**Adversarial Attacks and Defense Analysis**

By Jonathan Francisco

**1. Objective**

The goal of this experimental work is to evaluate the effectiveness of defensive techniques against security threats in a machine learning model. This idea stems from the paper from Qiang Liu et al on A Survey *on Security Threats and Defensive Techniques of Machine Learning: A Data Driven View* where they take a look at different adversarial models and defensive techniques to counter them. The end state of this study is to quantify the impact of security threats on machine learning models and demonstrate how defensive techniques can mitigate these threats. The effectiveness of the defensive techniques will be evaluated by comparing the original image, the attacked image, and the defended image.

**2. Experiment setup**

This experiment will start with a machine learning model that labels pictures of animals and tests the model against new raw data getting a percentage of confidence in the label. This program will be written utilizing python and the MobileNetV2 model and will introduce a security attack utilizing the Fast Gradient Sign Attack. After the attack, I will introduce a defensive technique to measure correctness compared to the control and attacked image. The final images will be shown, and the confidence percentages will be printed to view the comparison.

**3. Description**

The provided code showcases an implementation of the Fast Gradient Sign Method (FGSM) for adversarial attacks and defenses using the MobileNetV2 model. The code begins by importing the necessary libraries and defining helper functions for image preprocessing, label extraction, and image display. To start, the MobileNetV2 model is loaded and configured for use. An input image is then loaded and preprocessed to prepare it for input into the model. The code obtains the predicted label for the preprocessed image using the MobileNetV2 model.

Next, the FGSM method is employed to create an adversarial pattern. This is achieved by calculating the gradient of the loss with respect to the input image. The code selects multiple epsilon values and generates adversarial examples by perturbing the input image based on the adversarial pattern. The original image and the generated adversarial examples are displayed to visualize the impact of the perturbations on the image. Moving forward, the code proceeds to execute the FGSM attack by iteratively applying the adversarial pattern to the input image. This process aims to make the image more susceptible to misclassification by the model.

To counteract the adversarial attack, a defense function is defined. This function aims to restore the original image by subtracting the perturbation from a defense image iteratively. The defended images are displayed to observe the effectiveness of the defense mechanism.

Finally, the regular image, attacked image, and defended image are evaluated and compared by obtaining predictions and confidence scores from the MobileNetV2 model. This allows for a quantitative analysis of the impact of the attack and defense on the model's performance.

**4. Experiment results**

**A screen shot of a screen

Description automatically generated with low confidence**A panda bear climbing a tree

Description automatically generated with medium confidenceThe following images show the input image and the adversarial pattern created from the input image. In this case we used a picture of a giant panda, as shown in figure-1 there is a

Figure – Input Image Figure - adversarial pattern

77.01% confidence from the MobileNetV2 Model that the input image is indeed a panda. Figure-2 shows the adversarial pattern created which is then applied to the input image in order to misclassify the input panda image.

A panda bear in a tree

Description automatically generated with medium confidenceA panda bear climbing a tree

Description automatically generated with low confidence

Figure - Adversarial image 1 Figure - Adversarial image 2

As shown above, the perturbations that were created from the adversarial model have then been applied to the input image. Figure-3 shows a perturbation with an epsilon of 0.1 and Figure-4 shows a perturbation with an epsilon of 0.15. As you can see, the increase of epsilon changes the level of misclassification confidence in the input image. The epsilons range in order to vary the amount of clutter created by the adversarial model. In practice, too large an amount of clutter would still lead to a higher confidence of misclassification, however, would be easier to discover. When evaluating the success of adversarial models, the largest effect isn’t always the goal. A smaller perturbation that still shows a misclassification in essence accomplishes the goal and has the added effect of fooling more models as the image still closely resembles the base input.

After the adversarial images were created, they are then passed through the FGSM Defense function. The goal of this function is to reverse the perturbation that could have been applied to the input image. In this case we use a smaller epsilon (shown in Figure-5 of 0.001) to clean the clutter that could have been applied by an FGSM attack. Figure-4 then shows the output image from the defense, successfully relabeling the image to match the input with a high percentage of confidence.

A panda bear climbing a tree

Description automatically generated with low confidence

Figure - Defended Image

This test has been tried with multiple pictures of a variety of different animals leading to similar results. However, one problem that can occur is if the model initially misclassifies the original image. If the original image is misclassified, the defended image will be as well. However, it would still match the original, this problem would have to be solved at the initial training model.

After the attack and defended images are generated, an evaluation is conducted to see the confidence level of each image. As shown in Figure-6 the initial confidence is at 77.01% and the label shows the model’s name for a giant panda. In comparison, applying the perturbation at an epsilon at 0.15 brings the confidence down to 39.6% and misclassifies the image as a titi. After applying the defense model, it not only labels the image correctly, but outperforms the original image confidence.

A screen shot of a computer

Description automatically generated with medium confidenceFigure - Comparison of Images

**5. Conclusion**

In conclusion, adversarial attacks, like the FGSM attack, can successfully misclassify images on trained models. This is important as more automation with visual processing comes to market. For example, new self-driving cars’ ability to read road signs can be attacked as physical perturbations can be applied on road signs, hindering performance, and ultimately causing unsafe driving conditions. However, defense models can and are effective in countering these attacks. These defense models can’t defend against all adversarial model attacks and have to be continuously adjusted to fit new and upcoming attacks. Overall, this machine learning and its attack and defense is a booming industry for both research and development. Future research in these fields of technology needs to include defensive and security techniques when developing new technologies.