ODSmoothGrad: Generating Saliency Maps for Object Detectors

Chul Gwon
Analytic Folk
chul@analyticfolk.com

Steven C. Howell
ARLIS
University of Maryland
College Park, MD
showell@arlis.umd.edu

Abstract

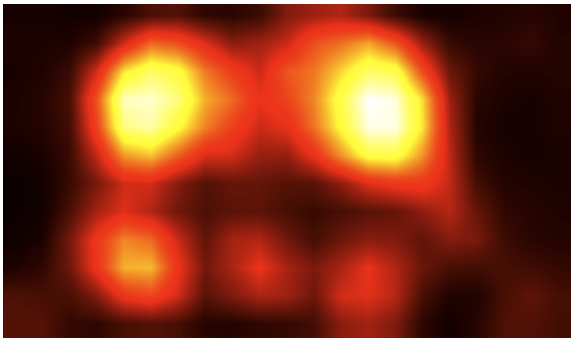
Techniques for generating saliency maps continue to be used for explainability of deep learning models, with efforts primarily applied to the image classification task. Such techniques, however, can also be applied to object detectors, not only with the classification scores, but also for the bounding box parameters, which are regressed values for which the relevant pixels contributing to these parameters can be identified. In this paper, we present ODSmoothGrad, a tool for generating saliency maps for the classification and the bounding box parameters in object detectors. Given the noisiness of saliency maps, we also apply the SmoothGrad algorithm [12] to visually enhance the pixels of interest. We demonstrate these capabilities on one-stage and two-stage object detectors, with comparisons using classifier-based techniques.

1. Introduction

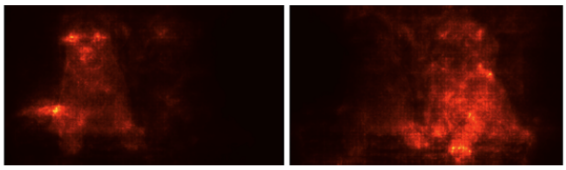
There is a significant amount of work on algorithms and tools for improving the explainability of models. As models become more complex, it becomes increasingly difficult to determine how the model achieved a particular result. Methods for obtaining saliency maps have long been used to highlight the parts of an image that provide the greatest contribution to a given output. The majority of the effort has been on the image classification problem, with some growing interest in the field of object detectors [7], although these approaches still focus on explainability of the classification results and ignore the bounding box values.

Extending the use of saliency maps to gain visibility into the decision-making process of object detectors is also relevant for explainability into these types of models. The class of a detected object, along with the bounding box parameters that localize it within the image, are all derived from the model. Confirming that these returned values are based on relevant features from the image is important for verifying model performance. The bounding box parameters

(a) Original Image



(b) Grad-CAM



(c) ODSmoothGrad

Figure 1. Saliency maps for classification and object detection for an image with two Labrador retrievers shown in (a). Classification-based maps generate a single map for the Labrador retriever class - the Grad-CAM mask using ResNet-101 is shown in (b). Object detection-based maps generate separate masks for each detected object rather than a single mask - ODSmoothGrad maps for the classification output of Faster R-CNN shown in (c).

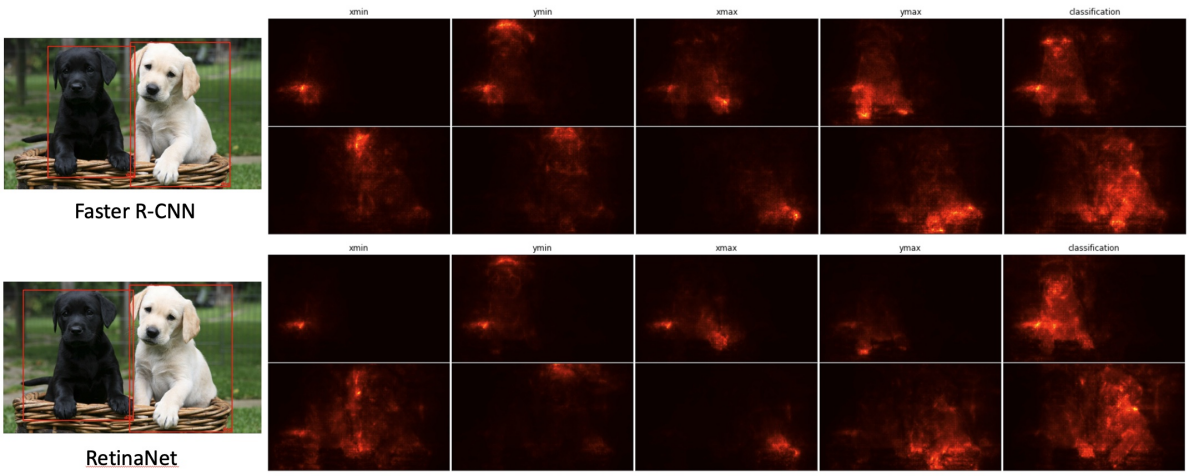


Figure 2. ODSmoothGrad saliency maps generated for Faster R-CNN and RetinaNet from the Detectron2 Model Zoo. The top row of each shows the saliency maps for the dog on the left, while the bottom row shows the dog on the right. The columns show the saliency map for x_{min} (left), y_{min} (top), x_{max} (right), y_{max} (bottom), and classification.

become particularly important when there are shifts in the predicted box as opposed to the ground truth annotation, or if ground truth annotations have intentionally been sized slightly larger to include some context. Another motivation for saliency maps with object detectors is that when there are multiple objects of the same class in a given image, the object detector will return a separate map for each detected object, while the classifier will return a single map for the entire image (Figure 1).

The contributions made by this paper include the application of saliency methods to object detectors to identify relevant pixels for both the classification and bounding box parameters, as well as demonstrating this capability on one-stage (RetinaNet [4]) and two-stage object detectors (Faster RCNN [8]). Our implementation applies the SmoothGrad algorithm [12] to improve visibility of relevant pixels.

2. Related Work

There is significant work in the field of object detection, including transformer-based object detectors [2]. For this study, we focused our work on anchor-based object detectors, particularly RetinaNet [4] and Faster R-CNN [8], to demonstrate the one-stage and two-stage detectors. Although these are no longer state-of-the-art for object detection benchmarks, the ability to train these models with more modest-sized datasets and hardware still make them popular options for detection.

As mentioned previously, the majority of the work in this area has been done for the image classification problem. Along with saliency-based approaches, recent work extended Layer-wise Relevance Propagation (LRP) specifically for use with SSDs [13] and YOLO5 [3]. The techniques applied here focus solely on the classification importance, rather than also including the relevance of the bounding box parameters themselves.

2.1. Saliency Methods with Classification

There are several published techniques that discuss the use of saliency methods, along with one that goes over the problems that particular saliency methods exhibit [1]. We provide a quick overview of some of them here, especially those relevant to our work and related work, but this is not intended to be an exhaustive list.

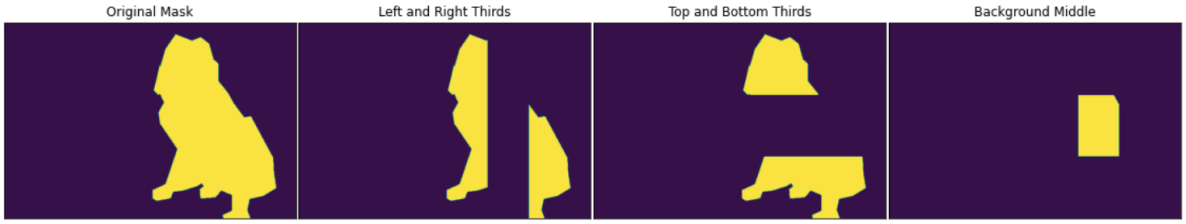
Class Saliency Extraction [11] begins with an image, with m rows and n columns, from which a saliency map $M \in R^{m \times n}$ is generated in the following manner: perform backward pass from the logit of interest to obtain derivative w; take the absolute value of each element of M; for color im- ages, take the max value across the channels for each pixel.

SmoothGrad [12] takes the average of multiple samples of class saliency extraction, injecting random noise into the image for each sample. The resulting map calculation looks as follows:

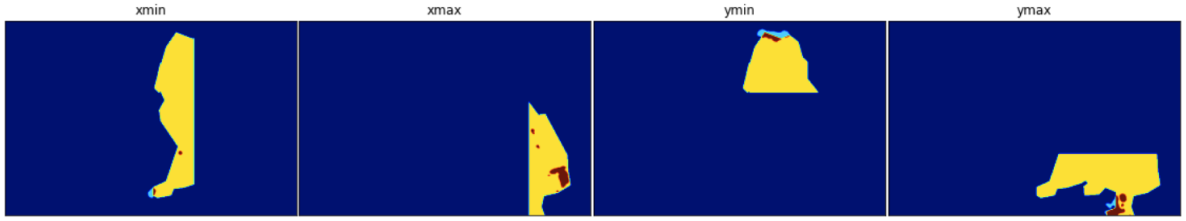
$$M_c(x) = \frac{1}{n} \sum_{1}^{n} M_c(x + \mathcal{N}(0, \sigma^2)) \tag{1}$$

where $M \in R^{m \times n}$, n is the number of samples, and $\mathcal{N}(0, \sigma^2)$ is Gaussian-distributed noise with a standard deviation σ.

Grad-CAM [10] performs a backward pass from the logit of particular class to the last convolutional layer in the CNN, and then average pools across the width and height dimensions to produce a weight value for each of the channels



(a) Original ground truth mask divided into thirds for validation



(b) Overlay of ground truth and prediction masks for the bounding box components

Figure 3. To validate the performance of the masks for the bounding box parameters, the original ground truth mask was divided into thirds, with the middle third used for background comparisons (a). The intersection-over-foreground (IOF) was then calculated between the saliency masks and the corresponding third of the ground truth mask (b). Original mask in yellow, saliency mask in cyan, and overlap region in dark red.

in the final layer with respect to this class. These weights are multiplied by the corresponding activation map, and the maps are combined and passed through a ReLU operation to create the final map. Since there are multiple downsampling operations that occur from the input image to the final convolutional layer, the width and height of the final layer is less than the initial image. To create a map of the same dimension as the original image, a bilinear upsampling is used, which still results in a coarse map. To achieve a pixel-level saliency map, Grad-CAM is combined with Guided Backpropagation to create Guided Grad-CAM.

2.2. Saliency Methods with Object Detection

For applying saliency methods to object detection, there have been three examples that have done similar work to ours, albeit using different techniques. The first is DetGrad-CAM [9], which applies Grad-CAM for all of the features and then sums across these features to produce a final saliency map. In this manner, DetGrad-CAM does allow for localized saliency maps, as expected from object detectors, but their work was restricted to classification improvements, and they tested it only on YOLOv2.

The second method is Spatial Sensitive Grad-CAM (SS-GradCAM) [15], which applies Grad-CAM to SSD. This technique only uses the classification output from the detected object, but does not take the bounding box parameters into consideration. It was also developed for use with the original SSD [6].

The third method takes a different approach than the gradient-based approaches discussed so far. D-RISE [7] generates a series of masks for a given image, and passes the image with the different masks applied through the object detector. The class probabilities generated by these masked images are used as the importance weights, and the masks are then combined based on these weights to produce the final importance map.

3. Method

We performed our experiment using two methods, the first being more true to the saliency extraction calculation from the logit, and the second using a simplified implementation that uses SmoothGrad and could be quickly generalized to other architectures.

For the first implementation, using RetinaNet [5], we found the anchor boxes with classification scores beyond a particular threshold. For each of these, we performed the backward pass against the classification logit as well as each of the bounding box parameters (xmin, ymin, xmax, ymax) to generate five saliency maps. Using this technique, we would not only be able to determine the saliency map for a particular class, but also the relevant pixels for other classes, as is possible with saliency map creation. The major problem with this implementation was that it was specific for RetinaNet, and so generalizing this for use across different object detectors would not be as straightforward.

For the second method, rather than going through the anchor boxes directly, we took the output tensors generated from the outputs from Detectron2 [14] and ran the backward pass from those directly. Although this goes against starting the backward pass from the logit, when combined with SmoothGrad, the resulting saliency maps were clearer

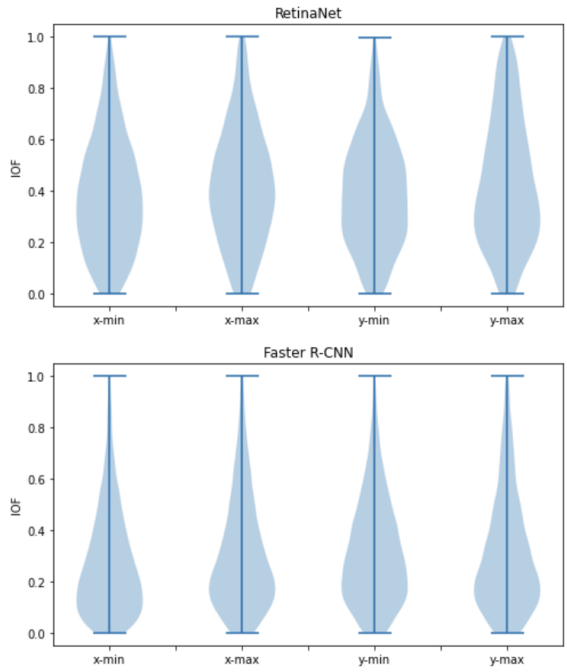


Figure 4. Violin plots showing the IOF values per detected object for RetinaNet and Faster R-CNN.

than with a single pass from our first implementation. For the SmoothGrad step, we chose a sample size $n = 20$, and a noise value $\sigma = 0.05$.

Between each of the sampling passes for SmoothGrad, we needed to align the bounding boxes from the detections. To accomplish this, we applied a threshold of 0.7 on the value of the intersection-over-union (IOU) between the bounding boxes of detected objects between each pass. For each detected object, we took averages of each of the saliency maps generated for the classification and bounding box parameters to produce the final results.

4. Experiments

4.1. Implementation

Whereas ODSmoothGrad could be applied against general object detector implementations, we specifically developed our library to work with Detectron2. We used RetinaNet and Faster RCNN from the Detectron2 Model Zoo, using implementations with the ResNet-101 backbone and a version of Faster RCNN that included the Feature Pyramid Network [4]. As shown in Figure 2, for each detected object, saliency maps are generated for the four bounding box parameters and the classification label.

4.2. Validation

To validate our method, we performed tests using the MSCOCO dataset for a random set of classes. We began by generating saliency maps using the SmoothGrad algorithm with the Detectron2 library [14]. Separate sets of saliency maps were generated for RetinaNet and Faster-RCNN to show viability using one-stage and two-stage object detectors. We also restrict the sample to high confidence detects, where the classification score is above 0.9. Next, for the purposes of validation, we created a binarized segmentation mask using the saliency map. To accomplish this, we used a very simplistic algorithm, where we first applied a 2σ Gaussian filter to smooth out the map, and then used pixels that were greater than a factor of 0.32 of the max pixel value as the foreground, while setting the remaining pixels to zero.

The next step involved using the ground truth segmentation polygons from MSCOCO. Direct comparisons against these polygons did not adequately demonstrate the capabilities of our technique, since the goal is to show that generating a saliency map for the bounding box parameters (such as x-min) would show greater significance around the corresponding parameter (the left side of the ground truth for x-min). As a result, we performed the following method of evaluating performance of the saliency map with the corresponding bounding box parameter:

- Divide the ground truth segmentation polygons into thirds in the x and y dimensions (Figure 3a)
- Calculate the intersection over foreground (IOF) of the first and last thirds of the ground truth polygon with the corresponding saliency masks from the min and max bounding box parameters (Figure 3b).
- Calculate the IOF of the saliency mask with the middle section of the ground truth polygon to determine background.

Comparing the saliency mask with the middle section is to demonstrate that the mask for the bounding box parameters do tend to be localized on the respective sides, rather than spreading out into other parts of the ground truth segmentation polygon.

The use of IOF, with the saliency segmentation mask area as the denominator, was used over Dice or Jaccard metrics to more appropriately capture relevance of the intersection. Dividing the ground truth polygon into thirds in the x and y dimensions provides an automated and consistent approach, but results in variations based on the size and orientation of the objects that skew the values of the aforementioned metrics.

5. Conclusions

The use of saliency methods is a popular way of achieving explainability of a model, and the extension of these methods into object detection algorithms provides visibility into their predictions. Applying these methods to both the

class prediction as well as the bounding box parameters allows confirmation that the results are based on the relevant features. The results from Figure 4 demonstrate a significant overlap between the generated saliency masks and the localized sections of the ground truth polygons. Some of the low values can be attributed to the imperfections in the ground truth segmentation of the objects, whereas others are the result of the saliency map picking up features in the image that do not directly correspond with the detected class.

6. Acknowledgements

We thank Dr. Tim Oates (UMBC) for review and comments on manuscript. This work was performed for CalypsoAI (https://calypsoai.com).

References

[1] Julius Adebayo, Justin Gilmer, Michael Muelly, Ian J. Goodfellow, Moritz Hardt, and Been Kim. Sanity checks for saliency maps. *CoRR*, abs/1810.03292, 2018. 2

[2] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. *CoRR*, abs/2005.12872, 2020. 2

[3] Apostolos Karasmanoglou, Marios Antonakakis, and Michalis Zervakis. Heatmap-based explanation of yolov5 object detection with layer-wise relevance propagation. In *2022 IEEE International Conference on Imaging Systems and Techniques (IST)*, pages 1–6, 2022. 2

[4] Tsung-Yi Lin, Piotr Dollár, Ross B. Girshick, Kaiming He, Bharath Hariharan, and Serge J. Belongie. Feature pyramid networks for object detection. *CoRR*, abs/1612.03144, 2016. 2, 4

[5] Tsung-Yi Lin, Priya Goyal, Ross B. Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. *CoRR*, abs/1708.02002, 2017. 3

[6] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C. Berg. Ssd: Single shot multibox detector. 2016. To appear. 3

[7] 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. *CoRR*, abs/2006.03204, 2020. 1, 3

[8] Shaoqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun. Faster R-CNN: towards real-time object detection with region proposal networks. *CoRR*, abs/1506.01497, 2015. 2

[9] Aniruddha Saha, Akshayvarun Subramanya, Koninika Patil, and Hamed Pirsiavash. Adversarial patches exploiting contextual reasoning in object detection. *CoRR*, abs/1910.00068, 2019. 3

[10] Ramprasaath R Selvaraju, Abhishek Das, Ramakrishna Vedantam, Michael Cogswell, Devi Parikh, and Dhruv Batra. Grad-cam: Why did you say that?, 2016. 2

[11] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. In *Workshop at International Conference on Learning Representations*, 2014. 2

[12] Daniel Smilkov, Nikhil Thorat, Been Kim, Fernanda B. Viégas, and Martin Wattenberg. Smoothgrad: removing noise by adding noise. *CoRR*, abs/1706.03825, 2017. 1, 2

[13] 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, 2019. 2

[14] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. https://github.com/facebookresearch/detectron2, 2019. 3, 4

[15] Toshinori Yamauchi and Masayoshi Ishikawa. Spatial sensitive grad-cam: Visual explanations for object detection by incorporating spatial sensitivity. In *2022 IEEE International Conference on Image Processing (ICIP)*, pages 256–260, 2022. 3