Grad CAM

anjanakr1916 & Karthika Sethunath

April 2022

1 Different CAM based models

1.1 Class Activation Maps

Technique for producing heat map to highlight class specific regions. After the convolution layers, consider there are k feature maps, each feature map have height v and width u. $A^k \in R^{(u,v)}$ Global average pooling turns feature maps into single number by taking the average of numbers in feature map. After the k numbers we turn these k numbers into classification decision using single fully connected layer. Each GAP(A1) is connected to each of output classes by weights.

$y^{cat} = \sum_{k=1}^{k} W_k^{cat} \frac{1}{z} \sum_{i=1}^{u} \sum_{j=1}^{v} A_{ij}^k$

Then, y^{cat} score $= W_1 * GAP(A1) + W_2 * GAP(A2) + W_3 * GAP(A3)$

Say $W_1, W_2, W_3$are weights connected to cat after GAP then y score is those weights multiplied to number produced by GAP.

$L_{cam}^c = W_1 * A1 + W_2 * A2 + W_3 * A3$. This output will be a matrix.

CAM-Based explanation provide visual explanation for a single input with a linear weighted combination of activation maps from convolutional layers. CAM creates localized visual explanations but is architecture sensitive, a global pooling layer is required to follow the convolutional layer of interest. Grad-CAM and its variations, e.g. Grad-CAM++, intend to generalize CAM to models without global pooling layers.

1.2 Grad CAM

To resolve the above mentioned problem, Grad-CAM extends the definition of α_k^c as the gradient of class confidence Y_c w.r.t. the activation map A_k.

$L_{Grad-cam}^c = ReLU(\sum_k \alpha_k^c A_k)$

where, $\alpha_k^c = \frac{1}{z} \sum_{i=1}^{u} \sum_{j=1}^{v} \frac{\partial y^c}{\partial A_{ij}^k}$

Disadvantages of Grad CAM: - Grad CAM's performance drops when localizing multiple occurences of the same class. - For a single object images, Grad - CAM heatmaps often does not capture the entire object in completeness.

1.3 Grad CAM++

Grad CAM++ is a more generalized visualization technique that addresses the limitations of Grad CAM. In Grad CAM if there are multiple occurrences of an object with slightly different orientations or views, different feature maps maybe activated with differing spatial footprints and the feature maps with lesser footprints fade away in the final map. This problem is addressed in Grad CAM++, by taking a weighted average of the pixel-wise gradients.

$w_k^c = \sum_i \sum_j \alpha_{ij}^{kc} * relu(\frac{\partial Y^c}{\partial A_{ij}^k})$

where, $\alpha_{ij}^{kc} = \frac{1}{\sum_{lm} \frac{\partial Y^c}{\partial A_{lm}^k}}$ if $\frac{\partial Y^c}{\partial A_{ij}^k} = 1$

presence of objects in all feature maps are highlighted with equal importance.

1.4 Smooth Grad CAM++

In this paper they introduce a smooth gradient that can help visually sharpen gradient-based sensitivity maps by taking random samples in a neighborhood of an input x, and averaging the resulting sensitivity maps. In this method a set of noised sample images (n) is taken by adding Gaussian noise to the input. The standard deviation is taken as 0.15. Values are varied according to the results we want. We take the average of all 1st, 2nd and 3rd order partial derivatives of all n noised inputs and apply the resulting averaged derivatives in computing α_{ij}^{kc} and w_c^k.

Let D_1^k, D_2^k, D_3^k denote matrices of 1st, 2nd and 3rd order partial derivatives respectively for feature map k. We compute α^{kc} as:

$$\alpha_{ij}^{kc} = \frac{\frac{1}{n} * \sum_1^n D_1^k}{2 * \frac{1}{n} * \sum_1^n D_2^k + \sum_a \sum_b A_{a,b}^k * \frac{1}{n} * \sum_1^n D_3^k}$$

substituting the averaged gradient in Grad-CAM++ equation, the weights w_c^k becomes:

$$w_k^c = \sum_i \sum_j \alpha_{ij}^{kc} * relu(\frac{\partial Y^c}{\partial A_{ij}^k})$$

Smooth Grad-CAM++ performs well in object localization and also in multiple occurrences of an object of same class.

1.5 Ablation CAM

Uses Ablation analysis to determine the importance of individual feature map unit with respect to class. Like grad cam we take the last layer say layer l. From l we can take the activation maps $A^1, A^2 ... A^k$. Then compute these activation maps specific weights $w_c^1, w_c^2 ... w_c^k$. We set the activation of spatial location (i,j) or activation map A^k to zero to get the slope.

$$W_k = \frac{f(A_{ij}) - f(0)}{A_{ij}^k}$$

We set all individual activation cell values of feature map Ak to zero and repeat all.The importance value can be the fraction drop in activation score of class c when Ak is removed.

$$L_{ablation-cam}^c = ReLU(\sum_k W_k^c A_k)$$

1.6 XGradCAM

In XGradCAM paper, we try to satisfy two basic axioms, i.e., sensitivity and conservation which is not considered in any of the CAM methods. - Sensitivity: The importance of each feature map should be equivalent to the score change caused by its removing. - Conservation: The responses of the map should be a redistribution of the class score.

XGradCAM is still a linear combination of feature maps. The only difference comes in the weights substituted. To meet the two axioms they formulate a minimization formula $\phi(w_c^k)$.

$$\phi(w_c^k) = \sum_{k=1}^k |S_c(F^l) - S_c(F^l - F^{lk}) - \sum_{xy}(w_c^k F^{lk}(x,y))| + |S_c(F^l) - \sum_{xy}(\sum_{k=1}^k (w_c^k F^{lk}(x,y)))|$$

Some of the terms are difficult to optimize by the variable w_c^k since there are no direct relationship between these terms and the k-th feature map of the target layer. Without considering terms, we get,

$$\alpha_c^k = \sum_{xy} \frac{F^{lk}(x,y)}{\sum_{x,y} F^{lk}(x,y)} * \frac{\partial S_c(F^l)}{^{lk}(x,y)}$$

2 Disadvantages of gradient based approaches

Assuming we have two activation maps A_l^i and A_l^j their corresponding weights were $\alpha_i^c > \alpha_j^c$ which means that the input region which generates feature map A_l^i is of more importance compared to that of the other region that generates A_l^j. However, it is easy to find counterexamples with false confidence in Grad-CAM: activation maps with higher weights show lower contribution to the network's output[]. This phenomenon may be caused by the global pooling operation on the top of the gradients and the gradient vanishing issue in the network.

Table 1: Weights

GradCAM variations	Weights
CAM	$y^{cat}score = W_1 * GAP(A1) + W_2 * GAP(A2) + W_3 * GAP(A3)$
GradCAM	$\alpha_k^c = \frac{1}{z}\sum_{i=1}^{u}\sum_{j=1}^{v}\frac{\partial y^c}{\partial A_{ij}^k}$
GradCAM++	$\alpha_{ij}^{kc} = \frac{1}{\sum_{lm}\frac{\partial Y^c}{\partial A_{lm}^k}}$ if $\frac{\partial Y^c}{\partial A_{ij}^k} = 1$
Smooth GradCAM++	$\alpha_{ij}^{kc} = \frac{\frac{1}{n}*\sum_1^n D_1^k}{2*\frac{1}{n}*\sum_1^n D_2^k + \sum_a\sum_b A_{a,b}^k * \frac{1}{n}*\sum_1^n D_3^k}$
Ablation CAM	$W_k = \frac{f(A_{ij}) - f(0)}{A_{ij}^k}$
XGrad CAM	$\alpha_c^k = \sum_{xy}\frac{F^{lk}(x,y)}{\sum_{x,y}F^{lk}(x,y)} * \frac{\partial S_c(F^l)}{lk(x,y)}$

3 RESNET-50

4 VGG-16

5 Evaluation metric

IOU - higher the better Confidence score - average drop is lower the better Percentage increase - higher the better

In IOU X-gradcam has the lowest value for VGG-16, in resnet-50 both gradcam and Xgradcam have the lowest value. In Confidence score Layer cam and gradcampp has lowest value. Smooth gradcampp has higher value in resnet-50 percentage increase and gradcampp has higher value in vgg-16.

6 Evaluating Visualizations

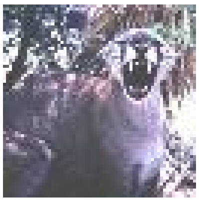
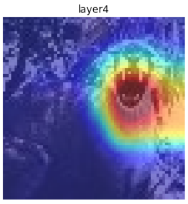
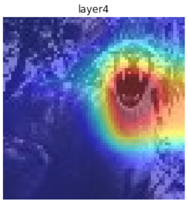
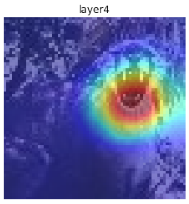
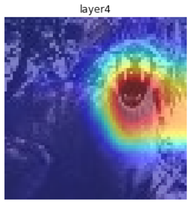
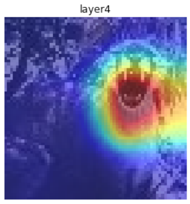
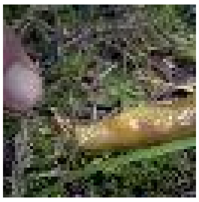
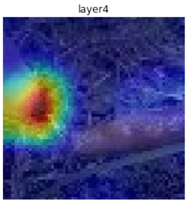
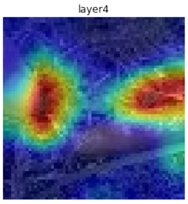
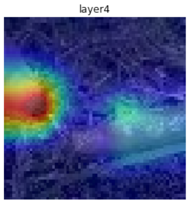
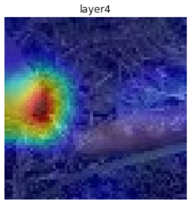
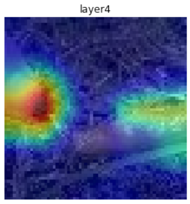
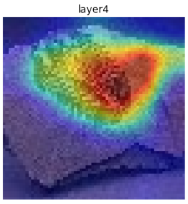
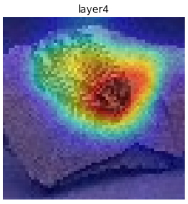
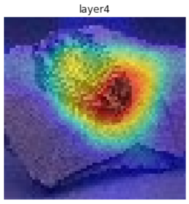
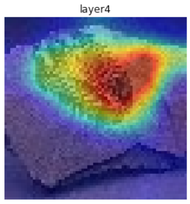
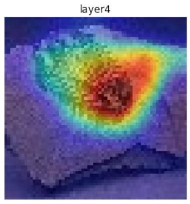
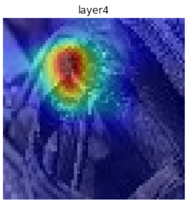
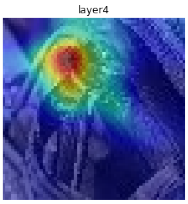
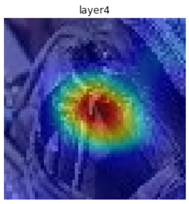
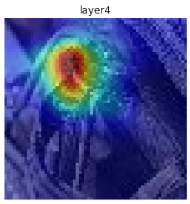
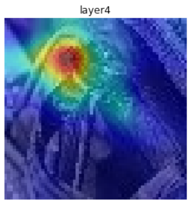
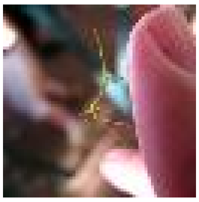
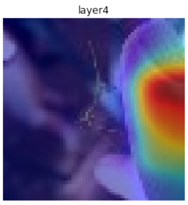
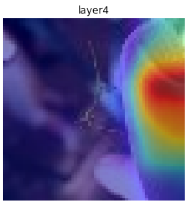
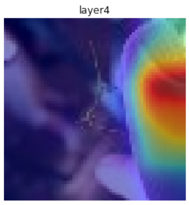
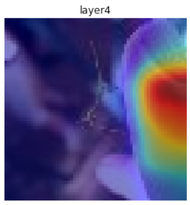
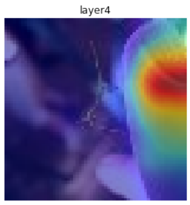
In this section we investigate the accuracy and authenticity of the different CAMs within resnet50 and Vgg16 architectures. We evaluate the performance of the proposed models by calculating the intersection of union. Intersection of Union(IOU) is an evaluation metric that is used to measure the accuracy of an object detector. In our experiment we use IOU to evaluate the ability of different CAM models to highlight important features of an image according to the target label.
Given an image, we first get the coordinates for the ground truth bounding box. According to the dataset we used(ImageNet), the coordinates for the ground truth bounding box was already provided. Then we pass the image through different CAM models which returns the activation map. Next we draw a bounding box around the highlighted features in the activation map. IOU is calculated by dividing the area of overlap by the area of union between the two boxes.
If the IOU value is equal to zero, then there is no overlap between the boxes and if the IOU value is equal to one, then there is a full overlap between the boxes. This method will help us see if the activated or highlighted features fall inside the ground truth bounding box.
For further understanding on the performance of the CAM models, we take the correctly classified images and plot the IOU values for the top prediction (i.e image with the highest score) and also plot the IOU values for the last prediction (i.e image with the least score). We then compare these values to examine the ability of the CAM models to highlight the relevant features which supposedly should fall inside the ground truth bounding box.

In figure 1 the IOU values for the top predictions are plotted for all CAM models(Vgg-16 architecture). In figure 2 the IOU values for the least predictions are plotted. In figure 1 we can observe that the values are spread out between 0 and 1. Most of the IOU values lie between 0.2 and 0.6 for all CAM models. In figure 2 we can observe that the IOU value for GradCAM and XGradCAM are mostly concentrated on the zero lines although there are a few values which goes beyond 0.5. While looking at the IOU values of LayerCAM and GradCAMPlusPlus the points are spread out and it lies within the range of 0.2 and 0.6.

In case of GradCAM and XGradCAM we can infer that the highlighted features for the least predicted category does not fall inside the ground truth bounding box. On the otherhand, in the case of LayerCAM and GradCAMPlusPlus, we can infer that even the highlighted features for the least predicted category falls inside the ground truth bounding box.

We do the same experiment for all the CAMS using resnet50 architecture.
From figure 4 we can infer that for LayerCAM and GradCAMPlusPLus the IOU values for the least predicted

Table 2: Table for true image

Image	GradCAM	GradCAM++	SmoothGradCAM++	XGradCAM	Layer CAM
					
					
					
					
					

category falls in the same range as the IOU values of the top predicted category. This means that the highlighted features for the top and least predicted category are similar and falls inside the ground truth bounding box. Whereas for GradCAM and XGradCAM the IOU values for the least predicted category falls mostly on the zero line.

References

[1] Ramaswamy, Harish Guruprasad. Ablation-cam: Visual explanations for deep convolutional network via gradient-free localization. Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. 2020.

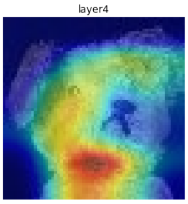
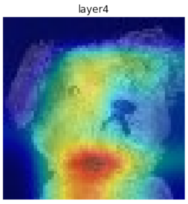
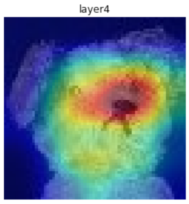
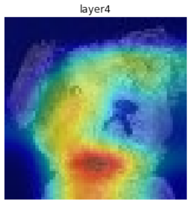
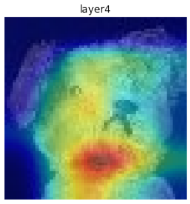
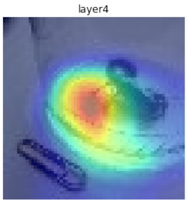
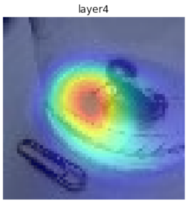
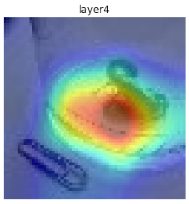
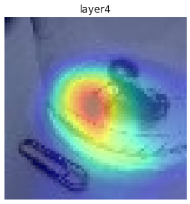
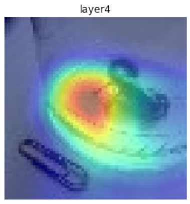
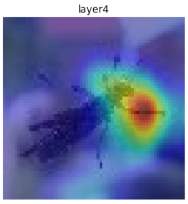
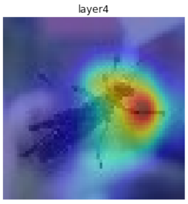
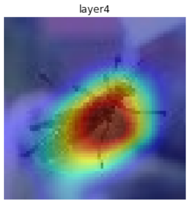
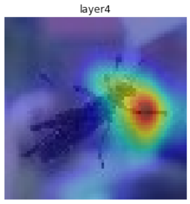
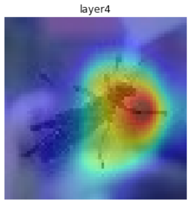
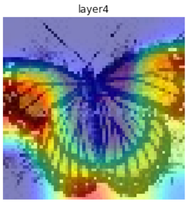
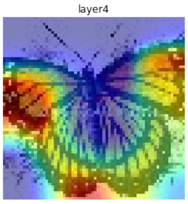
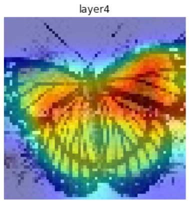
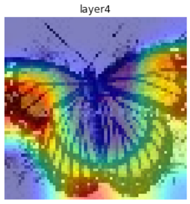
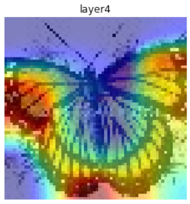
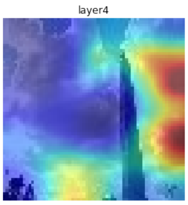
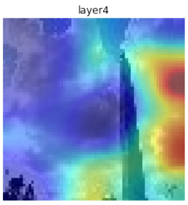
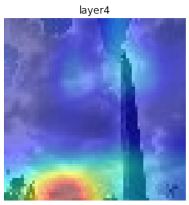
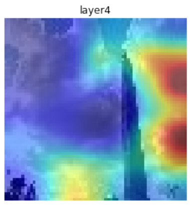
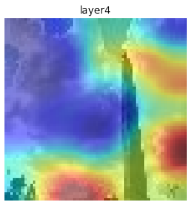
[2] Selvaraju, R.R., Cogswell, M., Das, A. et al. Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization.. Int J Comput Vis 128, 336–359 (2020).

[3] Aditya Chattopadhyay Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks. 2018 IEEE winter conference on applications of computer vision (WACV). IEEE, 2018.

[4] Omeiza, Daniel, et al. Smooth grad-cam++: An enhanced inference level visualization technique for deep convolutional neural network models. arXiv preprint arXiv:1908.01224 (2019).

[5] Fu, Ruigang, et al. Axiom-based grad-cam: Towards accurate visualization and explanation of cnns. arXiv preprint arXiv:2008.02312 (2020).

Table 3: Table for false image

Image	GradCAM	GradCAM++	SmoothGradCAM++	XGradCAM	Layer CAM
					
					
					
					
					

[6] Wang, Haofan, et al. "Score-CAM: Score-weighted visual explanations for convolutional neural networks." Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops. 2020.

Figure 1: IOU values of the top predictions (Vgg-16 architecture)

Table 4: Table showing Activation map of true class for false prediction

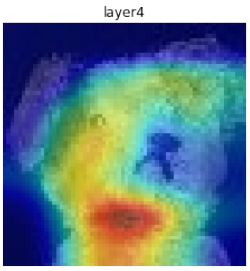
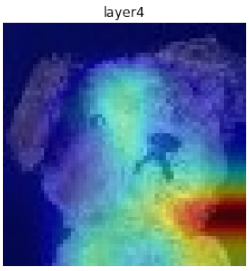
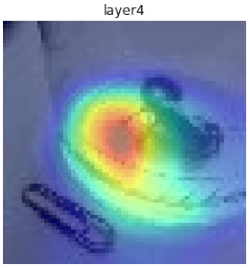
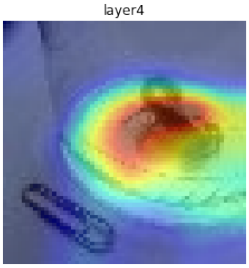
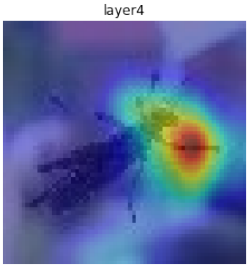
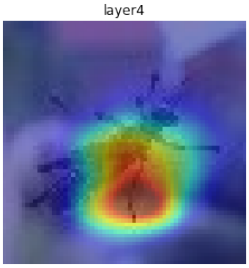
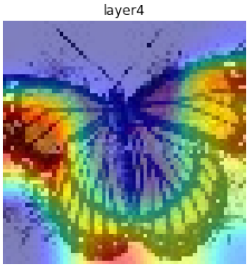
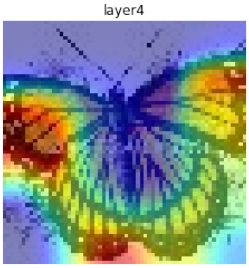
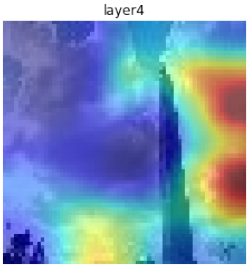
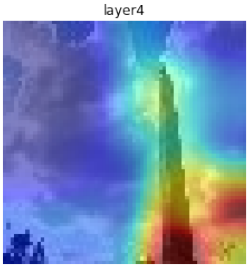
Predicted class	Predicted activation map	Actual class	Activation Map of actual class
Backpack, back pack, knapsack, packsack, rucksack, haversack		teddy, teddy bear	
Goldfish, Carassius auratus		scorpion	
nil		grasshopper, hopper	
goldfish, Carassius auratus		monarch, monarch butterfly, milkweed butterfly, Danaus plexippusr	
nil		obelisk	

Table 5: Table for true image

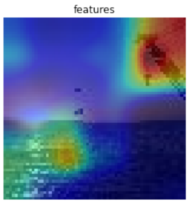
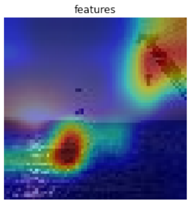
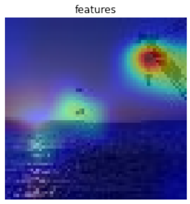
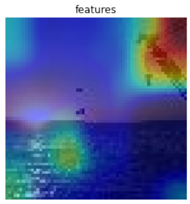
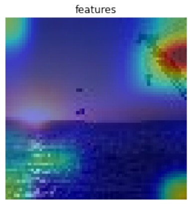
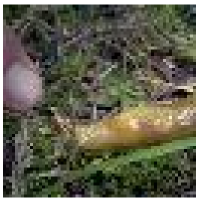
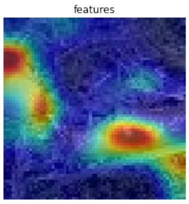
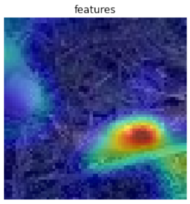
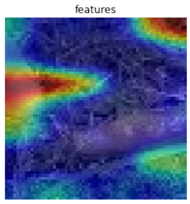
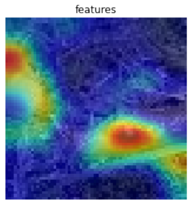
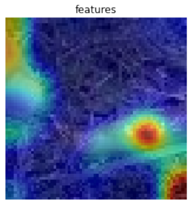
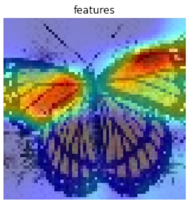
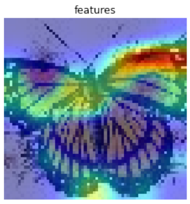
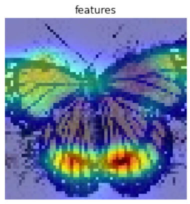
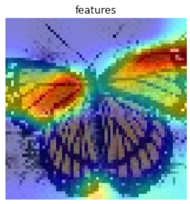
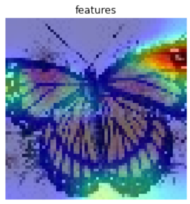
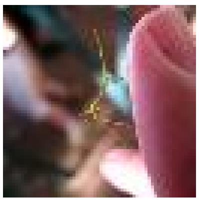
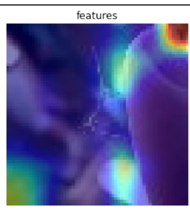
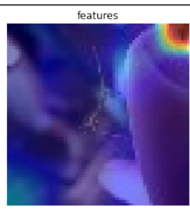
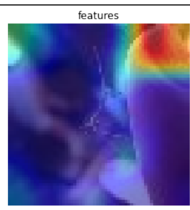
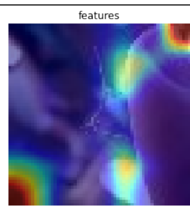
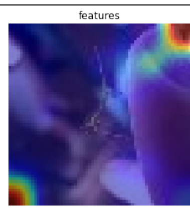
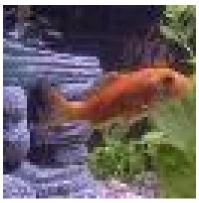
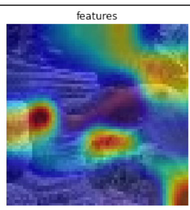
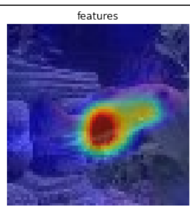
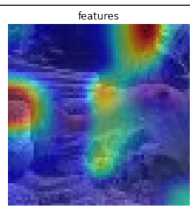
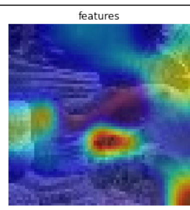
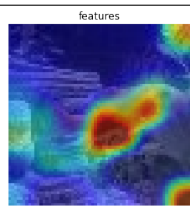
Image	GradCAM	GradCAM++	SmoothGradCAM++	XGradCAM	Layer CAM
					
					
					
					
					

Table 6: Table for false image

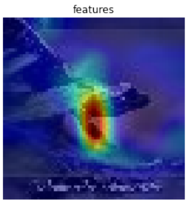
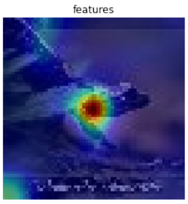
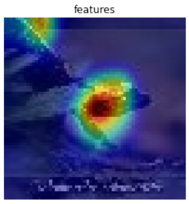
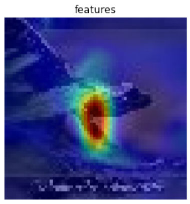
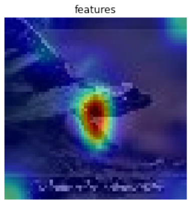
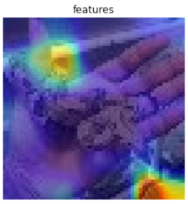
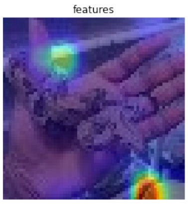
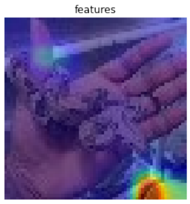
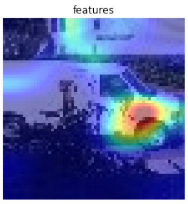
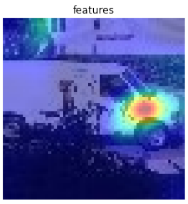
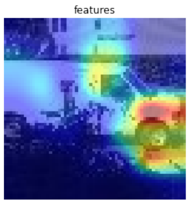
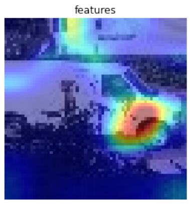
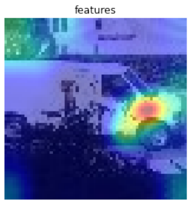
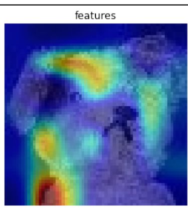
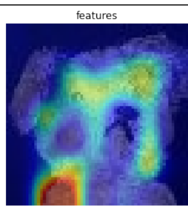
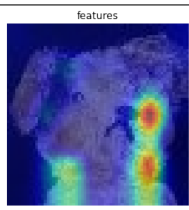
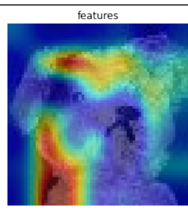
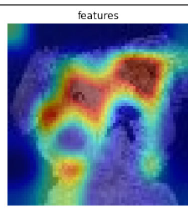
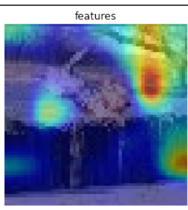
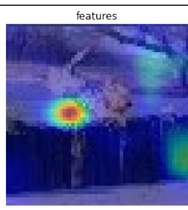
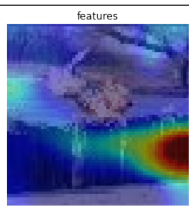
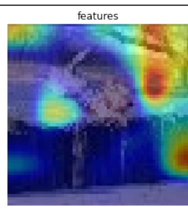
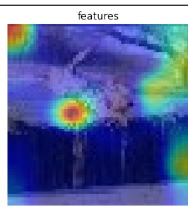
Image	GradCAM	GradCAM++	SmoothGradCAM++	XGradCAM	Layer CAM
					
					
					
					
					

Table 7: Table showing Activation map of true class for false prediction

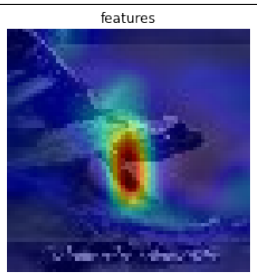
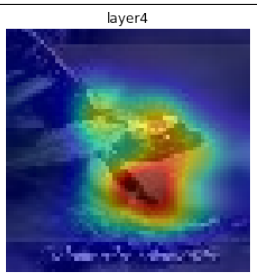
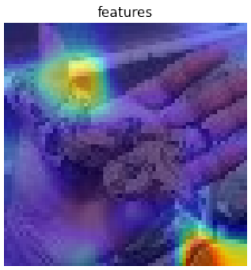
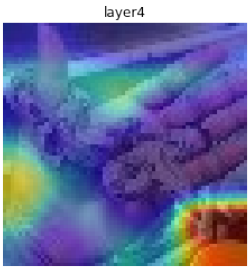
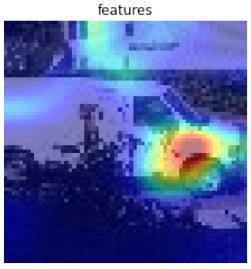
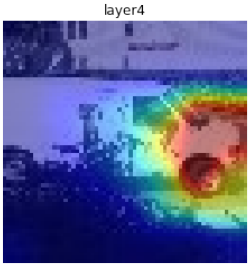
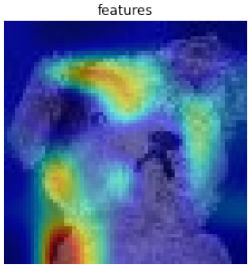
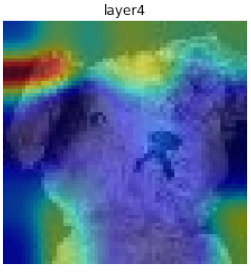
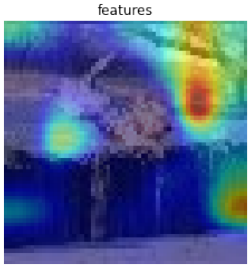
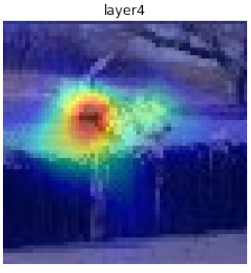
Predicted class	Predicted activation map	Actual class	Activation Map of actual class
Triceratops		European fire salamander, Salamandra salamandra	
Tabby, tabby cat		Boa constrictor, Constrictor constrictor	
Lake land terrier		Moving van	
Coral reef		Teddy, teddy bear	
Fountain		Yorkshire terrier	

Table 8: Evaluation metric

CAM	IOU			Confidence score			percentage increase in confidence		
	RESNET50	VGG-16	ViT	RESNET50	VGG-16	ViT	RESNET50	VGG-16	ViT
Gradcam	0.1329	0.1152		66.4950	65.6806		8.0655	8.2282	
Gradcampp	0.1435	0.1266		59.3698	64.0925		11.8388	8.5740	
Smoothgradcampp	0.1774			55.7187			14.4019		
Xgradcam	0.1329	0.1149		66.4950	66.0275		8.0655	7.7400	
Layercam	0.1477	0.1261		55.4633	64.1698		14.0663	8.5333	

Figure 2: IOU values of the last predictions (Vgg-16 architecture)

Figure 3: IOU values of the top predictions (resnet-50 architecture)

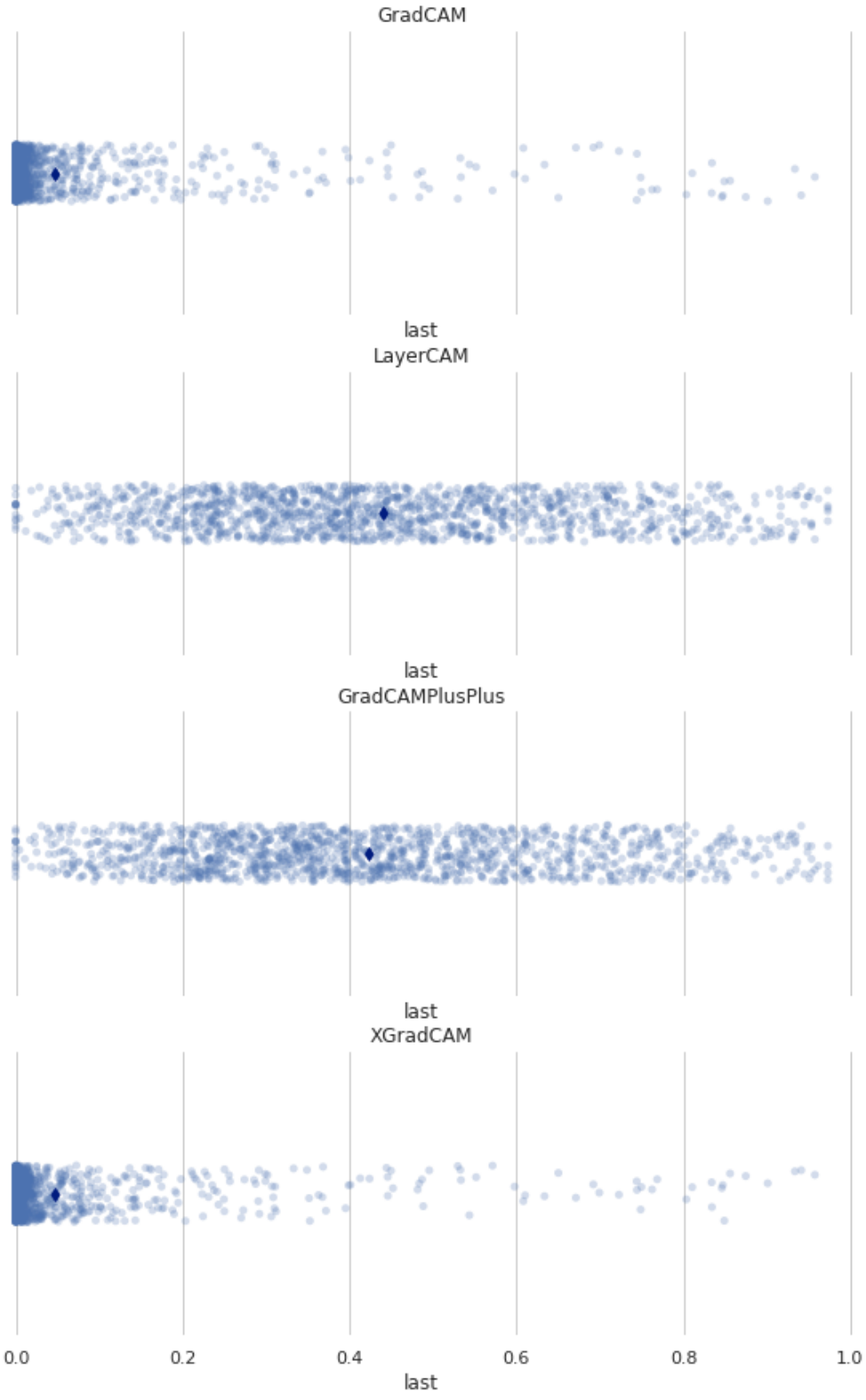


Figure 4: IOU values of the last predictions (resnet-50 architecture)

Figure 5: IOU values of the top predictions (resnet-50 architecture)

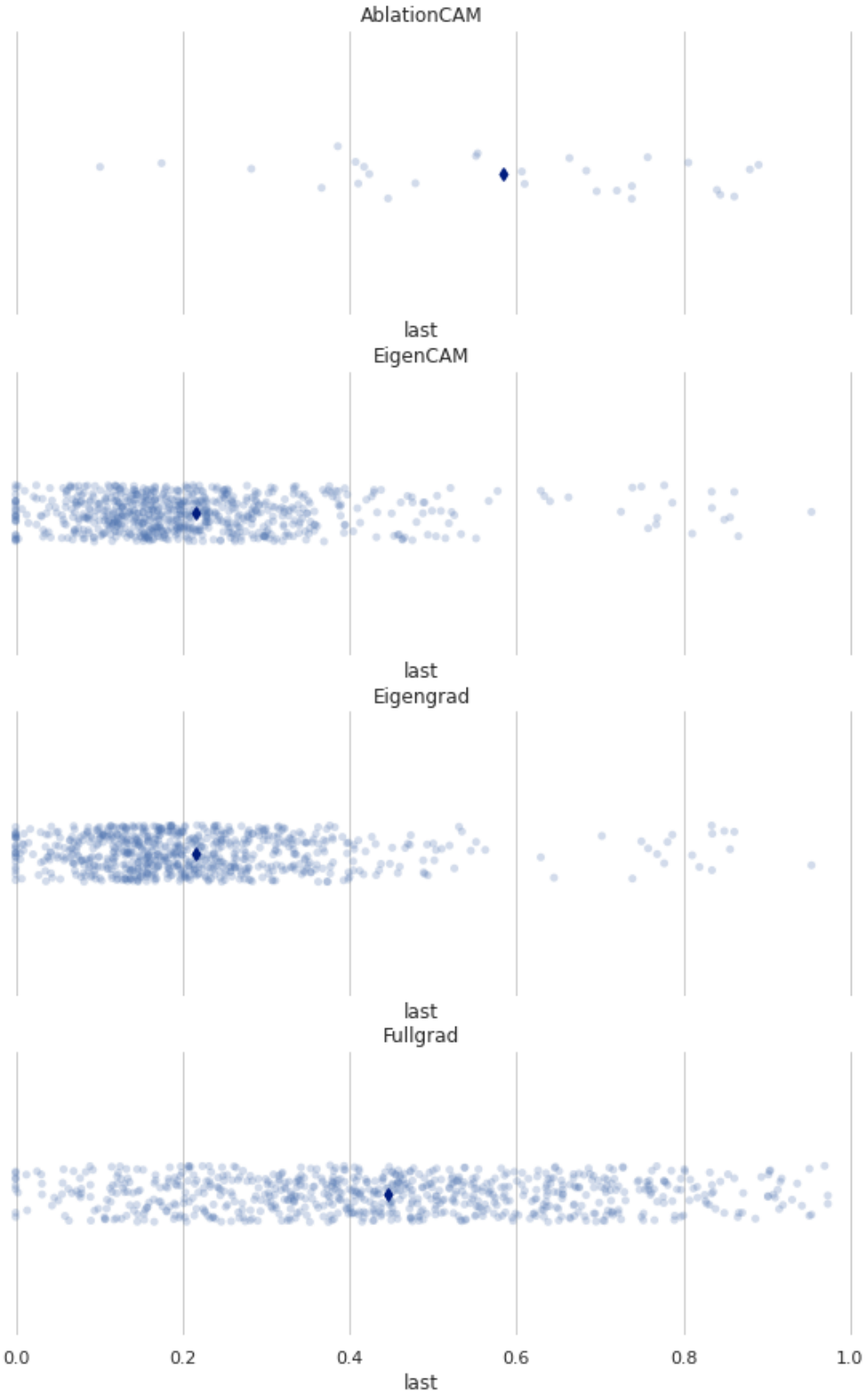


Figure 6: IOU values of the last predictions (resnet-50 architecture)

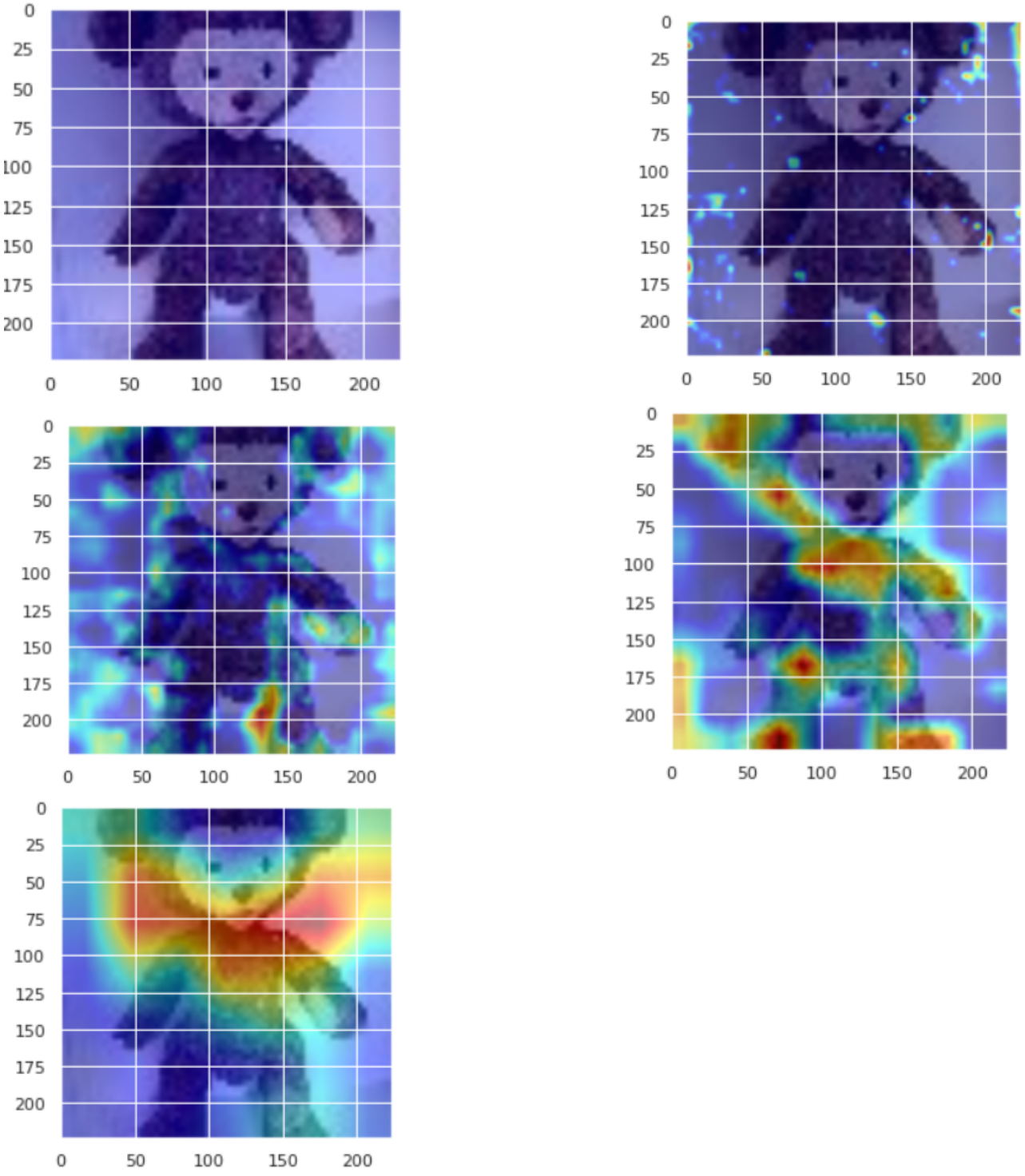


Figure 7: Renet50 Layer

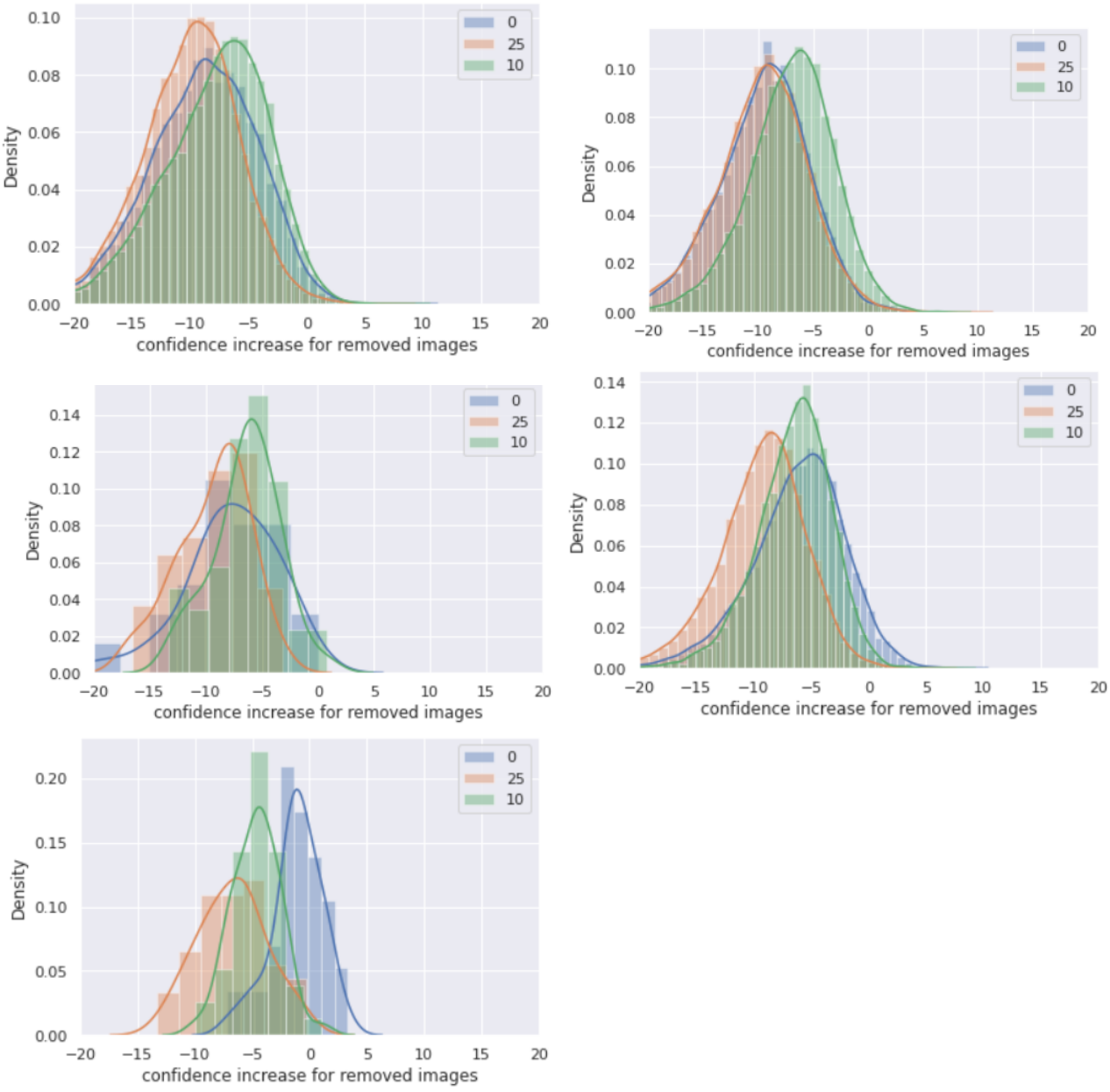


Figure 8: Resnet50 Layer

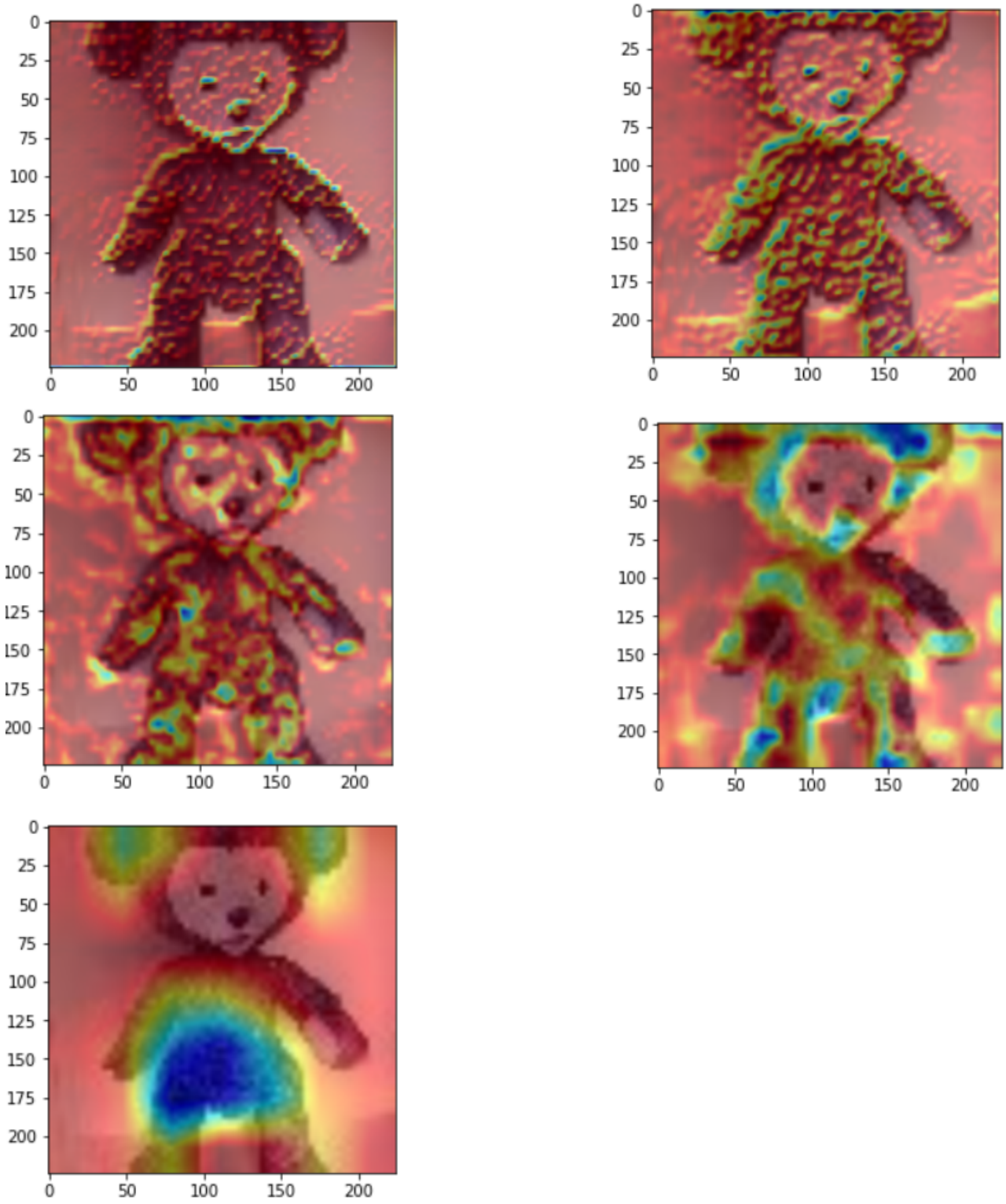


Figure 9: vgg16 Layer

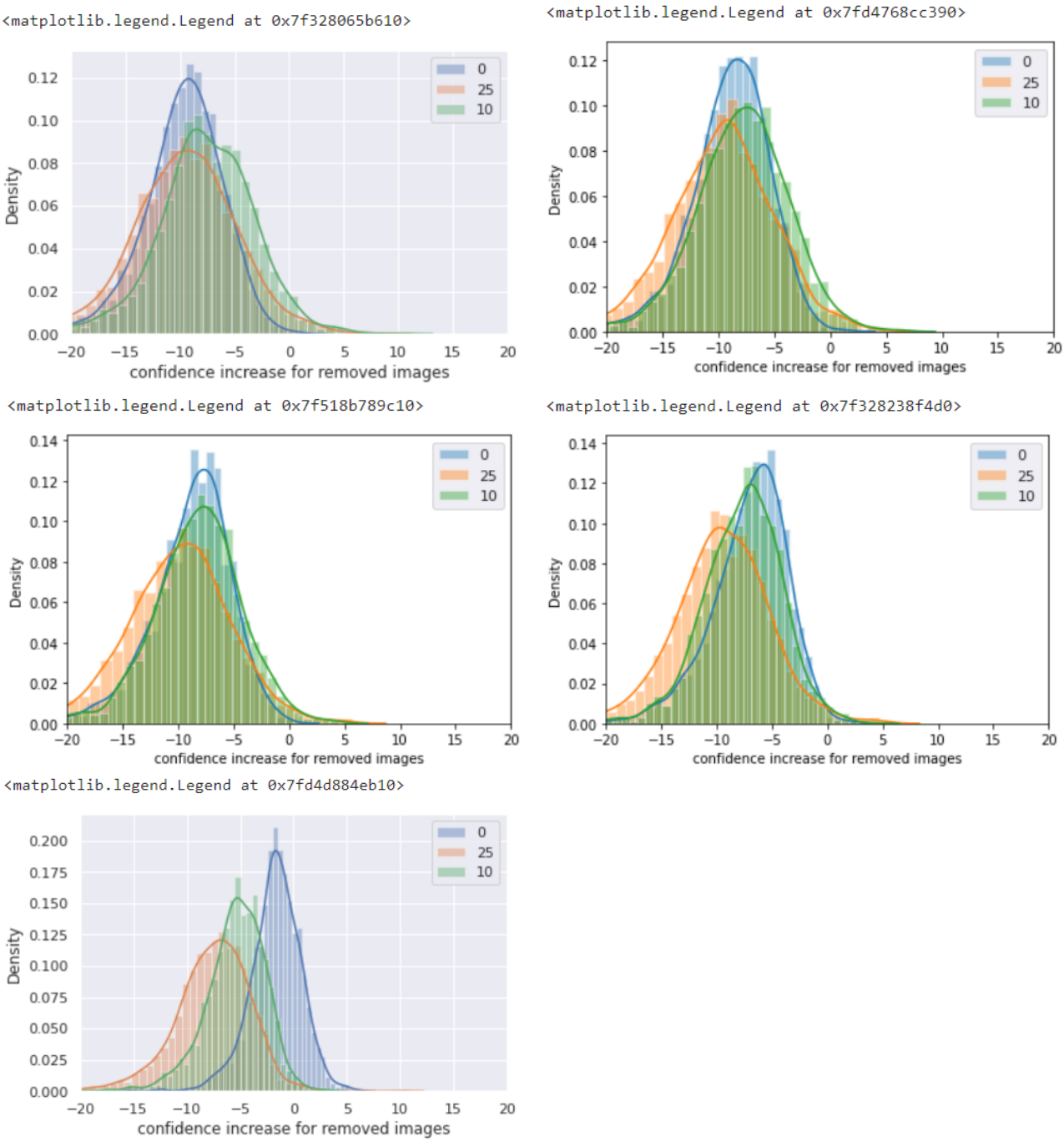
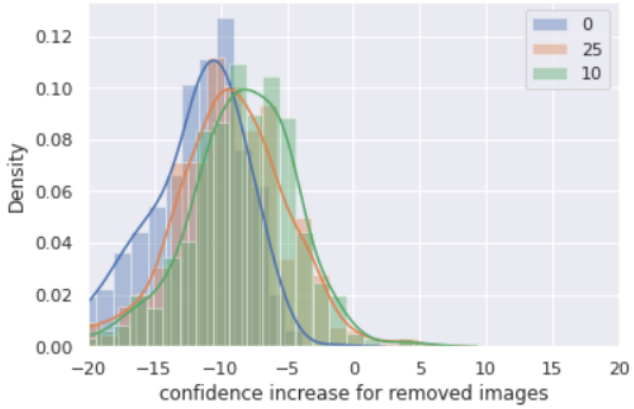


Figure 10: vgg16 Layer

<matplotlib.legend.Legend at 0x7f30bcadf710>



<matplotlib.legend.Legend at 0x7f30bcb2c250>

Figure 11: resnet50 and vgg16 with guided gradcam Layer