Grad-CAM++: Generalized Gradient-based Visual Explanations for Deep Convolutional Networks

Aditya Chattopadhay*
IIT Hyderabad
adityac@iith.ac.in

Anirban Sarkar*
IIT Hyderabad
cs16resch11006@iith.ac.in

Prantik Howlader*
Cisco Systems, Bangalore
prhowlad@cisco.com

Vineeth N Balasubramanian
IIT Hyderabad
vineethnb@iith.ac.in

Abstract

Over the last decade, Convolutional Neural Network (CNN) models have been highly successful in solving complex vision based problems. However, deep models are perceived as "black box" methods considering the lack of understanding of their internal functioning. There has been a significant recent interest to develop explainable deep learning models, and this paper is an effort in this direction. Building on a recently proposed method called Grad-CAM, we propose Grad-CAM++ to provide better visual explanations of CNN model predictions (when compared to Grad-CAM), in terms of better localization of objects as well as explaining occurrences of multiple objects of a class in a single image. We provide a mathematical explanation for the proposed method, Grad-CAM++, which uses a weighted combination of the positive partial derivatives of the last convolutional layer feature maps with respect to a specific class score as weights to generate a visual explanation for the class label under consideration. Our extensive experiments and evaluations, both subjective and objective, on standard datasets showed that Grad-CAM++ indeed provides better visual explanations for a given CNN architecture when compared to Grad-CAM.

1. Introduction

In recent years, the dramatic progress of machine learning in the form of deep neural networks has opened up new Artificial Intelligence (AI) capabilities in real-world applications. It is no new fact that deep learning models offer tremendous benefits with impressive results in tasks like object detection, speech recognition, machine translation to name a few. However, the connectionist approach of deep learning is fundamentally different from earlier AI systems where the predominant reasoning methods were logical and symbolic. These early systems could generate a trace of their inference steps, which then became the basis for explanation. On the other hand, the effectiveness of todays intelligent systems is limited by the inability to explain its decisions and actions to human users. This issue is especially important for risk-sensitive applications such as security, clinical decision support or autonomous navigation.

Recent advancements in AI promise to build autonomous systems that perceive, learn, decide and act on their own. However, machine learning today, especially deep learning models, attempt to approximate the function that transforms a given input into the expected output with no explicit knowledge of the function itself, or of the rationale involved in providing a particular output. This makes troubleshooting difficult in case of erroneous behaviour. Moreover, these algorithms are trained on a limited amount of data which most often is very different from real-world data. To this end, various techniques have been undertaken by researchers over the last few years to overcome the dilemma of blindly using the deep learning models. One technique is to rationalize/justify the decision of a model by training another deep model which comes up with explanations as to why the model behaved the way it did. Another emerging perspective to such explainable methods is to probe the black-box models by trying to change the input intellectually and analyzing the model's response to it.

While there have been some promising early efforts in this area, these are cursory and the field of explainable deep learning has a long way to go, considering the difficulty and variety in the problem scope. Zhou *et al*. [19] showed that various layers of the CNN (Convolutional Neural Network) behave as unsupervised object detectors by a new technique called CAM (Class Activation Mapping). By using a global average pooling [8] layer, and visualizing the

*All three authors contributed equally

978-1-5386-4886-5/18/$31.00 ©2018 IEEE
DOI 10.1109/WACV.2018.00097

weighted combination of the resulting feature maps at the penultimate (pre-softmax) layer, they were able to obtain heat maps that explain which parts of an input image were looked at by the CNN for assigning a label. However, this technique is constrained to only visualizing the last convolutional layer and also involves retraining a linear classifier for each class. Similar methods were examined with different pooling layers such as global max pooling [10] and log-sum-exp pooling [11].

Selvaraju *et al.* [13] came up with an efficient generalization of CAM, known as Grad-CAM, which fuses class-discriminative property of CAM with existing pixel-space gradient visualization techniques such as Guided Back-propagation [16] and Deconvolution [18], which highlight fine-grained details in the image. Therefore, Grad-CAM makes CNN-based models more transparent by visualizing input regions with high resolution details that are important for predictions.

The visualization generated by Grad-CAM explains the CNN based model prediction with fine-grained details of the predicted class, but lacks at localizing multiple occurrences of the same class. Also the localization of the heatmap generated by Grad-CAM is not very accurate with respect to covering the class region in an image. Therefore we propose Grad-CAM++ which covers these shortcomings to a great extent. The summarization of our contribution are listed as follows:

- Grad-CAM++ produces heatmaps at all locations of a class in case there are more than one, where the particular class can be positioned in a scattered or attached manner in the image. Thus, the heatmap generated better explains the model's behaviour when multiple instances of a single class are present in an image. This is shown with appropriate examples and comparisons with Grad-CAM results in section 3.

- Grad-CAM++ can localize the predicted class more accurately than Grad-CAM, which increases faithfulness to the model. We generated heatmaps for both the techniques (Grad-CAM and Grad-CAM++) and fused it with the actual image via point-wise multiplication. The behaviour of the confidence score (of the deep network) for that particular class, when presented with the original image and heatmap-fused image, can be analyzed to conclude which method generates a more class-specific heatmap. A lower or no drop in class score would indicate a higher localization of class-discriminative regions for a particular class. We provide results for this experiment in Section 4.1 which show a better localization capacity of Grad-CAM++ over Grad-CAM.

- We generated heatmaps with both Grad-CAM++ and Grad-CAM for a considerable number of images and merged them with the original image to generate regions important for the model's decision. These occluded images were given to human subjects along with their class labels and asked which of the two generated images contain more information about that class. Grad-CAM++ performed better than Grad-CAM. This experiment validates Grad-CAM++'s ability to correctly explain a deep network. We have provided the results for this particular experiment in Section 4.2.

This remainder of this paper is organized as follows. Section 2 discusses recent research efforts towards explainability in deep neural network models. A formal derivation of our method is provided in Section 3 with the relevant equations. In section 4, results from empiricial studies are presented which clearly show that Grad-CAM++ is a more robust generalization of Grad-CAM.

2. Related Work

Deep learning models have been highly successful in producing impressive results. However, the real bottleneck in accepting most of these techniques for real-life applications is the interpretability problem. Usually, these models are treated as black boxes without any knowledge of their internal workings. Recently, several methods have been developed to visualize Convolutional Networks and understand what is learnt by each neuron. The most straight-forward technique is to visualize the layer activations, where they generally show sparse and localized patterns. Visualizing the Convolution filters is also useful as as they reveal the nature of content extracted by the network in a given layer. Another method is to visualize the images which maximally activate a certain neuron in a trained network. This involves a large dataset of images to be feed forwarded through the network to understand what that neuron is looking for.

Zeiler & Fergus [18] proposed the deconvolution approach to better understand what the higher layers in a given network has learnt. "Deconvnet" makes data flow from a neuron activation in the higher layers, down to the image. In this process, parts of the image that strongly activate that neuron gets highlighted. Later, Springenberg *et al.* [16] extended this work to a new method called *guided backpropagation* which helped understand the impact of each neuron in a deep network w.r.t the input image. These visualization techniques were compared in [9]. Yosinski *et al.* [17] proposed a method to synthesize the input image, that causes a specific unit in a neural network to have a high activation, for visualizing the functionality of the unit. A more guided approach to synthesizing input images that maximally activate a neuron was proposed by Simonyan *et al.* [14]. In this work, they generated class-specific saliency maps, by per-

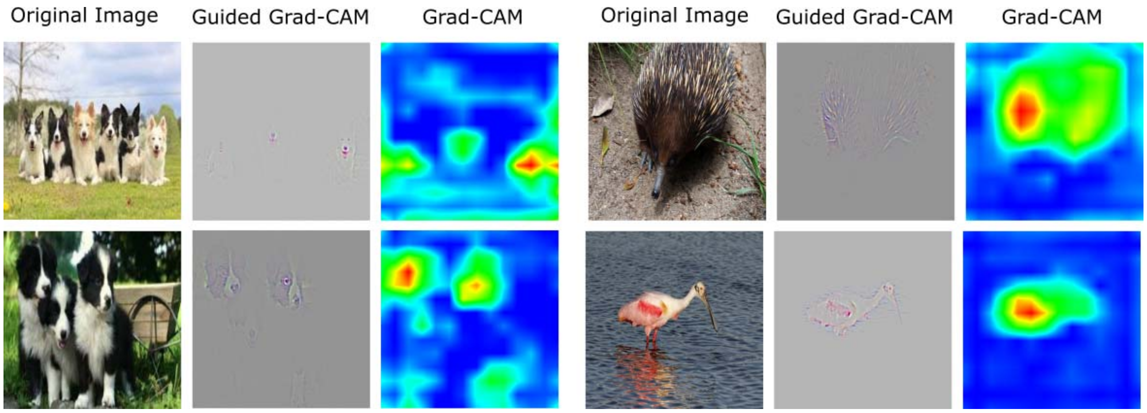

Figure 1. Weaknesses of Grad-CAM (a) multiple occurrences of the same class (Columns 1-3), and (b) localization capability of an object in an image (Columns 4-6). Note: All dogs are not visible for the first image under column 1. Full portion of the dogs are not visible for the second image under column 1. Both the classes are not visible in entirety under column 4 (the hedgehog's nose and the bird's legs are missing in the generated saliency maps).

forming a gradient ascent in pixel space to reach a maxima. This synthesized image serves as a class-specific visualization and enables us to delve deeper inside a CNN.

More recently, Ribeiro *et al*. [12] introduced a method called LIME (Local Interpretable Model-Agnostic Explanations) which makes a local approximation to the complex decision surface of any deep model with simpler interpretable classifiers like sparse linear models or shallow decision trees. For every test point, analyzing the weights of the sparse linear model gives some intuition to the non-expert as to the relevance of that feature in that particular prediction. In another approach, Al-Shedivat *et al*. [1] proposed contextual explanation networks (CENs), a class of models that jointly learns to predict and explain its decision. Unlike the existing posthoc model-explanation tools, CENs combine deep networks with context-specific probabilistic models and construct explanations in the form of locally-correct hypotheses. Konam [5] developed an algorithm to detect specific neurons which are responsible for decisions taken by a network and additionally locate patches of an input image which maximally activate those neurons. Lengerich *et al*. [7] proposed a different route towards explainability of CNNs. Instead of explaining the decision in terms of the input, they developed statistical metrics to evaluate the relation between the hidden representations in a network and its prediction. Another recent work [4], focusing on interpretability for self-driving cars, trained a visual attention model followed by a CNN model to obtain potentially salient image regions and applied causal filtering to find true input regions that actually influence the output.

In spite of the recent developments to make deep learning models explainable, we are far away from the desired goal. The focus of the ongoing research in this area is to develop algorithms that can generate interpretable explanations of the results of deep models used across domains, which can then be investigated upon failures to make them more robust. Another important reason is to build trust in these systems for their proper integration into our daily lives.

Our work in this paper is mainly inspired by two algorithms, namely CAM [19] and Grad-CAM [13]. Both CAM and Grad-CAM base their method on a fundamental assumption that the final score Y^c for a particular class c can be written as a linear combination of its global average pooled last convolutional layer feature maps A^k.

$$Y^c = \sum_k w_k^c \cdot \sum_i \sum_j A_{ij}^k \tag{1}$$

Each spatial location (i, j) in the class-specific saliency map L^c is then calculated as:

$$L_{ij}^c = \sum_k w_k^c . A_{ij}^k \tag{2}$$

L_{ij}^c directly correlates with the importance of a particular spatial location (i, j) for a particular class c and thus functions as a visual explanation of the class predicted by the network. CAM estimates these weights w_k^c by training a linear classifier for each class c using the activation maps of the last convolutional layer generated for a given image. CAM however has some limitations. It's explainability prowess is limited to CNNs with a Global Average Pooling (GAP) penultimate layer, and requires retraining of multiple linear classifiers (one for each class).

Grad-CAM was built to address these issues. In their paper[13], the authors show that with this assumption (Eq. 1), the weights w_k^c for a particular feature map A^k and class c is equal to:

$$w_k^c = Z . \frac{\partial Y^c}{\partial A_{ij}^k} \quad \forall \{i, j | i, j \in A^k\} \tag{3}$$

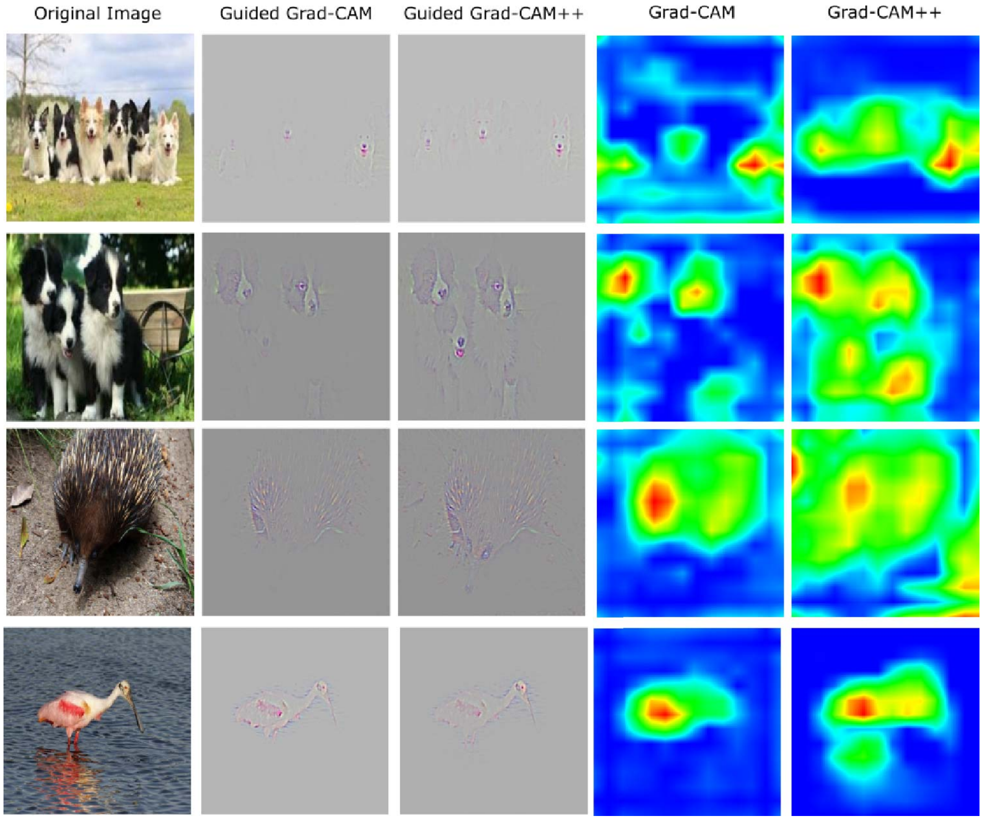

Figure 2. Success of Grad-CAM++ for (a) multiple occurrences of the same class (Rows 1-2), and (b) localization capability of an object in an image (Rows 3-4). Note: All dogs are better visible with more coverage, shown in the heatmap as well as Guided Grad-CAM++ for input images of rows 1 and 2. Full region of the class is visible for input images of rows 3 and 4 (namely nose of the hedgehog and the legs of the bird).

where Z is a constant (number of pixels in the activation map). This helps in generalizing CAM to any deep architecture with a CNN block, without any retraining or architectural modification, where the final Y^c is a differentiable function of the activation maps A^k. However, this formulation (Eq. 3) makes the weights w_k^c independent of the positions (i, j) of a particular activation map A^k. The authors work around this limitation by taking a global average pool of the partial derivatives ∂A_{ij}^k, i.e. -

$$w_k^c = \frac{1}{Z} \sum_i \sum_j \frac{\partial Y^c}{\partial A_{ij}^k} \tag{4}$$

To obtain fine-grained pixel-scale representations, the proposers of Grad-CAM upsample and fuse their class-specific saliency map L^c via point-wise multiplication with the visualizations generated by Guided Backpropagation. This visualization is referred to as Guided Grad-CAM. This approach however has some shortcomings as illustrated in Fig. 1. Grad-CAM fails to properly localize objects in an image if the image contains multiple occurrences of the same class. This is a serious issue as multiple occurrences of the same object in an image is a very common occurrence in the real world. Another consequence of an unweighted average of partial derivatives is that often, the localization doesn't correspond to the entire object, but bits and parts of it. This might hamper the user's trust in the model (which Grad-CAM eventually tries explaining). These issues impede Grad-CAM's premise of making a deep CNN more transparent. In this work, we suggest a simple modification to Grad-CAM which addresses the above mentioned issues and consequently serves as a better explanation algorithm for a given CNN architecture.

3. Grad-CAM++ Methodology

Building upon the works of Grad-CAM and CAM, we propose a more generalized method - Grad-CAM++. We reformulate (Eq.1) by explicitly coding the structure of the weights w_k^c as -

$$w_k^c = \sum_i \sum_j \alpha_{ij}^{kc}.relu(\frac{\partial Y^c}{\partial A_{ij}^k}) \tag{5}$$

The idea behind this formulation is that w_k^c captures the importance of a particular activation map A^k. Previous works in pixel-space visualization like Deconvolution [18] and Guided Backpropogation [16] have shown the importance of positive gradients in producing saliency maps for a particular convolutional layer. A positive gradient at location (i, j) for an activation map A^k implies that increasing intensity of pixel (i, j) would have a positive influence over the class score Y^c. Thus, a linear combinations of the positive partial derivatives w.r.t. each pixel in an activation map A^k would capture the importance of that map for class c. This structure ensures that the weights w_k^c are a weighted average of the gradients as opposed to a global average (Eq. 4). An empirical verification of the ”positive gradients” assumption is presented in the Supplementary Section 6.1.

We now formally derive a method for obtaining the gradient weights α_{ij}^{kc} for a particular class c and activation map k. Let Y^c be the score of a particular class c. From Eq.1 and Eq. 5

$$Y^c = \sum_k [\sum_i \sum_j \{\sum_a \sum_b \alpha_{ab}^{kc}.relu(\frac{\partial Y^c}{\partial A_{ab}^k})\} A_{ij}^k] \tag{6}$$

Here, (i, j) and (a, b) are the same iterators over the entire activation map A^k. Without loss of generality, we drop the $relu$ as it only functions as a threshold for allowing the gradients to flow back. Taking partial derivative w.r.t. A_{ij}^k on both sides:

$$\frac{\partial Y^c}{\partial A_{ij}^k} = \sum_a \sum_b \alpha_{ab}^{kc} \cdot \frac{\partial Y^c}{\partial A_{ab}^k} + \sum_a \sum_b A_{ab}^k \{\alpha_{ij}^{kc} \cdot \frac{\partial^2 Y^c}{(\partial A_{ij}^k)^2}\} \tag{7}$$

Taking partial derivative w.r.t A_{ij}^k on both sides:

$$\frac{\partial^2 Y^c}{(\partial A_{ij}^k)^2} = 2.\alpha_{ij}^{kc} \cdot \frac{\partial^2 Y^c}{(\partial A_{ij}^k)^2} + \sum_a \sum_b A_{ab}^k \{\alpha_{ij}^{kc} \cdot \frac{\partial^3 Y^c}{(\partial A_{ij}^k)^3}\} \tag{8}$$

Rearranging terms, we get:

$$\alpha_{ij}^{kc} = \frac{\frac{\partial^2 Y^c}{(\partial A_{ij}^k)^2}}{2\frac{\partial^2 Y^c}{(\partial A_{ij}^k)^2} + \sum_a \sum_b A_{ab}^k \{\frac{\partial^3 Y^c}{(\partial A_{ij}^k)^3}\}} \tag{9}$$

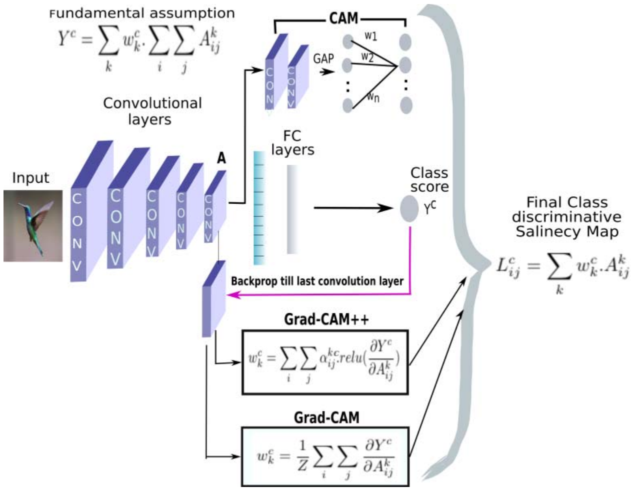
Fundamental assumption
$Y^c = \sum_k w_k^c . \sum_i \sum_j A_{ij}^k$
CAM
GAP
w1
w2
wn
Convolutional layers
FC layers
A
Input
CONV
CONV
CONV
CONV
CONV
CONV
CONV
Class score
Y^c
Final Class discriminative Salinecy Map
$L_{ij}^c = \sum_k w_k^c . A_{ij}^k$
Backprop till last convolution layer
Grad-CAM++
$w_k^c = \sum_i \sum_j \alpha_{ij}^{kc} relu(\frac{\partial Y^c}{\partial A_{ij}^k})$
Grad-CAM
$w_k^c = \frac{1}{Z} \sum_i \sum_j \frac{\partial Y^c}{\partial A_{ij}^k}$

Figure 3. An overview of all the three methods – CAM, Grad-CAM, Grad-CAM++ – with their respective computation expressions.

If $\forall i, j, \quad \alpha_{ij}^{kc} = Z^{-1}$, Grad-CAM++ reduces to the formulation for Grad-CAM (Eq. 4) Thus, Grad-CAM++, as its name suggests, can be viewed as a generalized version of Grad-CAM. In principle, the class score Y^c can be any prediction (object identification, classification, annotation, etc. to name a few); the only constraint being that Y^c must be a smooth function. For this reason, unlike Grad-CAM (which takes the penultimate layer representation as their class score Y^c), we directly take the softmax layer representation of class c, as the softmax function is infinitely differentiable. The time overhead for calculating higher-order derivatives would be of the same order as Grad-CAM, as only the diagonal terms are used (no cross higher-order derivatives). The class-discriminative saliency maps for a given image, L^c is then calculated as a linear combination of the forward activation maps, followed by a $relu$ layer. Each spatial element in the saliency map L^c is then computed as:

$$L_{ij}^c = relu(\sum_k w_k^c . A_{ij}^k) \tag{10}$$

To generate class-discriminative saliency maps along with the richness of pixels-space gradient visualization methods, we pointwise multiply the up-sampled (to image resolution) saliency map L^c with the pixel-space visualization generated by Guided Backpropagation. This technique is reminiscent to the technique adopted by Grad-CAM. The representations thus generated are called Guided Grad-CAM++.

In Fig. 2 we illustrate visually how taking a weighted combination of positive partial derivatives (Eq. 5) instead of a global average (Eq. 4) solves the problem of identifying multiple occurrences of the same class in an image and improper object localization. A bird’s eye view of all the three methods – CAM, Grad-CAM, Grad-CAM++ – is presented in Fig. 3.

4. Empirical Evaluation of Generated Explanations

A good explanation must be consistent with the model prediction (faithfulness) and display results in a human interpretable format (human trust). In addition to the visual results shown in Fig. 2, we also conducted other experiments to compare our method, Grad-CAM++ to Grad-CAM both objectively (faithfulness) and subjectively (human

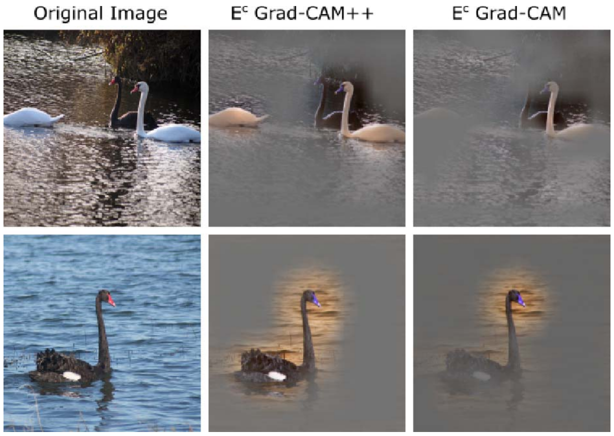

Figure 4. Example explanation maps for 2 images generated by Grad-CAM++ and Grad-CAM.

trust). We now present these experiments and results in this section. For all experiments, we used an off-the-shelf VGG-16 [15] model from the Caffe Model Zoo [3], as Grad-CAM performed almost all their experiments with VGG-16. Implementation of our method is available at https://github.com/adityac94/Grad_CAM_plus_plus.

4.1. Objective Evaluation of Performance based on Object Recognition

In this section, we evaluate the faithfulness of the explanations generated by Grad-CAM++ over PASCAL VOC 2007 val set (this is the set of images used by Grad-CAM for evaluating their visualizations) and compare the results to Grad-CAM. We choose this dataset in particular as it contains images with multiple classes. For every image, a corresponding explanation map E^c is generated by point-wise multiplication of the class-discriminative saliency maps (upsampled to image resolution) with the original image:

$$E^c = L^c \circ I \tag{11}$$

where $\circ$ refers to a Hadamard product (point-wise multiplication), I refers to the input image and L^c is the class-discriminative saliency maps as defined in Eq. 10. The class c used in the experiments for each image was the class predicted by the model. This was done for both the E^cs generated from Grad-CAM++ and Grad-CAM. Some example explanation maps are shown in Fig. 4. To put it simply, the explanation maps occlude parts of the image according to their importance in model decision-making as determined by Grad-CAM++/Grad-CAM. Our experiment setup consists of three different metrics: (i)Average drop% , (ii) % increase in confidence and (iii) Win%.

(i) Average drop%: A good explanation map for a class should highlight the regions that are most important. A deep CNN model looks for different patterns in the entire image space before making a decision. Occluding parts of an image would mostly lower the confidence of the model in it's decision (as opposed to the entire image). However, as the explanation maps keep the most important regions for a particular class in an image intact, this fall is expected to be less. This metric compares the average % drop in the model's confidence for a particular class in an image after occlusion. For example, let's say the model predicted an object Tiger in an image with confidence 0.8. When shown the explanation map for the same image, the model's confidence in the class Tiger fell to 0.4. Then the % drop in model confidence would be 50%. This value is averaged over the entire dataset.

Method	**Grad-CAM++**	Grad-CAM
- Average drop% *(Lower is better)*	**57.02**	61.22
- % incr. in confidence *(Higher is better)*	**14.98**	13.63
- Win% *(Higher is better)*	**58.01**	41.99

Table 1. Results for objective evaluation of the explanations generated by both Grad-CAM++ and Grad-CAM on the PASCAL VOC 2007 validation set (2510 images) ("incr" denotes increase). These results clearly indicate a superior performance of the generalized Grad-CAM++.

(ii) % increase in confidence: However, sometimes it is possible that the entirety of patterns the deep CNN looks for is in the most discriminative part highlighted by the explanation maps. In this scenario, there is an increase in the model's confidence for that particular class. This metric measures the number of times in the entire dataset, the model's confidence increased upon occluding unimportant regions. This value is expressed as a percentage.

(iii) Win%: While the first two metrics (Average drop% and % increase in confidence) evaluate the capability of an explanation map in highlighting the influential regions of an image, this metric explicitly compares the contrastive effectiveness of the explanation maps generated by Grad-CAM++ and Grad-CAM. It measures the number of times in a given set of images, the fall in the model's confidence for an explanation map generated by Grad-CAM++ is less than the respective explanation map generated by Grad-CAM. This value is expressed as a percentage. A lower fall would indicate that the salient features of an image preserved by Grad-CAM++ explanation map are more *model-appropriate* than the respective salient features captured by the Grad-CAM explanation map.

Method	**Grad-CAM++**	Grad-CAM
- Average drop% *(Lower is better)*	**43.31**	46.43
- % incr. in confidence *(Higher is better)*	**15.44**	13.67
- Win% *(Higher is better)*	**60.24**	39.76

Table 2. Results for objective evaluation of the explanations generated by both Grad-CAM++ and Grad-CAM on the ImageNet (ILSVRC2012) validation set ("incr" denotes increase). We took random subsets of 2510 images to maintain consistency with the PASCAL VOC 2007 dataset and averaged out the results. The results futher substantiate our claim that Grad-CAM++ improves upon the performance of Grad-CAM.

Method	**Grad-CAM++**	Grad-CAM
- Average drop% *(Lower is better)*	**60.29**	69.45
- % incr. in confidence *(Higher is better)*	**17.65**	2.94
- Win% *(Higher is better)*	**70.59**	29.41

Table 3. Results for objective evaluations generated by both Grad-CAM++ and Grad-CAM on a randomly chosen subset of the PASCAL VOC 2007 val dataset such that each image contains multiple occurances of a single class ("incr" denotes increase). Grad-CAM++ achieves a much higher improvement than Grad-CAM as this dataset is specifically targeted at Grad-CAM's point of failure.

The results of this experiment on the VOC val dataset is depicted in Table 1. Grad-CAM++ performs better than Grad-CAM in all three metrics. A higher % increase in confidence and a lower average drop% is consistent with our hypothesis that re-formulating the weights as Eq.5 helps in better localizing the most discriminative regions of an image (for a particular class c) as compared to global average pooling of the gradients (Eq. 4).

Having established an improved performance of Grad-CAM++ over Grad-CAM for the VOC 2007 val set, we redo the experiment for randomly sampled 2510 images from ImageNet (ILSVRC2012) val set, where most images have just one object(class). The rationale behind this setup was to evaluate whether Grad-CAM++ works better than Grad-CAM only for multiple object images or the improvement is universal. We randomly chose three subsets of size 2510 to maintain consistency with the previous experiment and to suppress the influence of random chance on our results (the entire validation set of ILSVRC2012 has 50,000 images, while the VOC 2007 dataset has 2510 val images). The results for this experiment is depicted in Table 2. Even in this scenario Grad-CAM++ outperforms Grad-CAM in all three metrics.

The first two experiments show that Grad-CAM++ does a better job at generating explanations that are more faithful to the deep network as compared to Grad-CAM. However, both these results only strengthen one of our arguments - better discriminative localization of object classes. Both the datasets - ImageNet and Pascal VOC, have relatively few images with multiple objects belonging to the same class. In Section 3, we posit that Grad-CAM++ addresses the problem of identifying multiple occurrences of the same class in an image and support that claim visually (Fig.2). To carry out a more objective evaluation in this setting, we randomly select a subset of 35 images from the PASCAL VOC 2007 validation dataset, such that each image contains multiple instances of a given object, and carry out the same experiment. The results are shown in Table 3. In this setting, Grad-CAM++'s improvement is magnified. Comparing the results with those in Table 1, we observe that Grad-CAM++ is a more robust explanation algorithm in settings with multiple occurrences of an object in a scene. The change in 'Average drop%' and '% incr. in confidence' changes by about $2.5\% - 3\%$. Comparatively the corresponding changes in Grad-CAM is huge, of about $8\% - 10\%$. Also, the third metric 'Win%', which measures the contrastive performance of the two algorithms rises from 60% to 70%. All three experiments in this section show the superior utility of Grad-CAM++ as an explanation algorithm over Grad-CAM. More empirical results showing the effectiveness of Grad-CAM++ for other CNN architectures, viz, AlexNet [6] and Resnet-50 [2] are discussed in Supplementary Section 6.2 and 6.3.

4.2. Evaluating Human Trust

The explainability prowess of any algorithm for a given deep model depends on two important factors - *faithfulness* and human *interpretability*. In the previous subsection, we explored "faithfulness", here we evaluate the "interpretability" of our explanations. Again, the explanations generated by Grad-CAM were treated as baseline for comparison. We generated explanation maps for all images in the ImageNet validation set for 5 classes, leading to a total of 250 images. These maps along with their corresponding original image were shown to 13 human subjects (who have no knowledge of deep learning whatsoever) and asked which explanation algorithm invokes more trust in the underlying model. The assumption in this work is that a pre-trained VGG-16 network [15] has learnt the hidden representations of the object categories adequately. Thus, the explanation algorithm that gets more votes from the subjects can be considered as doing a better job of invoking human trust in the underlying VGG-16 model. To further substantiate our claim, we chose 5 classes which have the highest F1-score for the validation dataset (above 0.94). As each class just has 50 images in the validation set, F1-score (which is the harmonic mean of pre-

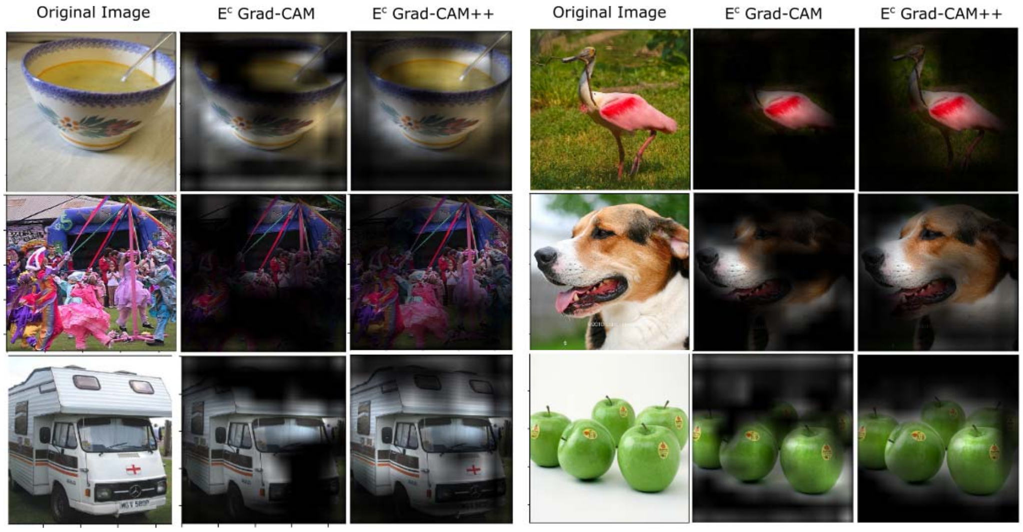

Figure 5. Example explanation maps for 6 images generated by Grad-CAM and Grad-CAM++.

cision and recall) is a better suited metric than classification error.

For each image, two explanation maps were generated, one from Grad-CAM and one from Grad-CAM++. Some visual examples of these explanation maps are presented in Fig. 5. The subjects were told the class of the image and asked to select the map they felt best described the object in the image (without telling them which one is GradCAM or GradCAM++). The subjects also had the option to select "same" if they felt both the generated explanation maps were similar. The responses for each image was normalized, such that the total score possible for each image is 1.0. To elaborate on this point, we obtained 13 responses for each image. For example, among the 13 responses, if 5 chose the explanation map generated by Grad-CAM++, 4 chose the explanation map generated by Grad-CAM and 4 chose the option "same", the respective scores from Grad-CAM++ and Grad-CAM would be 0.38 and 0.31 (with the remaining being "same"). These normalized scores were then added. So, the total achievable score is 250. Grad-CAM++ achieved a score of **109.69** as compared to **56.08** of Grad-CAM. The remaining 84.23 was characterized as "same" by the human subjects. This empirical study provides strong evidence for our hypothesis that weighing the partial derivatives (Eq. 5) helps in better image localization (as most images in the ImageNet validation dataset have just one instance of an object). A better localization invokes a greater trust in the deep model that makes the decision. As Grad-CAM++ is a generalization of Grad-CAM, it performs similar to Grad-CAM in about 33.69% cases. This experiment shows that the explanations generated by Grad-CAM++ prove better human interpretability when compared to Grad-CAM.

5. Conclusions

In this work, we propose a generalized class-discriminative approach for visual explanations of CNN based architectures, Grad-CAM++. We provide a formal derivation for our method and show that it is a simple, yet effective generalization of Grad-CAM. Our method addresses the shortcomings of Grad-CAM - multiple occurrences of same class in an image and poor object localizations. This was demonstrated via visual examples. We validated the effectiveness of our method both objectively (faithfulness to the model it's trying to explain) and subjectively (invoke human trust). We performed our experiments with a standard VGG-16 model and datasets (ImageNet and Pascal VOC). Extensive human studies revealed the effectiveness of Grad-CAM++ in explaining the model over Grad-CAM. Objective evaluation of the explanations generated, also show an improvement in favour of Grad-CAM++ over Grad-CAM. In theory, as a generalized version of Grad-CAM, Grad-CAM++ can be applied to any scenario where Grad-CAM is used. Future work involves evaluating our method for other tasks such as image captioning, visual question answering and video applications, which have CNNs as an integral module.

References

[1] M. Al-Shedivat, A. Dubey, and E. P. Xing. Contextual explanation networks. *arXiv preprint arXiv:1705.10301*, 2017. 3

[2] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 7

[3] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the 22nd ACM international conference on Multimedia*, pages 675–678. ACM, 2014. 6

[4] J. Kim and J. Canny. Interpretable learning for self-driving cars by visualizing causal attention. *arXiv preprint arXiv:1703.10631*, 2017. 3

[5] S. Konam. Vision-based navigation and deep-learning explanation for autonomy. In *Masters thesis, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA.*, 2017. 3

[6] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012. 7

[7] B. J. Lengerich, S. Konam, E. P. Xing, S. Rosenthal, and M. Veloso. Visual explanations for convolutional neural networks via input resampling. *arXiv preprint arXiv:1707.09641*, 2017. 3

[8] M. Lin, Q. Chen, and S. Yan. Network in network. *arXiv preprint arXiv:1312.4400*, 2013. 1

[9] A. Mahendran and A. Vedaldi. Salient deconvolutional networks. In *European Conference on Computer Vision*, pages 120–135. Springer, 2016. 2

[10] M. Oquab, L. Bottou, I. Laptev, and J. Sivic. Is object localization for free?-weakly-supervised learning with convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 685–694, 2015. 2

[11] P. O. Pinheiro and R. Collobert. From image-level to pixel-level labeling with convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1713–1721, 2015. 2

[12] M. T. Ribeiro, S. Singh, and C. Guestrin. Why should i trust you?: Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1135–1144. ACM, 2016. 3

[13] R. R. Selvaraju, A. Das, R. Vedantam, M. Cogswell, D. Parikh, and D. Batra. Grad-cam: Why did you say that? visual explanations from deep networks via gradient-based localization. *arXiv preprint arXiv:1610.02391*, 2016. 2, 3

[14] K. Simonyan, A. Vedaldi, and A. Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2013. 2

[15] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 6, 7

[16] J. T. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller. Striving for simplicity: The all convolutional net. *arXiv preprint arXiv:1412.6806*, 2014. 2, 5

[17] J. Yosinski, J. Clune, A. M. Nguyen, T. J. Fuchs, and H. Lipson. Understanding neural networks through deep visualization. *CoRR*, abs/1506.06579, 2015. 2

[18] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pages 818–833. Springer, 2014. 2, 5

[19] 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*, pages 2921–2929, 2016. 1, 3