Diverse Explanations for Object Detectors with Nesterov-Accelerated iGOS++

Mingqi Jiang
jiangmi@oregonstate.edu
Saeed Khorram
khorrams@oregonstate.edu
Li Fuxin
lif@oregonstate.edu

Collaborative Robotics and Intelligent
Systems Institute
Oregon State University
Corvallis, USA

Abstract

Object detection is a crucial task in computer vision, with applications ranging from autonomous driving to surveillance systems. However, few have approached the problem of explaining object detections to gain more insights. In this paper, we extend iGOS++, an explanation algorithm of image classification models, to the task of object detection. Our extension consists of two novel aspects. The first is to utilize Nesterov Accelerated Gradient (NAG) to improve the optimization with integrated gradients. This significantly improves over the line search used in the original work in terms of both speed and quality. Besides, we propose to generate diverse explanations via different initializations of the optimization algorithm, which can better showcase the robustness of the network under different occlusions. To evaluate the effectiveness of our algorithm, we conduct experiments on the MS COCO and PASCAL VOC datasets. Results demonstrate that our approach significantly outperforms existing methods in terms of both explanation quality and speed. Besides, the diverse explanations it generates give more insight into the (sometimes erroneous) mechanisms underlying deep object detectors.

1 Introduction

Saliency map, also known as heatmap explanations have been popular in recent years. These explanations highlight areas in an image that are important for deep network classification, which helps us gain more insights into those networks. However, most existing work focus solely on image classification, which limits their application in downstream tasks such as object detection and instance segmentation. Furthermore, most heatmap approaches generate only a single heatmap for each image, including recent work [15] that attempts to explain object detectors. This could fall short of providing a complete picture of the network it is attempting to explain, since the network may exhibit robustness under different kinds of occlusions [21], whereas a single heatmap can be seen as only one type of occlusion.

When extending explanation algorithms to detection/segmentation tasks, speed and resolution are two important aspects to consider. Detection/segmentation algorithms usually operate at considerably higher resolutions than classification networks, and more often need

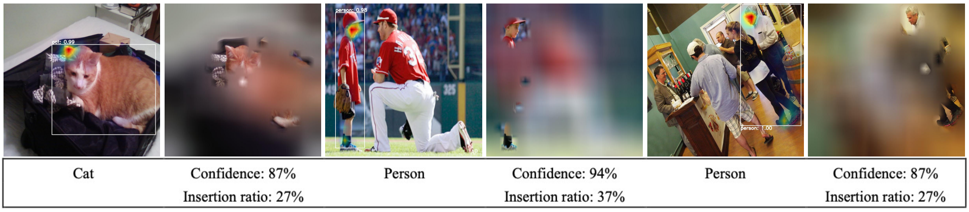


Figure 1: We present examples generated by our proposed NAG-iGOS++ approach at 25×25 resolution using Mask R-CNN. In the generated explanations, importance is indicated by a color gradient ranging from strong (red) to weak (blue). Our approach reveals that the model often focuses only on a subset of object parts, such as ears of the cat, or the head and feet of the persons. The explanation in the last two columns revealed that although the bounding box is detected correctly, the network was erroneously looking at the head and feet of two different persons, indicating its lack of deep understanding of the concept of a person.

to detect objects of much smaller scale. This calls for the explanation algorithm to be able to run efficiently, and be of sufficient resolution to properly explain detections/segmentations that might be very small. However, the state-of-the-art [15] for explaining detection algorithms needs more than 150 seconds per image, and resolutions of 16×16 or 25×25 in prior work often fall short of the need in explaining the predictions on small objects.

In this paper we extend the iGOS++ algorithm [8] into explaining detection networks. iGOS++ is capable of avoiding adversarial masks at higher resolutions that helps us to generate explanations of much higher resolutions than prior work. To overcome speed challenges, we propose to replace the line search algorithm used in [8] with Nesterov Accelerated Gradient (NAG), leading to a 2× speedup. Our final algorithm is over 3× more efficient than the state-of-the-art [15]. To provide multifaceted explanations for each detection, we propose optimizing with **multiple initializations**, enabling the generation of different high resolution explanations even for the same detection. Experiment results demonstrate that this approach significantly enhances the quality of explanations while maintaining efficiency. Additionally, our explanations sometimes reveal (Fig. 1) that the network may focus on a subset of the detection region and may erroneously merge different parts of two distinct objects.

In summary, in this paper, we make the following contributions:

- We extend iGOS++ to object detection tasks. Results show that the performance and speed of our method significantly improves over state-of-the-art.
- We propose to use Nesterov Accelerated Gradient (NAG) in iGOS++ to replace the line search, which speeds up the algorithm by 2× and improves the performance.
- We propose a scheme that initializes iGOS++ with multiple starting masks, which further improved performance and makes the algorithm capable of generating multiple explanations for a single detection. The generated explanations can give more insights to the robustness of the network to different occlusions.

2 Related work

2.1 Saliency Maps

Heatmap visualization approaches can be primarily categorized into *gradient-based* and *perturbation-based* approaches. Gradient-based approaches primarily employ (modified) gradients of the model with respect to the input features or activations to gauge their importance to the prediction of the network. [22] directly outputs the magnitude of the gradient of the class-specific outputs with respect to the input features as saliency maps. Grad-CAM [20] calculates the gradient with respect to the last convolution layer activations which extends CAM [28]. SmoothGrad [23] adds noise to the input image to produce a more robust explanation. Integrated Gradients [24] computes the sum of gradients at multiple locations along a straight line between a baseline image and the input image to determine the contribution of each pixel to the final prediction. However, gradient-based methods often face limitations such as insensitivity to class-specific parameters, vulnerability to adversarial attacks, and inflexibility in generating explanations at desired resolutions.

Perturbation-based approaches introduce modifications to the input image and observe the corresponding changes to the model prediction. [27] occludes patches of pixels over the input image and observes the resulting change in the output of the model. Later, methods such as LIME (Local Interpretable Model-Agnostic Explanations) [18] and RISE (Randomized Input Sampling for Explanation) [14] were introduced to explain the predictions made by black-box models. Score-CAM [26] uses a weighted combination of the activation maps based on their forward-pass score on the target class to generate the attribution maps. [19] restricts the flow of information by perturbing the activation rather than input features. Despite their merits, perturbation-based methods *can* shift images off their original data manifold and be computationally expensive, limiting their suitability for real-time applications [18].

Methods such as meaningful perturbations [3, 4] and Integrated-Gradient Saliency Maps (I-GOS) combine both perturbation-based and gradient-based methods in order to apply informed perturbations rather than random ones [14]. I-GOS combines [4] and [24] which utilizes integrated gradients as descent directions in an optimization algorithm. It uses a smoothness regularization term to reduce noise and improve the visual quality of the generated heatmaps. Subsequently, iGOS++ enhances I-GOS by considering both the removal and preservation of evidence during optimization and introduces a Bilateral Total Variation term to reduce heatmap dispersion, thereby improving the quality of the generated heatmaps.

2.2 Explanations for Object Detection Models

Recently, visualizing the decisions made by object detection models has gained more attention. [5] uses gradient backpropagation to approximately estimates SHAP (Shapley additive explanations) [12] values for assessing feature importance and proposes the "Explain to Fix" (E2X) framework. [25] calculates the correlation between the output of the model and each input feature for every predicted bounding box in the SSD detector [11], and create a heatmap highlighting the input features that are highly correlated to the output. More recently, D-RISE [15] was proposed for generating visual explanations for object detectors, which extends the masking technique RISE [14]. [7] proposed using pixel-wise feature attribution from approximate SHAP [12] in object detection pipelines for satellite images.

3 Method

3.1 Revisiting iGOS++

We build on top of iGOS++ [8], a recently proposed state-of-the-art heatmap visualization approach. This method identifies the most important areas of an input image for a given black-box network f outputting the class-conditional probability $f_c(I_0)$ (henceforth referred to as *prediction confidence*) for an input image I_0 on each class c. It optimizes for a deletion mask M_D and an insertion mask M_I to identify the areas that have significant impact on the prediction confidence of the model. Specifically, the deletion mask M_D is optimized to identify the areas that would lead to a significant decrease in the prediction confidence when removed from the input image I_0. Simultaneously, the insertion mask M_I is optimized to identify the evidence that would lead to high prediction confidence. Finally, an Hadamard product of the two masks give us the final mask M_{DI} for the target class c, on which it optimizes for both the deletion and the insertion losses. More formally:

$$\begin{aligned} \min_{M=(M_D,M_I)} \; & F_c(I_0,M) = f_c(\Phi(I_0,\tilde{I_0},M_D)) - f_c(\Phi(I_0,\tilde{I_0},1-M_I)) \\ & \qquad + f_c(\Phi(I_0,\tilde{I_0},M_{DI})) - f_c(\Phi(I_0,\tilde{I_0},1-M_{DI})) + g(M_{DI})) \\ s.t. \quad & g(M_{DI})) = \lambda_1 ||1{-}M_{DI}||_1 + \lambda_2 \mathrm{BTV}(M_{DI}), \quad \Phi(I_0,\tilde{I_0},M) = I_0 \odot M + \tilde{I_0} \odot (1-M) \\ & M_{DI} = M_D \odot M_I; \qquad 0 \leq M_D, M_I \leq 1 \end{aligned} \tag{1}$$

where $\tilde{I_0}$ is a baseline image with $f_c(\tilde{I_0})$ close to zero, $\odot$ is the Hadamard product, λ_1 and λ_2 are hyperparameters controlling the contribution of individual constraints used in the regularization term $g(.)$. The final solution to the optimization problem is M_{DI}. The approach of using these masks M_D, M_I and M_{DI} were shown to improve the performance of the explanation compared with only using M_D. Different from previous algorithms, the **Bilateral Total Variation (BTV)** term is used to prevent the heatmap from being scattered:

$$\mathrm{BTV} = \sum_{u \in \Lambda} e^{-\nabla I(u)^2/\sigma^2} ||\nabla M(u)||_\beta^\beta \tag{2}$$

where $M(u)$ and $I(u)$ represent the mask and the input image value at pixel u from the set of all pixels Λ, respectively, β and σ are hyperparameters.

Integrated Gradient. The optimization problem in eq.(1) is highly non-convex. To overcome slow convergence and local optima issues associated with optimizing using gradient descent [4], [8] used integrated gradient (IG) [24] as the descent direction. The IG of $f_c(M)$ with respect to M is given by:

$$\nabla^{IG}_{I_0} f_c(M) = \frac{1}{S} \sum_{S=1}^{S} \frac{\partial f_c(\Phi(I_0,\tilde{I_0},\frac{s}{S}M))}{\partial M} \tag{3}$$

where it accumulates the conventional gradient along the straight path from the disturbed image to the baseline. Since the baseline is the globally optimal solution for the deletion loss function, utilizing integrated gradients leads the algorithm towards that direction and sometimes can steer away from the local optima that gradient descent [16] tends to obtain.

Backtracking Line Search. To determine the appropriate step size for updating the mask, [8, 16] adapted the Armijo-Goldstein condition [4] and utilized backtracking line search (LS) to keep $F_c(I_0,M)$ minimized:

$$\sum_{S=1}^{S} F_c(\frac{s}{S}(M^k - \alpha^k \cdot TG(M^k))) - \sum_{S=1}^{S} F_c(\frac{s}{S}M^k) \leq -\alpha^k \cdot \beta \cdot TG(M^k)^T TG(M^k) \tag{4}$$

where TG represents the total gradient of $f_c(I_0,M)$, α^k denotes the step size at iteration k, and β is a parameter between 0 and 1.

3.2 NAG-iGOS++: Nesterov Accelerated Gradient Adaptation of iGOS++ to Object Detection

Unlike the classification problem, in object detection and instance segmentation tasks, the combination of the original image and the baseline image may change the generated bounding box proposals. To address this issue, we fix the bounding box proposal associated with each object for two-stage detectors and utilize the classification head of the proposal region to calculate the integrated gradient. For one-stage detectors, we fix the anchor location with the size for each detected object. By doing so, we ensure that the integrated gradient is calculated based on the correct proposal region.

Nesterov Accelerated Gradient. In [8], a backtracking line search was employed to determine the appropriate step size for updating the mask. However, this requires evaluating the integrated gradient of the network multiple times, leading to significant computational cost. There are numerous optimization algorithms that are popular recently in machine learning and deep learning to replace gradient descent. We experimented with several algorithms, including Adaptive Moment Estimation (Adam) [9], Nesterov accelerated gradient (NAG) [13], and Nesterov-accelerated Adaptive Moment Estimation (Nadam) [1]. The results showed that Adam and Nadam did not have better performance than LS, and tend to generate scattered noise pixels in the heatmaps. Only NAG demonstrated good performance and a faster convergence rate than LS. We update the mask with NAG as:

$$\begin{aligned} \omega^{k+1} &= M^k - \alpha \cdot TG(M^k) \\ M^{k+1} &= \omega^{k+1} + \varepsilon(\omega^{k+1} - \omega^k) \end{aligned} \tag{5}$$

where ε is in the range of [0,1), and in our experiments, we set $\varepsilon = k/(k+3)$. α is the learning rate that we set to be the same for all images at the same resolution.

Diverse Initializations. [21] showed that a deep network may be able to correctly and confidently classify images under multiple different occlusions, indicating that there may not exist a unique heatmap for each image/network pair. Hence, conventional heatmap approaches that generate a single heatmap may only provide a part of the complete picture. In [21], multiple solutions are found via a beam search algorithm, which severely limits the resolution of the obtained heatmaps (often to only 7×7).

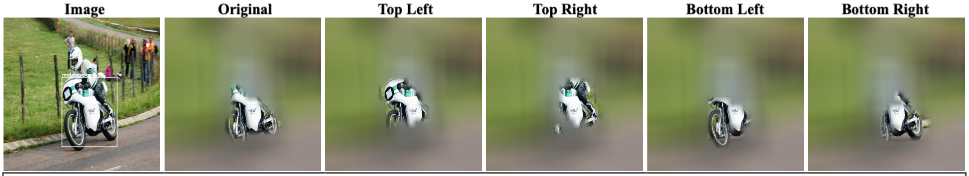
In this paper, we propose to utilize multiple diverse *initializations* to locate diverse heatmaps that may explain the same network. This overcomes the resolution limitation of [21] and allows us to locate diverse high-resolution heatmaps. We propose to generate K^2 different initializations of the heatmap optimization algorithm on a $K \times K$ grid, initializing the mask with nonzero values in only one cell for each initialization. Such diverse initializations could generate different optimization results if there are multiple ways the deep network could output confident predictions. Figure 2 demonstrates the impact of different initializations on the heatmap generated from the same object detection.

4 Experiments

We utilized a pre-trained Mask R-CNN [6] model[1] and the YOLOv3-SPP [17][2] implemented in PyTorch as our base models for all qualitative and quantitative experiments, using the

[1] https://github.com/pytorch/vision

[2] https://github.com/ultralytics/yolov3



Confidence:	90.11%	89.58%	80.25%	80.87%	88.29%
Insertion ratio:	63.16%	57.89%	57.89%	52.63%	52.63%

Figure 2: Examples generated by NAG-iGOS++ without and with initialization using different regions of predicted mask in insertion tasks using Mask R-CNN as the baseline model can be seen in the generated heatmaps. One can see that the network can generate a confident prediction from each different region, highlighting the need for multiple explanations.

2017 Val set of MS-COCO [10] and the Val set of PASCAL VOC 2012 [2]. Detections with a predicted score of 0.5 or greater were considered. For Mask R-CNN, input images were resized to the 800×800 resolution, and heatmaps were generated at three different resolutions: 16×16, 25×25, and 100×100. For YOLOv3-SPP, input images were resized to the 512×512 resolution, and heatmaps were generated at two different resolutions: 16×16 and 64×64. This was done to facilitate a fair comparison with other baselines; D-RISE uses a 16×16 resolution for quantitative results [15], while Grad-CAM was evaluated on Mask R-CNN at the 25×25 resolution and YOLOv3-SPP at the 16×16 resolution. We included a 100×100 resolution for Mask R-CNN and 64×64 for YOLOv3-SPP (both $1/8$ of the input image size) to illustrate the flexibility of choosing the resolution for our method, and note that a higher resolution is better for the explanation of smaller objects. Inference time is computed on a NVidia Quadro RTX 8000. For multiple initializations, we choose $K = 2$.

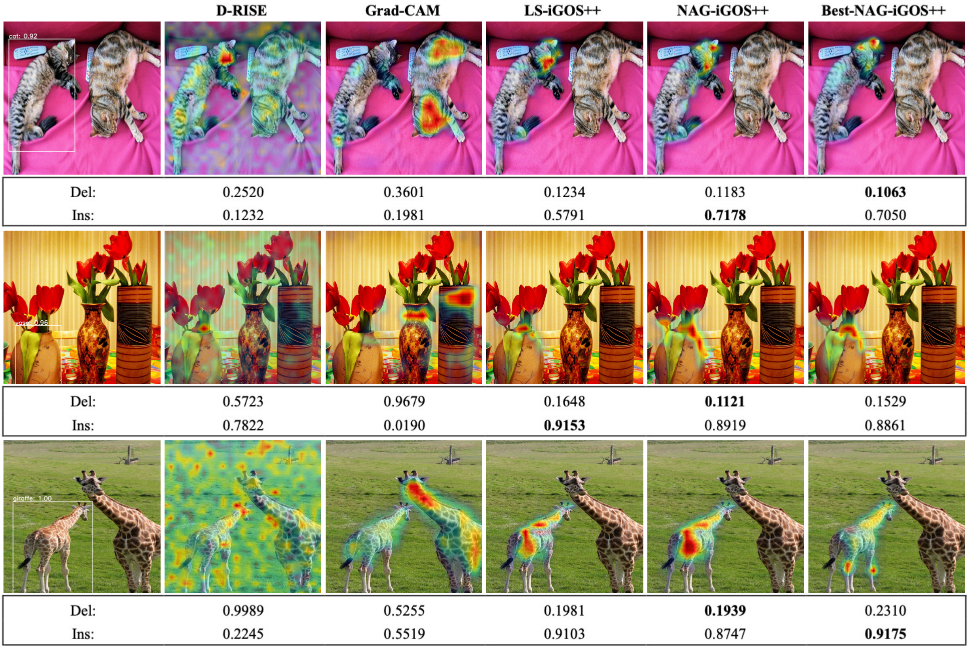
4.1 Metrics

We evaluate the capability of heatmaps in capturing the most important regions of an image using the Deletion and Insertion metrics [14]. These metrics involve substituting patches of pixels from a baseline image and evaluating the impact on the prediction confidence of the network. The *Deletion* metric measures the decrease in prediction confidence as salient regions in the input image are substituted with the baseline, whereas the *Insertion* metric measures the rate at which the original confidence can be restored if relevant evidence is reintroduced to the baseline. The Deletion/Insertion scores are the area under the curve (AUC) of the prediction confidences, with lower deletion scores and higher insertion scores indicating better performance. In all of our experiments, we use a highly blurred version of the original image as the baseline, which has been shown to be helpful to keep the perturbed images in the computation of Insertion/Deletion to stay on the natural image manifold [14, 16]. During the evaluation, we fix the bounding box proposal corresponding to the detection instead of generating new proposals from the masked image for two-stage detectors. For one-stage detectors, we also use the same anchor location with the same size for each object.

How many pixels should be inserted or deleted? Unlike the classification task, object detection and segmentation involve many small objects, so evaluating the heatmap by inserting and deleting all pixels of the entire image can be unfair to objects of varying sizes. A heatmap on a small object could quickly reach high confidence by merely inserting a small area, resulting in an artificially high Insertion score and low Deletion score. To address this, we normalize the amount of deleted and inserted pixels based on the size of the object predicted by the model. Specifically, a maximum of 3 times the number of predicted mask/box pixel values are inserted and deleted. This accounts for the possibility that the background information outside the bounding box may be utilized during the prediction.

Resolution	16 × 16			25 × 25			100 × 100		
Method	Del ↓	Ins ↑	Time(s)	Del ↓	Ins ↑	Time(s)	Del ↓	Ins ↑	Time(s)
D-RISE	0.6422	0.6322	220	– –	– –	– –	– –	– –	– –
Grad-CAM	– –	– –	– –	0.7839	0.3048	**7**	– –	– –	– –
LS-iGOS++	0.5630	0.6692	146	0.4685	0.6210	138	0.2370	0.5455	115
NAG-iGOS++	0.5577	0.6760	**62**	0.4641	0.6285	62	0.2380	0.5478	**62**
Best-NAG-iGOS++	**0.5388**	**0.6952**	248	**0.4399**	**0.6500**	248	**0.2048**	**0.5950**	248

Table 1: Quantitative comparison in terms of Deletion (lower is better), Insertion (higher is better), and runtime on the MSCOCO dataset using Mask R-CNN. LS-iGOS++ and NAG-iGOS++ use the backtracking line search optimization method and NAG, respectively. Best-NAG-iGOS++ uses multiple initializations with NAG



	D-RISE	Grad-CAM	LS-iGOS++	NAG-iGOS++	Best-NAG-iGOS++
Del:	0.2520	0.3601	0.1234	0.1183	**0.1063**
Ins:	0.1232	0.1981	0.5791	**0.7178**	0.7050
Del:	0.5723	0.9679	0.1648	**0.1121**	0.1529
Ins:	0.7822	0.0190	**0.9153**	0.8919	0.8861
Del:	0.9989	0.5255	0.1981	**0.1939**	0.2310
Ins:	0.2245	0.5519	0.9103	0.8747	**0.9175**

Figure 3: Visualization of the heatmap generated from different methods at the 25x25 resolution on Mask R-CNN detections

4.2 Results and Analysis

Insertion and Deletion Scores. Table 1 and 2 present the results of iGOS++ with different initialization and optimization methods, compared with state-of-the-art approaches on the MS-COCO dataset. It shows that our methods have superior performance over Grad-CAM and D-RISE at the respective resolutions of each method. For Best-NAG-iGOS++, we use the mask/box with the maximal difference between its insertion and deletion scores for evaluation. Note that these scores can be measured from the image and the model alone, hence the maximum can be selected using only the image and the model.

Note that this paper and [15] employ distinct methodologies for computing the Insertion and Deletion scores. Our approach employs highly blurred original images as the baselines for score computation, whereas [15] sets perturbed pixels to zero during deletion score calculation. Furthermore, [15] computes scores using the pixels of the entire image, which could potentially result in biased outcomes for objects of different sizes. Fig. 3 illustrates that the heatmaps produced by the D-RISE method can contain a considerable amount of noise. This noise can be attributed to the stochasticity of the approach. The heatmap visualization of Grad-CAM indicates that its poor insertion and deletion scores can be attributed to its tendency to highlight all regions belonging to the same class of objects in the entire image, even when only the score head of the target region is utilized for the backpropagation.

Resolution	16 × 16			64 × 64		
Method	Del ↓	Ins ↑	Time(s)	Del ↓	Ins ↑	Time(s)
D-RISE	0.4985	0.4953	70	– –	– –	– –
Grad-CAM	0.6980	0.2210	**2**	– –	– –	– –
LS-iGOS++	0.4804	0.4833	40	0.2475	0.2968	41
NAG-iGOS++	0.4688	0.4922	14	0.2384	0.3201	**14**
Best-NAG-iGOS++	**0.4015**	**0.5403**	56	**0.1704**	**0.3846**	56

Table 2: Quantitative comparison in terms of Deletion (lower is better), Insertion (higher is better), and run time on the MSCOCO dataset using YOLOv3-SPP. LS-iGOS++ and NAG-iGOS++ use the backtracking line search optimization method and NAG, respectively. Best-NAG-iGOS++ uses multiple initializations with NAG

Runtime. In Table 1 and 2, we present the average runtime per image for D-RISE, Grad-CAM, and all iGOS++ variants for explaining Mask R-CNN and YOLOv3-SPP on the MS-COCO dataset. All iGOS++ variants have a maximum of 5 iterations, while D-RISE has a maximum of 5000 iterations. The iGOS++ variant using line search is almost twice as fast as D-RISE, and the iGOS++ variant using NAG is about three times as fast as D-RISE.

Optimization	16 × 16			25 × 25			100 × 100		
	Del ↓	Ins ↑	Time(s)	Del ↓	Ins ↑	Time(s)	Del ↓	Ins ↑	Time(s)
LS	0.5630	0.6692	146	0.4685	0.6210	138	0.2370	0.5455	115
Adam	0.5747	0.6439	62	0.5040	0.5877	62	0.2997	0.4329	62
Nadam	0.6272	0.5895	62	0.5843	0.5280	62	0.4621	0.2951	62
NAG	0.5577	0.6760	**62**	0.4641	0.6285	**62**	0.2380	0.5478	**62**

Table 3: Ablation study on optimization methods used in iGOS++ on the MSCOCO dataset using Mask R-CNN

Ablation Study on Optimization Algorithms. We present results using NAG, Adam, and Nadam optimization methods in Table 3. We observe all these methods are significantly faster than line search (LS). However, among the three optimization methods, only NAG demonstrates improved performance in terms of both insertion score and deletion score when compared to the LS optimization method. It is also noteworthy that NAG maintained a consistent speed at different resolutions, while the LS method ran slightly faster at higher resolutions. This could be attributed to the fact that suitable update step sizes are easier to find at higher resolutions.

Ablation Study on initialization K. In Table 4, we show results utilizing different K values. Larger K gives more explanations which could be beneficial, but has small benefits on the metrics at the cost of slower runtime.

Best-	16 × 16			25 × 25			100 × 100		
NAG-iGOS++	Del ↓	Ins ↑	Time(s)	Del ↓	Ins ↑	Time(s)	Del ↓	Ins ↑	Time(s)
K = 2	0.5388	0.6952	**248**	0.4399	0.6285	**248**	0.2048	0.5950	**248**
K = 3	**0.5237**	**0.7044**	558	**0.4248**	**0.6601**	558	**0.1936**	**0.6036**	558

Table 4: Ablation with multiple initialization K on the MSCOCO dataset using Mask R-CNN

4.3 Multiple initializations

In Table 1-2, the results for Best-NAG-iGOS++ indicates the best result out of NAG-iGOS++ using four different initializations. The heatmap selected is based on the largest difference observed between the Insertion and Deletion scores. Fig. 4-5 display a few qualitative examples of the heatmaps from different initializations. We present various interpretations for the same object, such as different parts of an elephant and truck, all of which yield a confidence level exceeding 80% according to the model's predictions. The results indicate that the network is robust to different occlusions of the object and can make a confident prediction whenever seeing enough parts. This information cannot be revealed with a single heatmap.

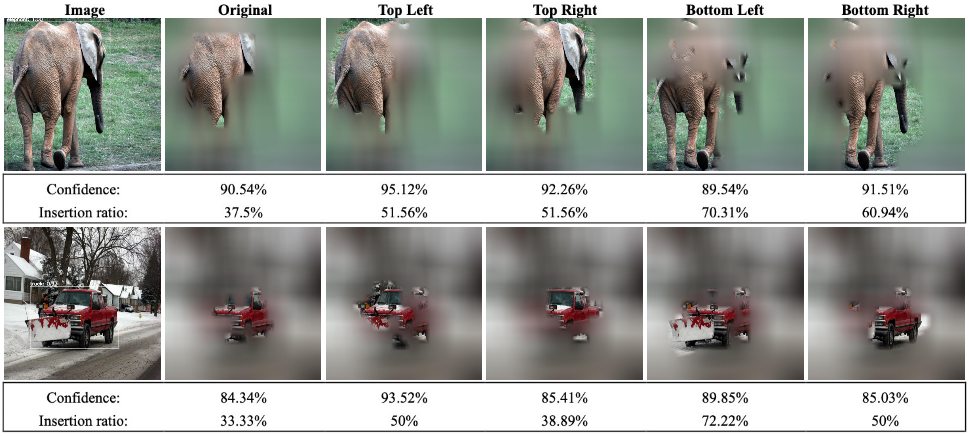


Image	Original	Top Left	Top Right	Bottom Left	Bottom Right
Confidence:	90.54%	95.12%	92.26%	89.54%	91.51%
Insertion ratio:	37.5%	51.56%	51.56%	70.31%	60.94%
Confidence:	84.34%	93.52%	85.41%	89.85%	85.03%
Insertion ratio:	33.33%	50%	38.89%	72.22%	50%

Figure 4: Examples generated by NAG-iGOS++ with different initializations on MSCOCO using Mask R-CNN. The regions not highlighted by the heatmap are blurred.

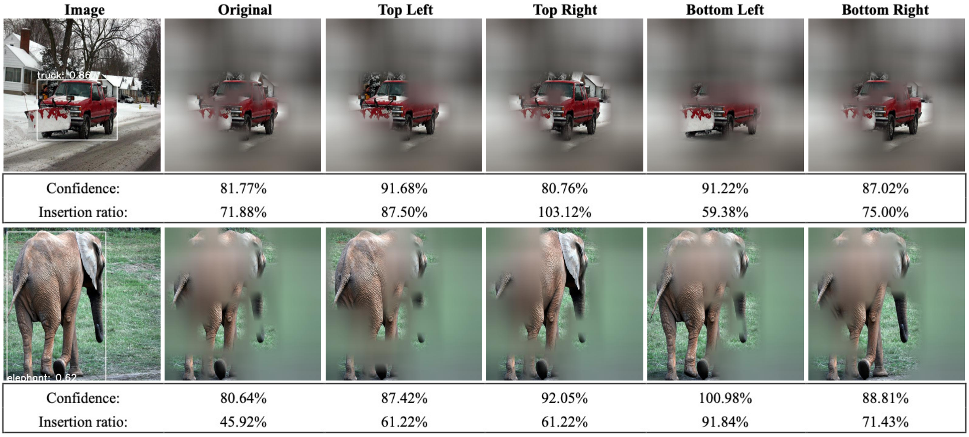


Image	Original	Top Left	Top Right	Bottom Left	Bottom Right
Confidence:	81.77%	91.68%	80.76%	91.22%	87.02%
Insertion ratio:	71.88%	87.50%	103.12%	59.38%	75.00%
Confidence:	80.64%	87.42%	92.05%	100.98%	88.81%
Insertion ratio:	45.92%	61.22%	61.22%	91.84%	71.43%

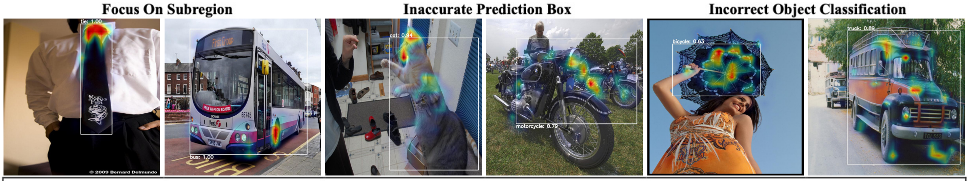
Figure 5: Examples generated by NAG-iGOS++ with different initializations on MSCOCO using YOLOv3-SPP. The regions not highlighted on the heatmap are blurred.

Comparing with Mask R-CNN, YOLO seems to be more focused on the corners of the detections. This is especially obvious in the truck image, where Mask R-CNN explanations all contained the center of the truck whereas all YOLO explanations (initialized from four corners, respectively) contained at least 3 corners of the box. This might be related to the one-stage nature of the YOLO detector that needs to locate the box extent in the same network as the classification, whereas in two-stage networks such as Mask R-CNN the box extent is mostly resolved in the anchor box stage and the second stage can just focus on classification.

4.4 More Visualizations using Mask R-CNN

We present the visualizations of a few more images in Fig 6. It shows that in many images the model tends to focus on specific parts or subregions of objects. Notably, the model consistently directs its attention to the knot on the tie and the tire of buses. This observation suggests a preference for distinctive local features.

Besides, from the two middle columns of Figure 6, we observe a tendency of the model to merge similar objects into a single bounding box. This merging behavior often leads to inaccurate localization and hampers the model's ability to precisely delineate individual instances. This phenomenon sheds light on the challenges faced by the model in accurately localizing and distinguishing between objects that share visual similarities. The visualizations in the last two columns show further insights into the model's behavior when it makes mistakes. For example, the rightmost object was incorrectly predicted to be a truck. The visualization hints that this inaccuracy might be attributed to the network's focus on the railings on top of the bus. More visualizations are shown in the supplementary materials.



Del:	0.1759	0.2039	0.1855	0.0972	0.0303	0.0479
Ins:	0.8201	0.8661	0.8538	0.8900	0.6406	0.8156

Figure 6: More visualizations using Mask R-CNN.

5 Conclusion

This paper presents NAG-iGOS++, an algorithm that extends the iGOS++ algorithm to explain object detection networks. We proposed to use Nesterov Accelerated Gradient which improved the efficiency and accuracy of explanations, and our multiple initializations provided a more complete picture of the object detection and segmentation networks to be explained. Through extensive experiments on the MSCOCO dataset, we demonstrate the superiority of our approach compared to existing methods in terms of both explanation quality and speed. The insights gained from our approach can also help in identifying errors and biases in the deep network, leading to improved performance and a better understanding of the inner working of the network.

Acknowledgements The work is partially supported by NSF-1751402, NSF-1927564, ONR-UWSC12017-BPO49408 and ONR-N00014-21-1-2052. The authors would also like to thank Dr. Qing Tao for helpful discussions.

References

[1] Timothy Dozat. Incorporating Nesterov Momentum into Adam. In *Proceedings of the 4th International Conference on Learning Representations*, pages 1–4, 2016.

[2] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88: 303–338, 2010.

[3] Ruth Fong, Mandela Patrick, and Andrea Vedaldi. Understanding deep networks via extremal perturbations and smooth masks. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*, pages 2950–2958. IEEE, 2019. doi: 10.1109/ICCV.2019.00304. URL https://doi.org/10.1109/ICCV.2019.00304.

[4] Ruth C Fong and Andrea Vedaldi. Interpretable explanations of black boxes by meaningful perturbation. In *Proceedings of the IEEE international conference on computer vision*, pages 3429–3437, 2017.

[5] Denis Gudovskiy, Alec Hodgkinson, Takuya Yamaguchi, Yasunori Ishii, and Sotaro Tsukizawa. Explain to fix: A framework to interpret and correct DNN object detector predictions. *arXiv preprint arXiv:1811.08011*, 2018.

[6] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.

[7] Hiroki Kawauchi and Takashi Fuse. Shap-based interpretable object detection method for satellite imagery. *Remote Sensing*, 14(9):1970, 2022.

[8] Saeed Khorram, Tyler Lawson, and Li Fuxin. IGOS++: Integrated Gradient Optimized Saliency by Bilateral Perturbations. In *Proceedings of the Conference on Health, Inference, and Learning*, CHIL '21, page 174–182, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450383592. doi: 10.1145/3450439.3451865. URL https://doi.org/10.1145/3450439.3451865.

[9] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[10] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014.

[11] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14*, pages 21–37. Springer, 2016.

[12] Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. *Advances in neural information processing systems*, 30, 2017.

[13] Yurii Nesterov. A method for unconstrained convex minimization problem with the rate of convergence o(1/k2). In *Doklady ANSSSR (translated as Soviet.Math.Docl.)*, volume 269, page 543–547, 1983.

[14] Vitali Petsiuk, Abir Das, and Kate Saenko. Rise: Randomized input sampling for explanation of black-box models. *arXiv preprint arXiv:1806.07421*, 2018.

[15] Vitali Petsiuk, Rajiv Jain, Varun Manjunatha, Vlad I Morariu, Ashutosh Mehra, Vicente Ordonez, and Kate Saenko. Black-box explanation of object detectors via saliency maps. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11443–11452, 2021.

[16] Zhongang Qi, Saeed Khorram, and Li Fuxin. Visualizing deep networks by optimizing with integrated gradients. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 11890–11898, 2020.

[17] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018.

[18] Marco Tulio Ribeiro, Sameer Singh, and Carlos 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, 2016.

[19] Karl Schulz, Leon Sixt, Federico Tombari, and Tim Landgraf. Restricting the flow: Information bottlenecks for attribution. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=S1xWh1rYwB.

[20] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017.

[21] Vivswan Shitole, Fuxin Li, Minsuk Kahng, Prasad Tadepalli, and Alan Fern. One Explanation is Not Enough: Structured Attention Graphs for Image Classification. volume 34, 2021.

[22] K. Simonyan, A. Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *International Conference On Learning Representations*, 2013.

[23] Daniel Smilkov, Nikhil Thorat, Been Kim, Fernanda Viégas, and Martin Wattenberg. Smoothgrad: removing noise by adding noise. *arXiv preprint arXiv:1706.03825*, 2017.

[24] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *International conference on machine learning*, pages 3319–3328. PMLR, 2017.

[25] Hideomi Tsunakawa, Yoshitaka Kameya, Hanju Lee, Yosuke Shinya, and Naoki Mitsumoto. Contrastive relevance propagation for interpreting predictions by a single-shot object detector. In *2019 International Joint Conference on Neural Networks (IJCNN)*, pages 1–9. IEEE, 2019.

[26] Haofan Wang, Zifan Wang, Mengnan Du, Fan Yang, Zijian Zhang, Sirui Ding, Piotr Mardziel, and Xia Hu. Score-cam: Score-weighted visual explanations for convolutional neural networks. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR Workshops 2020, Seattle, WA, USA, June 14-19, 2020*, pages 111–119. Computer Vision Foundation / IEEE, 2020. doi: 10.1109/CVPRW50498.2020.00020. URL https://openaccess.thecvf.com/content_CVPRW_2020/html/w1/Wang_Score-CAM_Score-Weighted_Visual_Explanations_for_Convolutional_Neural_Networks_CVPRW_2020_paper.html.

[27] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part I 13*, pages 818–833. Springer, 2014.

[28] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning Deep Features for Discriminative Localization. In *Computer Vision and Pattern Recognition*, 2016.