Evaluating CNN Predictions using Visualization Techniques

Syed Usama Amer (amers@carleton.edu)
Undergraduate Student, 300 North College Street
Northfield, MN 55057 USA

Kaixing Wu (wuk@carleton.edu)
Undergraduate Student, 300 North College Street
Northfield, MN 55057 USA

Abstract

Convolutional Neural Network models have produced remarkable results in image classification. Nevertheless, there is no clear understanding of why they perform so well. We use two different techniques that offer visual data to understand Convolutional Neural Networks. These approaches are known as Occlusion Sensitivity and Gradient-Weighted Class Activation Mapping. We also collect data from humans to configure what portions of an image they use to make decisions with respect to classification. We compare the results from Occlusion Sensitivity, Class Activation Mapping and human data, and find out that CNN classify images based on the similar portions that human used to classify things. Moreover, we also find out that CNN behaves differently when classifying the irregular inputs, such as adversarial examples and noise images. We give some possible explanations and believe that further research is needed to explain this different behavior.

Keywords: Convolutional Neural Network Classification; Occlusion Sensitivity; Grad-CAM; Visualizing Network; Adversarial Examples;

Introduction

Convolutional Neural Networks (CNNs) are part of a breed of deep networks that have produced previously unimaginable improvements in performance with respect to a range of computer vision tasks. These tasks include, but are not limited to, image classification, object detection, semantic segmentation and visual question answering. The aforementioned progress in performing computer vision tasks with an almost unerring certitude can be traced to several factors (Zeiler & Fergus, 2013) :

1. Ubiquity and easy availability of extraordinarily large data sets
2. Advancements in implementations of GPU, which has made the training of vast datasets a viable option
3. Utilization of better strategies to regularize models

Despite improvements by leaps and bounds in performance, due to the factors listed above, there is very little insight into as to what precisely happens under the hood that allows for such spectacular results. It is very difficult to even begin decomposing the internal operations, that cumulatively produce particular behaviour, into intuitive and comprehensible components. We need better understanding of CNNs for several reasons, namely, it will make developers less reliant on trial and error as they have a better understanding of the components needed to solve a particular problem. Further, it is important to shed the mystique that surrounds Artificial Intelligence (AI) systems because it breeds mistrust amongst users for them. In the case where AI systems perform as well as, if not better than humans, explanations of the underlying processes allow us to teach humans to be better at making predictions. (Selvaraju et al, 2016)

As such, the key questions we are investigating falls under the category of visualizing and understanding convolutional networks with respect to computer vision related tasks. Our experiment contains two parts: the first part is to evaluate the reason CNN make correct predictions for regular inputs by visualization; the second part is to evaluate the reason CNN make wrong predictions for irregular inputs by visualization.

As for the first part, we attempt to visualize the most salient regions on an image that contribute most prominently towards the correct predictions with respect to several different inputs. We implement three experiments in order to investigate our question. Two of these experiments are concerned with evaluating the performance of CNNs by utilizing two different approaches. The third experiment makes use of human subjects to offer some insight into what visual regions humans make use of in order to identify and categorize the inputs under scrutiny. This experiment is used to evaluate the results of the Occlusion Sensitivity experiment and will be discussed in the same section. We compare the results of these experiments in an attempt to gain better insight into the similarities and differences that govern the results produced.

As for the second part, we first use some adversarial images made by FGSM attack as inputs and compare the Grad-CAM results of adversarial images to the results of original images. Then, we compare the performance of different models and offer some explanations to the results. Lastly, we use the noise images as inputs and use Grad-CAM results to figure out the reason CNN make wrong predictions.

Background

Occlusion Sensitivity & Human Experiment Our first experiment, makes use of a sensitivity technique pioneered by Zeiler & Fergus (2013), analyzes classified output by occluding parts of the input image. We call this the regular input experiment. We pass the image under consideration several hundred times through a pre-trained CNN to produce a heat map that is coloured according to the distribution of probabilities produced at each run at every occluded part of the image. This allows us to reveals which parts of the image are most important for its correct classification. We were drawn to making use of this technique because it seems a very intuitive and natural way of testing what parts of an image the CNN thinks are chiefly responsible for producing the correct output. We do not tamper with the internal processes of the CNN, but simply give it different stratas of information with respect to the same image. Changing the input while keeping other things constant gives us an insight as to how the CNN reacts to different portions of the same input. Therefore, in this experiment, there is greater emphasis on the correctness of the model itself. In order to evaluate our results, we asked a group of ten human subjects to select regions on an image that they deemed most important for making decisions with respect to classification of our chosen input images. They indicated the regions by drawing upon them with a shape of their choice. We wanted to have some human data to which we could compare our results vis a vis CNNs.

Grad-CAM Our approach for the second experiment is far more theoretically intensive. It was devised by Selvaraju et al (2016). The technique is called Gradient-Weighted Class Activation Mapping or more commonly known as Grad-CAM. In essence, the approach "uses the class-specific gradient information flowing into the final convolutional layer of a CNN to produce a coarse localization map of the important regions in the image." (Selvaraju et al, 2016) This approach allows for insight into cases where our classifier fails, something our occlusion experiment cannot offer us, and demonstrates that reasonable explanations can account for seemingly absurd classifications. The criteria for an insightful visual explanation that informs the principles upon which the Grad-CAM technique is created are as follows: (i) Any good visualization technique should be class-discriminative, that is, it should convincingly localize and hone in on the category in the image (ii) It should be high-resolution, that is, it should be able to aptly capture the most minute details.

Selvaraju et al (2016) discovered that the Grad-CAM approach is naturally suitable for the first criterion for it is highly class-discriminative. Unfortunately, the approach is not very high-resolution. To that end, they took inspiration from other approaches in the field, such as Guided Backpropagation and Deconvolution, which produce high-resolution results that lack the class-discriminative attributes of Grad-CAM. In a remarkable breakthrough, they are able to pull and combine the theoretical strands that give rise to the disparate strengths of the aforementioned discrete models. They demonstrate that it is possible to "fuse existing pixel-space gradient visualizations with Grad-CAM to create guided Grad-CAM visualizations that are both high-resolution and class-discriminative." (Selvaraju et al, 2016）

ResNet and VGG16 ResNet and VGG16 are CNN models that can be used for image classification. ResNet (Kaiming He et al., 2015) is a residual learning framework to ease the training of networks that are substantially deeper than those used previously. VGG16 (Karen Simonyan et al., 2014) is a 16-layer-deep Convolutional Networks for large-scale image recognition. In general, ResNet has a better accuracy than VGG16. We use the imagenet pretrained models that outputs a distribution of probability of 1000 class label and use the label of top1 probability as the prediction label.

Adversarial Examples Adversarial examples are inputs to machine learning models that an attacker has intentionally designed to cause the model to make a mistake, such as wrong prediction. Usually, the modification of the image is unnoticeable to human eyes but may fool the model.

Fast Gradient Sign Method (FGSM) Fast Gradient Sign Method is an adversarial attack method created by Ian Goodfellow et al. (2014). It computes the adversarial examples efficiently by adding a perturbation in the direction of the sign of the gradient of the cost function with respect to the adversarial example. It is a one iteration but useful white box attack and usually can fool the image classifiers given the pretrained network model .

Experiment 1: Regular Inputs

In this section, we will investigate the results of the Occlusion Sensitivity experiment. A key question that arises when we try to classify images using CNNs is that whether or not the model is well and truly honing in on the location of the object or relying on other elements in the image to make its classification. In order to investigate this question, we carry out this experiment, due to the limited computational power available to us, on a small sample of four images to draw any conclusions. To compare our results to human data, we carry out an experiment to see which regions humans privilege over others to make their classification decisions.

Methods

We make use of a pre-trained VGG16 model in order to carry out this experiment. We carried the experiment out in Google Collab environment because after much experimentation it was the only stable environment we could find. We write our own function to occlude parts of the image. The function takes in the path of an image and modifies it so that it can be processed by our VGG16 model.We first make a prediction with the correct image to check if the model performs well and it does. Then we proceed to occlude parts of the image by setting the occlusion window. We iterate over all occluded pixels and store the probability of the prediction made by the classifier at each row and column on the image in a two-dimensional array called heatmap. We write another function to generate the heatmap of each image. The function also produces an image wherein we make the heatmap translucent and superimpose the original image on top of it to produce a better direct comparison.

For our experiment to collect data from humans, we asked a group of randomly selected people to look at the same set of pictures and select regions which drew them in and allowed them to classify the image. They could draw any shape on the image to indicate the area of interest. One of us then looked at all ten images produced for each classification. Based on estimations from the naked eye we produced one image for each classification wherein care was taken to note the aggregate of all regions that were privileged over others to make decisions by our volunteers. We realize the limitations of this method, namely, we are introducing imprecision into the proceedings by relying on our own faculties. We tried to find the dataset for human attention for classification ground truth heatmap, but according to the research of Sara Hooker et.al (2018), we know that, “Assessing the correctness of importance rankings is complicated by the lack of ground truth. If we knew the true importance of each pixel in the image to the model prediction, estimating feature importance would reduce to a traditional supervised learning problem.” Therefore, in the future, we believe a better experiment can be designed: we could make 20*20 grids on top of every image and let 100 students click the grids that contribute most towards the classification and build a ground truth heatmap based on those results. With more data, this design would have yielded results that would be more reliable. Nevertheless, limitations with respect to time and resources, have led us to carry out this experiment in such a manner. Further, we feel that at the very least, this allows us to have some yardstick to compare our results to. As such, our approach should be seen as heuristic with respect to human data.

Due to dearth of space, we will only include heatmaps of one image. The rest will be appended to the appendix.We will also include the one corresponding image produced by aggregating the regions privileged by human subjects to illustrate that results from our model and human subjects are quite similar. The appendix will include the rest of the aggregated images. Since the actual images produced by human subjects amount to a total of forty. We have included them in a separate folder under the label “raw_images.”

Results & Discussion

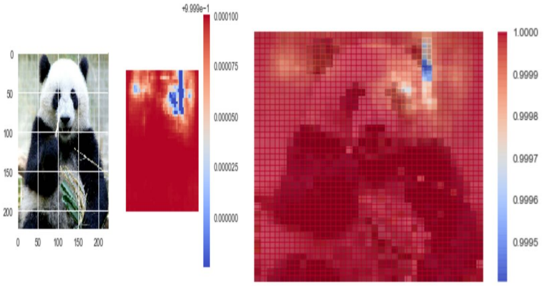


Figure 1. On the left side, we have one image that is the original image. The other is heatmap generated with respect to that image. On the right hand-side we have superimposed version. (Care must be taken to note that the pictures are not aligned in the superimposed versions)

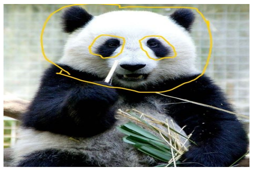


Figure 2. The general region which humans use to identify panda is quite similar to the CNN.

Our results demonstrated convincingly that the model is indeed localizing the objects we are trying to classify because the probability of the correct class decreases significantly when portions of the image where the object is located are occluded. It seems that humans have a wider field they use to make classification decisions.Nevertheless, there is sufficient overlap between our model and huma data to conclude humans and CNN localize the object to make decisions of classification.

Experiment 2: Irregular Inputs

In this section, we will investigate the Class Activation Map (CAM) of the irregular inputs using Grad-Cam. The main goal of this experiment is find out the reason the classification of the CNN classifiers may go wrong on irregular inputs by using Grad-CAM visualization. We believe there are two types of the irregular inputs, one is the modified images, such as adversarial examples that were calculated to fool the classifiers and another one is noise images. Therefore, our experiment contains two parts: first, we will compare the Class Activation Map of adversarial images against the original images and then, we will use noise images as input to generate the Class Activation Map.

Due to the time limits, we do not have enough human data to generate the heatmap as ground truth, so our analysis on based on subjective judgement toward the appearance of the results images. Moreover, due to the computing power limit, we will only select six images as examples for this experiment, so our analysis is based on small samples and we will try to draw a possible explanations without our own data support but from the work of other researchers.

1. Adversarial images for single model

Methods:

In this experiment, we use FGSM method to generate adversarial images using pretrained VGG16 as the target model with the parameter of epsilon 4. Once the adversarial images were generated, we use them and the original images as input to feed in the Grad-CAM jupyter notebooks. In the jupyter notebooks, we first use classifier model to get the original labels and combine them as a label_batch. We then feed the label_batch and the image_batch into Grad-CAM for VGG16 to get the Class Activation Maps.

The goal of this experiment is to analysis the changes of Class Activation Map between the original images and the adversarial images.

Results & Discussion

For full results, please see the append pdf file--gradCAM_tensorflow_VGG16. Here I will display the VGG16 results for pandas and agama as examples for the discussion in Figure 3 and Figure 4.

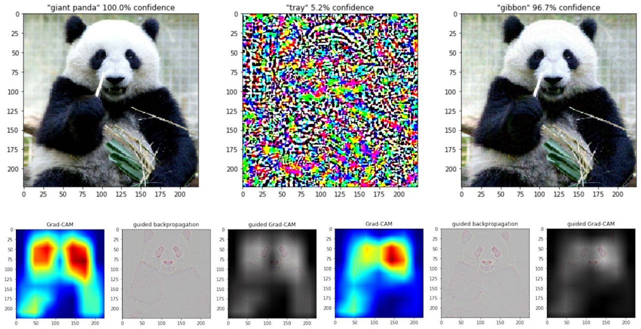


Figure 3. Comparison of original panda image(left) with FGSM modified panda image(right).

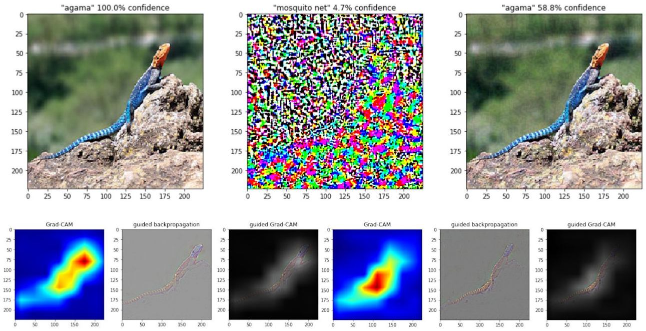


Figure 4. Comparison of original agama image(left) with FGSM modified agama image(right).

For the results, we can tell that the effects of FGSM can be different based on different images. In pandas image, the FGSM attack indeed fools the VGG16 classifier but not all images. In agama image, the attack only reduces the top 1 probability without changing any labels.

Moreover, from the Grad-CAM panda result, we can see that the focus has been changed (shifted towards the left eyes). This could means that the classifier is judging the image based on a different standard than human. While in the Grad-CAM agama result, the focus is still on the main body of the agama (shifting a little down from the head to the body) but the classifier can still recognize this body as agama excerpt with lower probability. We believe that this shift of the attention could be helpful in understanding the adversarial examples but further research is needed.

The reasons that adversarial examples can fool the model are still under research. The first possible explanation is that it is due to extreme nonlinearity of deep neural networks, perhaps combined with insufficient model averaging and insufficient regularization of the purely supervised learning problem (Szegedy et al. 2013). The second theory is that the models have linear behavior in high-dimensional spaces which caused this vulnerability (Goodfellow et al., 2014). The last possible answer is a Boundary Tilting Perspective (Tanay et al., 2016). In this hypothesis, "a class boundary can intersect this submanifold such that the two classes are well separated, but will also extend beyond it. Under certain circumstances, the boundary might be lying very close to the data, such that small perturbations directed towards the boundary might cross it."

2. Adversarial images cross model comparison.

Methods:

We use the same batch_image and batch_label in the previous experiment as inputs and visualize the Class Activation Maps from VGG16 and ResNet 101 model. The goal of this experiment is to know whether the FGSM attack method for VGG16 can fool ResNet 101. We will use both the classification score and the Grad-CAM result for comparison to figure out the reason.

Results & Discussion

For full results, please see the append gradCAM_tensorflow pdf files.. Here we display the VGG16 and ResNet 101 results for panda as examples for the discussion in Figure 5.

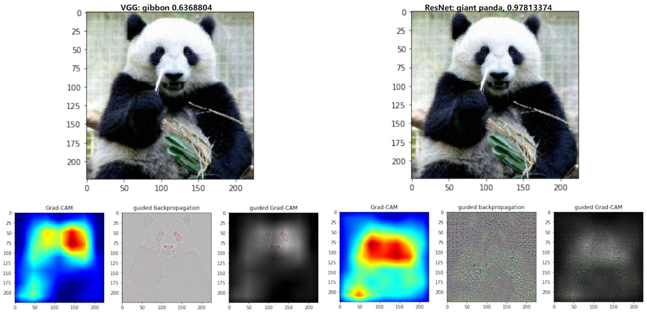


Figure 5. Comparison of CAM result on VGG16(left) with ResNet 101(right) on the FGSM panda image.

From the comparison, we can see that the FGSM attack on VGG16 did not fool the ResNet 101 classifier on panda images, since the ResNet 101 can still predict that the FGSM panda image as 97.8% of being a panda. From Figure 3, we can see that the important area shifted back to the two eyes of the panda, just like the Grad-CAM results for VGG16 on original panda image in Figure 1, while the Grad-CAM results for VGG16 on FGSM panda image focus more on the left eyes.

Moreover, we notice that ResNet generally has a better attack resistance toward the FGSM adversarial on VGG16. Though, we do not have strong data support, from the paper of Cihang Xie et al. (2018), we believe some relevant reasons are that: First, the FGSM attack is calculated based on VGG and the transferability is not so great, according to the Appendix Table 1; Second, the FGSM attack is "a traditional single-step attacks that tends to underfit to the specific network parameters θ due to inaccurate linear appropriation of the loss L(X, y_true; θ), thus cannot reach high success rates".

3. Noise images for VGG16 classification

Methods:

The noise images are computer-generated noise that the classifiers never seen before. CNN classifiers will sometimes classify those noise image as some class labels X with high probability, while we can distinguish the noise from actual images of X easily. We will use the Perlin Noise Image (Kenneth T. Co et al., 2018) and the Universal Perturbation Image (SM Moosavi-Dezfooli et al., 2016) as inputs and analysis the Class Activation Map outputs. One thing to notice is that in both of the original papers, Perlin Noise images and Universal Perturbation Images(UAP) were used to add in one original image to make an adversarial example, but here we will just use the raw noise images as inputs. The goal of this experiment is to analysis the reason that noise images can confuse classifiers using the Class Activation Map of these noise images.

Results & Discussion:

Here are the Grad-CAM VGG16 results for the Perlin Noise image and Universal Perturbation Image(UAP) in Figure 6.

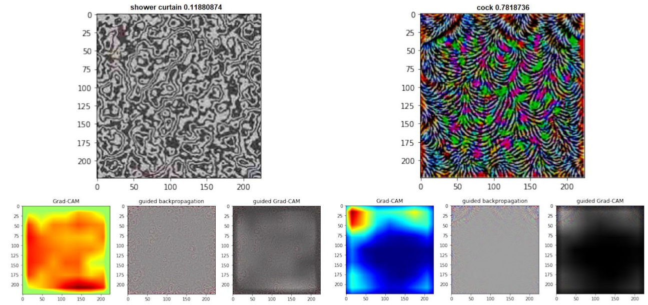


Figure 6. CAM for VGG16 results on Perlin noise(left) and UAP(right).

The result shows that the VGG16 classify the noise image into wrong label, especially for the Universal Perturbation Image that calculated based on the VGG16 model. In general, we believe that since there is no such a label as noise, so the CNN can not classify correctly.

Moreover, our guess is that the classification function is learned by the ImageNet training data and we can imagine that the classification space is drawn by the ImageNet data. However, the noise images are not been seen by the model during the training set and many noise images are very different from all other regular images in the image space (We can image the noise images are sitting in some unique subspace corner, far away from others ImageNet images in the image space). Therefore, the classification function may not work so well on these noise images.

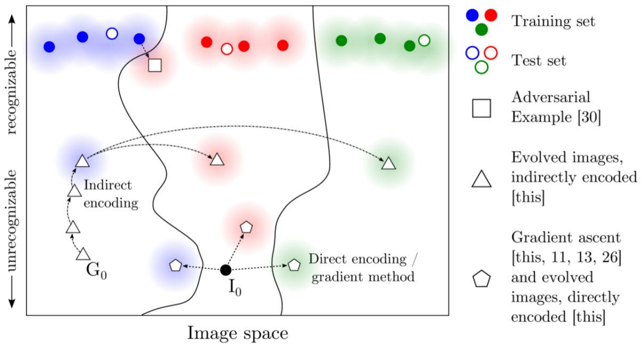

recognizable
unrecognizable
Training set
Test set
Adversarial Example [30]
Evolved images, indirectly encoded [this]
Gradient ascent [this, 11, 13, 26] and evolved images, directly encoded [this]
Indirect encoding
G_0
I_0
Direct encoding / gradient method
Image space

Figure 7. Representation of the image space from Nguyen, A et al., 2014

Similar idea was discussed in Nguyen, A et al., (2014). From this paper, we know that “in a high-dimensional input space, the area a discriminative model allocates to a class may be much larger than the area occupied by training examples for that class (see Figure 7). Synthetic images, such as noise image, far from the decision boundary and deep into a classification region may produce high confidence predictions even though they are far from the natural images in the class.”

For the perlin noise and UAP noise, the classification score seems to be a little high(above 10%) and the individual reason could be following:

From the Grad-CAM result of Perlin noise, we can see that the focus is basically the whole image. From the discussion in Kenneth T. Co et al., (2018), we believe one reason could be that “On a low level, the gradient construction of Perlin noise may have some relation with the gradient descent when neural networks are trained with back propagation. On a higher level, Perlin noise is used for generating realistic and natural-looking images, it captures patterns and characteristics that are associated with real images. Neural networks learn geometric correlations between pixels and features when they are trained; Perlin noise exploits this by generating and subtly adding these realistic features or patterns that allow it to mislead the classifier.”

From the Grad-CAM result of UAP, we can see that the focus is mainly on the top left corner and the classifier outputs a cock label with 78% confidence. From the discussion in SM Moosavi-Dezfooli et al. (2016), this attack “suggests that the universal perturbation exploits some geometric correlations between different parts of the decision boundary of the classifier. More precisely, our data suggests the existence of a subspace S of low dimension d that contains most normal vectors to the decision boundary in regions surrounding natural images. They hypothesize that the existence of universal perturbations fooling most natural images is partly due to the existence of such a low-dimensional subspace that captures the correlations among different regions of the decision boundary. In fact, this subspace “collects” normals to the decision boundary in different regions, and perturbations belonging to this subspace are therefore likely to fool datapoints.”

Conclusion

By now, it must be evident that our two parts of experiment, in very different ways, allow us to evaluate and understand the CNN predictions by visualizing the most salient regions on an image that contribute most prominently towards the top 1 predictions.

The first part allows us to vary the information we provide with respect to the same input to the same CNN and allows us to visualize how this variance in occlusion impacts the certitute with which our CNN makes predictions. The second experiment keeps the input the same but attempts to suss out the internal representations generated through the convolutional layers to divine the correct answer. This allows us to also see the computational process through which the CNN arrives at a particular answer.

Further, we investigate the reasons behind the CNN incorrect predictions given irregular inputs. For adversarial examples, we observed a shift of attention between original image and adversarial images. We also cited three plausible explanations for why adversarial examples can fool CNN. For cross-model comparison, we believe that different model have different resistance toward same attack. Lastly, we observed that mistake of CNN prediction on noise image and we derive a hypothesis based on the discussion of Nguyen, A et al., 2014.

Acknowledgments

Here we present our gratitude the guidance of Professor Anna Rafferty and the Carleton students who contributed to our collection of human attention data.

We also give thanks to the authors of open-source code: https://github.com/rodgzilla/machine_learning_adversarial_examples
https://github.com/insikk/Grad-CAM-tensorflow

References

Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., & Batra, D. (2017, October). Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization. In *ICCV* (pp. 618-626).

Zeiler, M. D., & Fergus, R. (2014, September). Visualizing and understanding convolutional networks. In *European conference on computer vision* (pp. 818-833). Springer, Cham

Co, K. T., Muñoz-González, L., & Lupu, E. C. (2018). Procedural Noise Adversarial Examples for Black-Box Attacks on Deep Neural Networks. arXiv preprint arXiv:1810.00470.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778).

Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.

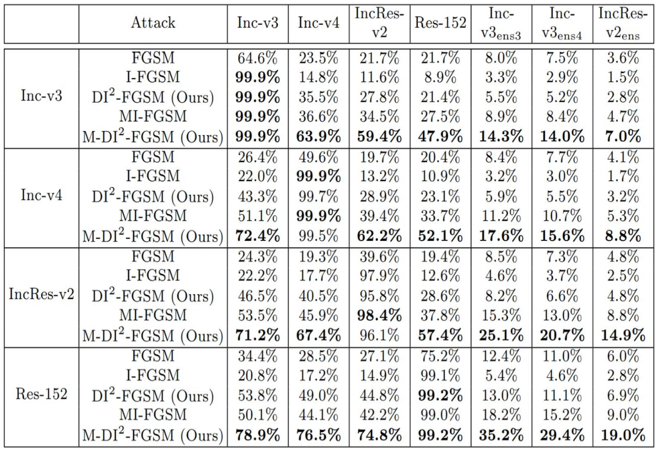
Goodfellow, I. J., Shlens, J., & Szegedy, C. Explaining and harnessing adversarial examples. arXiv preprint arXiv:1412.6572.

Moosavi-Dezfooli, Seyed-Mohsen, et al. "Universal adversarial perturbations." arXiv preprint (2017).

Nguyen, A., Yosinski, J., & Clune, J. (2015). Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 427-436).

Sara Hooker, Dumitru Erhan et al. "Evaluating Feature Importance Estimates." arXiv preprint arXiv:1806.10758

Appendix



	Attack	Inc-v3	Inc-v4	IncRes-v2	Res-152	Inc-$v3_{ens3}$	Inc-$v3_{ens4}$	IncRes-$v2_{ens}$
Inc-v3	FGSM	64.6%	23.5%	21.7%	21.7%	8.0%	7.5%	3.6%
	I-FGSM	**99.9%**	14.8%	11.6%	8.9%	3.3%	2.9%	1.5%
	DI^2-FGSM (Ours)	**99.9%**	35.5%	27.8%	21.4%	5.5%	5.2%	2.8%
	MI-FGSM	**99.9%**	36.6%	34.5%	27.5%	8.9%	8.4%	4.7%
	M-DI^2-FGSM (Ours)	**99.9%**	**63.9%**	**59.4%**	**47.9%**	**14.3%**	**14.0%**	**7.0%**
Inc-v4	FGSM	26.4%	49.6%	19.7%	20.4%	8.4%	7.7%	4.1%
	I-FGSM	22.0%	**99.9%**	13.2%	10.9%	3.2%	3.0%	1.7%
	DI^2-FGSM (Ours)	43.3%	99.7%	28.9%	23.1%	5.9%	5.5%	3.2%
	MI-FGSM	51.1%	**99.9%**	39.4%	33.7%	11.2%	10.7%	5.3%
	M-DI^2-FGSM (Ours)	**72.4%**	99.5%	**62.2%**	**52.1%**	**17.6%**	**15.6%**	**8.8%**
IncRes-v2	FGSM	24.3%	19.3%	39.6%	19.4%	8.5%	7.3%	4.8%
	I-FGSM	22.2%	17.7%	97.9%	12.6%	4.6%	3.7%	2.5%
	DI^2-FGSM (Ours)	46.5%	40.5%	95.8%	28.6%	8.2%	6.6%	4.8%
	MI-FGSM	53.5%	45.9%	**98.4%**	37.8%	15.3%	13.0%	8.8%
	M-DI^2-FGSM (Ours)	**71.2%**	**67.4%**	96.1%	**57.4%**	**25.1%**	**20.7%**	**14.9%**
Res-152	FGSM	34.4%	28.5%	27.1%	75.2%	12.4%	11.0%	6.0%
	I-FGSM	20.8%	17.2%	14.9%	99.1%	5.4%	4.6%	2.8%
	DI^2-FGSM (Ours)	53.8%	49.0%	44.8%	**99.2%**	13.0%	11.1%	6.9%
	MI-FGSM	50.1%	44.1%	42.2%	99.0%	18.2%	15.2%	9.0%
	M-DI^2-FGSM (Ours)	**78.9%**	**76.5%**	**74.8%**	**99.2%**	**35.2%**	**29.4%**	**19.0%**

Table 1.The success rates on seven networks using different method based on a single network from Cihang Xie et al. 2018.

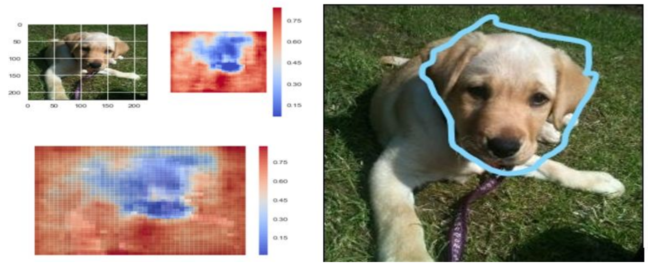


Figure 8: The heat map from occlusion experiment with the superimposed image and the aggregate image for human data.

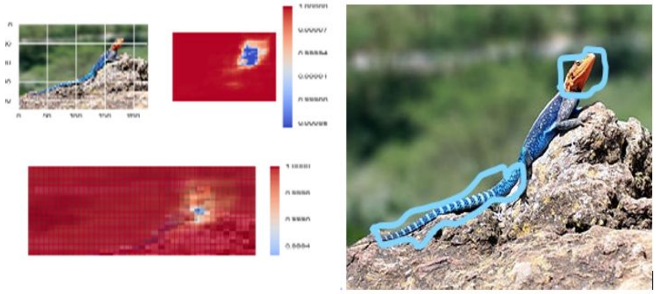


Figure 9: The heat map from occlusion experiment with the superimposed image and the aggregate image for human data.

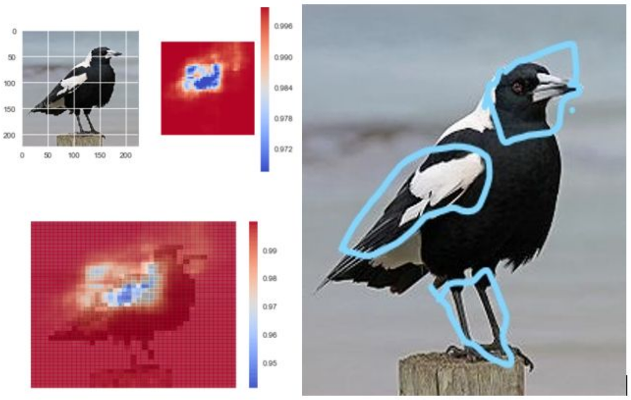


Figure 10: The heat map from occlusion experiment with the superimposed image and the aggregate image for human data.