Visualizing Deep Networks by Optimizing with Integrated Gradients

Zhongang Qi, Saeed Khorram, Li Fuxin
School of Electrical Engineering and Computer Science,
Oregon State University
{qiz,khorrams,lif}@oregonstate.edu

Abstract

Understanding and interpreting the decisions made by deep learning models is valuable in many domains. In computer vision, computing heatmaps from a deep network is a popular approach for visualizing and understanding deep networks. However, heatmaps that do not correlate with the network may mislead human, hence the performance of heatmaps in providing a faithful explanation to the underlying deep network is crucial. In this paper, we propose I-GOS, which optimizes for a heatmap so that the classification scores on the masked image would maximally decrease. The main novelty of the approach is to compute descent directions based on the integrated gradients instead of the normal gradient, which avoids local optima and speeds up convergence. Compared with previous approaches, our method can flexibly compute heatmaps at any resolution for different user needs. Extensive experiments on several benchmark datasets show that the heatmaps produced by our approach are more correlated with the decision of the underlying deep network, in comparison with other state-of-the-art approaches.

1. Introduction

In recent years, there has been a lot of focus on explaining deep neural networks [19, 15, 6, 3, 31]. Explainability is important for humans to trust the deep learning model, especially in crucial decision-making scenarios. In the computer vision domain, one of the most important explanation techniques is the heatmap approach [28, 23, 21, 29], which focuses on generating heatmaps that highlight parts of the input image that are most important to the decision of the deep networks on a particular classification target.

Some heatmap approaches achieve good visual qualities for human understanding, such as several one-step backpropagation-based visualizations including Guided Backpropagation (GBP) [25] and the deconvolutional network (DeconvNet) [28]. These approaches utilize the gradient or variants of the gradient and backpropagate them back to the input image, in order to decide which pixels are more relevant to the change of the deep network prediction. However, whether they are actually correlated to the decision-making of the network is not that clear [16]. [16] proves that GBP and DeconvNet are essentially doing (partial) image recovery, and thus generate more human-interpretable visualizations that highlight object boundaries, which do not necessarily represent what the model has truly learned.

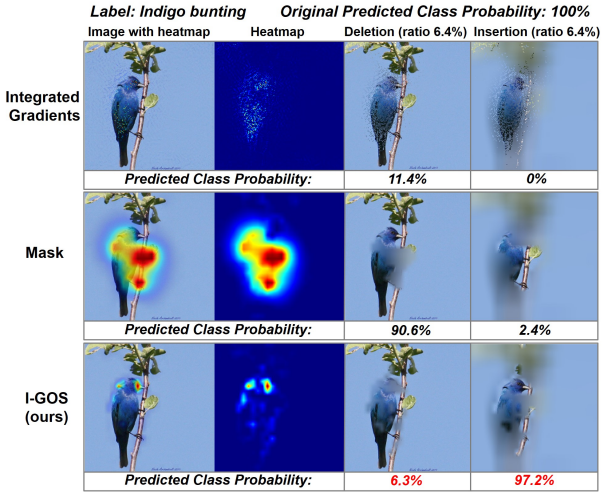


Figure 1. Heatmap visualizations can be verified by testing the CNN on deletion images (column 3), which blur the highlighted areas of the image, and insertion images (column 4), which blur areas not highlighted on the heatmap. The first two rows show that Integrated Gradients [26], Mask [8] may fail on these evaluations. Using heatmap generated from our I-GOS, CNN no longer classifies the deletion image to the same category (column 3), and classifies the insertion image correctly with only few pixels revealed (column 4), showing the correlation between the I-GOS heatmap and CNN decision making. For all approaches the same amount of pixels (6.4% in this figure) were blurred/revealed. (Best viewed in color)

An issue with these one-step approaches is that they only reflect infinitesimal changes of the prediction of a deep network. In the highly nonlinear function estimated by the deep network, such infinitesimal changes are not neces-

sarily reflective of changes large enough to alter the decision of the deep network. [18] proposed evaluation metrics based on masking the image with heatmaps and verifying whether the masking will indeed change deep network predictions. Ideally, if the highlighted regions for a category are removed from the image, the deep network should no longer predict that category. This is measured by the *deletion* metric. On the other hand, the network should predict a category only using the regions highlighted by the heatmap, which is measured by the *insertion* metric (Fig. 1).

If these are the goals of a heatmap, a natural idea would be to directly optimize them. The mask approach proposed in [8] generates heatmaps by solving an optimization problem, which aims to find the smallest and smoothest area that maximally decrease the output of a neural network, directly optimizing the *deletion* metric. It can generate very good heatmaps, but usually takes a long time to converge, and sometimes the optimization can be stuck in a bad local optimum due to the strong nonconvexity of the solution space.

Another approach called integrated gradients [26] claim that any change in the output can be reflected in their heatmap. The basic idea is to explicitly find the image that has the lowest prediction score – a completely grey image, or a highly blurred image usually would not be predicted to any category by a deep network, and then integrate the gradients on the entire line between the grey/blurred image to the original image to generate a heatmap. However, the heatmaps generated by integrated gradients are normally diffuse, thus difficult for human to understand (Fig. 1).

In this paper, we propose a novel visualization approach I-GOS (Integrated-Gradients Optimized Saliency) which utilizes the integrated gradients to improve the mask optimization approach in [8]. The idea is that the direction provided by the integrated gradients may lead better towards the global optimum than the normal gradient which may tend to lead to local optima. Hence, we replace the gradient in mask optimization with the integrated gradients. Due to the high cost of computing the integrated gradients, we employ a line-search based gradient-projection method to maximally utilize each computation of the integrated gradients. Our approach generates better heatmaps (Fig. 1) and utilizes less computational time than the original mask optimization, as line search is more efficient in finding appropriate step sizes, allowing significantly less iterations to be used. We highlight our contributions as follows:

(1) We developed a novel heatmap visualization approach I-GOS, which optimizes a mask using the integrated gradients as descent steps.

(2) Through regularization and perturbation we better avoided generating adversarial masks at higher resolutions, enabling more detailed heatmaps that are more correlated with the decision-making of the model.

(3) Extensive evaluations show that the proposed approach performs better than the state-of-the-art approaches, especially in the *insertion* and *deletion* metrics.

2. Related Work

There are several different types of the visualization techniques for generating heatmaps for a deep network. We classify them into one-step backpropagation-based approaches [28, 23, 25, 22, 26, 2, 29, 21], and perturbation-based approaches, e.g., [30, 4, 8, 18].

The basic idea of one-step backpropagation-based visualizations is to backpropagate the output of a deep neural network back to the input space using the gradient or its variants. DeconvNet [28], Saliency Maps (using the gradient) [23], and GBP [25] are similar approaches, with the difference among them in the way they deal with the ReLU layer. LRP [2] and DeepLIFT [22] compute the contributions of each input feature to the prediction. Excitation BP [29] passes along top-down signals downwards in the network hierarchy via a probabilistic Winner-Take-All process. GradCAM [21] uses the gradients of a target concept, flowing only into the final convolutional layer to produce a coarse localization map. [1] analyzes various backpropagation-based methods, and provides a unified view to explore the connections among them.

The perturbation-based methods firstly perturb parts of the input, and then run a forward pass to see which ones are most important to preserve the final decision. The earliest approach, [30], utilized a grey patch to occlude part of the image. This approach is direct but very slow, usually taking hours for a single image [1]. An improvement is to introduce a mask, and solve for the optimal mask as an optimization problem [4, 8]. [4] develop a trainable masking model that can produce the masks in a single forward pass. However, it is difficult to train a mask model, and different models need to be trained for different networks. [8] directly solves the optimization, and find the mask iteratively. Instead of only occluding one patch of the image, RISE [18] generates thousands of randomized input masks simultaneously, and averages them by their output scores. However, it consumes significant time and GPU memory.

Another seemingly related but different domain is the saliency map from human fixation [11]. Fixation Prediction [12, 13] aims to identify the fixation points that human viewers would focus on at first glance of a given image. When predicting eye fixation, the algorithm is guessing the regions humans are looking at, while our goal is to explain what the deep models focus on for a given image to make decisions. Deep models may use completely different mechanisms to classify than humans, hence human fixations should not be used to train or evaluate heatmap models.

3. Model Formulation

3.1. Gradient and Mask Optimization

Gradient and its variants are often utilized in visualization tools to demonstrate the importance of each dimension of the input. The motivation of it comes from the linearization of the model. Suppose a black-box deep network f predicts a score $f_c(I)$ on class c (usually the logits of a class before the softmax layer) from an image I. Assume f is smooth at the current image I_0, then a local approximation can be obtained using the first-order Taylor expansion:

$$f_c(I) \approx f_c(I_0) + \langle \nabla f_c(I_0), I - I_0 \rangle, \tag{1}$$

The gradient $\nabla f_c(I_0)$ is indicative of the local change that can be made to $f_c(I_0)$ if a small perturbation is added to it, and hence can be visualized as an indication of salient image regions to provide a local explanation for image I_0 [23]. In [22], the heatmap is computed by multiplying the gradient feature-wise with the input itself, i.e., $\nabla f_c(I_0) \odot I_0$, to improve the sharpness of heatmaps.

However, gradient only illustrates the infinitesimal change of the function $f_c(I)$ at I_0, which is not necessarily indicative of the salient regions that lead to a significant change on $f_c(I)$, especially when the function is highly nonlinear. What we would expect is that the heatmaps indicate the areas that would really change the classification result significantly. In [8], a perturbation based approach is proposed which introduces a mask M as the heatmap to perturb the input I_0. M is optimized by solving the following objective function:

$$\begin{aligned} &\underset{M}{\text{argmin}} \;\; F_c(I_0, M) = f_c\big(\Phi(I_0, M)\big) + g(M), \\ &\text{where} \;\; g(M) = \lambda_1 ||\mathbf{1} - M||_1 + \lambda_2 \text{TV}(M), \\ &\qquad \Phi(I_0, M) = I_0 \odot M + \tilde{I}_0 \odot (\mathbf{1} - M), \\ &\qquad \mathbf{0} \le M \le \mathbf{1}, \end{aligned} \tag{2}$$

In (2), M is a matrix which has the same shape as the input image I_0 and whose elements are all in $[0, 1]$; $\tilde{I}_0$ is a baseline image with the same shape as I_0, which should have a low score on the class c, $f_c\big(\tilde{I}_0\big) \approx \min_I f_c(I)$, and in practice either a constant image, random noise, or a highly blurred version of I_0. This optimization seeks to find a deletion mask that significantly decreases the output score $f_c\big(\Phi(I_0, M)\big)$, i.e., $f_c\big(I_0 \odot M + \tilde{I}_0 \odot (\mathbf{1} - M)\big) \ll f_c(I_0)$ under the regularization of $g(M)$. $g(M)$ contains two regularization terms, with the first term on the magnitude of M, and the second term a total-variation (TV) norm to make M more piecewise-smooth.

Although this approach of optimizing a mask performs significantly better than the gradient method, there exist inevitable drawbacks when using a traditional first-order algorithm to solve the optimization. First, it is slow, usually taking hundreds of iterations to obtain the heatmap for each image. Second, since the model f_c is highly nonlinear in most cases, optimizing (2) may only achieve a local optimum, with no guarantee that it indicates the right direction for a significant change related to the output class. Fig. 1 and Fig. 3 show some heatmaps generated by the mask approach.

3.2. Integrated Gradients

Note that the problem of finding the mask is not a conventional non-convex optimization problem. For $F_c(I_0, M) = f_c(I_0, M) + g(M)$, we (approximately) know the global minimum (or, at least a reasonably small value) of $f_c(I_0, M)$ in a baseline image $\tilde{I}_0$, which corresponds to $M = \mathbf{0}$. The integrated gradients approach [26] considers the straight-line path from the baseline $\tilde{I}_0$ to the input I_0. Instead of evaluating the gradient at the provided input I_0 only, the integrated gradients would be obtained by accumulating all the gradients along the path:

$$IG_i(I_0) = (I_0^i - \tilde{I}_0^i) \cdot \int_{\alpha=0}^{1} \frac{\partial f_c\big(\tilde{I}_0 + \alpha(I_0 - \tilde{I}_0)\big)}{\partial I_0^i} d\alpha, \tag{3}$$

where $IG(I_0) = \nabla_{I_0}^{IG} f_c(I_0)$ is the integrated gradients of f_c at I_0; i represents the i-th pixel. [26] proved that it satisfies an axiom called completeness that the integrated gradients for all pixels add up to the difference between the output of f_c at the input I_0 and the baseline $\tilde{I}_0$, if f_c is differentiable almost everywhere:

$$\sum_i IG_i(I_0) = f_c(I_0) - f_c(\tilde{I}_0), \tag{4}$$

where the summation sums over all pixels in $IG(I_0)$. The completeness axiom shows that if the baseline has a near-zero score, the integrated gradients can be interpreted as the prediction function of the input $f_c(I_0)$, which means all changes in $f_c(I_0)$ are reflected in the integrated gradients.

In practice, the integral in (3) is approximated via a summation. We sum the gradients at points occurring at sufficiently small intervals along the straight-line path from the input M to a baseline $\tilde{M} = \mathbf{0}$:

$$\nabla^{IG} f_c(M) = \frac{1}{S} \sum_{s=1}^{S} \frac{\partial f_c\left(\Phi\left(I_0, \frac{s}{S} M\right)\right)}{\partial M}, \tag{5}$$

where S is a constant, usually 20.

Integrated gradients have some nice theoretical properties and perform better than the gradient-based approaches. However, the heatmap generated by the integrated gradients is still diffuse. We speculate that the reason maybe that changes on some pixels in $IG(I_0)$ may not be very important to $f_c(I_0)$, or their contributions cancel out each other. Fig. 1 and Fig. 3 show some heatmaps generated by the integrated gradients approach where a grey zero image is utilized as the baseline. We can see that the integrated gradient contains many false positives in the area wherever the pixels have a large value of $I_0^i - \tilde{I}_0^i$ (either the white or the black pixels).

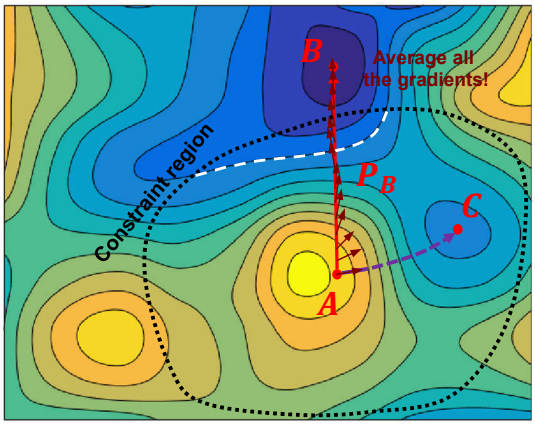


Figure 2. (Best viewed in color) Suppose we are optimizing in a region with a start point A, a local optimum C, and a baseline B which is the unconstrained global optimum; the area within the black dashed line is the constraint region which is decided by the constraint terms $g(I, M)$ and the bound constraints $\mathbf{0} \leq M \leq \mathbf{1}$, we may find a better solution by always moving towards B rather than following the gradient and end up at C.

3.3. Integrated Gradients Optimized Heatmaps

We believe the above two approaches can be combined for a better heatmap approach. The integrated gradient naturally provides a better direction than the gradient in that it points more directly to the global optimum of a part of the objective function. One can view the convex constraint function $g(M)$ as equivalent to the Lagrangian of a constrained optimization approach with constraints $\|\mathbf{1} - M\|_1 \leq B_1$ and $TV(M) \leq B_2$, B_1 and B_2 being positive constants, hence consider the optimization problem (2) to be a constrained minimization problem on $f_c(\Phi(I_0, M))$. In this case, we know the unconstrained solution in $M = \mathbf{0}$ is outside the constraint region. We speculate that an optimization algorithm may be better than gradient descent if it directly attempts to move to the unconstrained optimum.

To illustrate this, Fig. 2 shows a 2D optimization with a starting point A, a local optimum C, and a baseline B. The area within the black dashed line is the constraint region which is decided by the constraint function $g(M)$ and the boundary of M. A first-order algorithm will follow the gradient descent direction (the purple line) to the local optimum C; while the integrated gradients computed along the path P_B from A to the baseline B may enable the optimization to reach an area better than C within the constraint region. We can see that the integrated gradients with an appropriate baseline have a global view of the space and may generate a better descent direction. In practice, the baseline does not need to be the global optimum. A good baseline near the global optimum could still improve over the local optimum achieved by gradient descent.

Hence, we utilize the integrated gradients to substitute the gradient of the partial objective $f_c(M)$ in optimization (2), and introduce a new visualization method called Integrated-Gradient Optimized Saliency (I-GOS). For the regularization terms $g(M)$ in optimization (2), we still compute the partial (sub)gradient with respect to M:

$$\nabla g(M) = \lambda_1 \cdot \frac{\partial ||\mathbf{1} - M||_1}{\partial M} + \lambda_2 \cdot \frac{\partial \text{TV}(M)}{\partial M}, \tag{6}$$

The total (sub)gradient of the optimization for M at each step is the combination of the integrated gradients for the $f_c(M)$ and the gradients of the regularization terms $g(M)$:

$$TG(M) = \nabla^{IG} f_c(M) + \nabla g(M), \tag{7}$$

Note that this is no longer a conventional optimization problem, since it contains 2 different types of gradients. The integrated gradients are utilized to indicate a direction for the partial objective $f_c(M)$; the gradients of the $g(M)$ are used to regularize this direction and prevent it to be diffuse.

3.4. Computing the step size

Since the time complexity of computing $\nabla^{IG} f_c(M)$ is high, we utilize a backtracking line search method and revise the Armijo condition [17] to help us compute the appropriate step size for the total gradient $TG(M)$ in formula (7). The Armijo condition tries to find a step size such that:

$$f(M_k + \alpha_k \cdot d_k) - f(M_k) \leq \alpha_k \cdot \beta \cdot \nabla f(M_k)^T d_k, \tag{8}$$

where d_k is the descent direction; α_k is the step size; β is a parameter in $(0, 1)$; $\nabla f(M_k)$ is the gradient of f at point M_k.

The descent direction d_k for our algorithm is set to be the inverse direction of the total gradient $TG(M_k)$. However, since $TG(M_k)$ contains the integrated gradients, it is uncertain whether $\nabla F_c(M_k)^T d_k = -\nabla F_c(M_k)^T TG(M_k)$ is negative or not. Hence, we replace $\nabla F_c(M_k)$ with $TG(M_k)$ and obtain a revised Armijo condition as follows:

$$F_c\Big(M_k - \alpha_k \cdot TG(M_k)\Big) - F_c(M_k) \leq -\alpha_k \cdot \beta \cdot TG(M_k)^T TG(M_k), \tag{9}$$

The detailed backtracking line search works as follows:

(1) Initialization: set the values of the parameter β, a decay η, a upper bound α_u and a lower bound α_l for the step size; let $j = 0$, and $\alpha^0 = \alpha_u$;

(2) Iteration: if α^j satisfies condition (9), or $\alpha_j \leq \alpha_l$, end iteration; else, let $\alpha^{j+1} = \alpha^j \eta$, $j = j + 1$, test condition (9) again with $P_{[0,1]}(M_k - \alpha_k \cdot TG(M_k))$, where $P_{[0,1]}(M)$ clips the mask values to the closed interval $[0, 1]$;

(3) Output: if $\alpha^j \leq \alpha_l$, the step size α_k for $TG(M_k)$ equals to the lower bound α_l; else, $\alpha_k = \alpha^j$

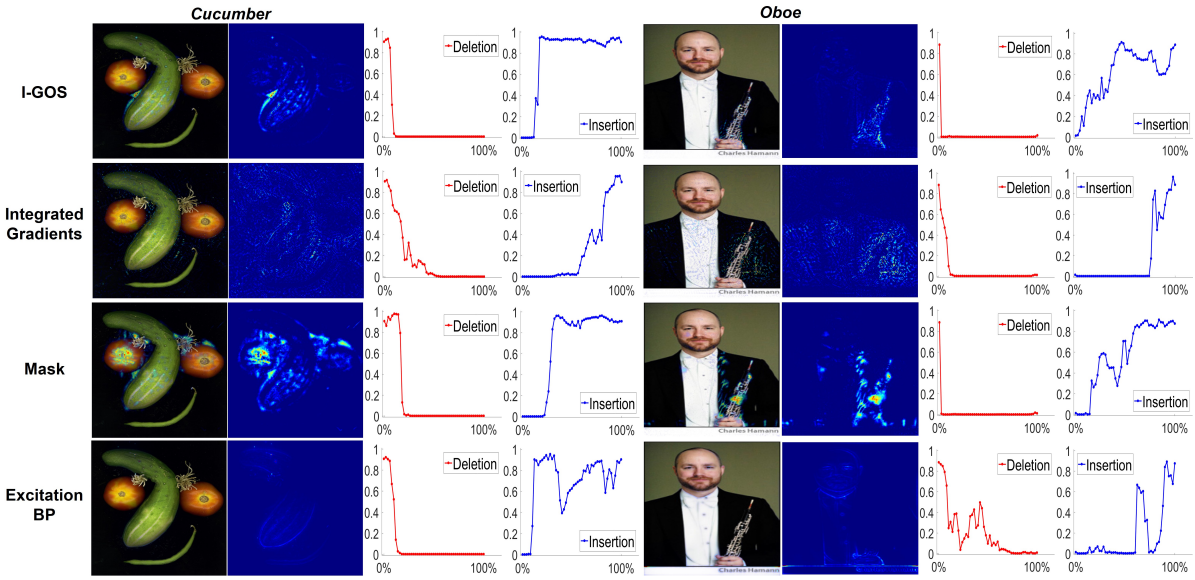


Figure 3. A comparison among different approaches with heatmaps of 224×224 resolution. The red plot illustrates how the CNN predicted probability drops with more areas masked, and the blue plot illustrates how the prediction increases with more areas revealed. The x axis for the red/blue plot represents the percentage of pixels masked/revealed; the y axis for the red/blue plot represents the predicted class probability. One can see with I-GOS the red curve drops earlier and the blue plot increase earlier, leading to more area under the insertion curve (insertion metric) and less area under the deletion curve (deletion metric). (Best viewed in color)

A projection step is needed in the iteration because the mask M_k is bounded by the closed interval $[0, 1]$. Since we have an integrated gradient in $TG(M)$, a large upper bound α_u and a small β are needed in order to obtain a large step that satisfies condition (9), similar to satisfying the Goldstein conditions for convergence in conventional Armijo-Goldstein line search.

Note that we cannot prove the convergence properties of the algorithm in non-convex optimization. However, the integrated gradient reduces to a scaling on the conventional gradient in a quadratic function (see supplementary material). In practice, it converges much faster than the original mask approach in [8] and we have never observed it diverging, although in some cases we do note that even with this approach the optimization stops at a local optimum. With the line search, we usually only run the iteration for $10 - 20$ steps. Intuitively, the irrelevant parts of the integrated gradients are controlled gradually by the regularization function $g(M)$ and only the parts that truly correlate with output scores would remain in the final heatmap.

3.5. Avoiding adversarial examples

Since the mask optimization (2) is similar to the adversarial optimization [27, 9] except the TV term, it is concerning whether the solution would merely be an adversarial attack to the original image rather than explaining the relevant information. An adversarial example is essentially a mask that drives the image off the natural image manifold, hence the approach in [8] utilize a blurred version of the original image as the baseline to avoid creating strong adversarial gradients off the image manifold. We follow [8] and also use a blurred image as the baseline. The total variation constraints also defeats adversarial masks by making the mask piecewise-smooth. We also added other methods to avoid finding an adversarial perturbation:

- When computing the integrated gradients using formula (5), we add different random noise n_s to I_0 at each point along the straight-line path:
- We set the resolution of the mask M be smaller than the shape of the input image I_0, upsample it before perturbing the input I_0, and rewrite formula (2) as:

$$M^* = \text{argmin } f_c\big(\Phi(I_0, \text{up}(M))\big) + \lambda_1 ||\mathbf{1} - M||_1 + \lambda_2 \text{TV}(M), \quad (10)$$

where up(M) upsamples M to the original resolution.

4. Experiments

4.1. Evaluation Metrics and Parameter Settings

Although many recent work focus on explainable machine learning, there is still no consensus about how to measure the explainability of a machine learning model. For the heatmaps, there exist several evaluation metrics, e.g., the pointing game [29], which measures the ability of a heatmap to focus on the ground truth object bounding box. However, such localization ability only represents human's understanding about the objects in the images, instead of the deep model's perspective of how to classify objects. There are plenty of evidences that deep learning

Table 1. Evaluation in terms of deletion (lower is better) and insertion (higher is better) scores on ImageNet dataset using the VGG19 model. GradCam can only generate 14×14 heatmaps for VGG19; RISE and Integrated Gradients can only generate 224×224 heatmaps.

	224×224		112×112		28×28		14× 14	
	Deletion	Insertion	Deletion	Insertion	Deletion	Insertion	Deletion	Insertion
Excitation BP [29]	0.2037	0.4728	0.2053	0.4966	0.2202	0.5256	0.2328	0.5452
Mask [8]	0.0482	0.4158	0.0728	0.4377	0.1056	0.5335	0.1753	0.5647
GradCam [21]	– –	– –	– –	– –	– –	– –	0.1527	0.5938
RISE [18]	0.1082	0.5139	– –	– –	– –	– –	– –	– –
Integrated Gradients [26]	0.0663	0.2551	– –	– –	– –	– –	– –	– –
I-GOS (ours)	**0.0336**	**0.5246**	**0.0609**	**0.5153**	**0.0899**	**0.5701**	**0.1213**	**0.6387**

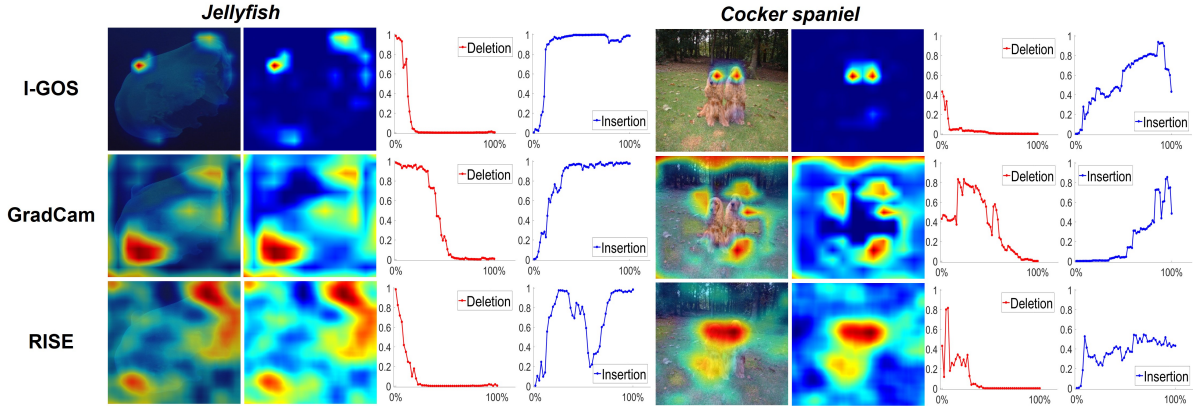


Figure 4. Comparisons between GradCam, RISE, and I-GOS, see Fig. 3 caption for explanations of the meaning of the curves.

sometimes uses background context for object classification which would invalidate pointing game evaluations. In [16], the authors also reveal that some backpropagation-based visualizations such as GBP and DeconvNet are essentially doing (partial) image recovery, and thus generate more human-interpretable visualizations that may score high on the pointing game but do not correlate with network results. Hence, the metrics from the pointing game may be not that convincing. Despite such deficiencies, we still treat it as a human-understandable metric to evaluate the performance of our approach. Following [18], if the most salient pixel lies inside the human annotated bounding box of an object, it is counted as a hit. The pointing game accuracy equals to $\frac{\#Hits}{\#Hits+\#Misses}$, averaged over all categories.

We follow [18] to adopt *deletion* and *insertion* as better metrics to evaluate the performance of the heatmaps generated by different approaches. The intuition behind the *deletion* metric is that the removal of the pixels most relevant to a class will cause the original class score dropping sharply. Only utilizing the *deletion* metric is not satisfactory enough since adversarial images can also achieve a quite good performance. Thus, the *insertion* metric is also needed as a supplementary. The intuition behind the *insertion* metric is that only keeping the most relevant pixels will retain the original score as much as possible, which can eliminate the disturbing from the adversarial attacks. The insertion metric would not score adversarial examples highly (see examples in supplementary material), since to achieve a good insertion score, the deep model needs to make a confident, consistent prediction using a small part of the image. In general, the *deletion-insertion* metrics is a fair metric pair to evaluate different visualization approaches.

In detail, for the *deletion* metric, we remove N pixels (dependent on the resolution of the mask) each time from the original image based on the values of the heatmap until no pixel is left, and replace the removed ones with the pixels of the same locations from a highly blurred version of the original image. The deletion score is the area under the curve (AUC) of the classification scores after softmax [18]). For the insertion metric, we do it inversely, which means we replace N pixels from the highly blurred image with the ones from the original image, based on the values of the heatmap until no pixel left. The insertion score is also the AUC of the classification scores for all the replaced images. In the experiments, we generate heatmaps with different resolutions, including 224×224, 112×112, 28×28, 14×14, and 7×7. And we compute the deletion and insertion scores based on heatmaps with the original resolutions before upsampling.

Three benchmark datasets are utilized in the experiments, including ImageNet [20], MSCOCO [14], and PASCAL VOC [7]. Four different deep networks, including VGG16, VGG19 [24], and ResNet50 [10], are tested as the base models. For the deletion and insertion task, we utilize the pretrained VGG19 and Resnet50 networks from the PyTorch model zoo to test $5,000$ randomly selected images from the validation set of ImageNet. For the pointing game, we utilize two pretrained VGG16 models from [18] to test $2,000$ randomly selected images from the validation set of MSCOCO, and $2,000$ randomly selected images from the

Table 2. Comparative evaluation in terms of deletion (lower is better) and insertion (higher is better) scores on the ImageNet dataset using ResNet50 as the base model. GradCam can only generate 7×7 heatmaps for ResNet50; RISE and Integrated Gradients can only generate 224×224 heatmaps.

	224×224		112×112		28×28		14×14		7×7	
	Deletion	Insertion	Deletion	Insertion	Deletion	Insertion	Deletion	Insertion	Deletion	Insertion
Mask [8]	0.0468	0.4962	0.0746	0.5090	0.1151	0.5559	0.1557	0.5959	0.2259	0.6003
GradCam [21]	– –	– –	– –	– –	– –	– –	– –	– –	0.1675	0.6521
RISE [18]	0.1196	0.5637	– –	– –	– –	– –	– –	– –	– –	– –
Integrated Gradients [26]	0.0907	0.2921	– –	– –	– –	– –	– –	– –	– –	– –
I-GOS (ours)	**0.0420**	**0.5846**	**0.0704**	**0.5943**	**0.1059**	**0.5986**	**0.1387**	**0.6387**	**0.1607**	**0.6632**

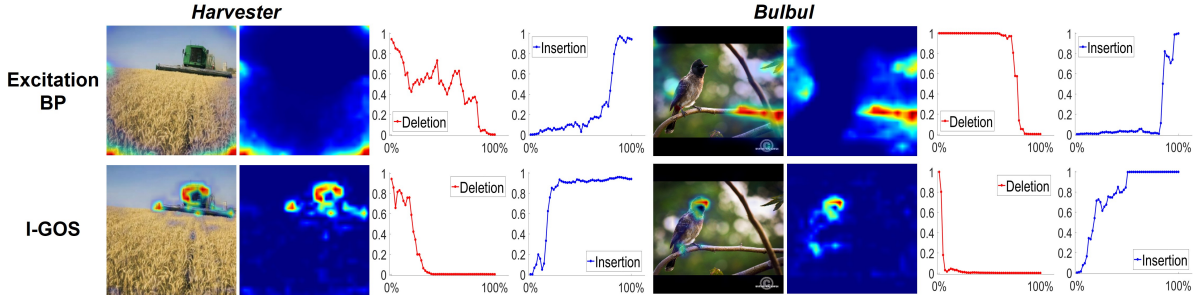


Figure 5. Comparison between Excitation BP and I-GOS in the size of 28×28. See Fig. 3 for explanations of the meaning of the curves.

test set of VOC07, respectively. In Eq. 9, $\beta = 0.0001$. λ_1 and λ_2 (Eq. 10) were fixed across all experiments under the same heatmap resolution. We downloaded and ran the code for most baselines, except for [26] which we implemented. All baselines were tuned to best performances. For all experiments we used the same pre-/post-processing with the same baseline image $\tilde{I}_0$. In [18], the authors used a less blurred image for insertion and a grey image for deletion. Here we used a more blurred image for both insertion and deletion, hence the insertion and deletion scores for RISE are bit different in our paper compared with theirs.

4.2. Results and Discussions

Deletion and Insertion: Table 1 and 2 show the comparative evaluations of I-GOS with other state-of-the-art approaches in terms of both *deletion* and *insertion* metrics on the ImageNet dataset using VGG19 and ResNet50 as the baseline model, respectively. Fig. 3 shows some comparison examples between different approaches on 224×224 heatmaps. From Table 1 and 2 we observe that our proposed approach I-GOS performs better than Excitation BP and Mask [8] in both deletion and insertion scores for heatmaps with all different resolutions. RISE and Integrated Gradients can only generate 224×224 heatmaps. GradCam can only generate 14×14 heatmap on VGG19, and 7×7 heatmap on Resnet50, respectively. And our approach also beats RISE, Integrated Gradient, and GradCam in both deletion and insertion scores on heatmaps with the same resolutions. Although Integrated Gradients has some good properties theoretically, it gets the worst insertion score among all the approaches, which indicates that it indeed contains lots of diffused pixels uncorrelated with the classification, as in the *Cucumber* and *Oboe* examples in Fig. 3. Excitation BP is a one-step backpropagation-based approach that is better than other one-step backpropagation-based approaches, and during the experiments we find that sometimes it just fires on the border and corner of the image instead of the contents, or on irrelevant parts of the image as argued in [16]. Thus, it performs the worst in the deletion task. RISE also suffers on the deletion score maybe because of the randomness on the masks it generates. Fig. 5 shows some visual comparisons between our approach and Excitation BP. Our approach performs better than GradCam for VGG19, and only slightly better for Resnet50. The reason is that for the 7×7 heatmap in Resnet50, it is difficult to increase the insertion score or decrease the deletion score further since there are only 49 pixels in the heatmap. For RISE, we followed [18] to generate $4,000$ 7×7 random samples for VGG, and $8,000$ 7×7 random samples for ResNet. Fig. 4 shows some visual comparisons between our approach, GradCAM, and RISE. From Fig. 4 we observe that sometimes GradCAM also fires on image border, corner, or irrelevant parts of the image (*cocker spaniel* in Fig. 4), which results in bad deletion and insertion scores. And the randomness on the mask indeed limits the performance of RISE (*Jellyfish* in Fig. 4).

Ablation Studies: We show the results of ablation studies in Table 3. From Table 3 we observe that without the TV term, insertion scores would indeed suffer significantly while deletion scores won't change much, indicating that the TV term is important to avoid adversarial masks. The random noise introduced in Sec 3.5 is very useful when the resolution of the mask is high (e.g, 224×224). From Fig. 6 we observe that I-GOS with noise can achieve much better insertion score than without noise for the same insertion ratio. When the resolution is low (e.g, 28×28), the noise

Table 3. The results of the ablation study on VGG19.

	224×224		28×28	
I-GOS	Deletion	Insertion	Deletion	Insertion
Ours	0.0336	**0.5246**	0.0899	**0.5701**
No TV term	**0.0308**	0.3712	**0.0841**	0.5181
No noise	0.0559	0.4194	0.0872	0.5634
Fixed step size	0.0393	0.5024	0.0906	0.5403

Table 4. Comparative evaluation in terms of running time (averaged on $5,000$ images) on ImageNet dataset using ResNet50 as the base model.

Running time (s)	224×224	112×112	28×28	14×14	7×7
Mask [8]	17.03	14.61	14.66	14.35	14.24
GradCam [21]	– –	– –	– –	– –	**<1**
RISE [18]	61.77	– –	– –	– –	– –
Integrated Gradients [26]	**<1**	– –	– –	– –	– –
I-GOS (ours)	6.07	**5.73**	**5.70**	**5.63**	5.62

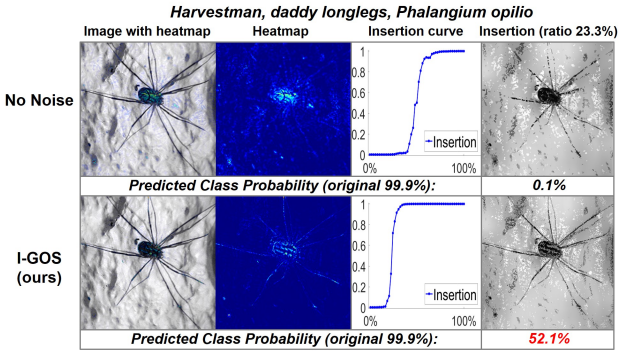


Figure 6. Examples from ablation studies in the size of 224×224. With noise added to the integrated gradient, the heatmap successfully reveals the entire legs of *daddy longlegs*, leading to better classification from the CNN, whereas without noise it is more adversarial (maybe merely by breaking each leg, CNN confidence is already reduced), leading to worse insertion metric.

is not that important since low resolution can already avoid adversarial examples. When we utilize a fixed step size (the step size is 1 in Table 3), both deletion and insertion scores become worse, showing the utility of the line search.

Running Time: In Table 4, we summarize the average running time for Mask, RISE, GradCam, Integrated Gradients, and I-GOS on ImageNet dataset using ResNet50 as the base model. For each approach, we only use one Nvidia 1080Ti GPU. For I-GOS, the maximal iteration is 15; for Mask, the maximal iteration is 500; for RISE, the number of random input samples is $8,000$. Combined with Table 2 and Table 4, we observe that our approach utilizes less time to generate better results than Mask and RISE. Especially, not a lot of iterations need to be used. For I-GOS, the number of iterations to converge is 13 and the time for each iteration is $0.38s$. The average running times for the backpropagation-based methods are all less than 1 second. However, our approach achieve much better performance than these approaches, especially with higher resolutions. To the best of our knowledge, our approach I-GOS is the fastest among the perturbation-based methods, as well as

Table 5. Mean accuracy (%) in the pointing game for VGG16 on MSCOCO and PASCAL VOC07, respectively.

Mean Acc (%)	MSCOCO	VOC07
AM [23]	37.10	76.00
Deconv [28]	38.60	75.50
MWP [29]	39.50	76.90
Excitation BP [29]	49.60	80.00
RISE [18]	**50.71**	**87.33**
Mask [8] (14×14)	40.03	79.45
Mask [8] (28×28)	43.24	77.57
I-GOS (ours) (14×14)	47.16	**85.81**
I-GOS (ours) (28×28)	**49.62**	83.61

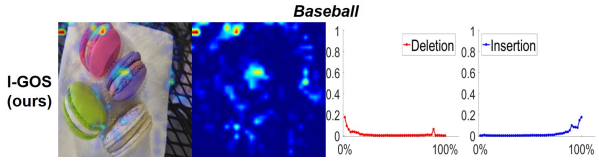


Figure 7. One failure case for I-GOS, insertion curve does not move until almost all pixels have been inserted.

the one with the best performance in deletion and insertion metrics among all heatmap approaches.

Failure Case: Fig. 7 shows one failure case, where the insertion score did not increase till the end. Our observation is that optimization-based methods such as I-GOS usually do not work well when the deep model's prediction confidence is very low (less than 0.01), or when the deep model makes a wrong prediction.

Pointing Game: Table 5 shows the comparative evaluations of I-GOS with other state-of-the-art approaches in terms of mean accuracy in the pointing game on MSCOCO and PASCAL, respectively. Here we utilize the same pretrained models from [18]. Hence, we list the pointing game accuracies reported in the paper except for Mask and our approach I-GOS. From Table 5 we observe that, I-GOS beats all the other compared approaches except for RISE, and it improves significantly over of the Mask. During the experiments we notice that, some object labels for MSCOCO and PASCAL in the pointing game have very small output scores for the pretrained VGG16 models, which affects the optimization greatly for both Mask and I-GOS. RISE does not seem to suffer from this. We believe RISE may be good at the pointing game, but its randomness would generally lead to a mask that is too diffuse, which significantly hurts its deletion and insertion scores (Table 1 and 2), while our approach generates a much more concise heatmap.

5. Conclusion

In this paper, we propose a novel visualization approach I-GOS, which utilizes integrated gradients to optimize for a heatmap. We show that the integrated gradients provides a better direction than the gradient when a good baseline is known for part of the objective of the optimization. The heatmaps generated by the proposed approach are human-

understandable and more correlated to the decision-making of the model. Extensive experiments are conducted on three benchmark datasets with four pretrained deep neural networks, which shows that I-GOS advances state-of-the-art deletion and insertion scores on all heatmap resolutions.

References

[1] M. Ancona, E. Ceolini, A. C. Öztireli, and M. H. Gross. A unified view of gradient-based attribution methods for deep neural networks. *CoRR*, abs/1711.06104, 2017. 2

[2] S. Bach, A. Binder, G. Montavon, F. Klauschen, K. Müller, and W. Samek. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PLoS One*, 10, 7 2015. 2

[3] D. Bau, B. Zhou, A. Khosla, A. Oliva, and A. Torralba. Network dissection: Quantifying interpretability of deep visual representations. In *Computer Vision and Pattern Recognition*, 2017. 1

[4] P. Dabkowski and Y. Gal. Real time image saliency for black box classifiers. In *NIPS*, 2017. 2

[5] Y. Dong, F. Liao, T. Pang, H. Su, J. Zhu, X. Hu, and J. Li. Boosting adversarial attacks with momentum. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 10

[6] E. Elenberg, A. G. Dimakis, M. Feldman, and A. Karbasi. Streaming weak submodularity: Interpreting neural networks on the fly. In *Advances in Neural Information Processing Systems*, pages 4044–4054, 2017. 1

[7] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results. http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html. 6

[8] R. C. Fong and A. Vedaldi. Interpretable explanations of black boxes by meaningful perturbation. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 3449–3457, Oct 2017. 1, 2, 3, 5, 6, 7, 8

[9] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 2672–2680. Curran Associates, Inc., 2014. 5

[10] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016. 6

[11] S. Johnson and T. Subha. A study on eye fixation prediction and salient object detection in supervised saliency. *Materials Today: Proceedings*, 4(2, Part B):4169 – 4181, 2017. International Conference on Computing, Communication, Nanophotonics, Nanoscience, Nanomaterials and Nanotechnology. 2

[12] S. S. S. Kruthiventi, V. Gudisa, J. H. Dholakiya, and R. Venkatesh Babu. Saliency unified: A deep architecture for simultaneous eye fixation prediction and salient object segmentation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016. 2

[13] M. Kummerer, T. S. A. Wallis, L. A. Gatys, and M. Bethge. Understanding low- and high-level contributions to fixation prediction. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017. 2

[14] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, editors, *Computer Vision – ECCV 2014*, pages 740–755, Cham, 2014. Springer International Publishing. 6

[15] S. M. Lundberg and S.-I. Lee. A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems*, pages 4765–4774, 2017. 1

[16] W. Nie, Y. Zhang, and A. Patel. A Theoretical Explanation for Perplexing Behaviors of Backpropagation-based Visualizations. *ArXiv e-prints*, May 2018. 1, 6, 7

[17] J. Nocedal and S. Wright. *Numerical Optimization*. Springer Series in Operations Research and Financial Engineering. Springer New York, 2000. 4

[18] V. Petsiuk, A. Das, and K. Saenko. RISE: Randomized Input Sampling for Explanation of Black-box Models. *ArXiv e-prints*, June 2018. 2, 6, 7, 8

[19] M. T. Ribeiro, S. Singh, and C. Guestrin. Why should i trust you?: Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1135–1144. ACM, 2016. 1

[20] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision*, 2015. 6

[21] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 618–626, 2017. 1, 2, 6, 7, 8

[22] A. Shrikumar, P. Greenside, A. Shcherbina, and A. Kundaje. Not just a black box: Learning important features through propagating activation differences. *CoRR*, abs/1605.01713, 2016. 2, 3

[23] K. Simonyan, A. Vedaldi, and A. Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *ICLR Workshop*, 2014. 1, 2, 3, 8

[24] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *International Conference on Learning Representations*, 2015. 6

[25] J. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller. Striving for simplicity: The all convolutional net. In *ICLR Workshop*, 2015. 1, 2

[26] M. Sundararajan, A. Taly, and Q. Yan. Axiomatic attribution for deep networks. In D. Precup and Y. W. Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, pages 3319–3328. PMLR, 2017. 1, 2, 3, 6, 7, 8

[27] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus. Intriguing properties of neural networks. In *International Conference on Learning Representations*, 2014. 5

[28] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, editors, *Computer Vision – ECCV 2014*, pages 818–833, Cham, 2014. Springer International Publishing. 1, 2, 8

[29] J. Zhang, Z. Lin, J. Brandt, X. Shen, and S. Sclaroff. Top-down neural attention by excitation backprop. In *European Conference on Computer Vision*, pages 543–559. Springer, 2016. 1, 2, 5, 6, 8

[30] B. Zhou, A. Khosla, À. Lapedriza, A. Oliva, and A. Torralba. Object detectors emerge in deep scene cnns. *CoRR*, abs/1412.6856, 2014. 2

[31] B. Zhou, Y. Sun, D. Bau, and A. Torralba. Interpretable basis decomposition for visual explanation. In *The European Conference on Computer Vision (ECCV)*, September 2018. 1

Supplementary Material

I. Properties of the Integrated Gradient in Quadratic Functions

Proposition 1. *The integrated gradients reduce to a scaling on the conventional gradient in a quadratic function if the baseline is the optimum.*

Proof. Given a quadratic function $f(\mathbf{x}) = \mathbf{x}^T A\mathbf{x} + b^T\mathbf{x} + c$, we have its conventional gradient as: $\nabla f(\mathbf{x}) = (A + A^T)\mathbf{x} + b$. Considering a straight-line path from the current point $\mathbf{x}_k$ to the baseline $\mathbf{x}_0$, for point $\mathbf{x}_s$ along the path, we have: $\mathbf{x}_s = \mathbf{x}_0 + \frac{s}{S}(\mathbf{x}_k - \mathbf{x}_0)$,

$$\begin{aligned}\nabla f(\mathbf{x}_s) &= (A + A^T)\mathbf{x}_s + b \\ &= (A + A^T)\left(\mathbf{x}_0 + \frac{s}{S}(\mathbf{x}_k - \mathbf{x}_0)\right) + b \\ &= \frac{s}{S}(A + A^T)\mathbf{x}_k + \frac{S-s}{S}(A + A^T)\mathbf{x}_0 + b \\ &= \frac{s}{S}\nabla f(\mathbf{x}_k) + \frac{S-s}{S}\nabla f(\mathbf{x}_0), \end{aligned} \tag{11}$$

Thus, we obtain the integrated gradient along the straight-line path as:

$$\begin{aligned}\nabla^{IG} f(\mathbf{x}_k) &= \frac{1}{S}\sum_{s=1}^{S} \nabla f(\mathbf{x}_s) \\ &= \frac{S+1}{2S}\nabla f(\mathbf{x}_k) + \frac{S-1}{2S}\nabla f(\mathbf{x}_0), \end{aligned} \tag{12}$$

When the baseline $\mathbf{x}_0$ is the optimum of the quadratic function, $\nabla f(\mathbf{x}_0) = 0$, and then

$$\nabla^{IG} f(\mathbf{x}_k) = \frac{S+1}{2S}\nabla f(\mathbf{x}_k). \tag{13}$$

Hence, the integrated gradients reduce to a scaling on the conventional gradient.

In this case, the revised Armijo condition also reduces to the conventional Armijo condition up to a constant. □

II. Adversarial Examples

Figure 8-9 shows some examples when using I-GOS to visualize adversarial examples. Here we utilize the MI-FGSM method [5] on VGG19 to generate adversarial examples. From Fig. 8-9 we observe that the heatmaps for the original images and for the adversarial examples generated by I-GOS are totally different. For the original image, I-GOS can often lead to a high classification confidence on the original class by inserting a small portion of the pixels. For the adversarial image though, almost the entire image needs to be inserted for CNN to predict the adversarial category. We note that we are not presenting I-GOS as a defense against adversarial attacks, and that specific attacks may be designed targeting the salient regions in the image. However, these figures show that the I-GOS heatmap and the insertion metric are robust against those full-image based attacks and not performing mere image reconstruction.

III. Deletion and Insertion Visualizations

Fig. 10 shows more comparison examples between different approaches on 224×224 heatmaps. Fig. 11 shows more visual comparisons between our approach, GradCAM, and RISE. From Fig. 10 we can see that, for Mask, it focuses on person instead of *Yawl* on the left image, and focuses on grass instead of *Impala* on the right image, indicating that sometimes the optimization can be stuck in a bad local optimum. From Fig. 11 we observe that sometimes GradCAM also fires on image border, corner, or irrelevant parts of the image (*Grey whale* in Fig. 11), which results in bad deletion and insertion scores. And the randomness on the mask indeed limits the performance of RISE (*West Highland white terrier* in Fig. 11).

Fig. 12-13 show some examples generated by our approach I-GOS in the deletion and insertion task using VGG19 as the baseline model. Fig. 14-15 show some examples generated by I-GOS in the deletion and insertion task using Resnet50 as the baseline model. The deletion or insertion image is generated by $I_0 \odot \text{up}(M) + \tilde{I}_0 \odot (\mathbf{1} - \text{up}(M))$, where the resolution of M is 28×28. For deletion image, we initialize the mask M as matrix of ones, then set the top N pixels in the mask to 0 based on the values of the heatmap, where the deletion ratio represents the proportion of pixels that are set to 0. For insertion image, we initialize mask M as matrix of zeros, then set the top N pixels in the mask to 1 based on the values of the heatmap, where the insertion ratio represents the proportion of pixels that are set to 1. In Fig. 12-15, the masked/revealed regions of the images may seem a little larger than the number of the deletion/insertion ratios. The reason is that after upsampling the mask M, some pixels on the border may have values between 0 and 1, resulting in larger regions to be masked or revealed. The predicted class probability is the output value after softmax for the same category using

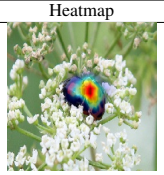
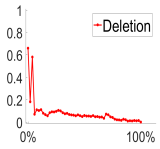
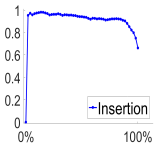
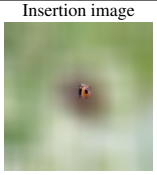
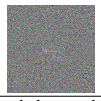
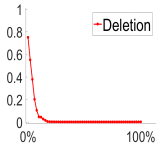
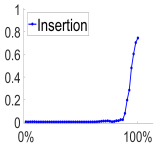
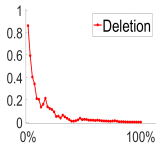
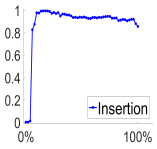
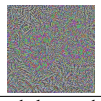
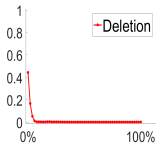
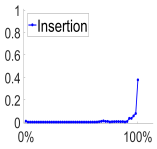
Label: *'Ladybug, ladybeetle'*	**Original image**	Heatmap	Deletion curve	Deletion image	Insertion curve	Insertion image
						
Predicted class probability	66%	– –	– –	10.1%	– –	51.9%
Deletion or Insertion ratio	– –	– –	– –	2.4%	– –	0.5%
Label: *'Persian cat'* Diff: 	**Adversarial example**	Heatmap	Deletion curve	Deletion image	Insertion curve	Insertion image
						
Predicted class probability	75%	– –	– –	14%	– –	37.8%
Deletion or Insertion ratio	– –	– –	– –	6.9%	– –	92.3%
Label: *'Teapot'*	**Original image**	Heatmap	Deletion curve	Deletion image	Insertion curve	Insertion image
						
Predicted class probability	86%	– –	– –	16.7%	– –	69.5%
Deletion or Insertion ratio	– –	– –	– –	9.7%	– –	5.2%
Label: *'Persian cat'* Diff: 	**Adversarial example**	Heatmap	Deletion curve	Deletion image	Insertion curve	Insertion image
						
Predicted class probability	44.3%	– –	– –	8.3%	– –	41.3%
Deletion or Insertion ratio	– –	– –	– –	3.4%	– –	98.9%

Figure 8. The top row are original images and their heatmaps generated by I-GOS; the bottom row are adversarial examples and their heatmaps generated by I-GOS. The red plot illustrates how the CNN predicted probability drops with more areas masked, and the blue plot illustrates how the prediction increases with more areas revealed. The x axis for the red/blue plot represents the percentage of pixels masked/revealed; the y axis for the red/blue plot represents the predicted class probability. One can see on normal images, CNN can classify with only the highlighted parts revealed, whereas on adversarial images one would almost need to insert the entire image to make the CNN to classify to the adversarial label.

the original image, the deletion image, and the insertion image as the input, respectively. From Fig. 12-15 we observe that the proposed approach I-GOS can utilize a low deletion ratio to achieve a low predicted class probability for the deletion task, and a low insertion ratio to achieve a high predicted class probability for the insertion task at the same time, indicating that I-GOS truly discovers the key features of the images that the CNN network is using. Especially, we realize that CNN is indeed fixating on very small regions in the image and very local features in many cases to make a prediction, e.g. in *Pomeranian*, the face of the dog is utmostly important. Without the face the prediction is reduced to almost zero, and with only the face and a rough outline of the dog, the prediction is almost perfect. The same can be said for *Eft*, *Black grouse*, *lighthouse* and *boxer*. Interestingly, for *Container ship* and *trailer truck*, their functional parts are extremely important to the classification. *Trailer truck* almost cannot be classified without the wheels (and could be classified with only the wheels), and *container ship* cannot be classified without the containers (and could be classified with almost only the containers and a rough outline of the ship).

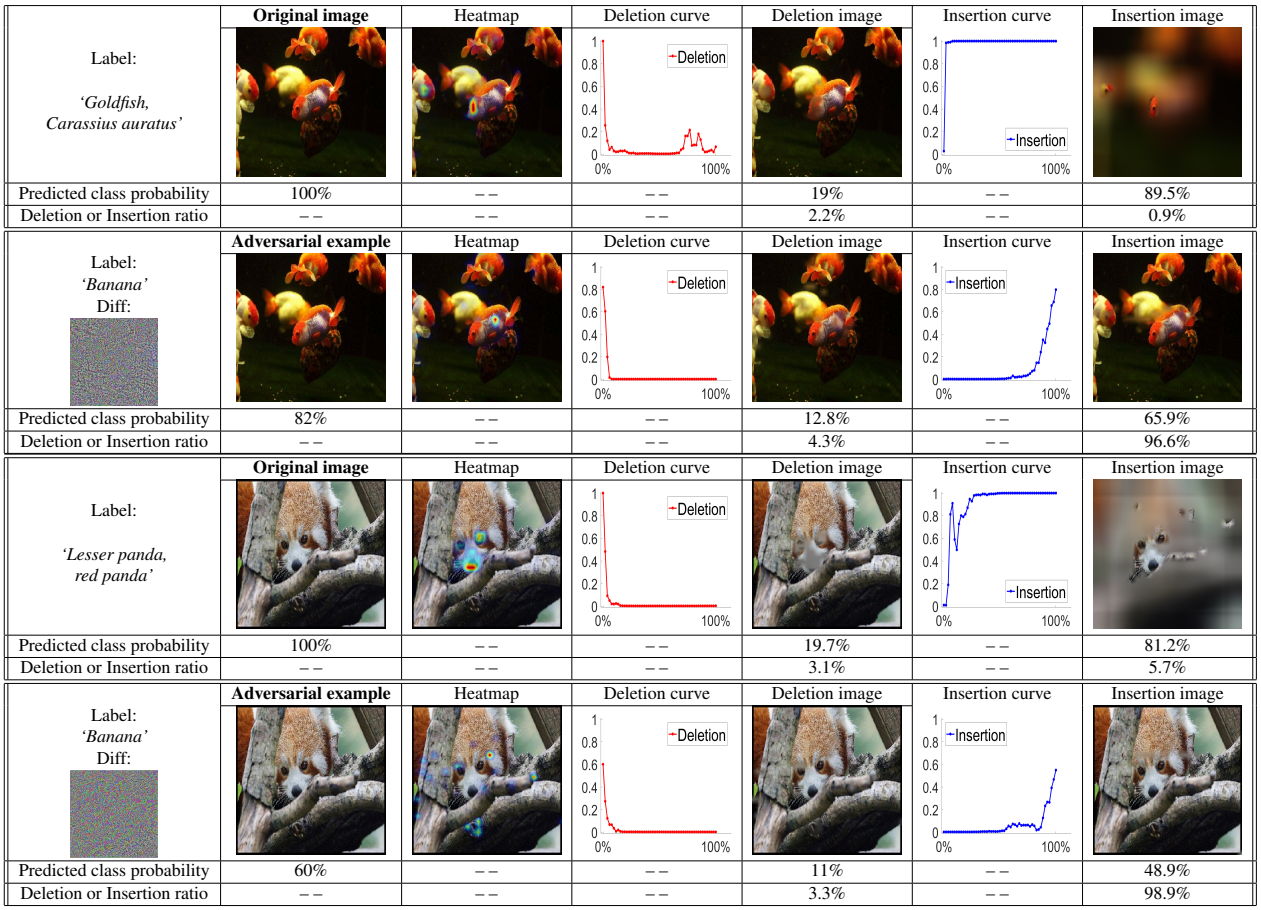


Figure 9. The top row are original images and their heatmaps generated by I-GOS; the bottom row are adversarial examples and their heatmaps generated by I-GOS, see Fig. 8 caption for explanations of the meaning of the curves. One can see on normal images, CNN can classify with only the highlighted parts revealed, whereas on adversarial images one would almost need to insert the entire image to make the CNN to classify to the adversarial label.

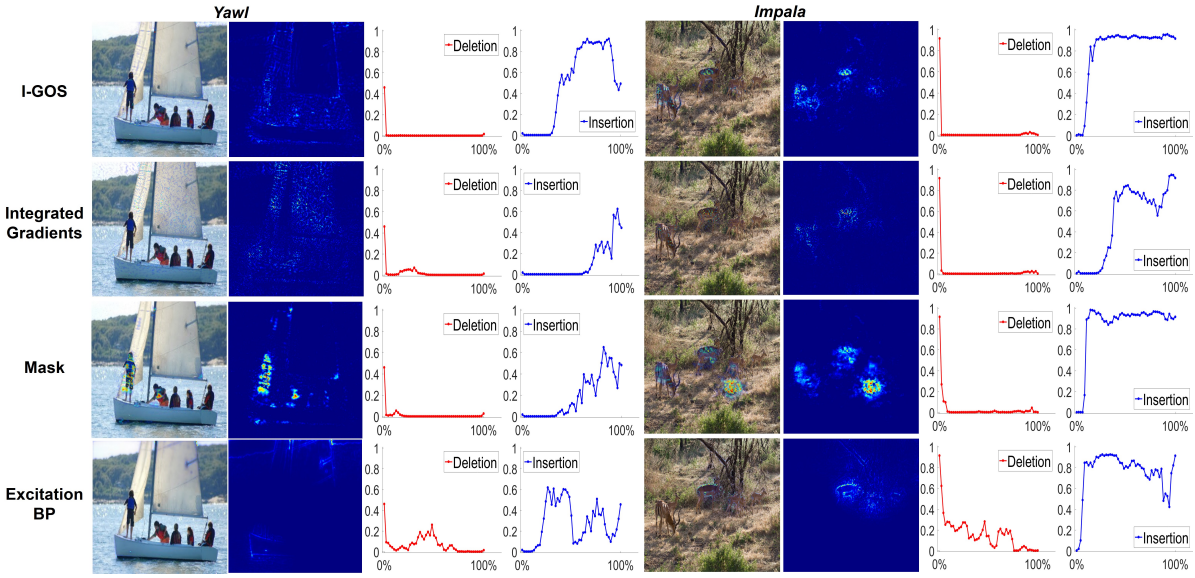


Figure 10. A comparison among different approaches with heatmaps of 224×224 resolution. The red plot illustrates how the CNN predicted probability drops with more areas masked, and the blue plot illustrates how the prediction increases with more areas revealed. The x axis for the red/blue plot represents the percentage of pixels masked/revealed; the y axis for the red/blue plot represents the predicted class probability. One can see with I-GOS the red curve drops earlier and the blue plot increases earlier, leading to more area under the insertion curve (insertion metric) and less area under the deletion curve (deletion metric). (Best viewed in color)

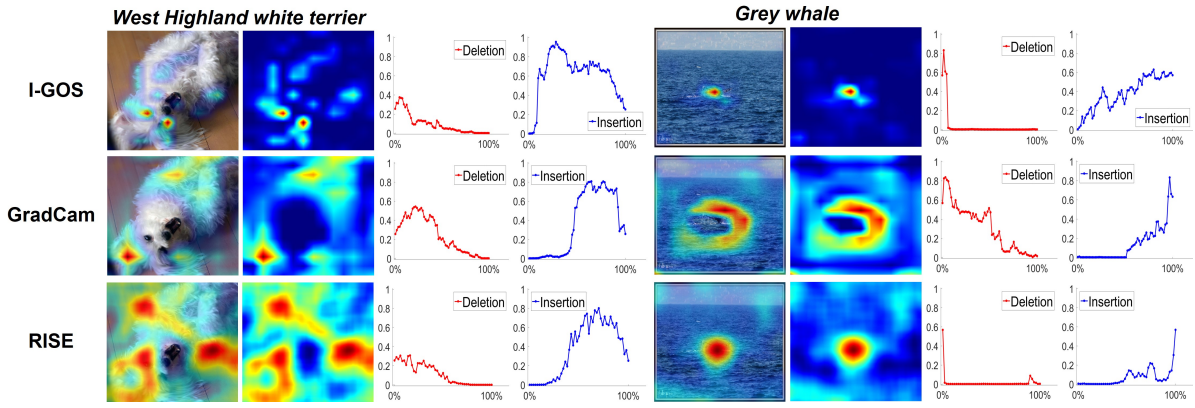
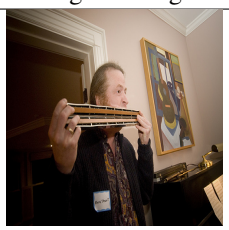
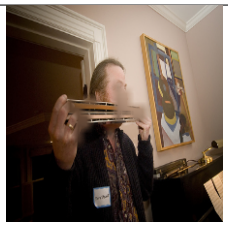
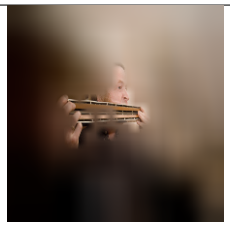
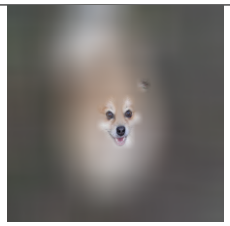


Figure 11. Comparisons between GradCam, RISE, and I-GOS, see Fig. 10 caption for explanations of the meaning of the curves.

Label	Original Image	Deletion	Insertion
27: 'Eft'			
Predicted class probability	99.6%	14%	97%
Deletion or Insertion ratio	– –	6.1%	1.5%

Label	Original Image	Deletion	Insertion
409: 'Analog clock'			
Predicted class probability	34.1%	4.3%	35.5%
Deletion or Insertion ratio	– –	1.9%	0.8%

Label	Original Image	Deletion	Insertion
593: 'Harmonica, mouth organ, harp, mouth harp'			
Predicted class probability	99.9%	11.9%	81.8%
Deletion or Insertion ratio	– –	3.1%	4.6%

Label	Original Image	Deletion	Insertion
259: 'Pomeranian'			
Predicted class probability	100%	4.8%	82.9%
Deletion or Insertion ratio	– –	3.4%	2.3%

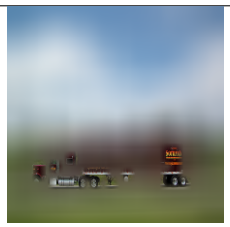
Label	Original Image	Deletion	Insertion
867: 'Trailer truck, tractor trailer, trucking rig, rig, articulated lorry, semi'			
Predicted class probability	99.7%	14.3%	86.2%
Deletion or Insertion ratio	– –	8.0%	3.1%

Figure 12. Examples generated by I-GOS in the deletion and insertion task using VGG19 as the baseline model.

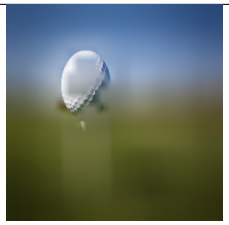
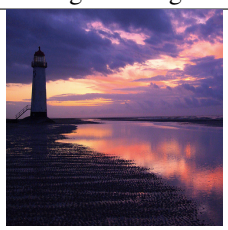
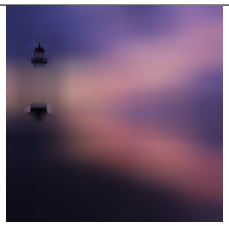
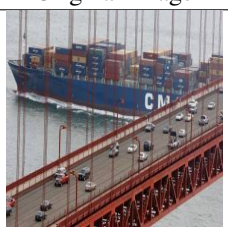
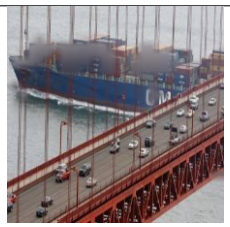
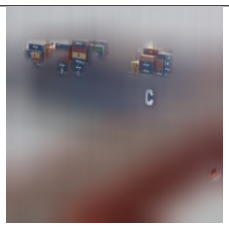
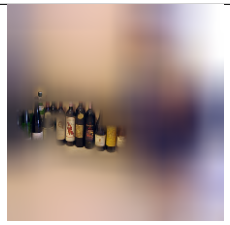
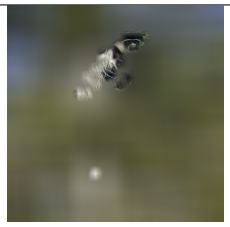
Label	Original Image	Deletion	Insertion
574: 'Golf ball'			
Predicted class probability	100%	19.6%	85.1%
Deletion or Insertion ratio	– –	21.0%	3.4%
Label	Original Image	Deletion	Insertion
80: 'Black grouse'			
Predicted class probability	99.5%	0.9%	99.7%
Deletion or Insertion ratio	– –	2.7%	0.8%
Label	Original Image	Deletion	Insertion
437: 'Beacon, lighthouse, beacon light, pharos'			
Predicted class probability	95.7%	12.3%	78.1%
Deletion or Insertion ratio	– –	0.8%	1.5%
Label	Original Image	Deletion	Insertion
259: 'African hunting dog, hyena dog, Cape hunting dog, Lycaon pictus'			
Predicted class probability	97.2%	15.7%	80.9%
Deletion or Insertion ratio	– –	3.4%	7.7%
Label	Original Image	Deletion	Insertion
510: 'Container ship, containership, container vessel'			
Predicted class probability	98.5%	19.4%	89.5%
Deletion or Insertion ratio	– –	7.7%	5.4%

Figure 13. Examples generated by I-GOS in the deletion and insertion task using VGG19 as the baseline model.

Label	Original Image	Deletion	Insertion
440: 'Beer bottle'			
Predicted class probability	52.1%	3.6%	47.9%
Deletion or Insertion ratio	– –	3.8%	5.9%

Label	Original Image	Deletion	Insertion
517: 'Crane'			
Predicted class probability	98.2%	18.8%	96.6%
Deletion or Insertion ratio	– –	4.1%	6.6%

Label	Original Image	Deletion	Insertion
920: 'Traffic light, traffic signal, stoplight'			
Predicted class probability	100%	18.7%	95.1%
Deletion or Insertion ratio	– –	23.5%	1.3%

Label	Original Image	Deletion	Insertion
375: 'Colobus, colobus monkey'			
Predicted class probability	100%	9.3%	85%
Deletion or Insertion ratio	– –	4.8%	3.1%

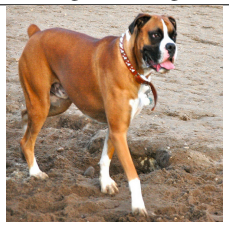
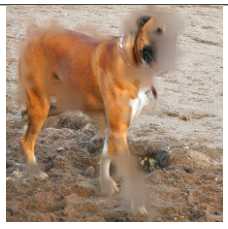
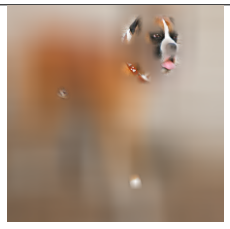
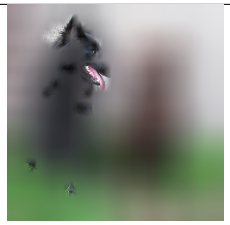
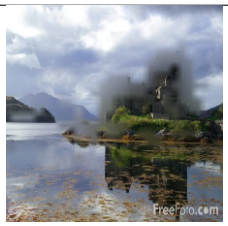
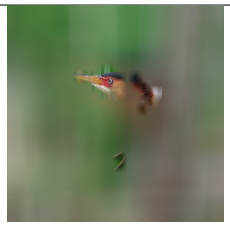
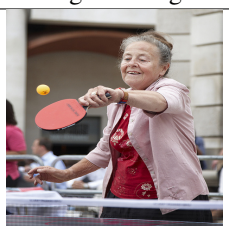
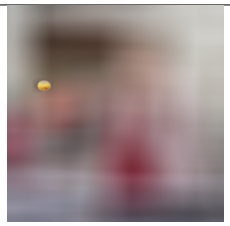
Label	Original Image	Deletion	Insertion
242: 'Boxer'			
Predicted class probability	99.4%	15.5%	83.2%
Deletion or Insertion ratio	– –	18.1%	2.8%

Figure 14. Examples generated by I-GOS in the deletion and insertion task using Resnet50 as the baseline model.

Label	Original Image	Deletion	Insertion
224: 'Groenendael'			
Predicted class probability	92.1%	10.6%	88.1%
Deletion or Insertion ratio	– –	3.3%	4.3%

Label	Original Image	Deletion	Insertion
483: 'Castle'			
Predicted class probability	100%	13.4%	80.4%
Deletion or Insertion ratio	– –	11.5%	2.8%

Label	Original Image	Deletion	Insertion
133: 'Bittern'			
Predicted class probability	98.5%	18.7%	81.7%
Deletion or Insertion ratio	– –	0.8%	3.1%

Label	Original Image	Deletion	Insertion
722: 'Ping-pong ball'			
Predicted class probability	99.8%	17.6%	94.9%
Deletion or Insertion ratio	– –	4.1%	0.3%

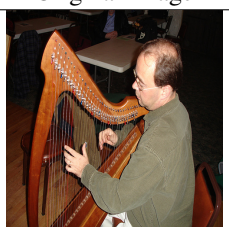
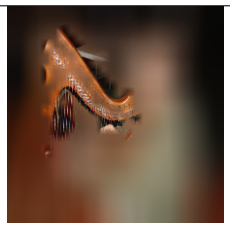
Label	Original Image	Deletion	Insertion
594: 'Harp'			
Predicted class probability	100%	14.7%	89.1%
Deletion or Insertion ratio	– –	21.4%	8.7%

Figure 15. Examples generated by I-GOS in the deletion and insertion task using Resnet50 as the baseline model.