8-1-2022

Exploratory Analysis of Effects of Data on Adversarial Perturbations

Hamza Ali Zafar

EXPLORATORY ANALYSIS OF EFFECTS OF DATA ON ADVERSARIAL PERTURBATIONS

A Thesis

presented in partial fulfillment of requirements

for the degree of Master of Science

in the Department of Computer and Information Science

The University of Mississippi

by

HAMZA ALI ZAFAR

August 2022

ABSTRACT

In recent years, the amount of data being produced has increased tremendously. This data has allowed us to create and train machine learning models that are being used nowadays. Though these models have proven to be very effective in classification and regression problems, they do have a vulnerability. This vulnerability is exploited by adding adversarial perturbations to the original data or in the deployed model. These adversarial attacks are done by making minute changes to the data to confuse the machine learning model. This can lead to models' misrepresentation and drop the accuracy. For the span of this paper, I have used various statistical and visualization analysis methods to find the effect of adversarial perturbations on the CIFAR10 image data. Some of these methods involved were mean, standard deviation, variance, covariance, the probability distribution of color channels, etc. This paper discusses the insights found during the analysis of the CIFAR10 image dataset and the future work to be expected in this field.

DEDICATION

This work is dedicated to everyone who helped me and guided me through my own time of stress and anxiety. Their help, support, and encouragement have been instrumental in completing this thesis, without whose tireless encouragement I would have given up long ago.

ACKNOWLEDGMENTS

I would like to thank my advisor Charles Fleming and my committee members for their help in editing the final document. I would also like to thank the Department of Computer and Information Science for their support and the Institute of Child Nutrition department without whom I could not finance my studies.

Lastly, I acknowledge the collegial support from my fellow students. You made this part of my life enjoyable and enriching.

TABLE OF CONTENTS

LIST OF FIGURES

FIGURE PAGE

CHAPTER I

INTRODUCTION

1. DATA

Data is defined as “facts and statistics collected together for reference or analysis”. In this age of the internet, the amount of data being produced has increased enormously [1]. According to the statistics mentioned over multiple sites online [2] [3] [4], In 2020, 1.7MB of data was being created every second, and around 2.5 quintillion bytes of data were being created every day. Similarly, more than 300 billion emails were sent and 500 million tweets were made every day. Big companies like Google, Facebook (Meta) and Amazon have around 1200 petabytes of data stored in their warehouses across the world [4]. To put in perspective, the data in digital form was around 40 times more than the stars in the observable universe as of 2020 and it has increased then [4]. This data which is now categorized as big data due to its extravagant size cannot be used in the same way classical data was used for analysis.

This big data is now used with modern techniques of data analysis which combines aspects of statistics, mathematics, programming, visualization, and distributed data computing to analyze enormous amounts of data and extract information. This in turn allows neural networks and other machine learning algorithms to train models from this data that can help us in the description, prediction, and prescription/post-scription of information from data like weather forecasting to ensure safety in case of flood and tornadoes or diagnose diseases using medical image data and many more.

These Modern tools and technologies have made the analysis of data more efficient for the humungous data that is available right now. Engineers and scientists use these techniques to manipulate the data to extract information which helps in building data tools applications and services. The process of acquiring information from data starts with data acquisition which involves gathering data from multiple sources.

These days the data comes from everywhere like IoT devices, servers, sensors, social media platforms, text data from blogs and websites, location data from GPS, images, videos, or audios being uploaded from mobile phones, satellites, or data from cars and so on. The big data streams coming from these sources are characterized by the following:

- Volume: Immense amount of data
- Velocity: Continuous amount of data streams
- Variety: Different types of data coming in like structured, semi-structured, or unstructured.

We have talked about the volume and the velocity of the data in the previous paragraphs. A variety of data comes in the form of structured, semi-structured, or unstructured formats. The structured format can be stored in tables like in RDBMS data. Semi-structured data can be stored in files like XML, JSON, or NoSQL databases. Text files, images, or videos are considered the unstructured data format.

Most of the data collected from these various sources are in raw format. This raw format is preprocessed and converted to useful information using data wrangling techniques to be used in multiple areas like machine learning model training, predicting values, forecasting, budgeting or analysis of data, etc. At the moment, this big data is used in all aspects of our life from

businesses, and social media networks to healthcare, education, government level, national security level, etc.

2. MACHINE LEARNING MODELS

This preprocessed data is the backbone of modern machine learning advancements and this increase in data collection has allowed machine learning models to evolve as well [5]. It started as part of the artificial intelligence subdomain but since the GPUs usage for parallelism to multiply matrices and calculating differentials used in the neural network was discovered [6] [7], they have taken over the modern artificial intelligence research and development. It is considered one of the main pillars of modern cutting-edge technologies.

Currently, neural networks are being used in every major industry for numerous tasks from automation, and classification to predicting and forecasting future events. Most of the apps and services we use nowadays have some type of machine learning involved in the back of the program. Common everyday examples include social media, manufacturing automation, and some of the major online platform's ways of recommending new stuff for you like Netflix, Spotify, YouTube, and Amazon use these recommendation systems. Broadly speaking, these machine learning models use the preprocessed sample data of user history as input and cluster it with similar data and users with similar interests to output predicted values for recommendations. Some other places these machine learning models are being used are image classification, regression modeling, clustering, anomaly detection, business intelligence, etc.

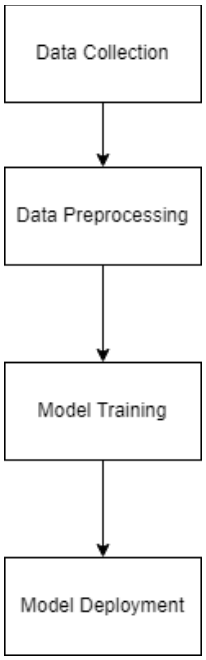

Figure 1: Machine Learning Model Workflow

These neural networks we see nowadays came from an idea of the human brain communication mechanism model which was first published in 1949 by Donald Hebb. His working of brain neurons is said to work on neuron excitement and firing to communicate with each other. This was combined with Samuel's machine learning efforts in 1957 by Frank Rosenblatt, who created perceptron, which led to multi-layer perceptron, and finally to feedforward neural networks and backpropagation, which is what the most modern neural networks are based on.

These models don't just take in data and output correct values, the parameters of the model need to be trained first. Backpropagation allows neurons to be trained using the error correction method which calculates the difference between predicted and the actual values and backpropagates the errors to adjust the parameters of the model. Some basic approaches used for

training the machine learning models are supervised learning, unsupervised learning, semi-supervised learning, and reinforcement learning [8]. In the case of supervised learning, the user provides the algorithms with labeled training data and calculates the loss between the actual value and output by the algorithm. Whereas unsupervised learning as the name suggests does not take in any labeled data, it learns from any meaningful connection or clusters on its own. The third approach is a mix of the first two where the user uses some labeled data but the model is allowed to find a correlation between data and output by itself. The last approach uses predefined rules which according to the output of the program provide positive and negative feedback to train the model.

This evolution of artificial intelligence has led us to take advantage of these models to understand some of the most complex relations at a deeper level. This was done by collecting data and using it in a way to answer the business, social and economic needs of the society and the people's unconscious and conscious necessities. Similarly, they are the main cause for advances in the field of health, Finance, and astronomy by providing humans a way to move into the future.

Although these machine learning models have their advantages, they also led us to learn some of the drawbacks of machine learning models. First and foremost, the training of these models can be pretty expensive. The larger the dataset and the complex the problem, the longer it takes to train the model to accurately predict the results. Another problem that can arise is if the collected data is not properly catered, it can have a bias as well. The bias can come due to negligence of minorities or the geographical place from where the data is being collected from. As the trained model depicts the data it was trained on, the need for a generalized dataset is very important for the model to work accurately in the real world. The most pressing disadvantage

and the topic this paper focuses on is the drawback that can cause the models to misclassify and lead to a calamity nowadays are adversarial vulnerabilities [9] [10] [11] [12].

3. ADVERSARIAL MACHINE LEARNING

Now that we know machine learning models are leading the advances in every field and are at the frontlines of the innovations we see and hear every day. Adversarial machine learning mixes machine learning and computer security where bypassing machine learning models is the task of the adversary. The human adversaries and parties with different agendas want to use the machine learning models and their vulnerabilities to try to get ahead of the other parties. These adversaries want to nullify the defenses used by their competition to take advantage of others' weaknesses to extract information or gain access. This in turn could result in consequences that can be a loss in terms of money, time, or even human life in some cases. A report from Gartner [13] predicted that around 30% of all cyberattacks will be caused by data poisoning or other adversarial attacks by 2022. Which makes these adversaries and their adversarial attacks on the machine learning models a huge problem.

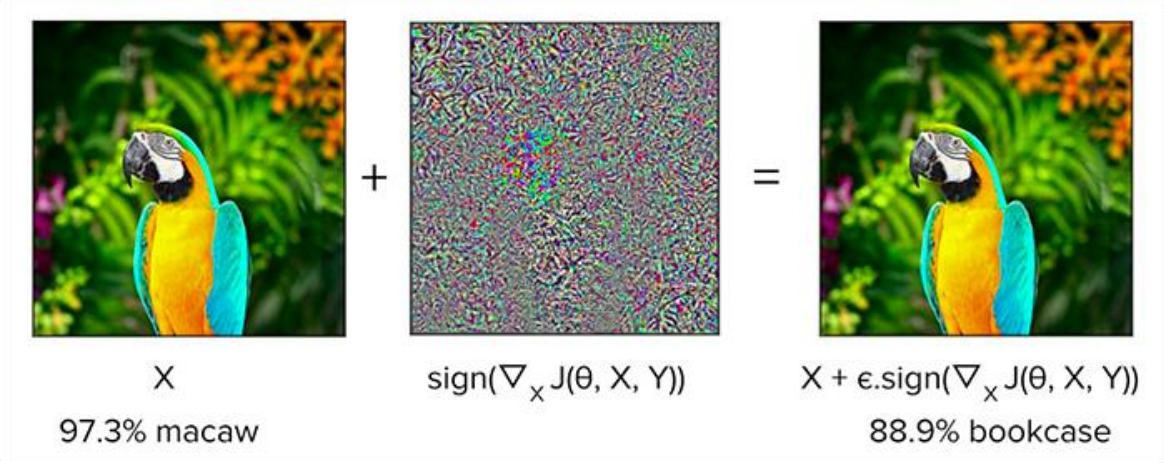

Figure 2: Image is taken from https://www.youtube.com/watch?v=oZYgaD004Dw

To put it in perspective, the GoogleNet model which performed and correctly classified MNIST test data with an accuracy of around 98% dropped to 18% using the Projected Gradient Descent (PGD) adversarial attack and it dropped to 1% using the DeepFool adversarial attack [14]. The figure 2 has a macaw on the left and on the right which looks similar to human eyes. But there is a specific difference which is shown in the middle image below. This noise in the middle image is specifically calculated using signed gradients to perturb the image below so the machine learning model can misclassify the macaw into a bookcase. Imagine this in a real-world scenario, like a paper discussed misclassifying a turtle into a rifle [15] and a stop sign into a speed limit sign [16] or Deepfake being used as a tool by adversary parties to misinform the mass population [17]. This could lead to massive threats not being detected in real life and can lead to a loss of money, time, and life if not handled carefully.

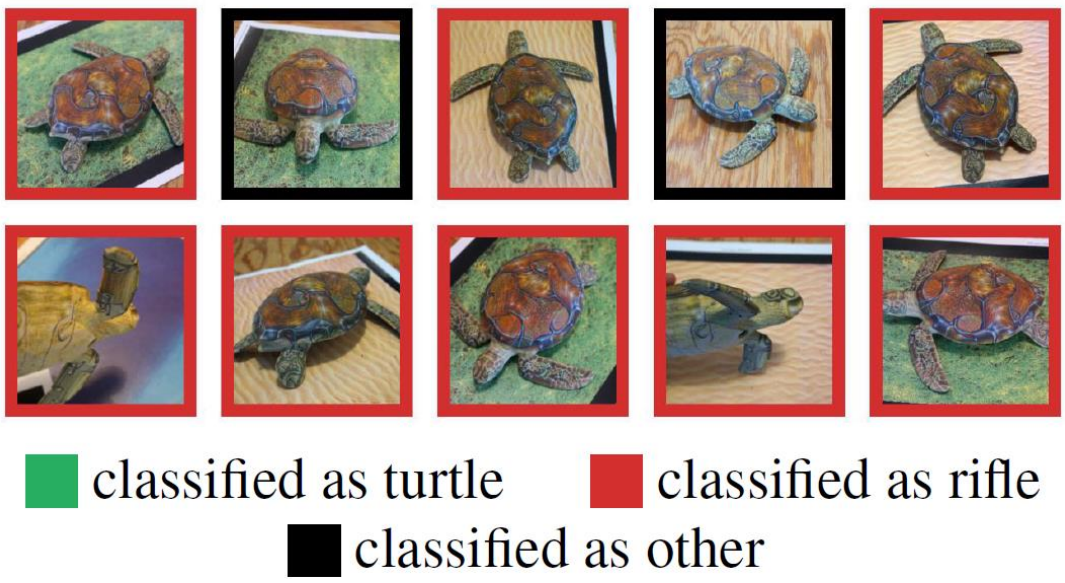

Figure 3: All above images with red borders were classified as a rifle

Figure 4: Stop sign classified as speed limit sign

These adversarial attacks make use of small patches to cause trained models to misidentify or poison data before or during the training phase to misclassify predicted output. Data poisoning is done on the training set to corrupt the model itself. This is done by inserting, modifying inputs in the training set, or changing the labels of inputs before training. The process of poisoning the training set takes a long time as the dataset nowadays are enormous. But the machine learning systems created afterward overlook the things the adversary wants the model to overlook. Evasion attacks are another type of adversarial attack which are done after the training phase and on the trained deployed models. They tend to find loopholes in the system's existing trained parameters to evade the security firewalls. These are the most common type of attacks. Evasion attacks are further divided into two subcategories.

- White-box attack
- Black-box attacks

White-box attacks have full knowledge of the gradients, parameters, and architecture of the machine learning model. They use this knowledge to attack the model. Black-box attacks do not have access to machine learning model parameters, gradients, and architecture. Some assume some knowledge of the machine learning algorithm used by the model but not the whole model. Attackers use examples to get an insight into the model. Even these subcategories can be of different types. These attacks can be either

- Non-targeted attacks
- Targeted attacks.

In a non-targeted attack, the goal is to lead the predicted output to any other output other than the actual value. In targeted attacks, the attacker tries to misclassify the output into a specific class label that is not the actual label.

These adversaries use data manipulation techniques described above which can remain undetected to human perception but fool the machine learning model. Although researchers have worked on improving the robustness of models and the data biasness but adversarial vulnerabilities continue to be a problem and no concrete solutions have been found for these vulnerabilities [18] [19]. Most robust models use adversarial examples training to make them robust to the attack but they are not considered the best solution. It can be seen as a two-player game, where when one algorithm is robust against an attack another attack comes along which can affect this algorithm, and so on.

This paper uses the backbone of these machine learning models which in this case is the data to explore the correlation between the data and the underlying adversarial vulnerabilities using the data manipulation techniques available to us in addition to new techniques discovered by the recent research [20]. There isn't much work done on the effects of data on these adversarial perturbations and that's what this paper is exploring in the next sections.

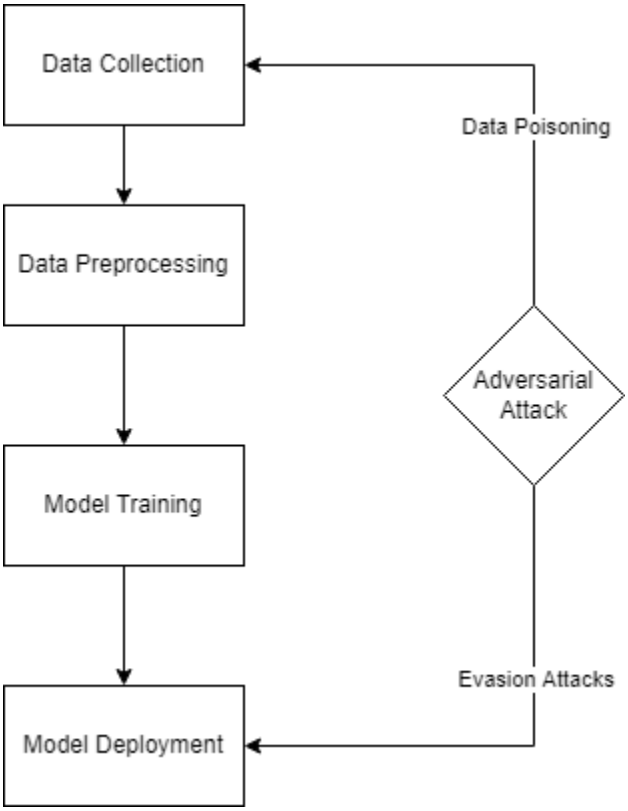

Figure 5: Adversarial Attacks

CHAPTER II

RELATED WORK

In this section, we go over some of the recent work already published which is related to our work and how our work is different from these already published papers. The use of GPUs for neural networks and the human-level accuracy of the machine learning models have acquired the attention of researchers in recent years [6] [7]. But now another phenomenon is getting attention as machine learning models are being deployed in real-world applications, that is the idea of adversarial vulnerabilities [12]. Previous work done by multiple research groups over the years has proposed multiple explanations. Some of these say we do not have a generalized dataset [21]. Some of these explanations said that the models have vulnerabilities due to their high dimensionality [22], another discussed that the adversarial vulnerabilities are not bugs but part of the features that the machine learning models learn [23]. In summary, previous work is only looking at the machine learning models, the generalization of the models, data features, and adversarial attacks separately. My work is focused on finding the correlation for adversarial vulnerabilities based on the data used for training and the analysis of how these adversarial vulnerabilities are affected by the data manipulation.

Ever since the advent of machine learning models, these models have carried a nature of ambiguity within themselves. What these models learn is still somewhat debatable. This black-box nature of these models offers researchers a wide area to work on and the reason we do not fully understand why adversarial vulnerabilities exist. There are different interpretations of how

the adversarial vulnerabilities occur and what are the causes behind it. As we discussed in the previous paragraph that some initial researchers suggested that adversaries exist because models are not generalized, while some suggested that it's because of the inherent problem of over-fitting [11].

There has been work done on creating multiple adversarial attacks in different settings such as white-box attacks or black-box attacks using different methods like the Fast gradient sign method (FGSM) [11], Projected gradient descent (PGD) [24] and Deep-fool, etc. In addition to creating these attacks, the researchers have also been providing some ideas on creating models that focus on defending against these attacks. Some defending options are training the model with adversarial examples with actual labels. This lets the model understand adversarial vulnerabilities a little better. Other options include rotations, translations, random resizing, and padding of the training set images to augment different scenarios that can occur in real-world applications [25]. There have been no concrete results on why these adversarial vulnerabilities occurrences and how these are related to the data the model is trained on during the training phase.

As discussed by the paper in [26], an attacker can make adversarial examples for black-box models without any information about the neural network being used behind that model just by querying the model. The attacker can use the transferability of adversarial examples which allows the adversary to use one adversarial example generated from one model to exploit different models with different architectures and training sets. This transferability is discussed by [27] in an empirical sense where the authors discuss adversarial subspaces which are orthogonal vectors whose linear combinations are adversarial perturbations. This paper discusses ways of finding these subspaces and how to defend against the adversarial examples.

A related work from the paper [28] that looks into the sensitivity of data distribution using synthetic transformations on the dataset showed that adversarially trained models achieve significantly different robust accuracy while standardly trained models' accuracies remained comparable to the actual on this new data. Although this paper looks at the data distribution effects on the trained neural network models, they do not compare the original data set on these models.

A similar paper [29] looks into what the machine learning models are learning from the data and how to improve robustness. They introduced a new version of ImageNet to look into the effects of texture information of the image on these convolutional neural networks. They were able to quantify the relation of shapes and textures on these convolutional neural networks showing that relying on shapes can greatly improve the robustness of these models. Although they did not look into the robustness of these shape-based models against the adversarial vulnerabilities.

A recent paper from MIT [23] suggests that a machine learning model not only learns the human perceptible features but some non-robust features as well which are not perceptible to humans. They set up a simple experiment to investigate this using the CIFAR-10 dataset. A few examples of the CIFAR-10 dataset are shown in Figure 8. They start by targeting each training set class with a class next to the actual class i.e., 0 class moved to 1, 9 class moved to 0, etc. through targeted adversarial perturbations. They then construct a new training set by labeling these adversarial perturbed images with their corresponding target class. This leads to a training set where each label looks incorrect to humans. Every dog is labeled as a cat and every cat as a bird and so on. Finally, they train a new machine learning classifier with this mislabeled data set.

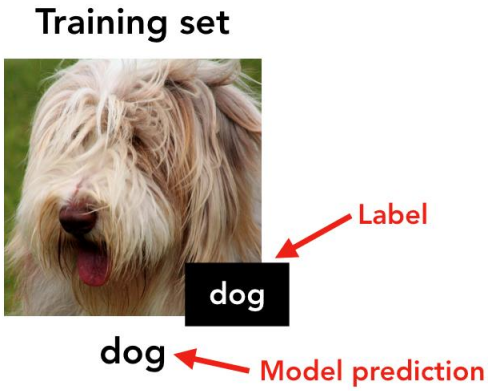

Figure 6: Actual training set image with the actual label,

image is taken from: http://gradientscience.org/adv/

Figure 7: New training set with mislabeled image,

image is taken from: http://gradientscience.org/adv/

This new classifier when evaluated with the actual test set against this mislabeled classifier, it returned 44% accuracy even though the classifier used was mislabeled. This is what they used to say that humans perceive the robust part of what the machine learning models learn and the other part these models learn are the non-robust features. Robust features remain the same even after the image is perturbed using the adversarial vulnerabilities. While non-robust

features change to the target image after being exposed to an adversarial attack. The fact that the original model generalized and got 44% accuracy on the actual test set is what the authors of the paper used to argue that non-robust features are there and can be used for generalization.

To further confirm that robust feature is also important and used by the classifier and why training the machine learning classifier with adversarial examples leads to a relatively robust model, they performed another experiment. Here they created another training set that restricted the features to contain only those used by the robust model. And trained the machine learning model on this. The resulting classifier although not trained on adversarial examples showed robustness to adversarial attacks. They suggested that robustness and non-robustness is due to the properties of the dataset itself. This MIT paper backs the hypothesis that there is a relation between data and the adversarial vulnerabilities which are not perceivable to humans and that is what my paper is focused on.

CHAPTER III

METHODOLOGY

This paper focuses on finding a relation between the data and the loopholes in machine learning caused by adversarial attacks. This section shares the methodology I used to perform the analysis for finding correlations between data and adversarial vulnerabilities. The approach for the said analysis started with selecting the dataset to be used for the experiments. A data set that is not too simple with enough classes and widely used in today's research. So, the data used in this paper for experimentation is taken from the CIFAR-10 data-set as it is used in almost all major machine learning research nowadays.

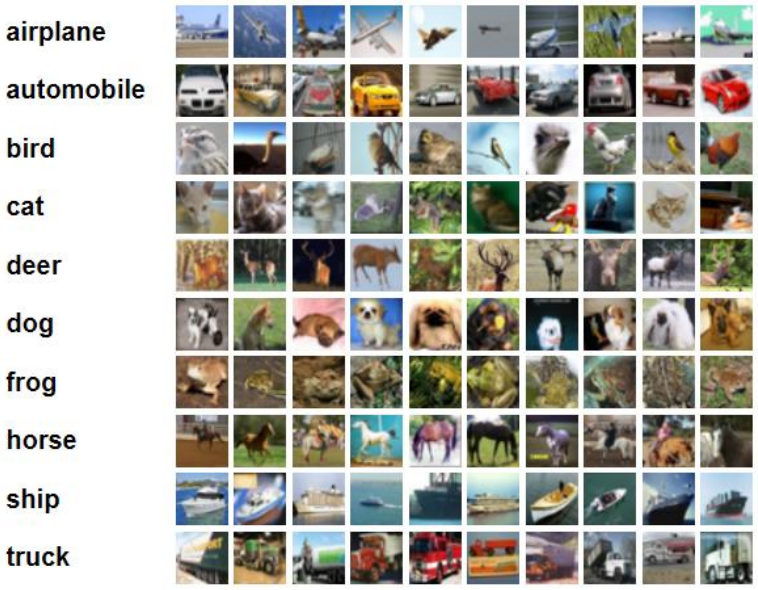

Figure 8: CIFAR-10 Data set

1. CIFAR-10 DATA-SET

The CIFAR-10 data-set contains 60000 32*32-colored images with 10 classes [30]. Each class has 6000 images. The data set is divided into 5 batches of 10000 images each for the training set and 10000 images for the test set. The test set contains randomly selected 1000 images per class from the 60000 images set. The 5 batches of the training set have randomly selected images where the number of classes in each batch is not kept equal for generality purposes. According to the data-set providers, the classes are completely mutually exclusive for example there is no overlap between automobiles and trucks. "Automobile" includes sedans, SUVs, and things of that sort. "Truck" includes only big trucks. Neither includes pickup trucks. The classes of the CIFAR-10 data-set are airplane, automobile, bird, cat, deer, dog, frog, horse, ship, and truck.

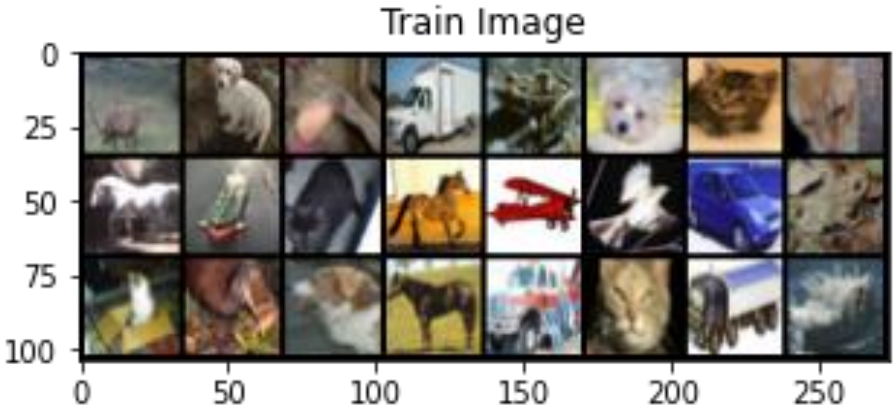

Figure 9: Batch input for training machine learning model

2. NEURAL NETWORK SELECTION

After selecting the data set the next step is to use a convolutional neural network that can achieve good accuracy but does not take too long to train to mimic a normal convolutional neural network. For the scope of this paper, this CIFAR-10 data-set is trained on a simple 9-layer convolutional neural network model to predict the class of an input image. This trained model takes in the 32*32 colored image as input, passes the input through a group of convolutional layers, group norm, and ELU activation in combination with dropout after every few layers and average pool at the last layer to output a vector of 10 values where each value is the probability for each class respectively.

For training the model on the CIFAR-10 data set, I used the cross-entropy loss function and trained the model using a 0.001 learning rate until the accuracy of the training set crossed 95%. Then evaluated the CIFAR-10 classifier model on the testing set of 10000 images to measure the accuracy. The convolutional neural network yielded an accuracy of 87.26% on the testing set.

3. ADVERSARIAL ATTACKS SELECTION

I used the most widely available adversarial attacks like Fast Gradient Sign Method FGSM [11] and the Projected Gradient Descent PGD [24] both of which are white-box non-targeted evasion attacks. These adversarial attacks were used to perturb the test set images of the CIFAR-10 dataset for the next step of analyzing the adversarial images and the actual images.

As mentioned in the paper [11], a fast adversarial example can be created for any image using the gradients and backpropagation. The following equation was introduced in the paper to generate the adversarial examples:

$$\boldsymbol{\eta} = \epsilon \text{sign}\left(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y)\right)$$

Here θ represents the parameters of the model, x represents the input to the model, y represents the target associated with x, and $\boldsymbol{J(\theta, x, y)}$ is the cost used to train the neural network. The steps involved in creating the perturbed image using the Fast gradient sign method were:

1. Taking an input image.
2. Making predictions on the image using a trained CNN.
3. Computing the loss of the prediction based on the true class label.
4. Calculating the gradients of the loss with respect to the input image.
5. Computing the sign of the gradient.
6. Using the signed gradient to construct the output adversarial image.

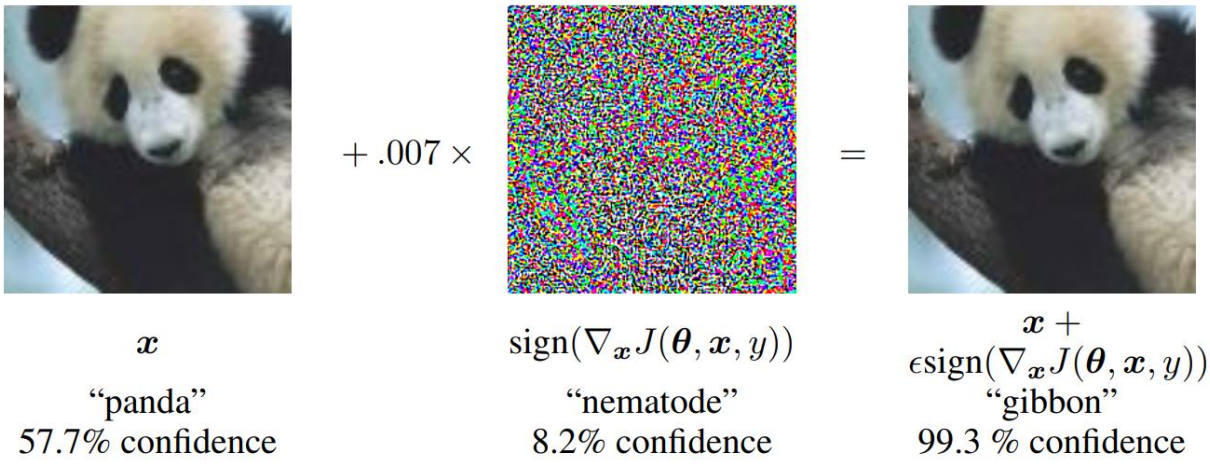

Figure 10: Creating a Fast adversarial example using the FGSM method defined above *[11]*

I implemented the above-mentioned method of FGSM using the ‘torchattacks’ library from python which provides all widely available attacks being used to generate adversarial images. Similarly, PGD which is an extension of the Basic Iterative Method (BIM) and FGSM where after each step the perturbation is projected back using a projection function is used to create the adversarial examples of the test set. The purpose of using two attacks was to compare how each affects the image visually or statistically and is there any correlation that shows up in one attack while not in the other. This could help in generalizing the exploratory analysis process.

4. ANALYSIS OF THE ORIGINAL AND ADVERSARIAL IMAGES

The next step in the process is to explore and analyze the adversarial images and their difference from the original images. For that purpose, I performed a set of analyses on these perturbed images and compared them with the original to find the changes in the data due to these adversarial perturbations. For the analyses of the data, this paper divided the analyses into two main approaches.

- **Statistical Analysis**

  Statistical analysis uses methods available in the field of statistics to calculate and analyze the original and adversarial images in this scenario. The process involves finding the population sample that is suited for the analysis which in this case is the test set of the CIFAR-10 dataset. We cannot use the training set the machine learning model was trained on this data. After that, the next step is to use different statistical methods to describe the spread and shape of the test set and use this description to make inferences about the data. The methods this paper used were:

- The measure of center tendency which comprises calculating mean, median, and mode.
- Variance or spread of the data per class
- Covariance between two different classes etc.

- **Visualization Analysis**

  Visualization analysis uses methods available to visualize data in graphs, charts, or other visual formats that can be useful in analysis and inferences. For the scope of this paper, it includes visualization of different plots like:

  - Heat-maps for correlation
  - Histograms of pixel and color values
  - The probability density function for each image pixel values

CHAPTER IV

RESULTS

1. NEURAL NETWORK EVALUATION

The experiment started by evaluating the machine learning classifier against training, testing, and adversarial testing sets. The training set accuracy of the model after training returned 96.01%. The model was then evaluated against the test set before applying the perturbations. The accuracy of the test set came out to be 87.26%. I then used this testing set to create adversarial perturbations and calculate accuracy for it. The accuracy of the testing set drops to 2.73% after perturbing test set images.

2. ANALYSIS USING HEATMAP

After the results from the evaluation, the next phase included creating adversarial images using FGSM, and PGD and applying statistical methods to these adversarial examples and the original images. I started with calculating mean values for the original and the adversarial images per class to compare if the mean moved from its original values for each class. To compare the means, I found the L2 distance for the means of the original test set images and the means of adversarial test set images. This showed no major change in the mean values from any set. The result can be seen in Figure *11* where the diagonal shows that there is no difference in the mean value of adversarial or original image means for each class, which shows that the difference between the perturbed and the original images are minimum. I used the heatmap to

find a correlation between the original and the adversarial images and check if the adversarial images mean for any class moved towards any other original images class mean. As it's evident from the Figure below that was not the case as no other value is black other than the diagonals.

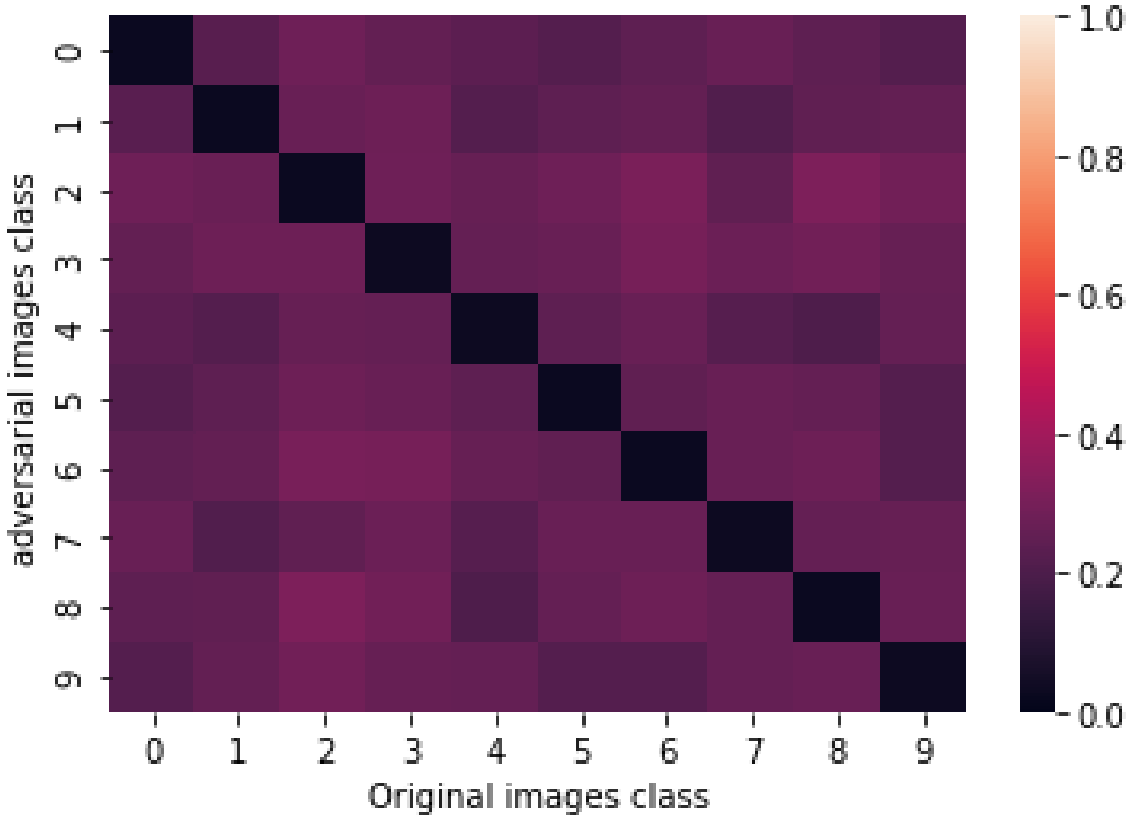

Figure 11: Difference in mean for original and adversarial images

The mean of each class image used in the above calculation was calculated using the torch mean methods for original and adversarial images and are shown in Figure *12*.

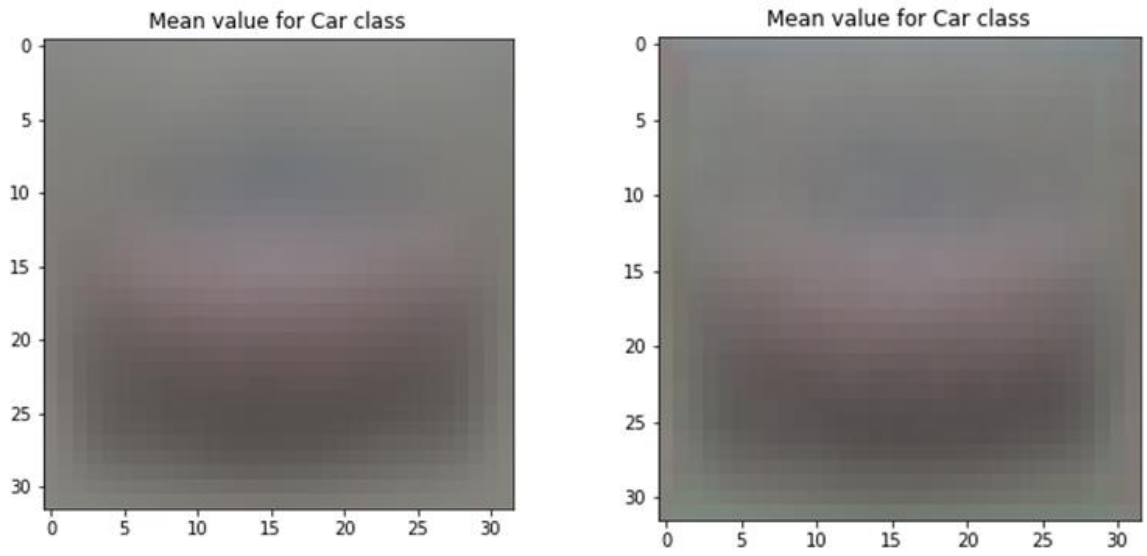

Figure 12: Mean value for original and adversarial car class

3. AVERAGE PERTURBATION FOR EACH CLASS

As the previous experiment showed that there was no change in the mean for original and adversarial values for each class, I moved on to perform another analysis. This analysis used average perturbation for each class to check whether a class requires less perturbation than the other to move to another class. The average perturbation calculated using FGSM ranged between 1.7141 for car class and 1.7331 for deer class. Table 1 shows the average perturbations for all classes calculated with FGSM. Similarly, the average perturbation calculated using PGD ranged between 1.4173 for bird class and 1.5111 for horse class. Table 2 shows the average perturbations for all classes calculated with PGD. The average perturbations were calculated by calculating the difference between the original and its adversarial counterpart for each class and taking the mean of these differences.

Table 1: Average Perturbation for each class using FGSM

Plane	Car	Bird	Cat	Deer	Dog	Frog	Horse	Ship	Truck
1.7171	1.7141	1.7274	1.7241	1.7331	1.7274	1.7256	1.7251	1.7271	1.7168

Table 2: Average Perturbation for each class using PGD

Plane	Car	Bird	Cat	Deer	Dog	Frog	Horse	Ship	Truck
1.4356	1.5041	1.4173	1.4273	1.4254	1.4630	1.4894	1.5111	1.4393	1.5026

Table 3: Predictions results on original and adversarial images and their difference using FGSM

Class	Original image predictions	Adversarial image prediction	Difference
Plane	934	678	256
Car	1061	1072	-11
Bird	1016	1411	-395
Cat	795	1207	-412
Deer	1066	1613	-547
Dog	1155	1019	136
Frog	1009	747	262
Horse	1010	738	272
Ship	1003	644	359
Truck	951	871	80

4. MOVEMENT OF THE ADVERSARIAL IMAGES IN DIFFERENT CLASSES

After looking at the average perturbation for each class. The next question that arose, as a result, was to check if these perturbations moving toward one class or not and where does the

adversary class move towards. So, I ran the original test set through the model and calculated how many for each class were correctly identified. This was done to get the number of accurately predicted values for each class. I then ran the same experiment with adversarial images as input to see how many of these adversarial images were identified in each class. The result from the above experiment was able to provide insight about adversarial images moving to a specific class plus the class where the most adversarial moved. As we can see in Table 3 the adversarial images moved towards the deer, cat, and bird class more as compared to others and moved away from the ship class when adversarial examples were created using FGSM.

Table 4: Predictions results on original and adversarial images and their difference using PGD

Class	Original image predictions	Adversarial image prediction	Difference
Plane	934	904	30
Car	1061	1018	43
Bird	1016	1327	-311
Cat	795	1183	-388
Deer	1066	1378	-312
Dog	1155	1227	-72
Frog	1009	626	383
Horse	1010	783	227
Ship	1003	605	398
Truck	951	949	2

Similarly, as seen in Table 4 when I used PGD adversarial attack the adversarial images still moved towards the deer, cat, and bird class more as compared to others but in addition to

moving away from the ship class, the adversarial images also moved away from frog class. To look at the reason for this, I looked at the confusion matrix for the original test set which is shown in Figure 13. Then calculated the count of each class that moved from predicted class to adversarial class using both FGSM and PGD methods. For example, as we can see in Figure 14, the class 0 which is the plane class had a total of 861 correctly predicted as we saw in Figure 13, when we apply adversarial perturbations using FGSM to these plane class images, 58 of them remain in the plane class 0, while 344 of these moved to bird class and 220 of these images moved to ship class. The heatmaps explaining the transfer of adversarial images from actual predicted to the results for adversarial are shown in Figure 14, and Figure 15.

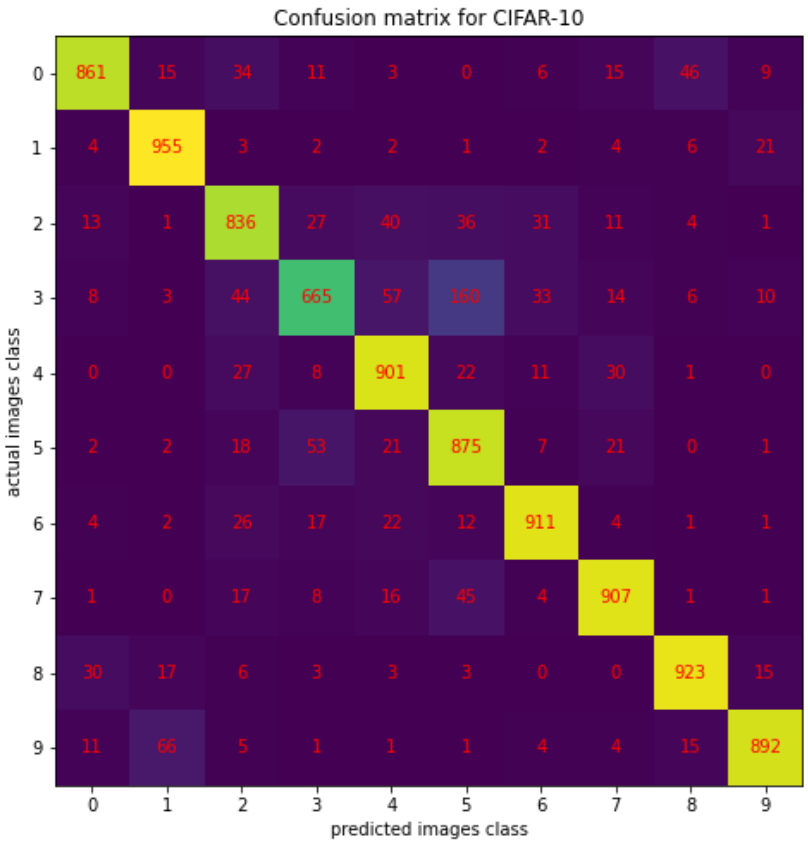

Figure 13: Confusion matrix for CIFAR-10 test set

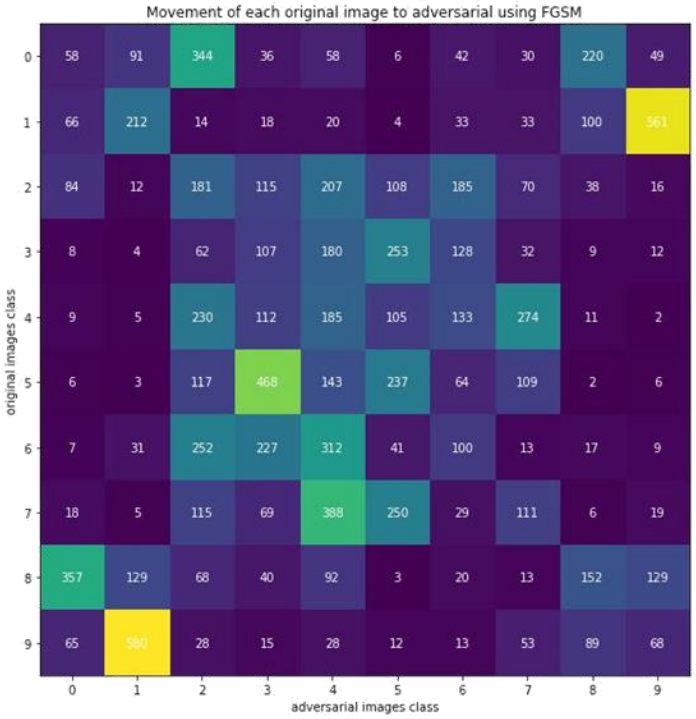

Figure 14: Movement of each original prediction to adversarial prediction using FGSM

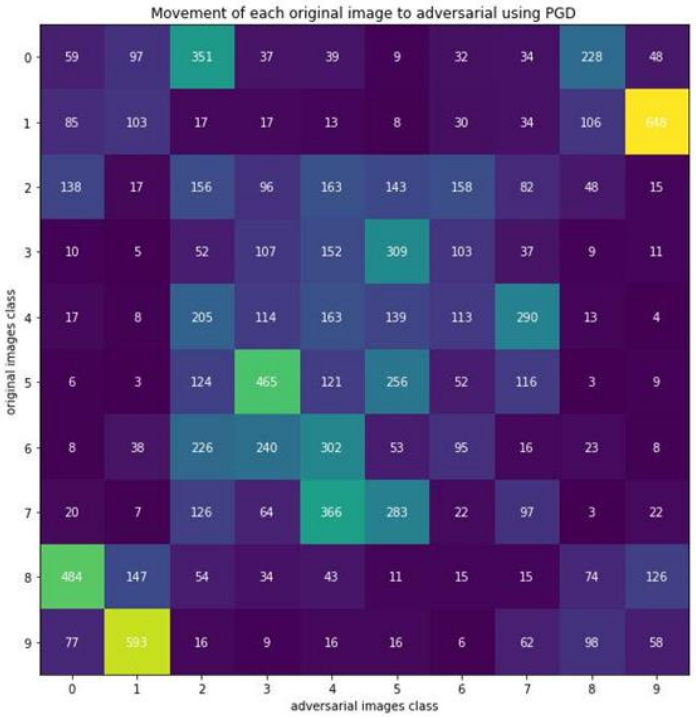

Figure 15: Movement of each original prediction to adversarial prediction using PGD

5. RELATION BETWEEN COLOR CHANNELS AND ADVERSARIAL IMAGES

Afterward, I looked at the RGB color channels as they may have some effects on the adversarial examples, I looked at the difference in the RGB color channels for original and adversarial images. The results as we can also see in the below figures showed that the adversarial images changed the pixel counts which were high in the original image to a more normalized state distributing it along with other color ranges. If a color pixel count had a spike in the original image, the adversarial attack distributes it to other values. These visual differences in color histograms can be seen in Figure 17, and Figure 18 which show the original and adversary counterpart of the same class image.

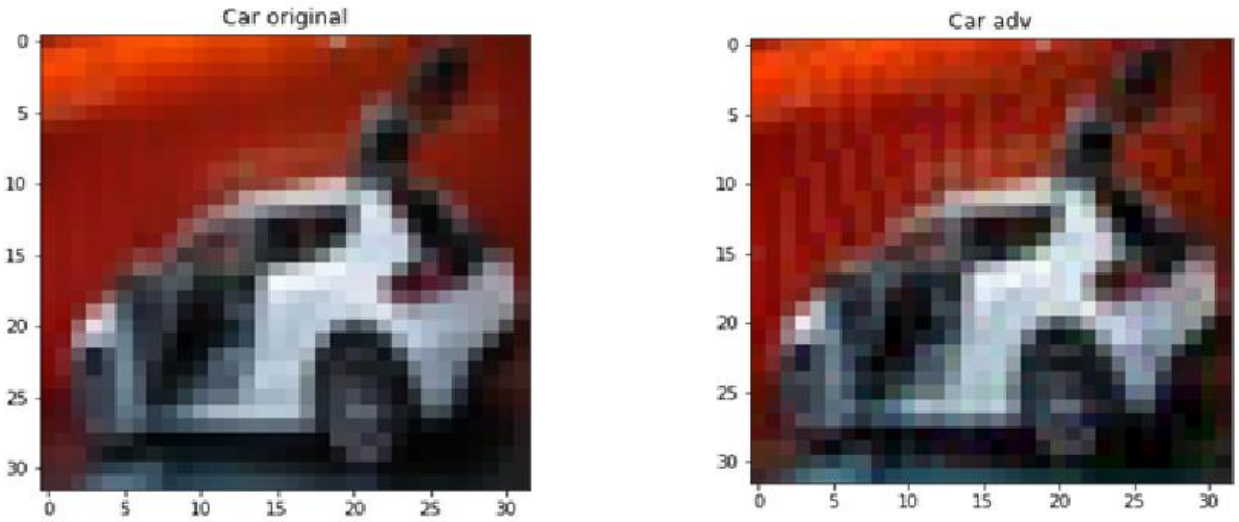

Figure 16: Original and the adversarial car image

The color-wise difference in Figure 19 shows the pixel counts which were changed in the adversarial image. This change in pixel count was further looked at in different graphs for each channel for the car image in Figure 20, Figure 21, and Figure 22. These figures show that the blue channel was changed the most in the car image.

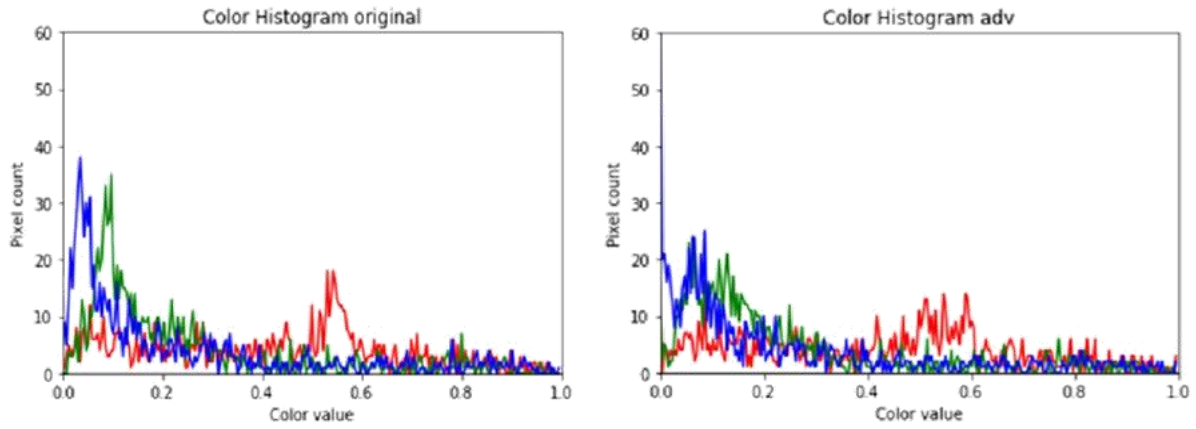

Figure 17: Color histogram for RGB channels of the original and adversarial car image

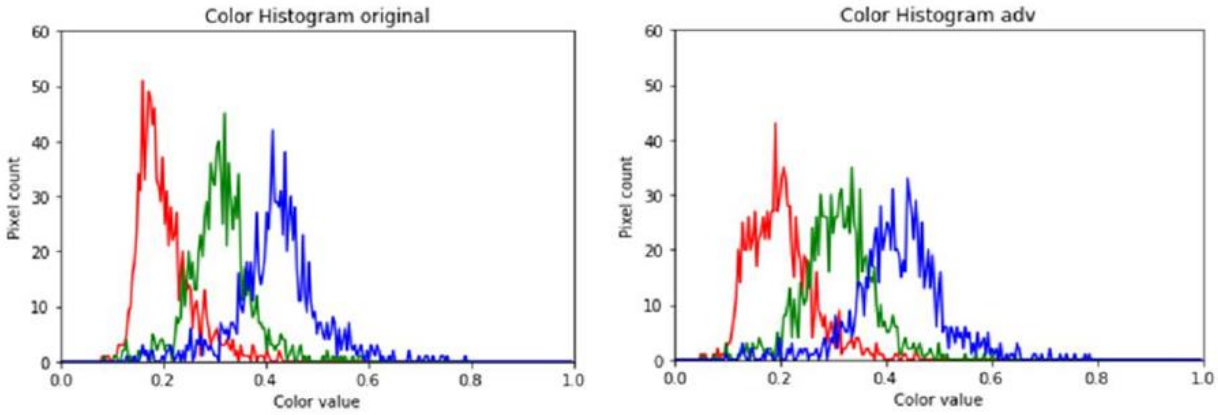

Figure 18: Color histogram for RGB channels of the original and adversarial deer image

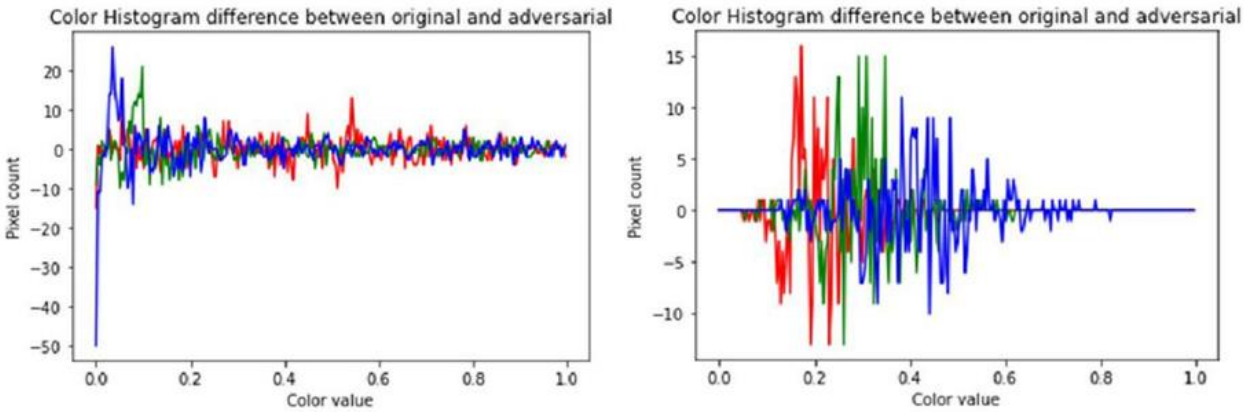

Figure 19: Color histogram difference b/w original and adversarial car image (left) and deer image (right)

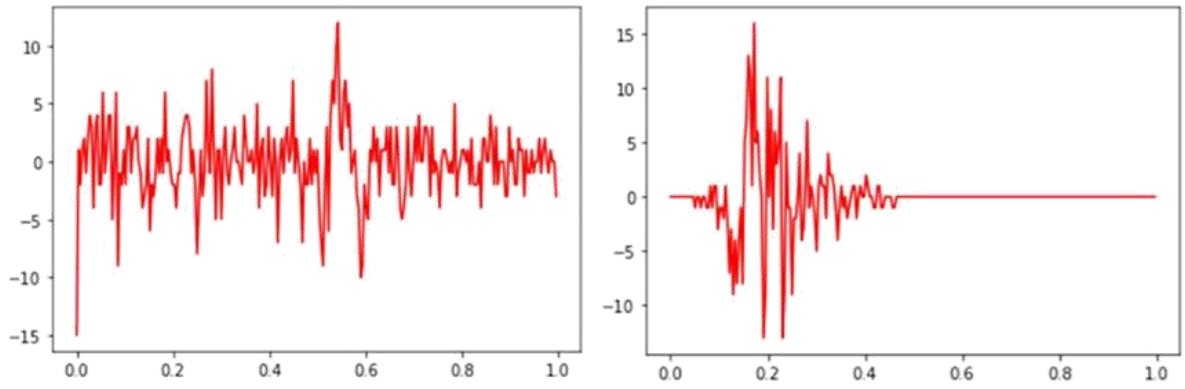

Figure 20: color difference b/w original and adversarial for the red channel car image (left) and deer image (right)

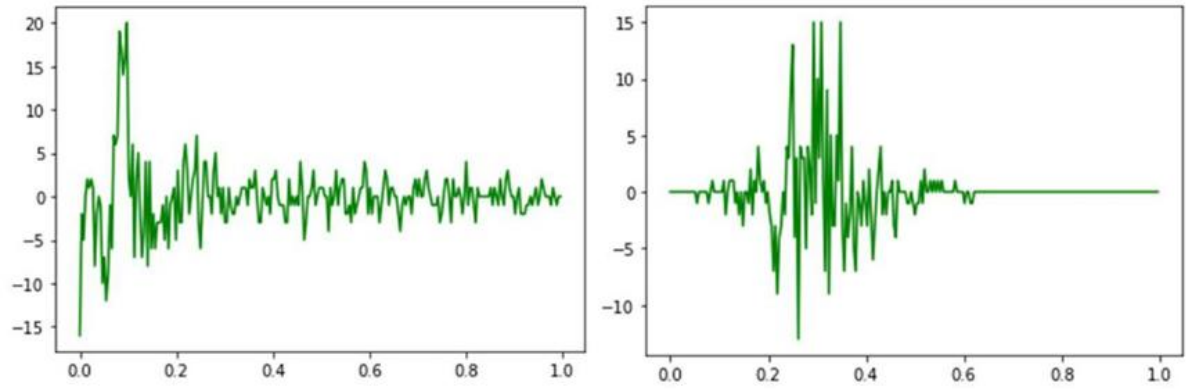

Figure 21: color difference b/w original and adversarial for the green channel car image (left) and deer image (right)

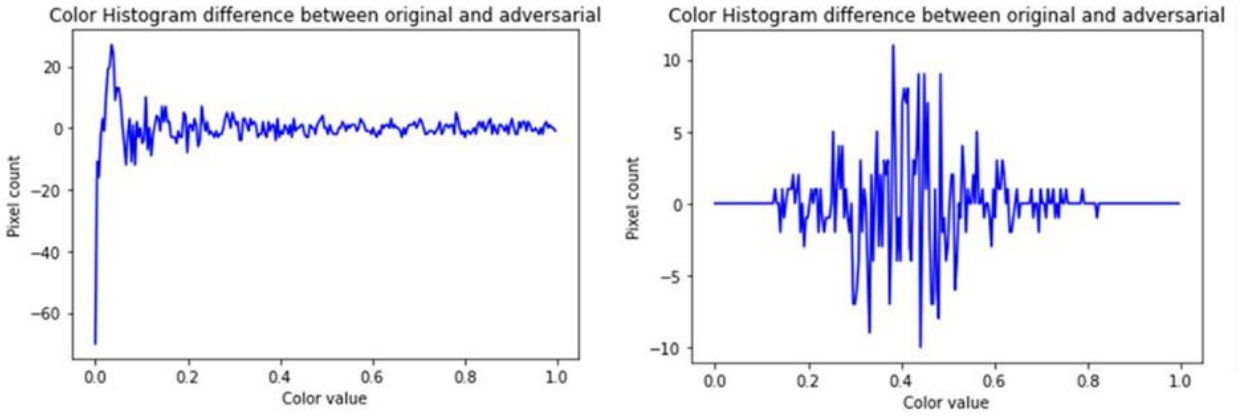

Figure 22: color difference b/w original and adversarial for the blue channel car image (left) and deer image (right)

6. PIXEL DIFFERENCE BETWEEN ORIGINAL AND ADVERSARIAL IMAGES

Similarly, the pixel value difference between the original and its adversarial images can also be seen in Figure *23*. Figure *24*, Figure *25*, and Figure *26* show the pixel-wise change between original and adversarial images for each color channel. This shows that the change is done throughout the image and the range of the change is between -0.03 and 0.03 for all channels. This result that many pixel values were changed instead of a few to make the adversarial examples show that pixel values may not be a good measure to finding a correlation between the images and the adversarial perturbations.

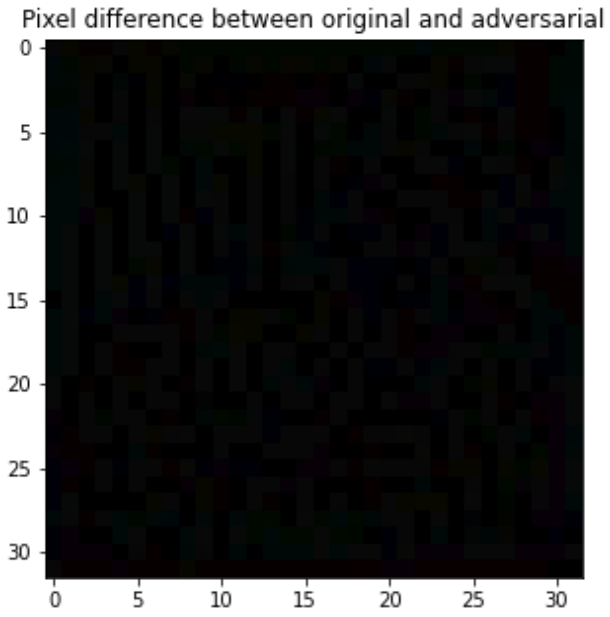

Figure 23: Pixel difference b/w original and adversarial image

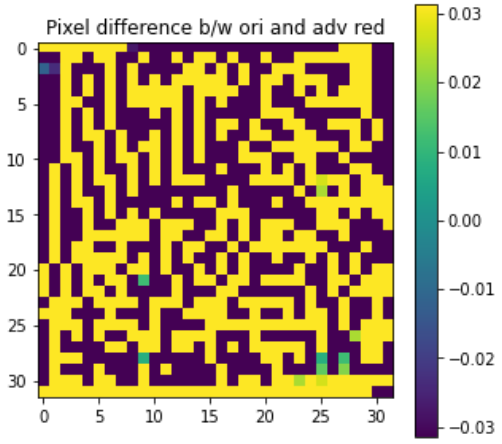

Figure 24: pixel difference b/w original and adversarial for the red channel

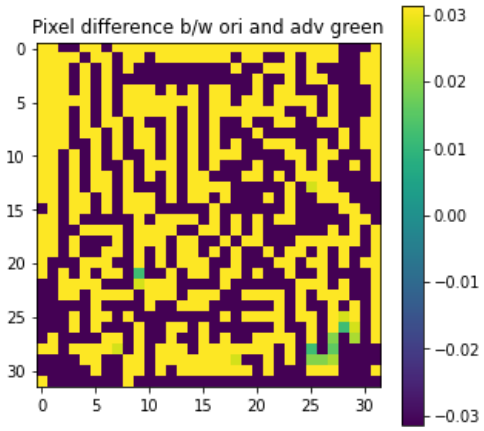

Figure 25: pixel difference b/w original and adversarial for the green channel

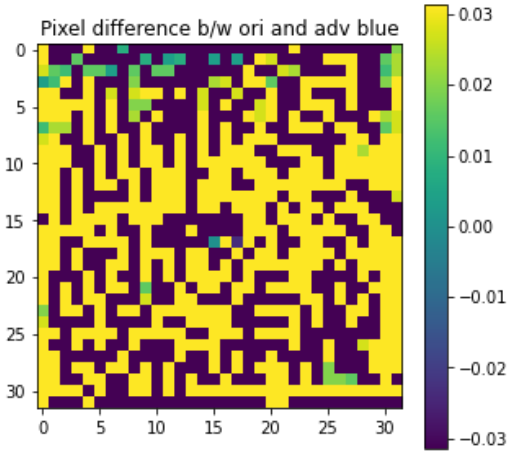

Figure 26: pixel difference b/w original and adversarial for the blue channel

7. PROBABILITY DISTRIBUTION OF ORIGINAL AND ADVERSARIAL IMAGES

Lastly, I looked at the probability distribution of the difference within each class for both original and adversarial image distribution. The results are shown in the figures below where Figure 27, and Figure 28 show the original and the adversarial deer and car class. Similarly, all other classes showed a similar distribution graph as the original for each class.

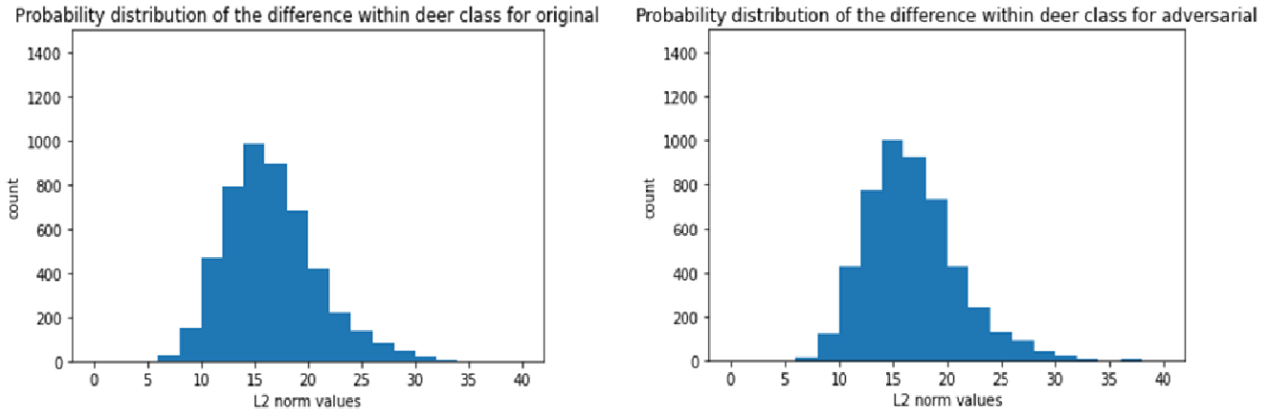

Figure 27: Probability distribution of the difference within deer class for original and adversarial images

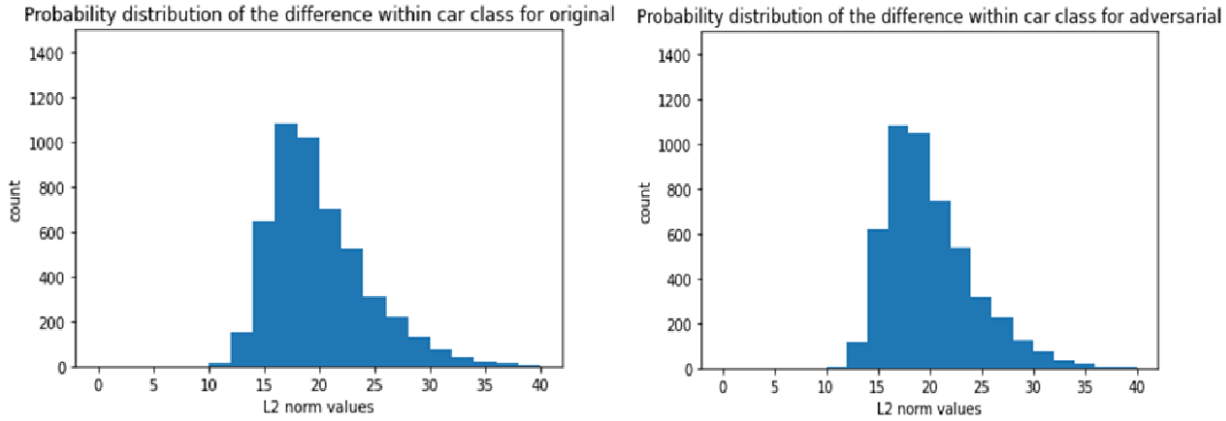

Figure 28: Probability distribution of the difference within car class for original and adversarial images

CHAPTER V

CONCLUSION/ FUTURE WORK

In summary, this paper looked at the different analysis methods available to find a correlation between original images and their adversarial counterparts. After applying these analysis methods, I have concluded that these normal statistical and visualization methods may not be able to identify the relation between the data and adversarial vulnerabilities. I may need to look into some advanced statistical and visualization experiments and other methods in the future to find the effects of data on the adversarial vulnerabilities not seen in the previous experiments. It could include diving deep to find a relationship between the RGB values changed and the Lumosity method as discussed in the results. Other future experiments can look into the following questions. Does applying edge detection help in identifying the relationship between these images or not. As edge detection is also used in feature maps for convolutional neural networks, so it may help in future work. Another question we can look into can be does the Segmentation of original and adversarial images remains the same or not or can I plot an Element-wise heat-map for changes from one class to another class to find the correlation between data and adversarial images created. Some far-fetched future work can also include finding if two different models trained on the same data have the same correlation between data and adversarial images and follow the hypothesis of transferability or not.

BIBLIOGRAPHY

BIBLIOGRAPHY

[1] J. Hurwitz, A. Nugent, F. Halper and M. Kaufman, "Big Data," *New York,* 2013.

[2] https://cloudtweaks.com/2015/03/how-much-data-is-produced-every-day/.

[3] https://techjury.net/blog/how-much-data-is-created-every-day/.

[4] https://seedscientific.com/how-much-data-is-created-every-day/.

[5] L. Ehrlinger, V. Haunschmid, D. Palazzini and C. Lettner, "A DaQL to Monitor Data Quality in Machine Learning Applications," in *Database and Expert Systems Applications*, Cham, Springer International Publishing, 2019, pp. 227-237.

[6] A. Moravanszky, "Linear algebra on the GPU," *Shader X,* vol. 2, 2003.

[7] K.-S. Oh and K. Jung, "GPU implementation of neural networks," *Pattern Recognition,* vol. 37, pp. 1311-1314, 2004.

[8] "Machine_learning," [Online]. Available: https://en.wikipedia.org/wiki/Machine_learning.

[9] L. Huang, A. D. Joseph, B. Nelson, B. I. Rubinstein and J. D. Tygar, "Adversarial Machine Learning," in *Association for Computing Machinery*, New York, 2011.

[10] N. Dalvi, P. Domingos, Mausam, S. Sanghai and D. Verma, "Adversarial Classification," in *Association for Computing Machinery*, New York, 2004.

[11] I. J. Goodfellow, J. Shlens and C. Szegedy, "Explaining and Harnessing Adversarial Examples," 2014.

[12] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow and R. Fergus, "Intriguing properties of neural networks," *arXiv preprint arXiv:1312.6199,* 2013.

[13] "Gatner's research," [Online]. Available: https://www.gartner.com/en/documents/3939991.

[14] "Adversarial attacks Benchmarks," [Online]. Available: https://github.com/iArunava/scratchai/tree/master/scratchai/attacks#benchmarks.

[15] A. Athalye, L. Engstrom, A. Ilyas and K. Kwok, "Synthesizing robust adversarial examples," in *International conference on machine learning*, PMLR, 2018, pp. 284--293.

[16] K. Eykholt, I. Evtimov, E. Fernandes, B. Li, A. Rahmati, C. Xiao, A. Prakash, T. Kohno and D. Song, "Robust physical-world attacks on deep learning visual classification," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 1625--1634.

[17] "Obama's deepfake," [Online]. Available: https://www.youtube.com/watch?v=cQ54GDm1eL0.

[18] M. Barreno, B. Nelson, R. Sears, A. Joseph and J. Tygar, "Can machine learning be secure?," *Proceedings of the 2006 ACM Symposium on Information, Computer and Communications Security, ASIACCS '06,* vol. 2006, pp. 16-25, 2006.

[19] Y. Deng and L. J. Karam, "Universal Adversarial Attack Via Enhanced Projected Gradient

Descent," in *2020 IEEE International Conference on Image Processing (ICIP)*, 2020, pp. 1241-1245.

[20] S. Huang, N. Papernot, I. Goodfellow, Y. Duan and P. Abbeel, "Adversarial Attacks on Neural Network Policies," 2017.

[21] L. Schmidt, S. Santurkar, D. Tsipras, K. Talwar and A. Madry, "Adversarially robust generalization requires more data," *Advances in neural information processing systems*, vol. 31, 2018.

[22] A. Shafahi, W. R. Huang, C. Studer, S. Feizi and T. Goldstein, "Are adversarial examples inevitable?," *arXiv preprint arXiv:1809.02104*, 2018.

[23] A. Ilyas, S. Santurkar, D. Tsipras, L. Engstrom, B. Tran and A. Madry, "Adversarial examples are not bugs, they are features," *Advances in neural information processing systems*, vol. 32, 2019.

[24] A. Madry, A. Makelov, L. Schmidt, D. Tsipras and A. Vladu, "Towards Deep Learning Models Resistant to Adversarial Attacks," 2017.

[25] C. Xie, J. Wang, Z. Zhang, Z. Ren and A. Yuille, "Mitigating adversarial effects through randomization," *arXiv preprint arXiv:1711.01991*, 2017.

[26] N. Papernot, P. McDaniel, I. Goodfellow, S. Jha, Z. B. Celik and A. Swami, "Practical black-box attacks against machine learning," in *Proceedings of the 2017 ACM on Asia conference on computer and communications security*, 2017, pp. 506-519.

[27] F. Tram{\`e}r, N. Papernot, I. Goodfellow, D. Boneh and P. McDaniel, "The space of transferable adversarial examples," *arXiv preprint arXiv:1704.03453,* 2017.

[28] G. W. Ding, K. Y. C. Lui, X. Jin, L. Wang and R. Huang, "On the Sensitivity of Adversarial Robustness to Input Data Distribution," *ICLR (poster),* 2019.

[29] R. Geirhos, P. Rubisch, C. Michaelis, M. Bethge, F. A. Wichmann and W. Brendel, "ImageNet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness," *arXiv preprint arXiv:1811.12231,* 2018.

[30] A. Krizhevsky, V. Nair and G. Hinton, "CIFAR-10 Dataset," [Online]. Available: https://www.cs.toronto.edu/~kriz/cifar.html.

VITA

Born in 1997, Hamza Zafar attended bachelors in Computer Science from National University of Sciences and Technology (NUST) in Pakistan and graduated from NUST in August 2018. After spending two years in industry back in Pakistan, joined University of Mississippi for a Masters in Engineering Science focused on Computer Science in January 2021 and received a Master of Science Degree in Engineering Science in August 2022.