# Efficient and Concise Explanations for Object Detection with Gaussian-Class Activation Mapping Explainer

Khanh Nguyen[†], Hung Nguyen[†,‡,*],
Khang Nguyen[†], Binh Truong[†], Tuong Phan[†,◇], Hung Cao[‡]
[†]Quy Nhon AI, FPT Software, Vietnam
[‡]Analytics Everywhere Lab, University of New Brunswick, Canada
[◇]University of Waterloo, Canada

**Abstract**

To address the challenges of providing quick and plausible explanations in Explainable AI (XAI) for object detection models, we introduce the Gaussian Class Activation Mapping Explainer (G-CAME). Our method efficiently generates concise saliency maps by utilizing activation maps from selected layers and applying a Gaussian kernel to emphasize critical image regions for the predicted object. Compared with other Region-based approaches, G-CAME significantly reduces explanation time to 0.5 seconds without compromising quality. Our evaluation of G-CAME, using Faster-RCNN and YOLOX on the MS-COCO 2017 dataset, demonstrates its ability to offer highly plausible and faithful explanations efficiently, especially in reducing the bias on the tiny object detection.

**Keywords:** Explainable AI, Object Detection, Class Activation Mapping

## 1. Introduction

In object detection, Deep Neural Networks (DNNs) [1] have significantly improved with the adoption of Convolution Neural Networks (CNNs). However, the deeper the network is, the more complex it is to understand, debug or improve, which can be a serious problem in critical areas [2]. To help humans have a deeper understanding of the model's decisions, several Explainable Artificial Intelligence (XAI) methods using saliency maps to highlight the important regions of input images have been introduced.

One simple and common way to explain the object detector is to ignore the model architecture and only consider the input and output. This approach aims to determine the importance of each region in the input image based on the change in the model's output. For example, Detector Randomized Input Sampling for Explanation (D-RISE) [3] estimates each region's effect on the input image by creating thousands of perturbed images, then feeds them into the model to predict and get the score for each perturbed mask. Another method is Surrogate Object Detection Explainer (SODEx) [4], an upgrade of Local Interpretable Model-Agnostic Explanations (LIME) [5], which also uses the same technique as D-RISE to explain object detectors. Although the results of both SODEx and D-RISE are compelling, the generation of a large number of perturbations slows the explanation generation considerably.

Other approaches, such as Class Activation Mapping (CAM) [6] and GradCAM [7], use the activation maps of a specific layer in the model's architecture as the main component to form the explanation. These methods are faster than the mentioned region-based but still have some meaningless information since the feature maps are not related to the target object [8]. Such methods can give a satisfactory result for the classification task. Still, they cannot be applied directly to the object detection task because these methods highlight all regions having the same target class and fail to focus on one specific region.

[*]hung.ntt@unb.ca

In this paper, we propose the *Gaussian Class Activation Mapping Explainer* (G-CAME), which can explain the classification and localization of the target objects. Our method extends the applicability of CAM-based XAI to object detectors. By adding the Gaussian kernel as the weight for each pixel in the feature map, G-CAME's final saliency map can explain each specific object. Our contributions can be summarized as follows:

(1) We propose the first CAM-based method tailored for object detection, G-CAME, which can explain object detectors as a saliency map for a specific target object. G-CAME can explain in a reasonably short time, which overcomes the existing methods' time constraints like D-RISE [3] and SODEx [4].
(2) We qualitatively and quantitatively evaluate our method with D-RISE on two main types of object detectors, namely YOLOX [9] (one-stage detector) and Faster-RCNN [10] (two-stage detector), and prove that our method can give a less noise, more accurate saliency map in a shorter time than D-RISE.

Our code is available at https://github.com/khanhnguyenuet/GCAME.

## 2. Explainable AI in Object Detection

Object detection, a field in computer vision (CV), involves models that are broadly classified into two categories: one-stage and two-stage models. One-stage models, such as the YOLO series [11], SSD [12], and RetinaNet [13], detect objects directly over a dense sampling of locations. In contrast, two-stage models like the R-CNN family [1], FPN [14], and R-FCN [15], involve a two-phase process. Initially, these models select Regions of Interest (ROI) from the feature extraction stage, followed by classification based on each proposed ROI.

While several XAI methods have been applied to analyze deep CNN models in classification tasks, their applicability in object detection is comparatively limited due to constraints in flexibility, suitability, and computational efficiency [16].

This section discusses two XAI types: Region-based saliency methods and CAM-based saliency methods. These methods are evaluated for their applicability in both classification and object detection tasks. A significant gap in current XAI methods, particularly in object detection, is identified, laying the groundwork for the introduction of our method.

### 2.1. Region-based saliency methods

Region-based saliency methods use masks to isolate specific regions of an input image, assessing their impact on the output by processing the masked input through the model and quantifying each region's influence. In classification, LIME [5] and its extension, RISE [17], are notable examples, where the latter employs thousands of masks to generate a composite saliency map. Recent advancements have adapted these methods for object detection. SODEx [4] applies LIME to explain object detectors, modifying the metric to focus on target bounding boxes. D-RISE [3] refines this by altering the computation of weighted scores for each random mask, specifically for object detection. D-CLOSE [18] further utilizes multiple levels of segmentation on the image and combines them to deliver more concise and consistent explanations. Region-based methods offer an intuitive approach as they do not necessitate the end-users in-depth understanding of the model's architecture.

However, a notable challenge is the sensitivity of these explanations to changes in hyperparameters, resulting in multiple potential explanations for a single object. Consequently, to achieve a clear and satisfactory explanation, careful finetuning hyperparameters is essential. Additionally, a significant drawback of region-based methods is the considerable amount of time required to generate an explanation.

### 2.2. CAM-based methods

CAM-based XAI, on the other hand, requires a detailed understanding of the model's architecture. Techniques such as CAM [6] and its successors, GradCAM [7], GradCAM++ [19], and XGradCAM [20], are noteworthy for producing detailed saliency maps. These methods utilize partial derivatives of feature maps in selected layers relative to the target class score. While CAM-based methods are generally more efficient than Region-based methods [21], their reliance on feature maps can result in less meaningful saliency maps. Additionally, these methods have primarily been developed for classification tasks, with no existing adaptations for object detection.

In light of these limitations, we introduce G-CAME, a novel CAM-based XAI method designed explicitly for object detection. G-CAME is the first of its kind to offer stable and rapid explanations for both one-stage and two-stage object detection models, addressing the shortcomings of existing approaches.

## 3. Proposed method

For a given image $I$ with size $h$ by $w$, an object detector $f$ and the prediction $d$ includes the bounding box and predicted class. We aim to provide a saliency map $S$ to explain why the model has that prediction. The saliency map $S$ has the same size as the input $I$. Each value $S_{(i,j)}$ shows the importance of each pixel $(i, j)$ in $I$, respectively, influencing $f$ to give prediction $d$. We propose a new method that helps to produce that saliency map in a white-box manner. Our method is inspired by GradCAM [7], which uses the class activation mapping technique to generate the explanation for the model's prediction. The main idea of our method is to use normal distribution combined with the CAM-based method to measure how one region in the input image affects the predicted output. Fig. 1 shows an overview of our method.

Due to their output difference, we cannot directly apply XAI methods for the classification model to the object detection model. In the classification task, the model only gives one prediction that shows the image's label. However, in the object detection task, the model gives multiple boxes with corresponding labels and the probabilities of objects. Most object detectors, like YOLO [11] and R-CNN [1], usually produce $N$ predicted bounding boxes in the format:

$$d_i = (x_1^i, y_1^i, x_2^i, y_2^i, p_{obj}^i, p_1^i, \ldots, p_C^i) \tag{3.1}$$

The prediction is encoded as a vector $d_i$ that consists of:

- Bounding box information: $(x_1^i, y_1^i, x_2^i, y_2^i)$ denotes the top-left and bottom-right corners of the predicted box.
- Objectness probability score: $p_{obj}^i \in [0, 1]$ denotes the probability of an object's occurrence in the predicted box.
- Class score information: $(p_1^i, \ldots, p_C^i)$ denotes the probability of $C$ classes in predicted box.

In almost all object detectors, such as Faster-RCNN [10], YOLOX [9], the anchor boxes technique is widely used to detect bounding boxes. G-CAME utilizes this technique to find and estimate the region related to the predicted box. Our method can be divided into 4 phases (Fig. 1) as follows: 1) Choosing target layers, 2) Object Locating, 3) Weighting Feature Map, and 4) Masking Target Region.

### 3.1. Target layers selection

**One-stage object detector (YOLOX)** For a one-stage object detector, such as YOLOX, we choose the final convolution layer in each branch of the model as the target layer to calculate the derivative, as convolutional layers naturally retain spatial information

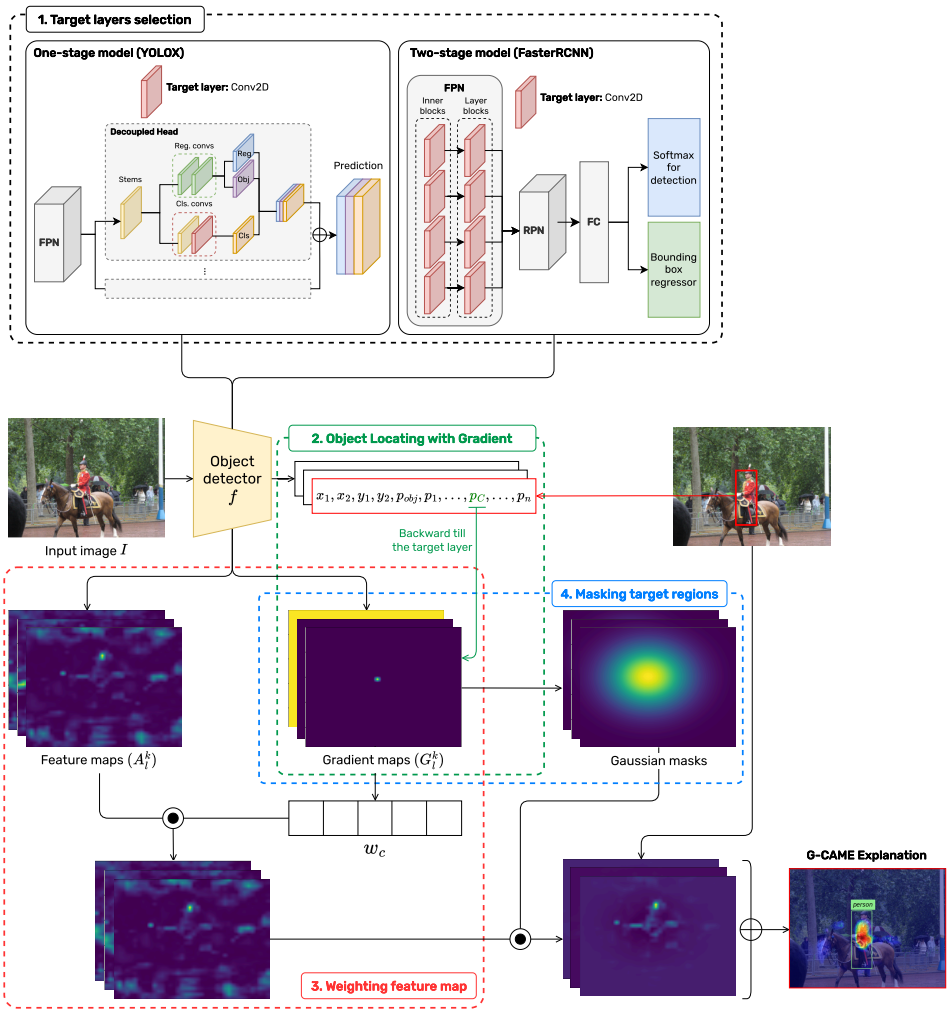



*Figure 1.* Overview of G-CAME method. We use the gradient-based technique to get the target object's location and weight for each feature map. We multiply element-wise with Gaussian kernel for each weighted feature map to remove unrelated regions. After applying the Gaussian kernel, the output saliency map is created by a linear combination of all weighted feature maps.

that is lost in fully connected layers. Hence, the last convolutional layers are expected to have the best compromise between high-level semantics and detailed spatial information [7]. The neurons in these layers look for semantic class-specific information in the image.

**Two-stage object detector (Faster-RCNN)** Two-stage object detectors, such as Faster-RCNN, contain two phases. In the first stage, the image is passed through stacked convolution layers in backbone layers and the Feature Pyramid Network (FPN) [14] which includes four branches to detect the different objects' sizes to extract features. Subsequently, the Region Proposal Network (RPN) identifies potential object-containing regions, which are then resized uniformly via the Region of Interest (ROI) Pooling layer. For a two-stage object detector, we utilize the convolution layers in the FPN network as the target layers to analyze because they are the last layers containing spatial information of the feature extractors.

### 3.2. **Object Locating with Gradient**

The anchor box technique is used in most detector models like YOLOX [9], Faster-RCNN [10], and PAFNet [22] to predict the bounding boxes. In the final feature map, each pixel predicts $N$ bounding boxes and one bounding box for the anchor-free technique. To get the correct pixel representing the box that we aim to explain, we take the derivative of the target box with the final feature map to get the location map $G_k^{l(c)}$ as the following formula:

$$G_k^{l(c)} = \frac{\partial S^c}{\partial A_k^l} \tag{3.2}$$

where $G_k^{l(c)}$ denotes the gradient map of layer $l$ for feature map $k$. $\frac{\partial S^c}{\partial A_k^l}$ is the derivative of the target class score $S^c$ with the feature map $A_k$. In the regression task of most one-stage object detectors, $1 \times 1$ Convolution is used for predicting the bounding box, so in the backward pass, we have the Gradient map $G$ having the value of 1 pixel.

In the two-stage object detector, such as Faster-RCNN, because the regression and classification tasks are in two separate branches, we tailor G-CAME for two-stage models as follows. First, we calculate the partial derivative of the class score according to each feature map of selected layers. Faster-RCNN has four branches of detecting objects, and we choose the last convolution layer of each branch to calculate the derivative. When we take the derivative of the class score to the target layer, the gradient map $G_k^{l(c)}$ has more than one pixel having value because anchor boxes are created in the next phase, namely the detecting phase. The ROI pooling layer replaces $1\times1$ Convolution, and they are in a separate branch from the classification stage. Thus, we cannot get the pixel representing the object's center through the gradient map. To solve this issue, we set the pixel with the highest value in the gradient map as the center of the Gaussian mask. We estimate that the area around the highest value pixel likely contains relevant features.

### 3.3. **Weighting feature map via Gradient-based method**

We adopt a gradient-based method as GradCAM [7] for the classification to get the weight for each feature map. As the value in the gradient map can be either positive or negative, we divide all $k$ feature maps into two parts ($k_1$ and $k_2$, $k_1 + k_2 = k$), the one with positive gradient $A_k^{c(+)}$ and another with negative gradient $A_k^{c(-)}$. $\alpha_k^c$ is the weight for each feature map $k$ of target layer $l$ calculated by taking the mean value of the gradient map $G_k^{l(c)}$. The negative $\alpha$ is considered to reduce the target score, so we sum two parts separately and then subtract the negative part from the positive one (as Eq. 3.5) to get a smoother saliency map, and then use the $ReLU$ function to remove the pixel not contributing to the prediction.

$$A_{k_2}^{c(-)} = \alpha_{k_2}^{c(-)} A_{k_2}^c \tag{3.3}$$

$$A_{k_1}^{c(+)} = \alpha_{k_1}^{c(+)} A_{k_1}^c \tag{3.4}$$

$$L_{\text{CAM}}^c = ReLU\left(\sum_{k_1} A_{k_1}^{c(+)} - \sum_{k_2} A_{k_2}^{c(-)}\right) \tag{3.5}$$

Because GradCAM can only explain classification models, it highlights all objects of the same class $c$. By detecting the target object's location, we can tailor G-CAME to the object detection problem by explaining only one target object.

### 3.4. **Masking target region with normal distribution**

To deal with the localization issue, we propose to use the normal distribution to estimate the region around the object's center. Because the gradient map shows the target object's

location, we estimate the object region around the pixel representing the object's center by using a Gaussian mask as the weight for each pixel in the weighted feature map $k$. The Gaussian kernel is defined as:

$$G_\sigma = \frac{1}{2\pi\sigma^2} \exp^{-\frac{(x^2+y^2)}{2\sigma^2}} \tag{3.6}$$

where the term $\sigma$ is the standard deviation of the value in the Gaussian kernel and controls the kernel size $\kappa$. $x$ and $y$ are two linear-space vectors filled with value in range $[1, \kappa]$ one vertically and another horizontally. The bigger $\sigma$ is, the larger highlighted region we get. For each feature map $k$ in layer $l$, we apply the Gaussian kernel to get the region of the target object and then sum all these weighted feature maps. In general, we slightly adjusted the weighting feature map (Eq. 3.5) to get the final saliency map as shown in Eq. 3.7:

$$L^c_{\text{GCAME}} = ReLU\left(\sum_{k_1} G_{\sigma(k_1)} \odot A^{c(+)}_{k_1} - \sum_{k_2} G_{\sigma(k_2)} \odot A^{c(-)}_{k_2}\right) \tag{3.7}$$

#### 3.4.1. **Choosing $\sigma$ for Gaussian mask**

The Gaussian masks are applied to all feature maps, with the kernel size being the size of each feature map, and the $\sigma$ is calculated as in Eq. 3.10.

$$R = \log \left| \frac{1}{Z} \sum_i \sum_j G^{l(c)}_k \right| \tag{3.8}$$

$$S = \sqrt{\frac{H \times W}{h \times w}} \tag{3.9}$$

$$\sigma = R \log S \times \frac{3}{\lfloor \frac{\sqrt{h \times w} - 1}{2} \rfloor} \tag{3.10}$$

where the $\sigma$ is combined by two terms. In the first term, we calculate the expansion factor with $R$ representing the importance of location map $G^{l(c)}_k$ and $S$ is the scale between the original image size $(H \times W)$ and the feature map size $(h \times w)$. We use the logarithm function to adjust the value of the first term so that its value can match the size of the gradient map. For multi-scale object detectors, we have a different $S$ for each scale level. In the second term, we choose Gaussian kernel size based on the $3\sigma$-rule [23] as the Eq. 3.11 and take the inverse value.

$$\kappa = 2 \times \lceil 3\sigma \rceil + 1 \tag{3.11}$$

#### 3.4.2. **Gaussian mask generation**

We generate each Gaussian mask with the following steps:

(1) Create a grid filled with value in range $[0, w]$ for the width and $[0, h]$ for the height ($w$ and $h$ is the size of the location map $G^{l(c)}_k$).
(2) Subtract the grid with value in position $(i_t, j_t)$ where $(i_t, j_t)$ is the center pixel of the target object on the location map.
(3) Apply Gaussian formula (Eq. 3.6) with $\sigma$ as the expansion factor as Eq. 3.10 to get the Gaussian distribution for all values in the grid.
(4) Normalize all values in range $[0, 1]$.

By normalizing all values in range $[0, 1]$, Gaussian masks only keep the region relating to the object we aim to explain and remove other unrelated regions in the weighted feature map.

## 4. Experiments and Results

We performed our experiment on the MS-COCO 2017 [24] dataset with 5000 validation images. The models in our experiment are YOLOX-l (one-stage model) and Faster-RCNN (two-stage model). All experiments and conducted on NVIDIA Tesla P100 GPU. G-CAME's inference time depends on the number of feature maps in selected layer $l$. Our experiments run on model YOLOX-l with 256 feature maps for roughly 0.5 second per object.

### 4.1. Sanity check

To validate whether the saliency map is a faithful explanation or not, we perform a sanity check [25] with Cascading Randomization and Independent Randomization. In the Cascading Randomization approach, we randomly choose five convolution layers as the test layers. Then, for each layer between the selected layer and the top layer, we remove the pre-trained weights, reinitialize with normal distribution, and perform G-CAME to get the explanation for the target object. In contrast to Independent Randomization, we only reinitialize the weight of the selected layer and retain other pre-trained weights. The sanity check results show that G-CAME is sensitive to model parameters and can produce valid results, as shown in Fig. 2.

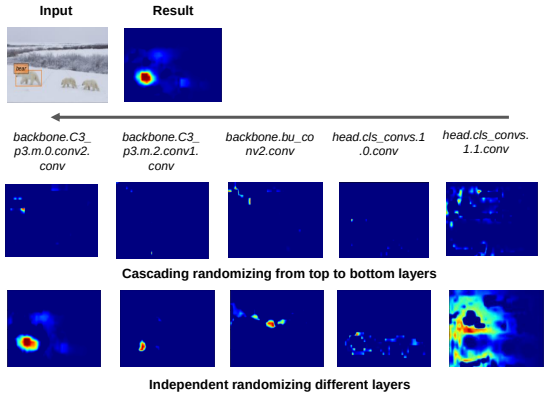



*Figure 2.* The result of Cascading Randomization and Independent Randomization for five layers from top to bottom of the YOLOX model. Chosen layers in the head part do not include the layer in the regression branch. The result shows G-CAME is sensitive to the model's parameters.

### 4.2. Qualitative evaluation

We performed a saliency map qualitative evaluation of G-CAME in comparison with D-RISE. We use D-RISE's default parameters [3], where each grid's size is $16 \times 16$, the probability of each grid's occurrence is 0.5, and the amount of samples for each image is 4000. For G-CAME, we choose the target layers as shown in Sec. 3.1 to calculate the derivative.

Fig. 3 shows the results of G-CAME compared with GradCAM and D-RISE. GradCAM is only applicable for the classification task, as it shows the saliency maps for all objects in the same class. Considering XAI methods for object detectors, where G-CAME and D-RISE can deliver the explanations for a specific object, G-CAME can generate saliency maps where the random noises are significantly reduced in comparison with D-RISE.

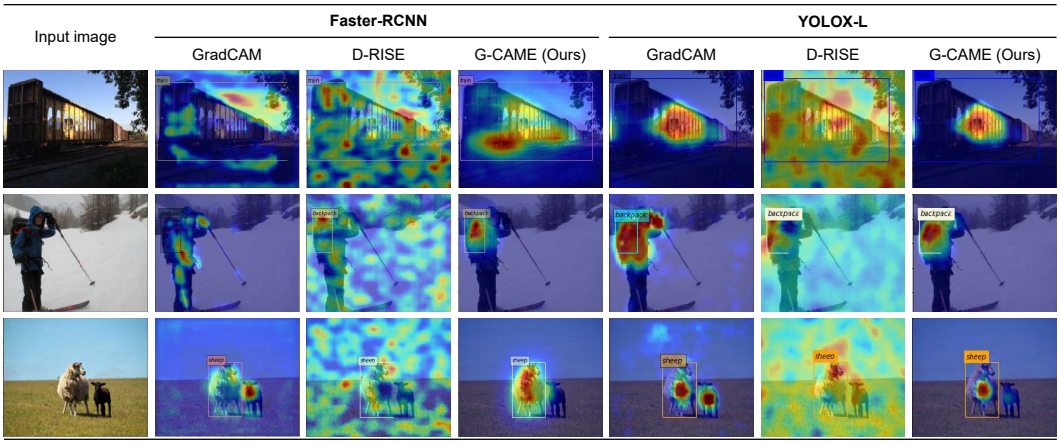



*Figure 3.* Visualization results of GradCAM, D-RISE, and G-CAME on samples of MS-COCO 2017 dataset. G-CAME can generate the least noisy saliency maps for explaining a specific object.

### 4.3. **Quantitative localization evaluation**

We use two standard metrics, Pointing Game [26] and Energy-based Pointing Game [27] to compare the correlation between an object's saliency map and human-labeled ground truth. The results are shown in Table 1.

#### 4.3.1. **Pointing Game (PG)**

To evaluate XAI methods via PG metric, firstly, we run the model on the dataset and get the bounding boxes that best match the ground truth for each class on each image. A *hit* is scored if the highest point of the saliency map lies inside the ground truth; otherwise, a *miss* is counted. The pointing game score for each image is calculated by

$$PG = \frac{\#Hits}{\#Hits + \#Misses} \tag{4.1}$$

This score should be high for a good explanation to evaluate an XAI method.

#### 4.3.2. **Energy-Based Pointing Game (EBPG)**

EBPG [27] calculates how much the energy of the saliency map falls inside the bounding box. Similar to the PG score, a good explanation is considered to have a higher EBPG. EBPG formula is defined as follows:

$$EBPG = \frac{\sum L^c_{(i,j)\in bbox}}{L^c_{(i,j)\in bbox} + L^c_{(i,j)\notin bbox}} \tag{4.2}$$

PG and EBPG results are reported in Table 1. Specifically, more than 65% energy of G-CAME's saliency map falls into the ground truth bounding box compared with only 18.4% of D-RISE. In other words, G-CAME drastically reduces noises in the saliency map. In PG evaluation, G-CAME also gives better results than D-RISE. 98% of the highest pixel lie inside the correct bounding box, while this number in D-RISE is 86%.

#### 4.3.3. **Bias in Tiny Object Detection**

Explaining tiny objects detected by the model can be a challenge for XAI methods. In particular, the saliency map may be biased toward the neighboring region. This issue can

| Method | D-RISE | G-CAME (Our) |
|---|---|---|
| PG%↑ (Overall \| Tiny object) | 0.86 \| 0.127 | **0.98 \| 0.158** |
| EBPG%↑ (Overall \| Tiny object) | 0.184 \| 0.009 | **0.671 \| 0.261** |

| Method | D-RISE | G-CAME (Our) |
|---|---|---|
| Confidence Drop%↑ | **42.3** | 36.8 |
| Information Drop%↓ | 31.58 | **29.15** |
| Running time(s)↓ | 252 | **0.435** |

*Table 1.* Comparison of D-RISE and G-CAME (Our) on the MS-COCO 2017 validation dataset with the YOLOX model. Evaluation metrics include PG%, EBPG%, Confidence Drop, Information Drop, and Running time. Higher or lower scores are better as indicated by ↑ / ↓. The best results are shown in bold.

worsen when multiple tiny objects partially or fully overlap because the saliency map stays in the same location for every object. In our experiments, we define the tiny object by calculating the ratio of the predicted bounding box area to the input image area (640×640 in YOLOX). An object is considered tiny when this ratio is less than or equal to 0.005. In Fig. 4, we compare G-CAME with D-RISE in explaining tiny object prediction for two cases. In the first case (Fig. 4a), we test the performance of D-RISE and G-CAME in explaining two tiny objects of the same class. The result shows that D-RISE fails to distinguish two "traffic lights", where the saliency maps are nearly identical. For the case of multiple objects with different classes overlapping (Fig. 4b), the saliency maps produced by D-RISE hardly focus on one specific target. The saliency corresponding to the "surfboard" even covers the "person", and so does the explanation of the "person". The problem can be the grid's size in D-RISE, but changing to a much smaller grid's size can make the detector unable to predict. In contrast, G-CAME can clearly show the target object's localization in both cases and reduce the saliency map's bias to unrelated regions. In detail, we evaluated our method only in explaining tiny object prediction with EBPG score. The MS-COCO 2017 validation dataset has more than 8000 tiny objects, and the results are reported in Table 1. Our method outperforms D-RISE with more than 26% energy of the saliency map falling into the predicted box, while this figure in D-RISE is only 0.9%. Especially, most of the energy in D-RISE's explanation does not focus on the correct target. In the PG score, instead of evaluating one pixel, we assess all pixels having the same value as the pixel with the highest value. The result also shows that G-CAME's explanation has better accuracy than D-RISE's.

### 4.4. **Quantitative faithfulness evaluation**

Another essential aspect of an XAI method is the ability to ensure the explanation's completeness and consistency for the model's predictions. In this section, we employ the Confidence Drop score and Information Drop score to evaluate G-CAME and D-RISE on the YOLOX model with the MS-COCO 2017 dataset.

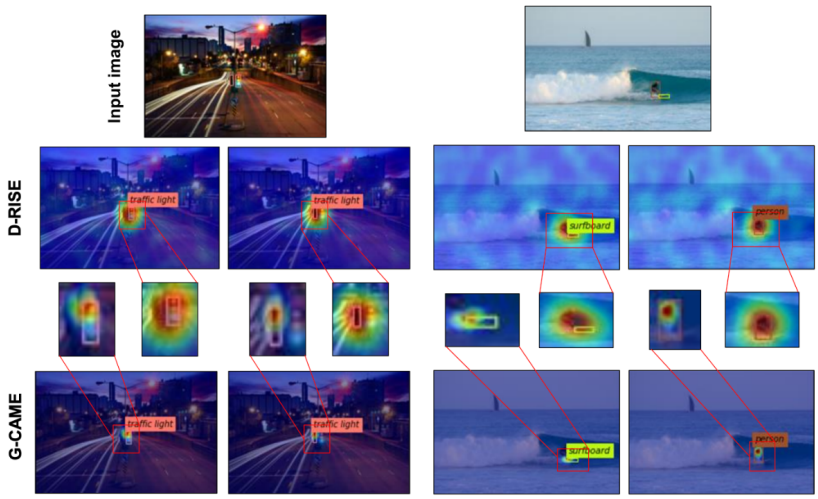
#### 4.4.1. **Confidence Drop**

We employ the Average Drop metric to evaluate the confidence change [19, 20, 28] in the model's prediction for the target object when using the explanation as the input. In other words, when we remove these important regions, the confidence score of the target box should be dropped. The Average Drop can be calculated by the formula:

$$AD = \frac{1}{N}\sum_{i=1}^{N}\frac{max(P_c(I_i) - P_c(\tilde{I}_i), 0)}{P_c(I_i)} \times 100 \tag{4.3}$$

where:

$$\tilde{I}_o = I \odot (1 - M_o) + \mu M_o \tag{4.4}$$




(a) Explanation for tiny objects of different class (b) Explanation for tiny objects of the same class

*Figure 4.* The saliency map of D-RISE and G-CAME for tiny objects prediction. We evaluate them on two cases: (a) tiny objects of the same class lying close together and (b) multiple tiny objects of different classes lying close together. In both cases, G-CAME can clearly distinguish each object in its explanations.

$$P_c(\tilde{I}) = IoU(L_i, L_j) \cdot p_{c(L_j)} \tag{4.5}$$

Here, we tailor the original formula of Average Drop for the object detection model. In Eq. 4.4, we create a new input image masked by the explanation $M$ of G-CAME. $\mu$ is the mean value of the original image. With the value of $M$, we only keep 20% of the pixel with the most significant value in the original explanation and set the rest as 0. Then, we can minimize the explanation's noise, and the saliency map can focus on the regions most influencing the prediction.

In Eq. 4.5, to compute probability $P_c(\tilde{I})$, we first calculate the pair-wise $IoU$ of the box $L_j$ predicted on perturbed image $\tilde{I}$ with the box $L_i$ predicted on the original image and take the one with the highest value. After that, we multiply the first term with the corresponding class score $p_{c(L_j)}$ of the box. In calculating $P_c(I_i)$, the $IoU$ equals 1, so the value remains the original confidence score. Hence, if the explanation is faithful, the confidence drop should increase. However, removing several pixels can penalize the method of producing the saliency map that has connected and coherent regions. Specifically, pixels representing the object's edges are more meaningful than others in the middle [29]. For example, pixels representing the dog's tail are easier to recognize than others lying on the dog's body.

#### 4.4.2. **Information Drop**

Besides the Confidence Drop score, we measure the faithfulness of the method via the Information Drop score. We compare the information level of the bokeh image, which is created by removing several pixels from the original image after applying the XAI method. To measure the bokeh image's information, we use WebP [30] format and calculate the Information Drop score by taking the proportion of the compressed size of the bokeh image to the original image [29].

### 4.5. **Evaluation**

Table 1 highlights the strengths of our G-CAME compared to D-RISE. D-RISE achieves a 42.3% Confidence Drop by spreading its saliency map across the image, leading to a significant but less targeted reduction in confidence. This approach contrasts with G-CAME, which maintains focus on the target object, resulting in a more modest confidence drop that signifies a precise and relevant explanation. Crucially, G-CAME outperforms D-RISE in Information Drop, scoring 29.1% against 31.58%, indicating superior preservation of the original image's content. Additionally, our method offers a significant speed advantage, delivering explanations in under a second, compared to D-RISE's four-minute runtime. These results demonstrate G-CAME's efficiency in providing focused, relevant, and quick explanations for object detection models.

## 5. **Conclusion**

In this paper, we proposed G-CAME, a novel CAM-based XAI method elevating the Gaussian kernel to explain one-stage and two-stage object detection models. The experiment's results show that our method can plausibly explain the model's predictions and reduce the bias in tiny object detection. Moreover, our method's runtime is relatively short, overcoming the time constraint of existing region-based methods and reducing the noise in the saliency map.

## Acknowledgment

This work was partially supported by the NBIF Talent Recruitment Fund (TRF2003-001) and the UNB-FCS Startup Fund (22-23 START UP/ H CAO).

## References

[1] R. Girshick, J. Donahue, T. Darrell, and J. Malik. "Rich feature hierarchies for accurate object detection and semantic segmentation". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2014, pp. 580–587.

[2] T. T. H. Nguyen, V. B. Truong, V. T. K. Nguyen, Q. H. Cao, and Q. K. Nguyen. "Towards Trust of Explainable AI in Thyroid Nodule Diagnosis". In: *arXiv preprint arXiv:2303.04731* (2023).

[3] V. Petsiuk, R. Jain, V. Manjunatha, V. I. Morariu, A. Mehra, V. Ordonez, and K. Saenko. "Black-box explanation of object detectors via saliency maps". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 11443–11452.

[4] J. H. Sejr, P. Schneider-Kamp, and N. Ayoub. "Surrogate Object Detection Explainer (SODEx) with YOLOv4 and LIME". In: *Machine Learning and Knowledge Extraction* 3.3 (2021), pp. 662–671.

[5] 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*. 2016, pp. 1135–1144.

[6] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba. "Learning deep features for discriminative localization". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 2921–2929.

[7] 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: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 618–626.

[8] Q. Zhang, L. Rao, and Y. Yang. "Group-CAM: group score-weighted visual explanations for deep convolutional networks". In: *arXiv preprint arXiv:2103.13859* (2021).

[9] Z. Ge, S. Liu, F. Wang, Z. Li, and J. Sun. "Yolox: Exceeding yolo series in 2021". In: *arXiv preprint arXiv:2107.08430* (2021).

[10] S. Ren, K. He, R. Girshick, and J. Sun. "Faster r-cnn: Towards real-time object detection with region proposal networks". In: *Advances in neural information processing systems* 28 (2015).

[11] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. "You only look once: Unified, real-time object detection". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 779–788.

[12] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg. "Ssd: Single shot multibox detector". In: *European conference on computer vision*. Springer. 2016, pp. 21–37.

[13] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár. "Focal loss for dense object detection". In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 2980–2988.

[14] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie. "Feature pyramid networks for object detection". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 2117–2125.

[15] J. Dai, Y. Li, K. He, and J. Sun. "R-fcn: Object detection via region-based fully convolutional networks". In: *Advances in neural information processing systems* 29 (2016).

[16] A. Grami. "The Gaussian Distribution". In: *Probability, Random Variables, Statistics, and Random Processes: Fundamentals & Applications*. Wiley, 2019, pp. 201–238. DOI: 10.1002/9781119300847.ch7. URL: https://ieeexplore.ieee.org/document/8689279.

[17] V. Petsiuk, A. Das, and K. Saenko. "Rise: Randomized input sampling for explanation of black-box models". In: *arXiv preprint arXiv:1806.07421* (2018).

[18] V. B. Truong, T. T. H. Nguyen, V. T. K. Nguyen, Q. K. Nguyen, and Q. H. Cao. "Towards Better Explanations for Object Detection". In: *arXiv preprint arXiv:2306.02744* (2023).

[19] A. Chattopadhay, A. Sarkar, P. Howlader, and V. N. Balasubramanian. "Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks". In: *2018 IEEE winter conference on applications of computer vision (WACV)*. IEEE. 2018, pp. 839–847.

[20] R. Fu, Q. Hu, X. Dong, Y. Guo, Y. Gao, and B. Li. "Axiom-based grad-cam: Towards accurate visualization and explanation of cnns". In: *arXiv preprint arXiv:2008.02312* (2020).

[21] H. T. T. Nguyen, H. Q. Cao, K. V. T. Nguyen, and N. D. K. Pham. "Evaluation of explainable artificial intelligence: Shap, lime, and cam". In: *Proceedings of the FPT AI Conference*. 2021, pp. 1–6.

[22] Y. Xin, G. Wang, M. Mao, Y. Feng, Q. Dang, Y. Ma, E. Ding, and S. Han. "Pafnet: An efficient anchor-free object detector guidance". In: *arXiv preprint arXiv:2104.13534* (2021).

[23] F. Pukelsheim. "The three sigma rule". In: *The American Statistician* 48.2 (1994), pp. 88–91.

[24] 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: *European conference on computer vision*. Springer. 2014, pp. 740–755.

[25] J. Adebayo, J. Gilmer, M. Muelly, I. Goodfellow, M. Hardt, and B. Kim. "Sanity checks for saliency maps". In: *Advances in neural information processing systems* 31 (2018).

[26] J. Zhang, S. A. Bargal, Z. Lin, J. Brandt, X. Shen, and S. Sclaroff. "Top-down neural attention by excitation backprop". In: *International Journal of Computer Vision* 126.10 (2018), pp. 1084–1102.

[27] H. Wang, Z. Wang, M. Du, F. Yang, Z. Zhang, S. Ding, P. Mardziel, and X. Hu. "Score-CAM: Score-weighted visual explanations for convolutional neural networks". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*. 2020, pp. 24–25.

[28] H. G. Ramaswamy et al. "Ablation-cam: Visual explanations for deep convolutional network via gradient-free localization". In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 2020, pp. 983–991.

[29] A. Kapishnikov, T. Bolukbasi, F. Viégas, and M. Terry. "Xrai: Better attributions through regions". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2019, pp. 4948–4957.

[30] *Google, WebP format*. https://developers.google.com/speed/webp.