM on Asia conference on computer and communications security*, pages 506–519, 2017.

[21] Omkar M. Parkhi, Andrea Vedaldi, Andrew Zisserman, and C. V. Jawahar. Cats and dogs. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2012.

[22] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

[23] Yatheendra Pravan. Project idea: Cat vs dog image classifier using cnn implemented using keras, Jul 2021.

[24] Victoria C Stodden. Reproducible research: Addressing the need for data and code sharing in computational science. 2010.

[25] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

[26] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *Proceedings of the 2014 International Conference on Learning Representations*, 2014.

[27] Dimitris Tsipras, Shibani Santurkar, Logan Engstrom, Alexander Turner, and Aleksander Madry. Robustness may be at odds with accuracy. *arXiv preprint arXiv:1805.12152*, 2018.

[28] Lei Wu, Zhanxing Zhu, Cheng Tai, et al. Understanding and enhancing the transferability of adversarial examples. *arXiv preprint arXiv:1802.09707*, 2018.

[29] Cihang Xie, Zhishuai Zhang, Yuyin Zhou, Song Bai, Jianyu Wang, Zhou Ren, and Alan L Yuille. Improving transferability of adversarial examples with input diversity. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2730–2739, 2019.

[30] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? *Advances in neural information processing systems*, 27, 2014.

A Non-Robust dataset cat-like dog

As explained in our second experiment in section 5.3, when creating the non-robust dataset one interesting observation is that there were very few dogs in the set D_{NR}. Among the very few dogs that made it into this dataset, most exhibited some cat-like features.

Below we added two of these images. Notice how these dogs have very pronounced, pointy ears, and round faces. These features can be associated with cat features.

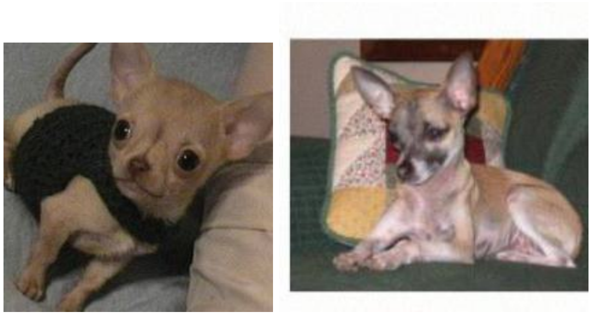

Figure 9: Two PGD-perturbed dog samples from the generated D_{NR}. Both of these images are classified as cats by the whole ensemble. Interestingly, these dogs seemingly have cat-like features such as being small, pronounced ears and round faces.