Analyzing the Effect of Adversarial Inputs on Saliency Maps

Zhi Qi Liu, 1003479354 **Zhiwei Liu, 1003493007** **Salar Hosseini, 1003142020**

Abstract

In machine learning, small perturbations applied to images can fool a model into making confident but incorrect predictions. These adversarial images are often indiscernible to the human eye, but can cause a model to fail catastrophically. In this work, saliency maps, namely GradCam, GuidedBackprop, and SmoothGrad, were used to compare the salient features of the original images from CIFAR-10 and their adversarial counterparts engineered using the Fast Gradient Sign Method (FGSM). For both untargetted and targetted attacks, quantitative measurements of similarity demonstrated that GradCam was the most successful at detecting adversarial inputs. Therefore, we propose that GradCam has the potential to be a tool for not only visualizing salient features, but also detecting adverserial attacks.

1 Introduction

Machine learning models continue to grow larger and more complex in design, making it increasingly difficult to understand the intuition behind the computations. Interpretability of neural nets is not only useful for debugging purposes but is also a key component to exposing potential biases in the model. Saliency maps have emerged as a promising method to understand what a model considers as relevant features in the inputs [1]. We propose to use different saliency map methods to understand the effects of adversarial inputs and how they affect a model's prediction ability. Adversarial inputs will be generated for the CIFAR-10 images and we will compare the saliency maps generated by SmoothGrad [2], GradCAM [3] and GuidedBackprop [4] on raw inputs versus adversarial inputs. Finding a successful salient model will help improve adversarial input detection and help us gain a better understanding of interpretability methods applied to machine learning.

2 Related Work

2.1 Explanation & Interpretability Maps

To improve the interpretability of complex CNNs, saliency methods have been developed to highlight the features that are supposedly most relevant to a network's prediction. Simonyan et al. showed that this saliency could be visualized by simply plotting the gradient [5] which is an indicator of how much changes in each input dimension would change the predictions. Another method, SmoothGrad [2] reduces the noise from using only the gradient by averaging saliency maps over several inputs which have Gaussian noise added to them. Expanding on the idea of using gradients for visualization is GradCam [3], which takes the gradient of the logit for a particular class with respect to the feature map after the last convolutional layer. This is motivated by the reasoning that the feature maps from the deeper stages of a CNN capture higher-level visual entities [6] and that the last convolution stage is the last to retain spatial information. GuidedBackprop [4] takes a different approach and returns the gradient map obtained by zeroing out negative gradients obtained while backpropagating through ReLU units. Other notable methods include Gradient $\odot$ Input [7] and Integrated Gradients [8].

Preprint. Under review.

2.2 Adversarial Inputs

Adversarial inputs [9] are formed by intentionally applying worst-case disturbances to data samples such that models confidently output incorrect predictions. The Fast Gradient Sign Method (FGSM) [9] constructs these disturbances as the sign of the gradient of the cost function used to train a neural network. Furthermore, adversarial inputs can be constructed with the intention to minimize the probability that the adversarial input results in a correct class prediction (untargetted attack), or to maximize the probability that the adversarial input results in a target class (targeted attack) [10].

3 Method

To evaluate the various saliency maps, we studied how adversarial examples in the problem domain of image classification affect the resulting maps. Primarily, a convolutional neural network, namely AlexNet [11], was used to perform multi-class image classification on the CIFAR-10 images dataset [12], which has 10 classes. Our approach can be formulated as:

1. Fine-tune an ImageNet-pretrained AlexNet on CIFAR-10 to perform image classification.
2. For each image in the reserved test set, compute class predictions, non-targetted adversarial examples that fool AlexNet into misclassifying the image, and targetted adversarial examples that fool AlexNet into predicting each possible class.
3. Compute the saliency maps with respect to the predicted targets of the original input (ie. real_pred) and corresponding adversarial examples (ie. fake_pred).
4. Analyze the pairs of saliency maps for each method quantitatively (eg. with similarity metrics) and qualitatively (eg. by considering the features that seem to be learned by the model).

To visually analyze the effect of an attack, saliency maps were generated using SmoothGrad [2] , GradCam [3] and GuidedBackProp [4]. The SmoothGrad explanation is defined in Equation 1, where noise vectors $g_i \sim \mathcal{N}(0, \sigma^2)$ are generated i.i.d. from a normal distribution.

$$\boxed{E_{SG} = \frac{1}{N}\sum_{i=1}^{N} E(x + g_i)} \tag{1}$$

GradCam explanations are derived from the feature map of the last convolution layer. Let A^k be this map, then the GradCam explanation is defined in Equation 2, where $\alpha_c^k = \frac{1}{Z}\sum_i \sum_j \frac{\partial S}{\partial A_{ij}^k}$ are the linear computation weights.

$$\boxed{E_{\text{GradCam}} = ReLU\left(\sum_k \alpha_c^k A^k\right)} \tag{2}$$

GuidedBackProp uses a different approach that aims to zero out negative gradients of ReLU units during backpropagation. Let $f^l = ReLU(f^{l-1})$ and the corresponding intermediate representation obtained during backpropagation $R^{l+1} = \frac{\partial f^{\text{out}}}{\partial f^{l+1}}$. Then with GuidedBackProp, the mask is computed according to Equation 3, where the indicator function is a mask that keeps only positive gradients and positive activations.

$$\boxed{R^l = \mathbb{1}_{R^{l+1} > 0 \text{ and } f^l > 0} R^{l+1}} \tag{3}$$

To generate an adversarial prediction, the Fast Gradient Sign Method [9] was used. It is defined in Equation 4, where the prediction X is generated by adding a fixed perturbation in a direction. For an untargetted attack, the direction is directly away from the source image and for a targetted attack, the direction is towards a predetermined (wrong) class label.

$$\boxed{X_{\text{Adversarial}} = X + \epsilon \cdot sign\left(\nabla_X J(X, Y)\right)} \tag{4}$$

While evaluating human perception is an active area of research, some well-known quantitative metrics are Spearman rank correlation, Mean of Structural Similarity Index (mSSIM), Pearson correlation of histogram of gradients (HoGs) [1]. These metrics were used to quantify the visual similarities of the true gradients and adversarial gradients.

4 Experiments and Results

4.1 Untargetted FGSM Attack

The saliency maps of an image before and after an adversarial attack were generated using SmoothGrad, GradCam and GuidedBackprop. To generate SmoothGrad saliency maps, $N = 10$ noisy maps were sampled. It should also be noted that the gradient maps returned by SmoothGrad and GuidedBackprop were converted to have a single channel (grayscale). This was done to have a fair quantitative comparison with GradCam which only returns a single channel, and for more clear visualization. A sample visualization of the different methods on non-targetted adversarial samples from three different perturbation (ϵ) levels is shown in Figure 1. Qualitatively, it can be observed that GradCam generates very coarse saliency maps, while SmoothGrad and GuidedBackprop generate much more fine-grained maps. As the value of ϵ increases, it becomes increasingly evident from the RGB images that an attack is being performed. Furthermore, it is visually clear that the saliency maps also become more perturbed. For the case of $\epsilon = 0.2$, the attack is not large enough to deviate the class prediction, but from only the noise in the image, the saliency maps already change noticeably. For the case of $\epsilon = 0.6$, the attack results in an incorrect prediction of 'frog' and due to taking gradients with respect to the wrong class, the saliency maps are perturbed even more significantly.

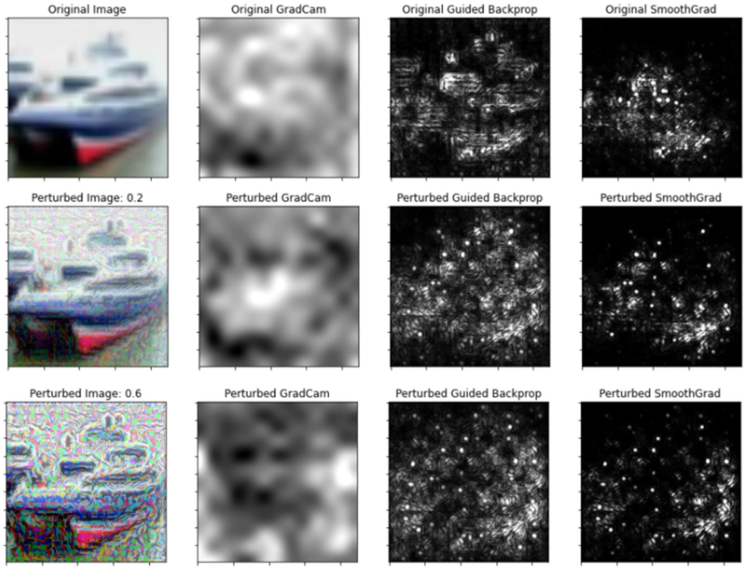

Figure 1: Sample image of a ship image from CIFAR-10 and visualizations of gradients generated using SmoothGrad, GradCam and GuidedBackpropagation on the original and adversarial images ($\epsilon = 0.2$ results in a prediction of 'ship', and $\epsilon = 0.6$ results in a prediction of 'frog'). See Figure 3 in the visualization section of the Appendix for full image.

To conduct a quantitative analysis, untargetted FGSM attacks with different perturbations (ϵ = {0.2, 0.4, 0.6, 0.8, 1.0}) were performed on each image in the test dataset and the saliency maps were computed on the image before and after the attack. The maps were compared using Spearman rank correlation, mSSIM and Pearson correlation of HoGs. Lower values from these metrics indicate more dissimilar saliency maps between the original and attacked images (i.e. more evident attacks), while higher values indicate less evident attacks. The mean performance on each metric across the entire dataset is shown in Figure 2.

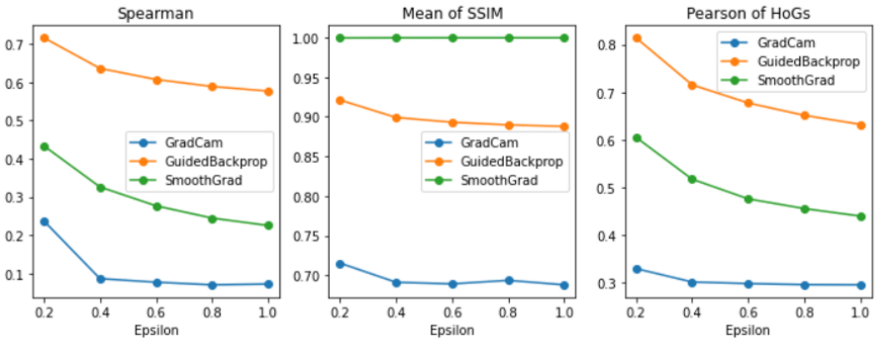

Figure 2: Quantified similarities of saliency maps across untargetted FGSM attacks with increasing level of perturbations.

As the level of perturbation increased, all saliency methods were able to generate increasingly dissimilar saliency maps which is expected since the adversarial predictions become increasingly different. Overall, GradCam achieves the lowest similarity across all metrics, achieving a Spearman correlation of ∼0.1, mSSIM of ∼0.7 and a Pearson correlation of ∼0.3 for its histogram of gradients. These are all significantly lower than the metrics obtained by other methods, suggesting that GradCam generates the most dissimilar saliency maps for an image before and after an untargetted FGSM across all perturbation levels.

4.2 Targetted FGSM Attack

Similar to the quantitative analysis conducted for untargeted adversarial attacks, targeted FGSM attacks were performed on the entire test dataset across all the different target class labels (10 classes for CIFAR-10) using a perturbation of $\epsilon = 0.4$. The mean performance of each saliency method across all similarity metrics is shown in Table 1.

Table 1: Mean Spearman, mSSIM and Pearson correlation of HoGs achieved by each saliency method across all class labels used for targeted FGSM attacks.

Adversarial Method	Spearman Rank Correlation	mSSIM	Pearson Correlation of HoGs
SmoothGrad	0.288	0.999	0.495
GradCam	0.0804	0.689	0.298
GuidedBackprop	0.602	0.895	0.668

As shown in Table 1, GradCam achieves the lowest mean similarity across all target classes for each of the metrics. Notably, it achieves a mean Spearman rank correlation of 0.0804 which indicates almost no correlation, indicating that GradCam is exceptionally good at detecting a targeted FGSM attack. Furthermore, an experiment was conducted to compare these correlation metrics across each of the different target classes, and it was observed that there is little variance in the correlation of the original and attacked saliency maps across the different classes. This result is shown in Figure 4 in Appendix D.

5 Conclusion

Adversarial attacks can fool a model into making incorrect predictions with confidence. Having examined SmoothGrad, GradCam and GuidedBackprop as promising saliency mapping methods to detect targetted and untargetted FGSM attacks, we have shown that GradCam consistently outperforms other methods in terms of Spearman rank correlation, mSSIM and Pearson correlation of HoGs.

References

[1] Julius Adebayo et al. *Sanity Checks for Saliency Maps*. 2020. arXiv: 1810.03292 [cs.CV].

[2] Daniel Smilkov et al. *SmoothGrad: removing noise by adding noise*. 2017. arXiv: 1706.03825 [cs.LG].

[3] Ramprasaath R Selvaraju et al. *Grad-CAM: Why did you say that?* 2017. arXiv: 1611.07450 [stat.ML].

[4] Jost Tobias Springenberg et al. *Striving for Simplicity: The All Convolutional Net*. 2015. arXiv: 1412.6806 [cs.LG].

[5] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. *Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps*. 2014. arXiv: 1312.6034 [cs.CV].

[6] Matthew D. Zeiler and R. Fergus. "Visualizing and Understanding Convolutional Networks". In: *ECCV*. 2014.

[7] Avanti Shrikumar et al. *Not Just a Black Box: Learning Important Features Through Propagating Activation Differences*. 2017. arXiv: 1605.01713 [cs.LG].

[8] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. *Axiomatic Attribution for Deep Networks*. 2017. arXiv: 1703.01365 [cs.LG].

[9] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. *Explaining and Harnessing Adversarial Examples*. 2015. arXiv: 1412.6572 [stat.ML].

[10] Wenqi Wei et al. *Adversarial Examples in Deep Learning: Characterization and Divergence*. 2018. arXiv: 1807.00051 [cs.LG].

[11] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. "Imagenet classification with deep convolutional neural networks". In: *Advances in Neural Information Processing Systems*, p. 2012.

[12] Alex Krizhevsky. "Learning Multiple Layers of Features from Tiny Images". In: *University of Toronto* (May 2012).

A Attribution

Zhi Qi Liu, 1003479354

- Implement Spearman rank correlation, SSIM, Pearson correlation of HoGs
- Write report

Zhiwei Liu, 1003493007

- Implement targetted/untargetted adversarial attacks
- Write report

Salar Hosseini, 1003142020

- Implement GradCam, SmoothGrad, GuidedBackprop
- Write report

B Source Code

Link to *pynb* file to reproduce experiments.

`https://github.com/zqallan/csc413-project`

C Visualizations for Untargetted Attack

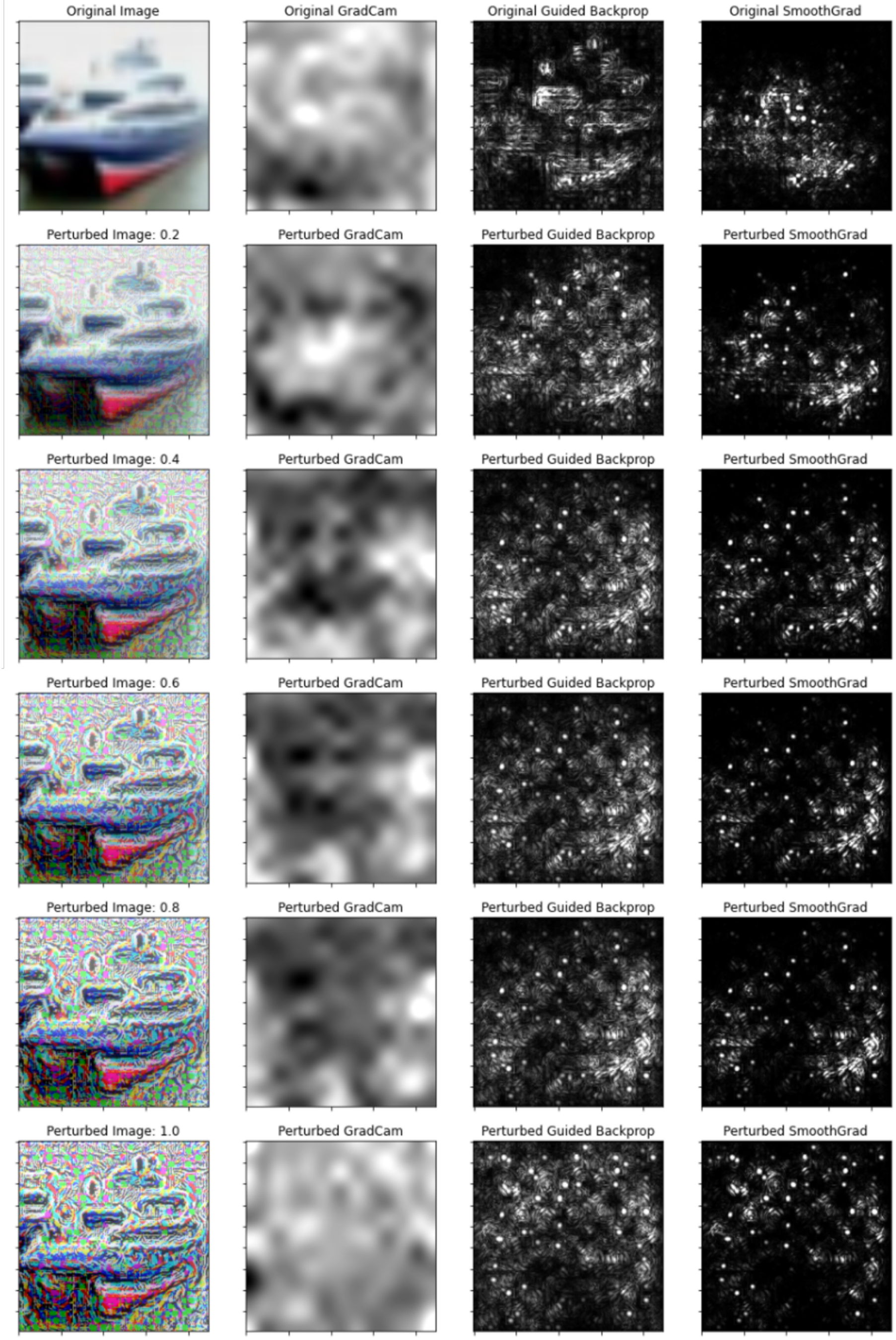

Figure 3: Full image of a ship image from CIFAR-10 and visualizations of gradients generated using SmoothGrad, GradCam and GuidedBackpropagation.

D Quantitative Metrics for Targetted Attacks

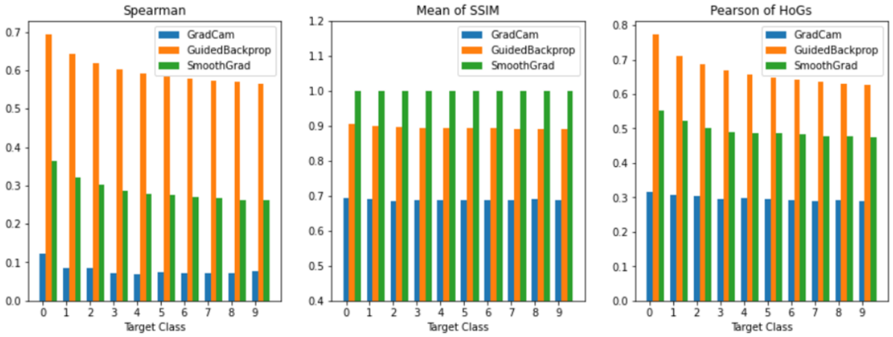

Figure 4: Spearman, mSSIM and Pearson correlation of HoGs achieved by each gradient method across all class labels used for FGSM attacks.