Foveation-based Mechanisms Alleviate Adversarial Examples of Deep Neural Networks

Yan Luo[1] **Xavier Boix**[1,2] **Gemma Roig**[2] **Tomaso Poggio**[2] **Qi Zhao**[1]
[1] Department of Electrical and Computer Engineering, National University of Singapore, Singapore
[2] CBMM, Massachusetts Institute of Technology & Istituto Italiano di Tecnologia, Cambridge, MA

Abstract

Adversarial examples are images with visually imperceptible perturbations that result in Deep Neural Networks (DNNs) fail. In this paper, we show that DNNs can recover a similar level of accuracy prior to adding the adversarial perturbation, by changing the image region in which the DNN is applied. This change of the input image region is what we call *foveation mechanism*. There are many foveations that can alleviate an adversarial example, *e.g.,* shifting or shrinking few pixels in any direction the image region in which the DNN is applied. This is effective because an adversarial example only affects the foveation for which it has been generated, and a very small subset of similar foveations. Our experiments in ImageNet using state-of-the-art DNNs, namely, AlexNet, GoogleNet and VGG, demonstrate that the effectiveness of the foveation mechanisms is because the DNNs are robust to the scale and location transformations of the object induced by the foveation, but this robustness does not generalize to the adversarial perturbations.

1 Introduction

The impressive generalization properties of Deep Neural Networks (DNNs) have been recently questioned. Szegedy *et al.* showed that DNNs fail to predict the object category in an image when some perturbations are added to the image, and surprisingly, these perturbations can be so small that are imperceptible to humans [1]. The perturbed images are the so-called *adversarial examples*.

Adversarial examples exist because, hypothetically, DNNs act as a high-dimensional linear classifier of the image [2]. The perturbation may be aligned with the linear classifier to counteract the correct prediction of the DNN, and since the dimensionality of the classifier is high, as it is equal to the number of pixels in the image, the perturbation can be spread among the many dimensions and make the perturbation of each pixel small, *c.f.* [2, 3]. This hypothesis of DNNs acting too linearly is supported by quantitative experiments in datasets where the target object is always centered and fixed to the same scale, namely, MNIST [4] and CIFAR [5].

In this paper, we further develop this hypothesis by analyzing adversarial examples using several DNN architectures for ImageNet, namely, AlexNet [6], GoogLeNet [7] and VGG [8], which do not assume a positioning and scale of the object as in previous works. To analyze the effects of the position of the object, we analyze the relationship between the region of the image in which the DNN is applied and the adversarial example. We call foveation mechanism the procedure to select the image region to apply the DNN for object recognition. Adversarial examples are generated for a specific foveation. In previous works, the adversarial examples are generated for a foveation of the whole image. We analyze the effect of changing the foveation in the DNNs for ImageNet.

Our experiments show that an adversarial example only affects the foveation for which it has been generated, and a very small subset of very similar foveations. There is a large number of foveations that are not affected by an adversarial example. For example, changing the foveation used to generate

29th Conference on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain.

the adversarial example by shifting or shrinking the foveation few pixels, is sufficient to obtain an accuracy close to the accuracy of the DNN before adding the adversarial perturbation.

The reason that explains why foveations alleviate adversarial examples, involves an observation that has never been reported before, about the properties of DNNs under changes of scale and translation of the objects. This observation is supported by a series of experiments, which also discard other possible intuitive explanations about why foveations alleviate adversarial examples, *e.g.,* a foveation can remove clutter, or a foveation removes part of the adversarial perturbation.

Our experiments show that the adversarial perturbation is not robust to the image transformation induced by the foveation. From the aforementioned linearity hypothesis by Goodfellow *et al.* [2], it follows that after foveating on an object from an adversarial example, the output of the DNN can be decomposed into two additive terms, one that comes from the foveated object without perturbation, *i.e.,* the object from the "clean" image, and another term from the foveated perturbation. Our experiments demonstrate that the robustness of the DNN to the scale and translation transformations by the foveation only applies to the term of the "clean" image, and not to the term of the adversarial perturbation. This is because DNNs for ImageNet are trained to be robust to scale and position transformations of the object, but not to transformations of perturbations.

Finally, we show that a foveation mechanism that has a random component to select the region to foveate, could be used in any DNN to alleviate adversarial examples. This is because there are many possible foveations that are not affected by an adversarial example (our experiments show that there are thousands of them), and it is very unlikely that the randomly chosen foveation is affected by the adversarial example, as this only affects a small subset of foveations. Note that this mechanism can be used without any additional cost together with the methods presented in previous works that alleviate the adversarial examples by re-training the DNNs [9, 2].

2 Foveation-based Mechanisms for Adversarial Examples

In this section, we introduce why adversarial examples can be alleviated with a foveation mechanism. We first review the current hypothesis in the literature that shows that adversarial examples are a consequence of DNNs acting as a high-dimensional linear classifier. This will serve as the basis to afterwards introduce our explanation.

2.1 Revisiting the High-dimensional Linearity Hypothesis

Let $\mathbf{x}$ be an image that contains an object whose object category is of class label ℓ. We denote as $f(\mathbf{x})$ the mapping of the DNN from the input image to the classification scores, and we use ϵ to denote a perturbation that produces a misclassification when added to the input image, *i.e.,* $\ell \notin C(f(\mathbf{x}+\epsilon))$ in which $C(\cdot)$ maps the classification scores to class labels. The set of all perturbations that produce a misclassification is denoted as $\mathcal{E}_{\mathbf{x}} = \{\epsilon |\ \ell \notin C(f(\mathbf{x}+\epsilon))\}$, and the perturbation with minimal norm in this set is defined as $\epsilon^\star$, *i.e.,* $\epsilon^\star = \arg\min_{\epsilon \in \mathcal{E}_{\mathbf{x}}} \|\epsilon\|$. Many of the perturbations in $\mathcal{E}_{\mathbf{x}}$ may be imperceptible.

Goodfellow *et al.* [2] showed that small perturbations may affect the classifier because, hypothetically, DNNs act as a high-dimensional linear classifier, of the same dimensionality as the image size. The learned parameters of the DNN cause that the DNN bypasses the non-linearities in the architecture, and the DNN acts as a linear classifier. Thus, $f(\mathbf{x}+\epsilon^\star)$ could be rewritten as $\mathbf{w}'\mathbf{x} + \mathbf{w}'\epsilon^\star$, where $\mathbf{w}'$ is the transpose of a high-dimensional linear classifier, which yields a classifier approximately equivalent to $f(\cdot)$. Then, the high-dimensionality of the linear classifier allows that $\mathbf{w}'\epsilon^\star$ could be a high value, but the average per pixel value of $\epsilon^\star$ is low, as the perturbation $\epsilon^\star$ can be spread among the many dimensions of the classifier (the image pixels), and make $\epsilon^\star$ imperceptible [2, 3].

A direct consequence of the linearity of the DNN is that multiplying the adversarial perturbation by a constant value larger than 1 also produces misclassification, since $c\mathbf{w}'\epsilon^\star$ is a factor c times larger than before. Thus, the set of adversarial examples, $\mathcal{E}_{\mathbf{x}}$, is at least a dense set of perturbations in the direction of $\epsilon^\star$. Another phenomenon that can be explained through the linearity of the DNN is that adversarial examples generated for a specific DNN model produce misclassification in other DNNs [1]. This is because the different DNNs are similar among them as they all act as a linear classifier.

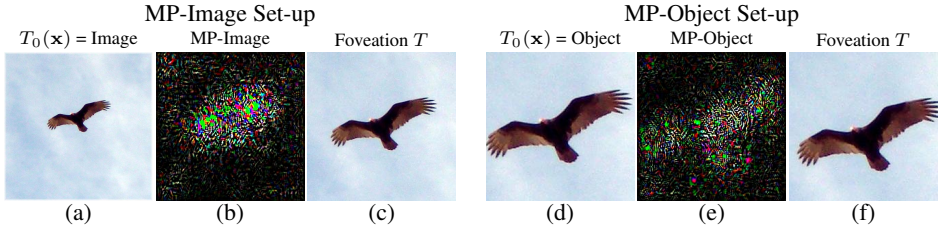

Figure 1: *Example of the two set-ups with AlexNet with BFGS perturbation.* (a) Foveation for *MP-Image*, (b) adversarial perturbation *MP-Image*, (c) *Saliency Crop MP-Image*; (d) foveation for *MP-Object*, (e) adversarial perturbation *MP-Object*, and (f) 1 *Shift MP-Object*.

2.2 Foveation Mechanisms Alleviate Adversarial Examples

We define a foveation as the transformation of the image to select the image region to apply the DNN. The foveation provides an input to the DNN that always has the same size, even if the size of the selected region varies. Note that the foveation discards the information from the image regions that are not selected. Without loss of generality, in this paper, we use as the foveation mechanism a crop of a region that includes most of the object or the whole object. In the experiments section, we use different instances of this foveation mechanism, which we detail later.

Our definition of foveation leads to the observation that DNNs that use the whole image as input to the DNN, are using a foveation mechanism that consist on always foveating at the whole image. Even in the simplest case of applying the DNN to the same image region, the DNN uses a foveation mechanism. Thus, *any* DNN for object recognition uses a foveation mechanism, as the DNN needs to be applied in an image region. We summarize this in Observation 1, which in fact, is a tautology given our definition of foveation:

Observation 1. *All DNNs for object recognition use a foveation mechanism to select an image region to apply the DNN.*

Thus, using a foveation mechanism in a DNN does not extend the DNN, because all DNNs already use a foveation mechanism.

Observation 1 shows that all adversarial examples are generated for a foveation of an image region. Let $T(\mathbf{x})$ be the image after the foveation, which is used as the input of the DNN. Also, let $T_0(\mathbf{x})$ be the foveation used to generate the adversarial example, *i.e.,* the adversarial example is the result of $\arg\min_{\epsilon \in \mathcal{E}_{T_0(\mathbf{x})}} \|\epsilon\|$. In previous works, the adversarial example is calculated for a DNN that takes the whole image as input, and hence, the dependency of the adversarial example and T_0 can be omitted, *i.e.,* in previous works $T_0(\mathbf{x})$ is equal to $\mathbf{x}$.

This leads to the question of whether the adversarial perturbation is also effective when a different foveation from $T_0(\mathbf{x})$ is used to evaluate the DNN, *i.e.,* does the adversarial example generated for $T_0(\mathbf{x})$ affect the DNN when the foveation used to evaluate the DNN, $T(\mathbf{x})$, is different from $T_0(\mathbf{x})$? If the answer to this question is negative, we may be able to design a system which is robust to the adversarial examples by using a procedure to select a foveation different from $T_0(\mathbf{x})$.

We have experimentally analyzed this question (the results are reported in the experiments section). We have found that for the DNNs for ImageNet we have tested, adversarial examples clearly do not generalize to different foveations.

Observation 2. *The adversarial examples in the literature only affect the foveation for which it has been generated, T_0, and very similar foveations. The rest of foveations are not affected by this adversarial example.*

Our experiments support Observation 2: A change of T_0 consisting on a small shift or shrink of few pixels produces that the adversarial example generated for T_0 does not negatively affect the DNN. In Fig. 1 we show two examples of adversarial perturbation for two different foveations. Also, the figure shows that different foveations alleviate these adversarial examples.

There could be several intuitive reasons to explain Observation 2, *e.g.,* the foveation T removes part of the adversarial perturbation. Yet, our experiments show that these kind of intuitive arguments

fail to fully explain Observation 2. The explanation is related to a new property that has never been reported, about the robustness of DNNs to changes of scale and translation of the objects in the image.

In order to introduce this property, we assume that the foveation mechanism does not introduce any non-linearity, *e.g.,* it does not modify pixel values from the original image, and the interpolations for re-sizing the image are linear. Also, we assume that the transformation produced by the foveation does not negatively affect the performance of the DNN. This is reasonable if the foveation does not discard the target object. This is because the foveation only produces a change of the position and scale of the object in the input of the DNN, and as it is well-know, the representations learned by the DNNs are robust to these changes of scale and position of the objects.

Given these two mild assumptions we just introduced, and also, that the foveation to generate the adversarial example, T_0, is different from the foveation we use to evaluate the DNN, T, we have that from the linearity hypothesis introduced by [2], $f(T(\mathbf{x}+\epsilon^\star))$ is equal to $\mathbf{w}'T(\mathbf{x}) + \mathbf{w}'T(\epsilon^\star)$. Since the representations learned by the DNNs are robust to changes of scale and position of the objects produced by the foveation, the term without the perturbation after the foveation, $\mathbf{w}'T(\mathbf{x})$, is not expected to have a lower accuracy than before, $\mathbf{w}'T_0(\mathbf{x})$.

Yet, we have found that using a different foveation mechanism from the adversarial example, reduces the alignment between the classifier and the perturbation, $\mathbf{w}'T(\epsilon^\star) \ll \mathbf{w}'T_0(\epsilon^\star)$. Note that DNNs are trained with objects at many different positions and scales, that produce representations robust to the transformations of the object, such as the transformation produced by changing T_0 to T. However, objects are visually very different from the adversarial perturbations, and the robustness of DNNs to transformations of objects does not generalize to the perturbations. This is summarized in the following observation, and validated in the results section.

Observation 3. *The robustness of the DNNs to changes of scale and position of the objects* $(\mathbf{w}'T(\mathbf{x}) \approx \mathbf{w}'T_0(\mathbf{x}))$ *does not generalize to the adversarial perturbations,* i.e., $\mathbf{w}'T(\epsilon^\star) \ll \mathbf{w}'T_0(\epsilon^\star)$.

Note that Observation 2 and 3 suggest that a DNN is robust to the adversarial examples if the algorithm to generate the adversarial example does not know the region of the foveation is which the DNN is applied. The algorithm to generate the adversarial example could try to estimate a foveation T_0 which is similar to the foveation used by the DNN, T. Yet, Observation 2 shows that there are many possible foveations T that are robust to the adversarial example. T could be set by randomly choosing a foveation among all the many possible ones, and as a result, it will be very unlikely that T_0 is similar to T. In the experiments, we show an example of a foveation mechanism that randomly selects a foveation among thousands of them, based on a saliency map.

Previous works to alleviate adversarial perturbations are based on training a DNN that learns to classify adversarial examples generated at the training phase [2, 9]. The foveation mechanism can be used in conjunction with these techniques, and it comes without any additional cost of re-training the DNN.

A remaining question is whether it exist an adversarial example that can affect multiple foveations at the same time. In this paper, we have investigated the properties of adversarial examples from previous works. Yet, it could be that there are other types of adversarial examples that have not been discovered yet, for which Observation 2 does not hold. A hint towards an answer is that the linearity of the DNNs may allow generating a perturbation that affects multiple foveations by summing perturbations that affect the different foveations. Our preliminary experiments show that this adversarial perturbation is not imperceptible, but there may be other more sophisticated procedures than the one we used, that are effective to generate imperceptible perturbation for multiple foveations.

3 Experimental Set-up

In this section we introduce the experimental details and the foveation mechanisms that we use.

We use ImageNet (ILSVRC) [10], which is a large-scale dataset for object classification and detection. We report our results on the validation set of ILSVRC 2012 dataset, which contains $50'000$ images. Each image is labeled with one of the $1'000$ possible object categories, and the ground truth bounding box around the object is also provided. We evaluate the accuracy performance with the standard protocol for image classification [11], taking the top 5 prediction error.

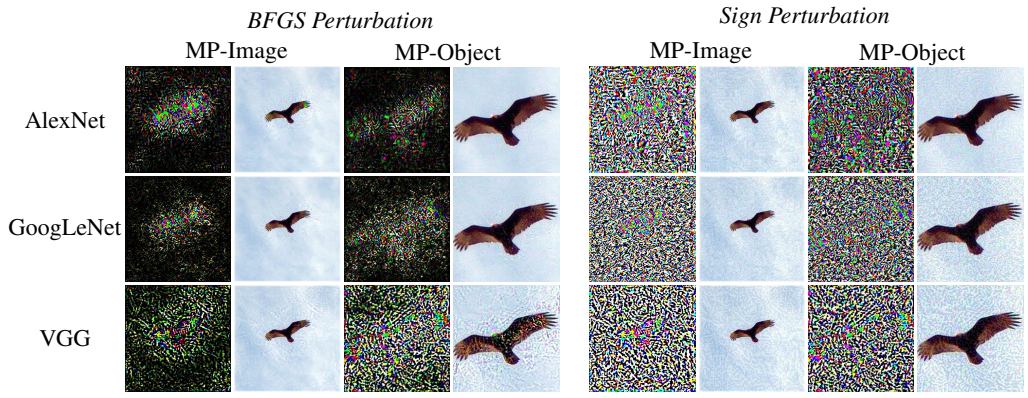

Figure 2: *Example of different DNNs' minimum perturbations.* Each row corresponds to a different DNN (AlexNet [6], GoogLeNet [7] and VGG [8]). The four first columns are for *BFGS*, and the last four columns for *Sign*. MP-Image ($T_0(\mathbf{x})$ = Image) and MP-Object ($T_0(\mathbf{x})$ = Object) is shown for each case. The adversarial example is displayed with the perturbation multiplied by a factor of 10.

We report results using 3 DNN which are representative of the state-of-the-art, namely AlexNet [6], GoogLeNet [7] and VGG [8]. We use the publicly available pre-trained models provided by the Caffe library [12]. We use the whole image as input to the DNN, resized to 227×227 pixels in AlexNet, and 224×224 in VGG and GoogLeNet. In order to improve the accuracy at test time, a commonly used strategy in the literature is to use multiple crops of the input image and then combine their classification scores. We do not use this strategy unless we explicitly mention it.

The adversarial perturbation consist on finding the perturbation with minimum norm that produces misclassification, under the top-5 criterion. This is an NP-hard problem, and the algorithms to compute the minimum perturbation are approximate. To generate the adversarial examples we use the two algorithms available in the literature. Namely, the algorithm by [1], that is based on an optimization with L-BFGS, and we also use the algorithm by [2], that uses the sign of the gradient of the loss function. We call them *BFGS* and *Sign*, respectively. For more details about the generation of the perturbation we refer to sec. 1.1 in the suppl. material, and to [1, 2]. We observed that the vast majority of the adversarial examples are imperceptible (sec. 2.1 in suppl. material).

We use two different foveations $T_0(\mathbf{x})$ to generate the adversarial example. In the first foveation, the adversarial perturbation is generated on the foveation of the whole image, as it has been done in previous works, *i.e.,* $T_0(\mathbf{x}) = \mathbf{x}$. In the second foveation, the adversarial perturbation is computed on the image with the cropped target object. Several qualitative examples of the perturbations are shown in Fig. 2. Observe that for *BFGS*, the perturbation is concentrated on the position of the object, while for *Sign*, the perturbation is spread through the image. Also, note that for each DNN architecture the perturbation is different. In the following, we introduce the foveations T that we use on the adversarial examples generated with the two different foveations $T_0(\mathbf{x})$ we just mentioned. In Fig. 2 in the suppl. material we show examples of the different foveations T.

Adversarial Example for the Foveation of the Whole Image, *i.e.,* $T_0(\mathbf{x})$ equal to the image (MP-Image) This adversarial example is generated for the whole image, $T_0(\mathbf{x}) = \mathbf{x}$, which is the foveation that has been analyzed in previous works. Any of the foveations T we use for this adversarial example may remove clutter, as they will remove part of the background (see Fig. 1a, b and c). We use the following foveations:

- *Object Crop MP-Image:* It crops the target object from the image using the ground truth bounding box. If there are multiple target objects, we crop each of them, and average the classification scores. Note that this foveation mechanism does not remove any part of the object, and it removes most of the clutter. This is the only foveation that guarantees that all the clutter is removed, and also, they may remove part of the target object.

- *Saliency Crop MP-Image:* This foveation is based on using a state-of-the-art saliency model to select 3 regions to crop from the most salient locations of the image. The crops are generated selecting three centroids of the saliency map, and the size of the region of the foveation is randomly choose to be between 60% and 100% of the image. This randomness is to show as a proof-of-concept that

Table 1: *Evaluation of the Foveation Mechanisms for BFGS.* Quantitative results of the top-5 accuracy with minimum perturbation. See suppl. material for the results with *Sign*.

	$T_0(\mathbf{x})$ = Image				$T_0(\mathbf{x})$ = Object			
	w/o MP-Image	MP-Image	Object Crop MP-Image	Saliency Crop MP-Image	w/o MP-Object	MP-Object	1 Shift MP-Object	Embedded MP-Object
ALX	78.4	0.5	78.0	72.4	81.9	1.8	76.8	75.1
GNT	87.4	1.0	83.1	80.7	89.4	2.8	83.3	84.2
VGG	85.4	0.6	82.6	78.8	91.2	3.1	81.5	78.1

a foveation mechanism that can not be predicted during the generation of the adversarial example, is effective to alleviate adversarial examples. We use the SALICON saliency map, which extracts the saliency map using the same DNNs we test [13]. We observed that these saliency maps are robust to the adversarial perturbations, since the adversarial examples are generated to produce misclassification for object recognition, but not to affect the saliency prediction. The classification accuracy is computed by averaging the confidences from the multiple crops.

- 10 *Crop MP-Image:* Another foveation strategy we evaluate is the 10 crops that is implemented in the Caffe library to boost the accuracy of the DNNs [12]. The crops are done on large regions of the image, which in most cases the target object is contained in the crops (each crop discards about 21% of the area of the image). The crops are always of the same size, and there are 5 of them (4 clamped at each corner of the image, and one in the center of the image). The images resulting from these 5 crops are flipped, which makes a total of 10 crops.

- 3 *Crop MP-Image:* This is the same as 10 *Crop MP* but only with 3 random crops selected among the 10 crops. It can be used for a fair comparison between *Saliency Crop MP-Image* with 10 *Crop MP-Image*, and to evaluate how much improvement comes from averaging multiple foveations.

Adversarial Example for the Foveation of the Target Object, *i.e.,* $T_0(\mathbf{x})$ equal to the Object (MP-Object) This adversarial example is generated to affect the foveation of cropping the target object with the ground truth bounding box (denoted as *MP-Object*, see Fig. 1d, e and f). Note that *MP-Object* is the adversarial example for an image with almost no clutter, and hence, any of the foveations mechanism we use do not remove clutter:

- *Embedded MP-Object:* It consists on embedding the image of the cropped object with the perturbation to the full image. Note that this foveation adds the clutter back to the image.

- 10 *Shift MP-Object:* It is based on the 10 crops of Caffe we introduced before for 10 *Crop MP*. In this case, each of the 10 crops is a shifted version of the target object, that removes part of the object and uses part of the background to do the crop (the foveation shifts a bit to the background, and it yields a cropped region with 12% of background, as shown in Fig. 1f). We set the size of the crops such that they do not modify the original scale of the object.

- 1 *Shift MP-Object:* It consists on selecting 1 random crop from the 10 shifts of 10 *Shift MP-Object*.

4 Results

In this section, we show experimental evidence that supports the Observations 2 and 3, and we demonstrate that foveations can be used to greatly alleviate the effect of the adversarial examples.

4.1 Evaluation of Observation 2

Table 1 (complete results in Table 1 in suppl. material) shows the accuracy of the adversarial examples, and also, it shows the accuracy after the foveation T. We can see that the adversarial example affects both MP-Image and MP-Object by producing an accuracy close to 0%. It is not exactly 0% because in some images the minimum norm of the perturbation is out of the range of the line search of the algorithm to calculate *BFGS* and *Sign* perturbations. We can also see that the accuracy for any different foveation from T_0 is almost the same as the accuracy without the adversarial perturbation, for both MP-Image and MP-Object. This result supports Observation 2, and it suggests that foveations are a powerful mechanism to alleviate the adversarial examples (note that it improves the accuracy from 0% to more than 70% for both MP-Image and MP-Object). *Object Crop MP-Image* produces

the biggest improvement over all foveation mechanisms, probably because it is the only foveation mechanism that guarantees that parts of the target object are not removed, and at the same time it removes the clutter.

In Fig. 3 and Table 2 both in the suppl. material, we extend the result in Table 1 by reporting the accuracy using different values of the norm of the perturbation (the perturbation has a fixed norm for all images). Similar conclusions can be extracted as in Table 1.

In order to provide further evidence to support Observation 2, in Fig. 3a (complete results in Fig. 4 in suppl. material), we provide the number of correctly classified images when we shrink the foveation of MP-Object a certain number of pixels. We can see that by simply shrinking 3 pixels, the accuracy almost completely hits the upper bound of the accuracy. We have observed the same for small shift of the foveation in any direction.

Recall that *Saliency Crop* randomly selects the region for the foveation based on a saliency map. The results we just showed, suggest that *Saliency Crop* could be robust to adversarial examples in general, because the algorithm to generate an adversarial example ignores which foveation is used due to the randomness in *Saliency Crop*. Since the result in Fig. 3a shows that a foveation that differs about 3 pixels from T_0 can alleviate the adversarial perturbation, the number of successful foveations is of the order of thousands (the number of foveations with a difference of 3 pixels in an image of 224×224 is about $5'600$, and *Saliency Crop* randomly select among a pool of 40% of these foveations). Thus, the probability that the algorithm that generates the adversarial example correctly guesses the foveation used by *Saliency Crop* is of the order of one divided by a thousand.

4.2 Evaluation of Observation 3

We now test the Observation 3 by analyzing the effect of the foveation to the classification scores of the adversarial perturbation and the image without the perturbation. Namely, we evaluate $f(T_0(\mathbf{x})) - f(T(\mathbf{x}))$ (the difference between the classification score after changing the foveation T_0 to T, for the image without perturbation), and $(f(T_0(\mathbf{x}+\epsilon^\star)) - f(T_0(\mathbf{x}))) - (f(T(\mathbf{x}+\epsilon^\star)) - f(T(\mathbf{x})))$ (the same as the previous term but for the adversarial perturbation). We do so for the classification score of the ground truth object category (without the last soft-max layer), and for both MP-Image and MP-Object. In Fig. 3b (complete results in Fig. 5 in suppl. material), we show the cumulative histogram of the number of images with a change of the classification score smaller than the indicated in the horizontal axis in the figure. We can see that the alignment between the perturbation and the classifier decreases after the foveation T, and that the term of the image without perturbation is not affected as much as the term of the perturbation (*BFGS* and *Sign* are compared independently). This result strongly supports Observation 3.

We now analyze other possible reasons that can also explain why foveations alleviate adversarial examples. We found that besides Observation 3, none of these reasons could provide a clearcut explanation of Observation 2.

Foveations remove clutter In Table 1 (complete results in Table 1 in suppl. material), we evaluate the improvement in the accuracy when the foveations remove clutter, by comparing the accuracy without adversarial perturbation for MP-Image and MP-Object (compare w/o MP-Image and w/o MP-Object). We can see that the foveation of the object slightly improves the accuracy in the absence of adversarial perturbation (about 5%). This suggests that removing clutter always improve the accuracy on average, independently of the Observation 3. However, note that the improvement from removing clutter can not be the main reason to explain Observation 2, since the accuracy of *MP-Object*, in which there is no clutter, significantly improves after adding back the clutter (*Embedded MP-Object*). Also, this improvement is almost the same as for the foveations that remove clutter in the *MP-Image* set-up.

Average of the classification score from multiple foveations Similarly, in the suppl. material in Fig. 3 and Table 2 and 3, we can see that the foveation slightly improves the accuracy when it averages the classification score from multiple foveations (see results for 10 *Crop MP-Image* and 10 *Sift MP-Object*). This suggests that averaging multiple foveations always improve the accuracy, independently of the Observation 3. Yet, we can see that this improvement of the accuracy is much smaller than the effect from Observation 3, by comparing the accuracy of 1 and 10 *Shift MP-Object*, and the accuracy of 1 and 10 *Crop MP-Image*.

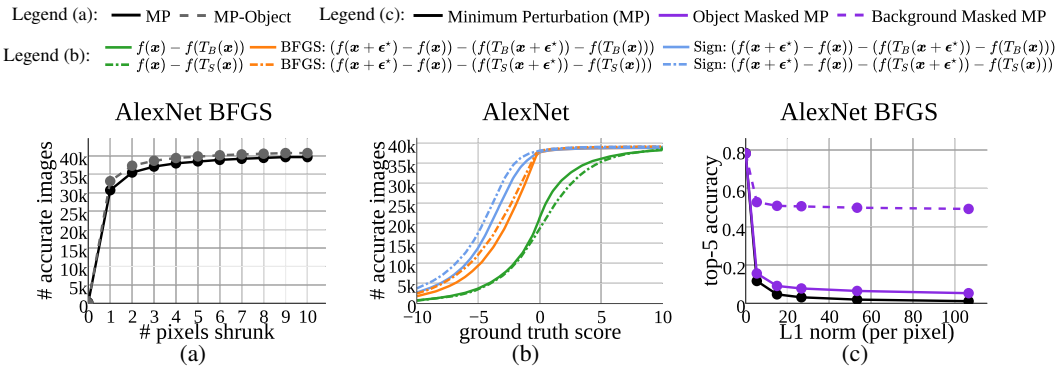

Figure 3: *Effect of the Foveation to the Perturbations.* See suppl. material for results in other DNN and *Sign*. (a) Accuracy after shrinking the foveation of the adversarial example, for AlexNet and *BFGS*, only the images that are correctly classified by the DNN are included; (b) Cumulative histogram of the number of images with a change of classification score smaller than the indicated in the horizontal axis. $T_B(\cdot)$ is the foveation with *Object Crop MP*, and $T_S(\cdot)$ is the foveation with 1 *Shift MP-Object*; (c) Accuracy of the masked perturbations for the AlexNet DNN, when varying the norm of *BFGS*.

Foveations remove part of the perturbation We now analyze the effect of removing part of the perturbation from different regions of the image. We report the accuracy that yields MP-Image when it is masked to be positioned only on the target object, or when it masked to be only on the background. Specifically, we create an image mask using the ground truth bounding box (denoted as *Object Masked* in Fig. 3), where the mask's pixels are 1 inside the bounding box, and 0 otherwise. Before MP-Image is added to the image to produce the adversarial example, we multiply (pixel-wise) the perturbation by the mask, such that the resulting perturbation is only positioned inside the bounding box. To analyze the perturbation that is in the background, we also create the inverted mask, denoted as *Background Masked*, in which the pixels outside the bounding box are 1, and 0 for the pixels inside.

In Fig. 3c (complete results in Fig. 6 in suppl. material), we report the accuracy performance for *BFGS* for *Object Masked* and *Background Masked*, and also for MP-Image, for different norms of the perturbation. We observe that the accuracy performance for the perturbation positioned only on the object, decreases in a similar way as *MP-Image*. When adding the perturbation only on the background, the accuracy decreases significantly less than for *Object Masked* (there is a difference between them of about 40% of accuracy). Note that this result is also clear for small values of the perturbation norm (which is the norm for the majority of minimal perturbations as shown in Fig. 1 in suppl. material). The result of this experiment suggests that removing or adding perturbation on the background does not affect the accuracy performance, and hence, the foveation mechanisms can not benefit from this.

5 Conclusions

We have shown that the transformation of the image produced by a foveation decreases the effect of the adversarial perturbation to the classification scores, because the robustness of the DNNs to transformations of objects does not generalize to the perturbations. This suggests that a system similar to [14], which integrates information from multiple fixations for object recognition, may be more robust to the phenomenon of the adversarial examples. Note that this system, is more similar to human vision than current DNN architectures. Yet, a remaining puzzle is the robustness of human perception to the perturbations. Some hints towards an answer could be that humans fixate their eyes on salient regions, and the eccentricity dependent resolution of the retina help eliminate the background outside the eye fixation.

References

[1] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. J. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," in *ICLR*, 2014.

[2] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in *ICLR*, 2015.

[3] A. Fawzi, O. Fawzi, and P. Frossard, "Fundamental limits on adversarial robustness," in *ICML*, 2015.

[4] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[5] A. Krizhevsky, "Learning multiple layers of features from tiny images," *Tech. Rep*, 2009.

[6] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *NIPS*, 2012.

[7] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *CVPR*, 2015.

[8] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *ICLR*, 2015.

[9] S. Gu and L. Rigazio, "Towards deep neural network architectures robust to adversarial examples," in *ICLR*, 2015.

[10] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *CVPR*, 2009.

[11] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. S. Bernstein, A. C. Berg, and L. Fei-Fei, "Imagenet large scale visual recognition challenge," *IJCV*, 2015.

[12] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," in *MM*, 2014.

[13] X. Huang, C. Shen, X. Boix, and Q. Zhao, "SALICON: Reducing the semantic gap in saliency prediction by adapting neural networks," in *ICCV*, 2015.

[14] V. Mnih, N. Heess, A. Graves, and K. Kavukcuoglu, "Recurrent models of visual attention," in *NIPS*, 2014.

1 Experimental Set-up

1.1 Generation of the Adversarial Perturbation

Before introducing the generation of the perturbation, we introduce a more specific mathematical notation, that later will serve to clarify the details about the perturbation generation. Recall that $\mathbf{x} \in \mathbb{R}^N$ is an image of size N pixels. This image contains an object whose object category is of class label $\ell \in \{1 \dots k\}$, where k is the number of object categories (*e.g.,* $k = 1000$ in ImageNet). $f : \mathbb{R}^N \rightarrow \mathbb{R}^k$ is the mapping from the input image to the classification scores returned by the CNN. Note that $f(\mathbf{x}) \in \mathbb{R}^k$ is a vector that contains the classification scores for all object categories. We use $e(f(\mathbf{x}), \ell)$ to denote a function to evaluate the error of the classification scores, where ℓ is the label of the ground truth object category. In ImageNet, this error function is based on the top-5 scores [11]. Thus, $e(f(\mathbf{x}), l)$ returns 0 when ℓ corresponds to one of the five highest scores in $f(\mathbf{x})$, otherwise $e(f(\mathbf{x}), l)$ returns 1.

We use $\epsilon \in \mathbb{R}^N$ to denote the perturbation image that produce a misclassification when added to the input image, *i.e.,* $e(f(\mathbf{x} + \epsilon), l) = 1$. The image $\mathbf{x} + \epsilon$ is the *adversarial example*. The set of all perturbation images that produce a misclassification of an image can be grouped together. We use $\mathcal{E}_{\mathbf{x}} \subset \mathbb{R}^N$ to denote such set, $\mathcal{E}_{\mathbf{x}} = \left\{\epsilon \in \mathbb{R}^N |\ e(f(\mathbf{x} + \epsilon), l) = 1\right\}$. Then, we define $\epsilon^\star$ as the perturbation with minimal norm to produce misclassification of the image, *i.e.,* $\epsilon^\star = \arg\min_{\epsilon \in \mathcal{E}_{\mathbf{x}}} \|\epsilon\|$. Observe that the optimal perturbation depends on the norm we choose to minimize. We will analyze the L_1 and L_∞ norms, since the perturbations we analyze optimize one of these two norms.

BFGS perturbation. As in [1], we approximate the perturbation $\epsilon^\star$ by using a box-constrained L-BFGS [1]. It consists on minimizing the L_1 norm of the perturbation, $\|\epsilon\|_1$, that produces a misclassification of the image using the top-5 accuracy criteria, *i.e.,* $e(f(\mathbf{x} + \epsilon), \ell) = 1$. To do so, we minimize the L_1 norm plus a loss function, which we denote as $\text{loss}(\epsilon, \{\mathbf{x}, \ell\})$. Let η be a constant that weights the L_1 norm with respect to the loss function.

Since the minimization should produce a misclassification, the loss function is based on the accuracy. Thus, the loss could be equal to $(1 - e(f(\mathbf{x} + \epsilon), \ell))$. Since directly minimizing the top-5 accuracy is difficult, because the derivative of $e(f(\mathbf{x} + \epsilon)$ is 0 for any ϵ except in the boundary of producing a misclassification, we use a hinge loss. Thus, $\text{loss}(\epsilon, \{\mathbf{x}, \ell\})$ is equal to 0 when the image is misclassified (which corresponds to the final objective), otherwise the loss function takes the value of the classification score of the class that we aim to misclassify, which can be expressed as $I(\ell)^T f(\mathbf{x}+\epsilon)$ where $I(\ell) \in \mathbb{R}^k$ is an indicator vector that is 1 in the entry corresponding to the class ℓ and 0 otherwise.

The minimization of $\eta\|\epsilon\|_1 + \text{loss}(\epsilon, \{\mathbf{x}, \ell\})$ is done with L-BFGS. This uses the gradient of the loss with respect to ϵ, which can be computed using the back-propagation routines used during training of the CNN. In order to further minimize the norm returned by L-BFGS, we do a line search of the norm given the perturbation by L-BFGS, *i.e.,* $\tilde{\epsilon} = \alpha\epsilon/\|\epsilon\|$, where α is a scalar factor.

In the experiments, we set $\eta = 10^{-6}$ because using this constant L-BFGS could find a perturbation that produces a misclassification in the majority of the images, except for approximately the 0.5% of the images. In all the CNNs tested, these images had a classification score higher than 0.9. To obtain a perturbation that produces a misclassification in these images, we apply the line-search method with the perturbation returned after stopping L-BFGS after one iteration.

Sign perturbation. It was introduced in [2]. The perturbation of the *Sign perturbation* is generated using $\text{sign}(\nabla\text{loss}(\epsilon, \{\mathbf{x}, \ell\}))$, which can be computed using back-propagation. Then, we use the line-search method to minimize the norm of the perturbation to misclassify all images.

1.2 When are Adversarial Perturbations Perceptible?

The perceptibility of adversarial examples is subjective. In Fig. 10, 12 and 14, we show examples of the same adversarial perturbation varying the L_1 norm per pixel, and the L_∞ norm, in Fig. 11, 13, 15. We can see that from a certain factor, the perturbation becomes clearly perceptible, and it occludes parts of the target object. This helps us approximately determine at what point the adversarial perturbations become perceptible. For *BFGS*, we can say that when the L_1 norm per pixel is higher

[1] `http://www.cs.ubc.ca/~schmidtm/Software/minFunc.html`

than 15 the perturbation becomes visible, and for L_∞ the threshold is 100. This difference is because the density of *BFGS* is not the same through all image, it is mainly in the same position of the object. For *Sign*, the threshold for both norms to make the perturbation visible is about 15, and it is the same for both norms because this perturbation is spread evenly through all the image. We use these values for the analysis.

In Fig. 4, we report the histogram of the norm of the different perturbations that produce misclassification. As we can see, for the vast majority of the images, there is an imperceptible adversarial perturbation.

1.3 Foveation Mechanisms Set-ups

In Fig. 5, we show an illustrative example of different foveation mechanisms introduced in the paper.

2 Results

In this supplementary material, we provide the complete results from the paper, which include *BFGS* and *Sign*, and also, the different DNNs for ImageNet. The conclusions extracted from these results are the same as the results reported in the paper.

2.1 Evaluation of Observation 2

Table 2 shows the accuracy for the foveation for which the perturbation is generated, T_0, and also, the accuracy after a different foveation is applied.

In Fig. 6, we report the accuracy using different values of the norm of the perturbation (the perturbation has a fixed norm for all images). In Table 3, we provide the numerical values for different norm of the perturbation. We can observe that when the perturbations are imperceptible or almost imperceptible, all the foveation mechanisms we introduce improve the accuracy between 30% to 40%, which further confirms that foveations considerably alleviate adversarial examples. In addition, these results also show that when we increase the value of the norm of the perturbation, the accuracy decreases. This is because large perturbations may bring the DNN to the non-linear region, and also, the perturbation is occluding the object rather than acting as an adversarial perturbation. In Table 3, we report the increase of the norm of the adversarial perturbation after the foveation T. Note that after the foveation the norm of the perturbation has to substantially increase to produce misclassification (about 5 times for *BFGS*, and between 5 to 8 times for *Sign*). Note that these conclusions as for Table 2, but they can not be directly compared, because in Table 2 the norm of the perturbation is the minimum norm for each image, and in Fig. 6 and Table 3 the perturbation in each image has the same norm.

In Table 4 we report the increase of the norm of the adversarial perturbation after the foveation. Note that after the foveation the norm of the perturbation has to substantially increase to produce misclassification (about 5 times for *BFGS*, and between 5 to 8 times for *Sign*).

Finally, in Fig. 7, we provide the number of correctly classified images when we shrink the foveation of MP-Object a certain number of pixels.

2.2 Evaluation of Observation 3

In Fig. 8, we show the cumulative histogram of the number of images with a change of the classification score smaller than the indicated in the horizontal axis in the figure. We can see that the alignment between the perturbation and the classifier decreases after the foveation T, and that the term of the image without perturbation is not affected as much as the term of the perturbation (*BFGS* and *Sign* are compared independently).

In Fig. 9 we report the accuracy performance for *BFGS* for *Object Masked* and *Background Masked*, and also for MP-Image, for different norms of the perturbation.

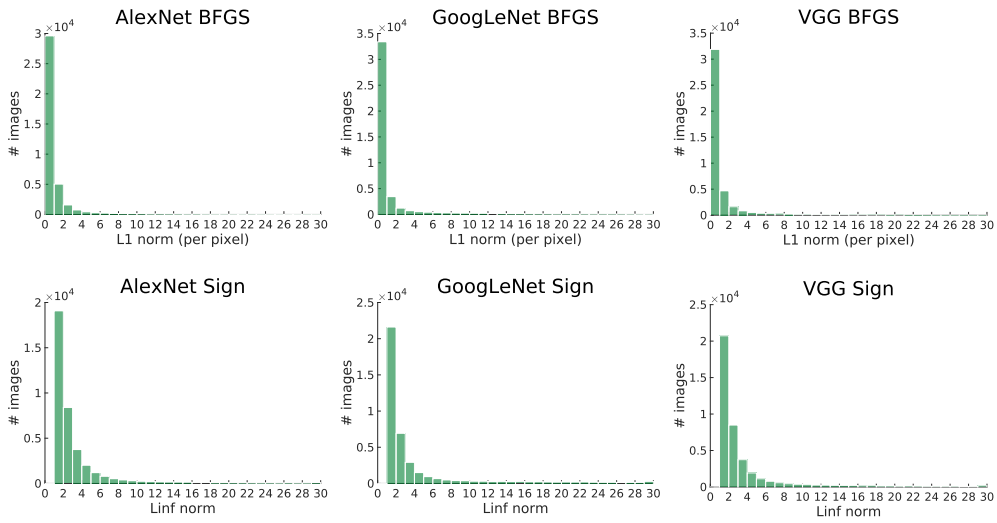

Figure 4: *Histogram of Norm of the Perturbations.*

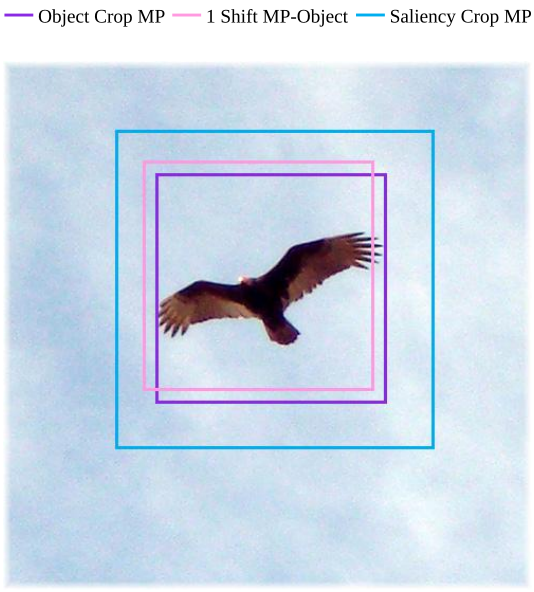

Figure 5: *Example of Different Foveations. Object Crop MP-Image* is the purple bounding box and 1 *Shift MP-Object* is the pink bounding box. The blue bounding box corresponds to a crop from *Saliency Crop MP-Image*. Note that the center crop of 10 *Shift MP-Object* is exactly the same as *Object Crop MP-Image*.

Table 2: *Evaluation of the Foveation Mechanisms.* Quantitative results of the top-5 accuracy with minimum perturbation.

	BFGS Minimum Perturbation							
	$T_0(\mathbf{x})$ = Image				$T_0(\mathbf{x})$ = Object			
	w/o MP-Image	MP-Image	Object Crop MP-Image	Saliency Crop MP-Image	w/o MP-Object	MP-Object	1 Shift MP-Object	Embedded MP-Object
ALX	78.4	0.5	78.0	72.4	81.9	1.8	76.8	75.1
GNT	87.4	1.0	83.1	80.7	89.4	2.8	83.3	84.2
VGG	85.4	0.6	82.6	78.8	91.2	3.1	81.5	78.1
	Sign Minimum Perturbation							
	$T_0(\mathbf{x})$ = Image				$T_0(\mathbf{x})$ = Object			
	w/o MP-Image	MP-Image	Object Crop MP-Image	Saliency Crop MP-Image	w/o MP-Object	MP-Object	1 Shift MP-Object	Embedded MP-Object
ALX	78.4	0	72.1	67.8	81.9	0.2	70.7	70.8
GNT	87.4	0.1	72.9	70.7	89.4	0.5	73.3	77.6
VGG	85.4	0.1	68.6	63.9	91.2	0.7	64.0	67.7

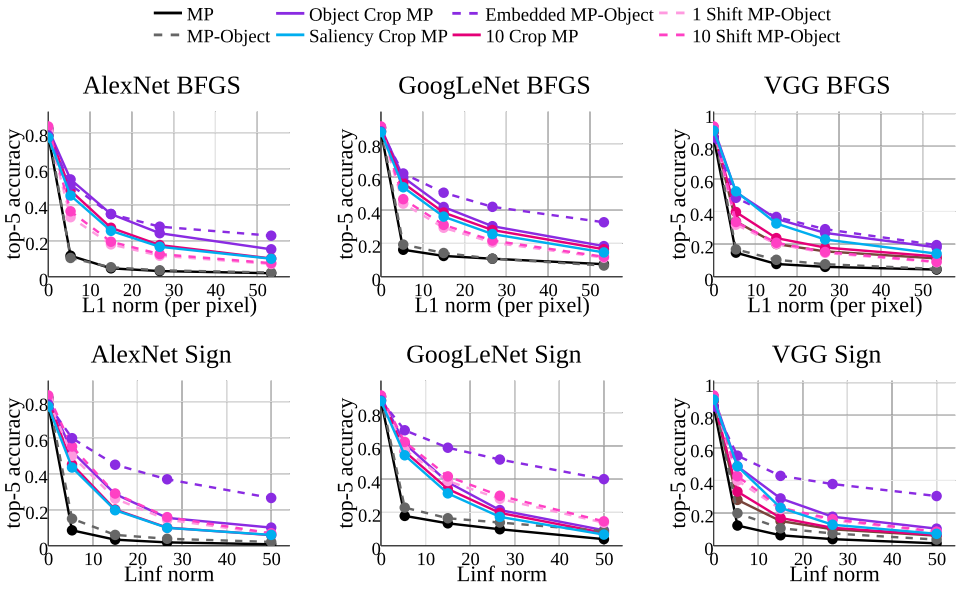

Figure 6: *Accuracy of the Foveations.* Accuracy for the three CNNs we evaluate, when varying the L_1 norm of *BFGS* and varying the L_∞ norm of *Sign*.

Table 3: *Evaluation of the Foveation Mechanisms.* Quantitative results of the top-5 accuracy in Fig. 6.

	No Perturbation ($L_1 = 0$)							
	MP-Image	Object Crop MP-Image	10 Crop MP-Image	Saliency Crop MP-Image	MP-Object	10 Shift MP-Object	1 Shift MP-Object	Embedded MP-Object
ALX	0.7841	0.8192	0.8111	0.7729	0.8192	0.8341	0.8123	0.7841
GNT	0.8736	0.8939	0.8951	0.8690	0.8939	0.9030	0.8910	0.8736
VGG	0.8536	0.9122	0.8912	0.8957	0.9122	0.9212	0.9132	0.8536
	BFGS Perturbation ($L_1 = 5.3$)							
	MP-Image	Object Crop MP-Image	10 Crop MP-Image	Saliency Crop MP-Image	MP-Object	10 Shift MP-Object	1 Shift MP-Object	Embedded MP-Object
ALX	0.1166	0.5401	0.4782	0.4508	0.1043	0.3632	0.3304	0.5100
GNT	0.1622	0.5972	0.5623	0.5377	0.1934	0.4662	0.4394	0.6204
VGG	0.1477	0.5180	0.3966	0.5234	0.1679	0.3355	0.3190	0.4838
	Sign Perturbation ($L_\infty = 5.3$)							
	Before Foveation	After Foveation			Before Foveation	After Foveation		
	MP-Image	Object Crop MP-Image	10 Crop MP-Image	Saliency Crop MP-Image	MP-Object	10 Shift MP-Object	1 Shift MP-Object	Embedded MP-Object
ALX	0.0855	0.5225	0.4515	0.4341	0.1501	0.5506	0.4989	0.5970
GNT	0.1793	0.6133	0.5666	0.5435	0.2292	0.6235	0.5763	0.6938
VGG	0.1238	0.4867	0.3319	0.4869	0.1995	0.4259	0.3995	0.5520

Table 4: *Evaluation of the Foveation Mechanisms.* Increase factor of the norm of the perturbation to produce misclassification after the foveation mechanisms. Only the images that are correctly classified before and after the foveation are included.

	AlexNet		GoogLeNet		VGG	
ratio	Sign	BFGS	Sign	BFGS	Sign	BFGS
Object Crop MP / MP	6.2484	14.4476	6.3432	25.8569	5.0216	17.1506
10 Shift MP-Object / MP-Object	4.9085	9.1912	5.6342	19.4249	2.9630	8.5588
1 Shift MP-Object / MP-Object	4.5336	8.4306	5.1256	18.1737	2.9676	8.8038

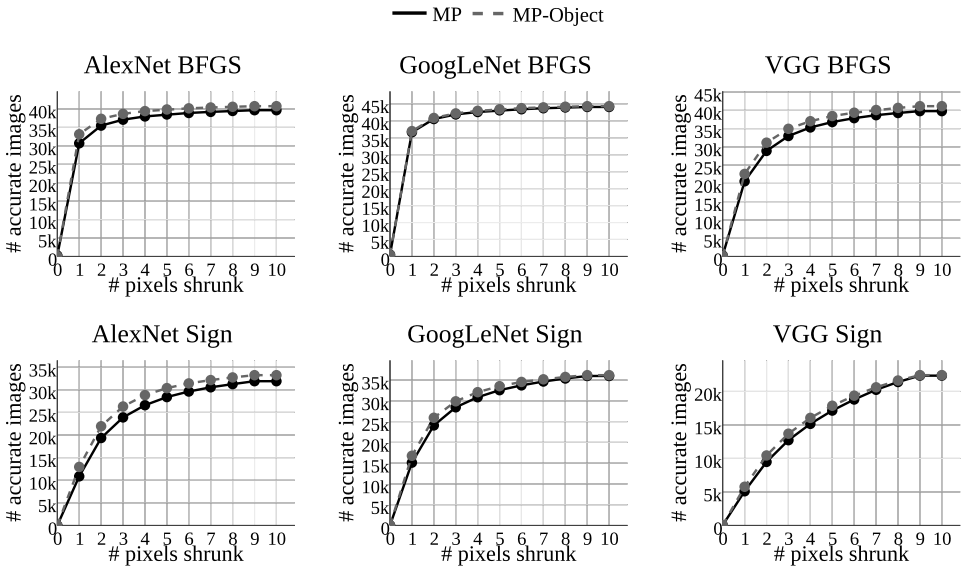

Figure 7: *Accuracy After Shrinking the Foveation.* Accuracy after shrinking the foveation of the adversarial example. Only the images that are correctly classified by the DNN are included

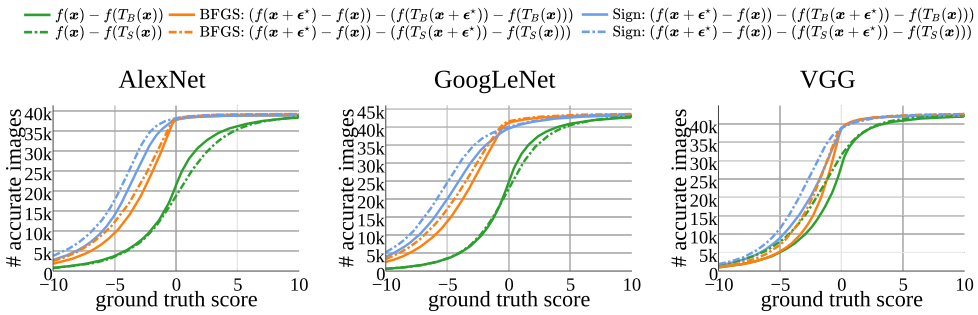

Figure 8: *Effect of the Foveation to the Perturbations.* Cumulative histogram of the number of images with a change of classification score smaller than the indicated in the horizontal axis. $T_B(\cdot)$ is the foveation with *Object Crop MP*, and $T_S(\cdot)$ is the foveation with 1 *Shift MP-Object*. Only the images that are correctly classified by the CNN are included.

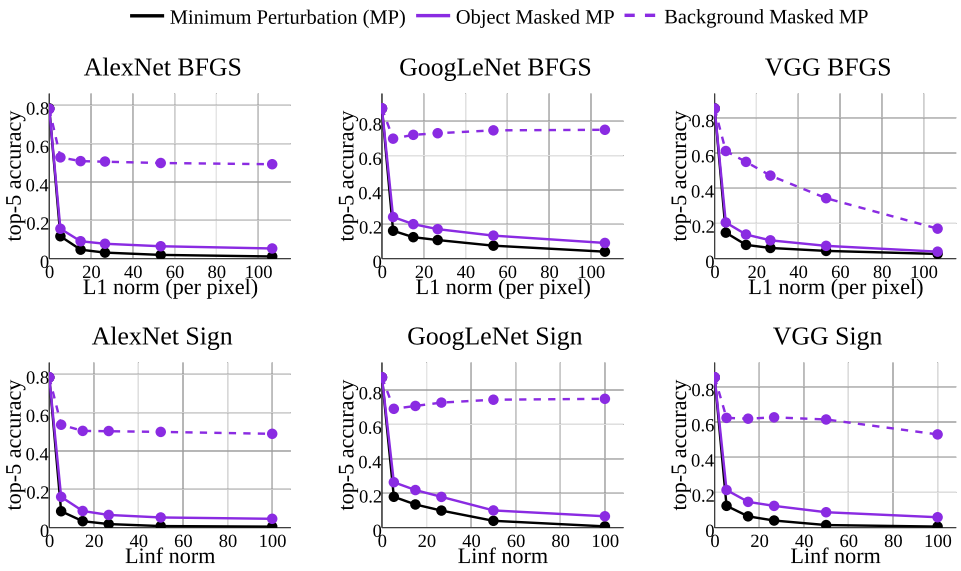

Figure 9: *Accuracy of the Masked Perturbations.* Accuracy for the three CNNs we evaluate, when varying the norm of *BFGS* (first row) and *Sign* (second row).

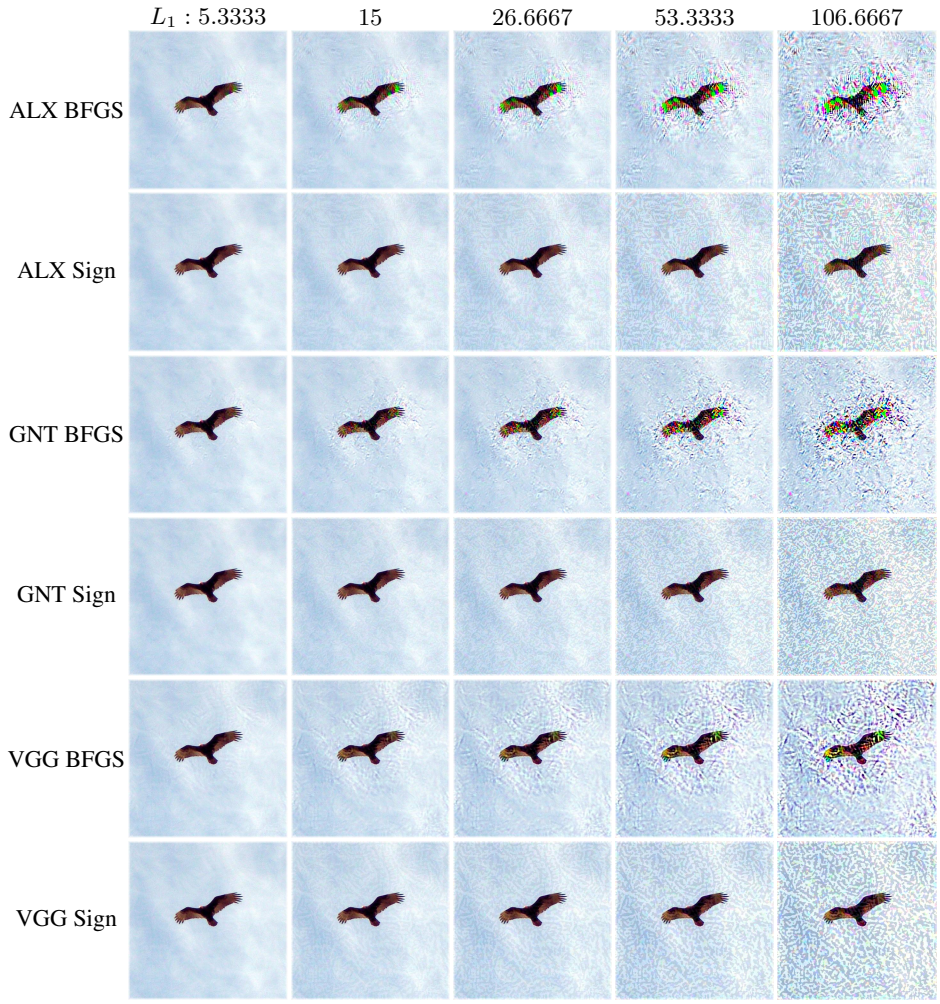

Figure 10: *Qualitative example of the perturbations changing the average per pixel of the L_1 norm.* We denote the perturbation as X Y, where X is the network that generated the perturbation - ALX (AlexNet), GNT (GoogLeNet), VGG- and Y indicates the *BFGS* or *Sign*.

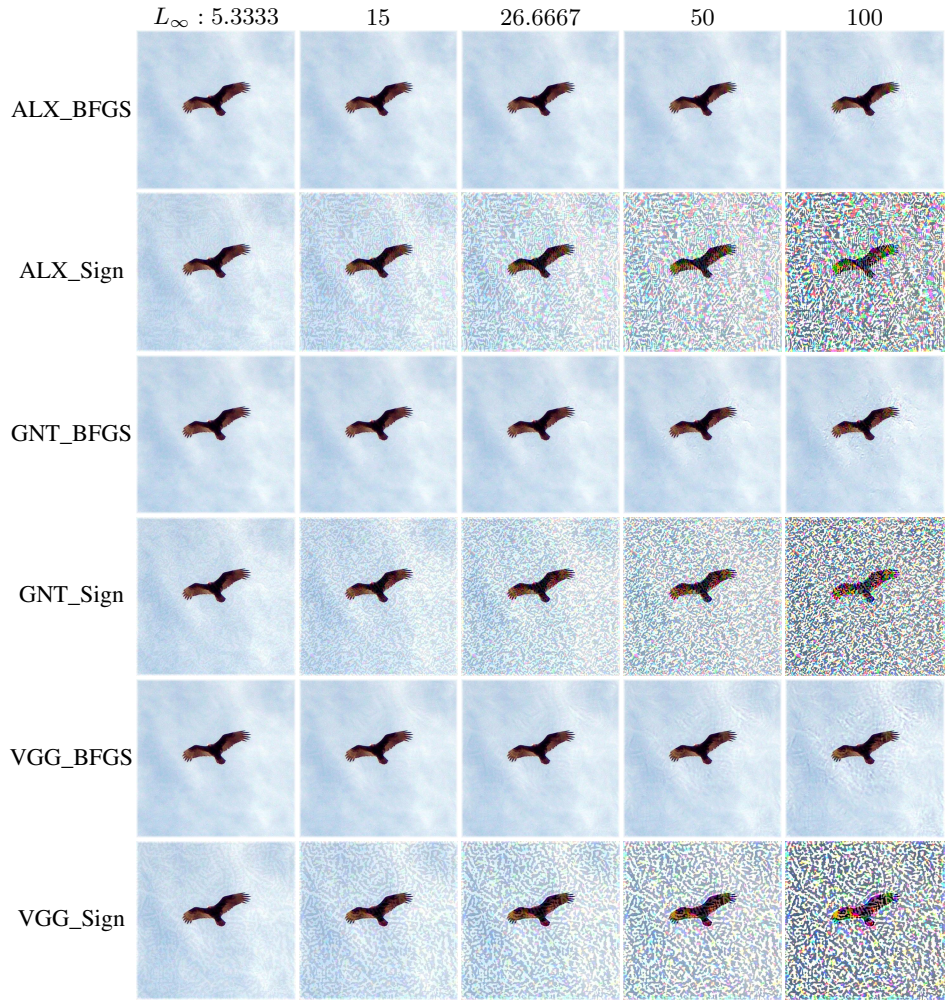

Figure 11: *Qualitative example of the perturbations changing the average per pixel of the L_∞ norm.* We denote the perturbation as X Y, where X is the network that generated the perturbation - ALX (AlexNet), GNT (GoogLeNet), VGG- and Y indicates the *BFGS* or *Sign*.

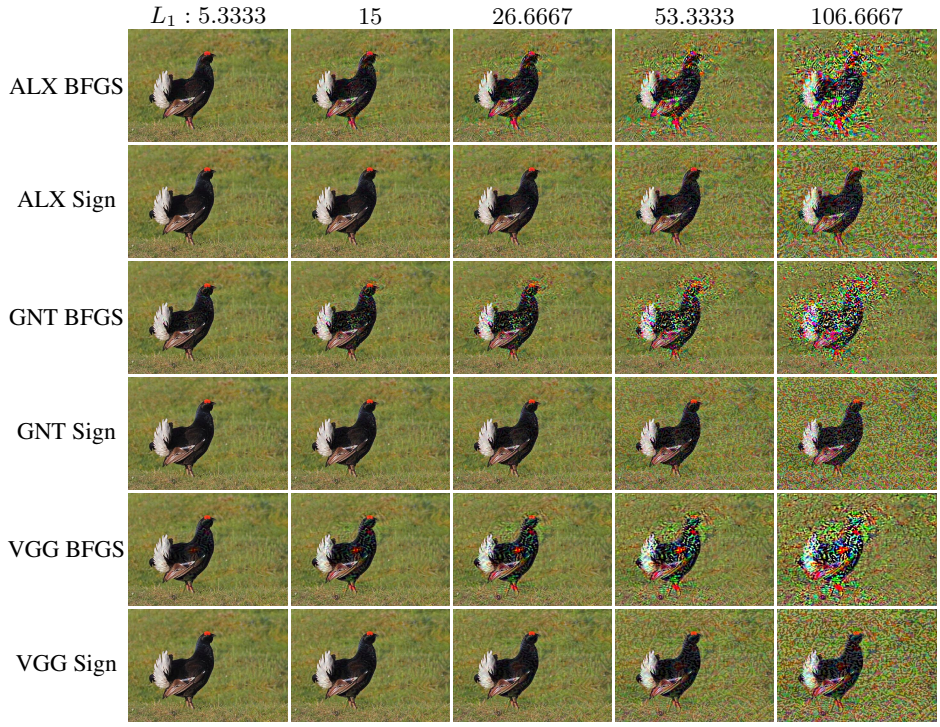

Figure 12: *Qualitative example of the perturbations changing the average per pixel of the L_1 norm.* We denote the perturbation as X Y, where X is the network that generated the perturbation - ALX (AlexNet), GNT (GoogLeNet), VGG- and Y indicates the *BFGS* or *Sign*.

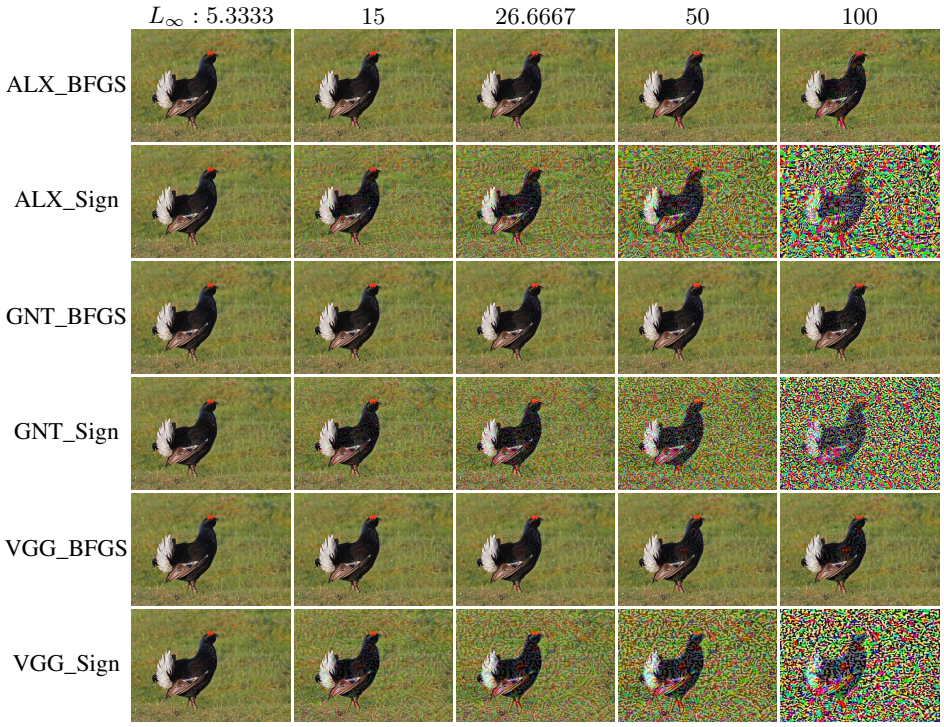

Figure 13: *Qualitative example of the perturbations changing the average per pixel of the L_∞ norm.* We denote the perturbation as X Y, where X is the network that generated the perturbation - ALX (AlexNet), GNT (GoogLeNet), VGG- and Y indicates the *BFGS* or *Sign*.

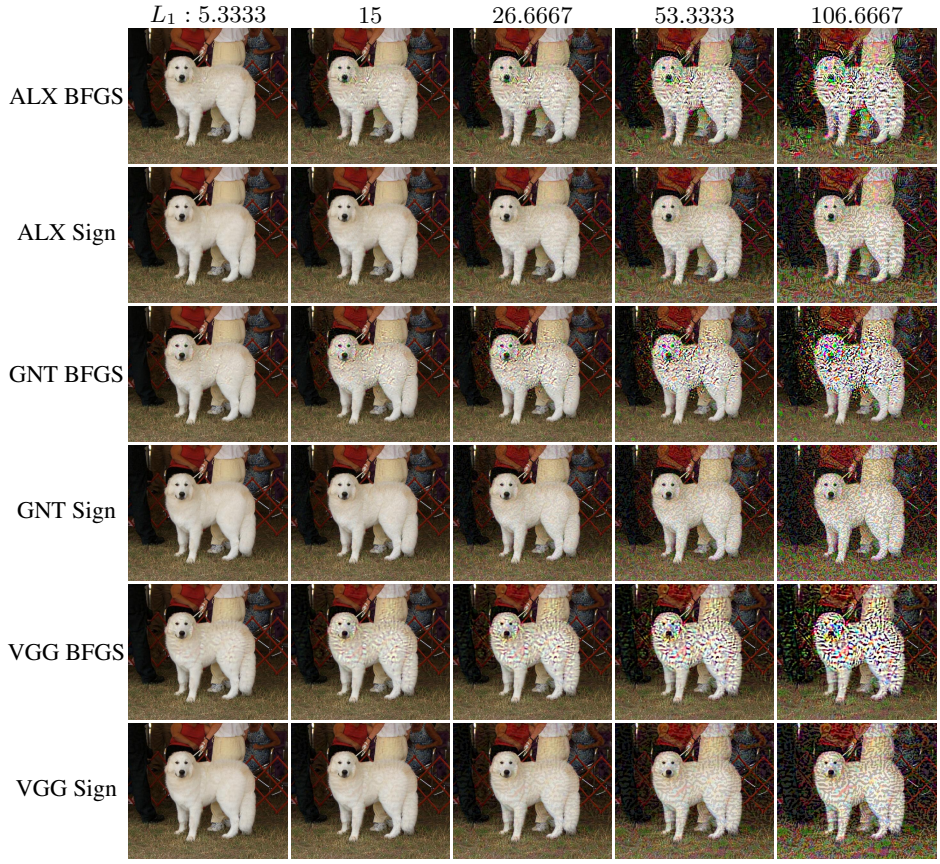

Figure 14: *Qualitative example of the perturbations changing the average per pixel of the L_1 norm.* We denote the perturbation as X Y, where X is the network that generated the perturbation - ALX (AlexNet), GNT (GoogLeNet), VGG- and Y indicates the *BFGS* or *Sign*.

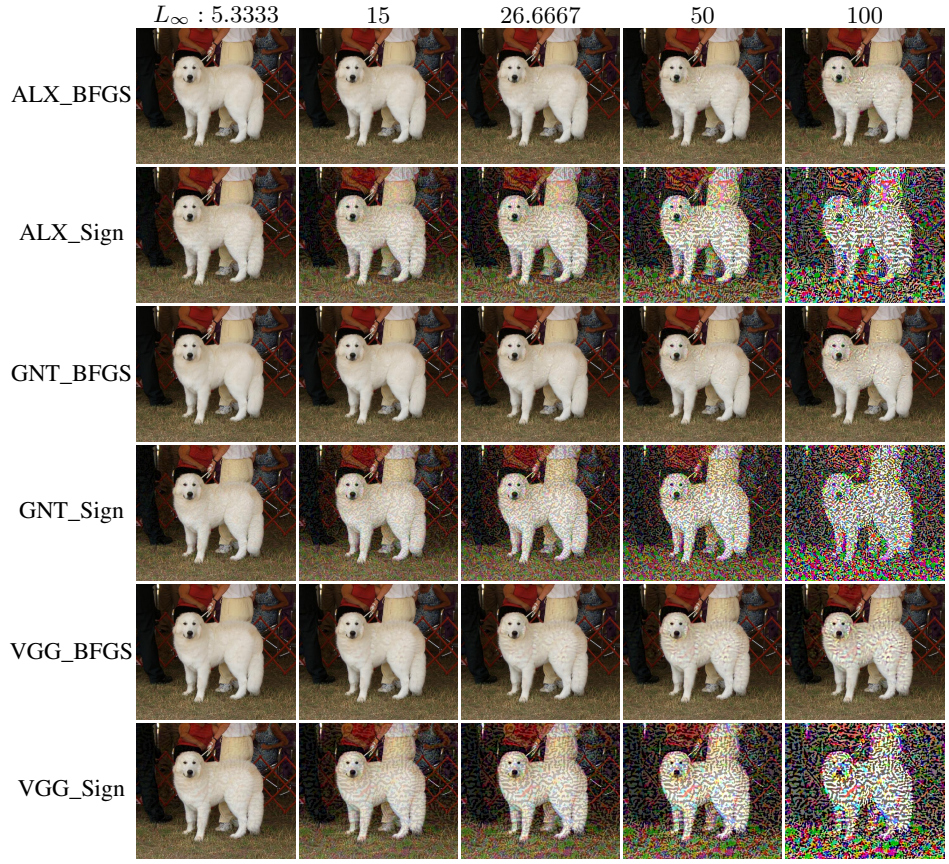

Figure 15: *Qualitative example of the perturbations changing the average per pixel of the L_∞ norm.* We denote the perturbation as X Y, where X is the network that generated the perturbation - ALX (AlexNet), GNT (GoogLeNet), VGG- and Y indicates the *BFGS* or *Sign*.