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

Team Name SAS: Serious About Science
Team Members
Vatanpreet Singh (2021702014)
Sudarshan S Harithas (2021701008)
Mohd Hozaifa Khan (2021701026)
Syed Abdul Hadi (2021900006)

1 Problem Statement

The Convolutional Neural Networks (CNN) have proved to be an effective tool in solving multitude of problems. But most of the CNNs appear to be a 'black box' and it is difficult to interpret the results and figure out their decision making process. Such has not been case with previous rule or heuristic based classification approaches. Not knowing the decision making track puts great risk of employing CNNs in daily use. *Class Activation Mapping* (CAM) is an approach that aims to to improve interpretability of CNNs by identifying the regions of an image relevant for the model for class predictions.

In this project we attempt to provide visual explanations of CNN decision making and we implement three novel approaches from scratch to solve the CAM problem. The state-of-the-art solution is provided by GRAD CAM++ [1] that not only demonstrates improvement in terms of localization accuracy,(i.e. the generated heat map covers the class region of the image), but also is able to better explain the predictions in case of multiple occurrences of a same class object in an image. For the purpose of bench marking, we further implement CAM [2] and GRAD CAM [3]. Moreover, we expand the horizon of our work by testing the efficacy of these algorithms on multiple CNN architectures.

2 Overview of Our current Progress

Our progress in comparison to the expected deliverables can be summarized as follows:

1. Algorithm Implementation: Three algorithms namely GRAD CAM, GRAD CAM++ and CAM have been completely implemented from scratch.

2. Testing with multiple CNN backbone Architectures: The original paper [1] tests the performance of the system using VGG-16 as the backbone architectures. We further test the performance of the system on multiple architectures such as ResNet-50, ResNet-101 and EfficientNet80. This experiment tests the generalizability of the above mentioned algorithms.

3. Qualitative and quantitative results: We have successfully replicated the qualitative and quantitative results that were demonstrated in the paper [1] [3].

3 Methodology

3.1 GRAD CAM ++

GRAD CAM++ is a generalization over two algorithms namely GRAD CAM [3] and CAM [2]. Both these methods express the final score Y^c for a given class **c** as a linear weighted combination of the *Global Average Pooling (GAP)* output of the last CNN layer feature map A^k. i.e.

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

In GRAD CAM++ the weights w_k^c is given by

$$w_k^c = \sum_i \sum_j \alpha_{ij}^{kc} Relu(\frac{\partial Y^c}{\partial A_k^{ij}}) \tag{2}$$

Where α_{ij}^{kc} are the weights of the spatial location (i, j) of the A^k feature map corresponding to class c and $Relu()$ is the *Rectified Linear Activation Function* and $\frac{\partial Y^c}{\partial A_k^{ij}}$ are the pixel wise gradients. The intuition behind the selection of positive gradients are that we require weights that are positively correlated with the output. If the weights are negative it implies that the given feature makes the image less likely to belong to a particular class, where as a 0 weight indicates that the observation of the feature has no impact on the final result.

The objective of GRAD CAM ++ is to determine an appropriate set of α_{ij}^{kc} such that the we get close to uniform positive weights w_k^c for all instances of the object in the image.

By substituting Eq. 2 in 1 we get

$$Y^c = \sum_k [\sum_i \sum_j (\sum_a \sum_b \alpha_{ab}^{kc} Relu(\frac{\partial Y^c}{\partial A_{ij}^k}))) A_{ij}^k] \tag{3}$$

By taking the first and second derivatives of Eq. 3 we get Eq. 4 and 5. The *Relu* function is dropped as it linear (with slope 1) for positive values and zero for negative, essentially it functions as a threshold operation and facilitates the flow of gradients.

$$\frac{\partial Y^c}{\partial A^k_{ij}} = \sum_a \sum_b A^k_{ab} \alpha^{kc}_{ij} \frac{\partial^2 Y^c}{(\partial A^k_{ij})^2} + (\sum_a \sum_b \alpha^{kc}_{ab} (\frac{\partial Y^c}{\partial A^k_{ab}})) \tag{4}$$

$$\frac{\partial^2 Y^c}{(\partial A^k_{ij})^2} = 2\alpha^{kc}_{ij} \frac{\partial^2 Y^c}{(\partial A^k_{ij})^2} + \sum_a \sum_b A^k_{ab} (\alpha^{kc}_{ij} \frac{\partial^3 Y^c}{(\partial A^k_{ij})^3}) \tag{5}$$

Rearranging Eq. 5 to obtain the weights α^{kc}_{ij} we get

$$\alpha^{kc}_{ij} = \frac{\frac{\partial^2 Y^c}{(\partial A^k_{ij})^2}}{2\frac{\partial^2 Y^c}{(\partial A^k_{ij})^2} + \sum_a \sum_b A^k_{ab} (\frac{\partial^3 Y^c}{(\partial A^k_{ij})^3})} \tag{6}$$

By substituting Eq. 6 in Eq. 2 we obtain a closed form solution to the α weights of GRAD CAM++.

The class-discriminative saliency maps for a given image, L_c is given by Eq. 7. The resulting map is up-sampled to image resolution for visualization

$$L^c_{ij} = Relu(\sum_k w^c_k A^k_{ij}) \tag{7}$$

3.1.1 Improving Computational Efficiency

The calculation of the higher order derivatives can be an expensive operation in general (considering calculation of the non-diagonal derivatives), it can be simplified by passing the penultimate layer scores S^c though an exponential function and the last layer having only linear or $Relu()$ activation. Therefore the score Y^c is given by

$$Y^c = \exp(S^c)$$

The first derivative is given by Eq. 8

$$\frac{\partial Y^c}{\partial A^k_{ij}} = \exp(S^c) \frac{\partial S^c}{\partial A^k_{ij}} \tag{8}$$

The quantity $\frac{\partial S^c}{\partial A^k_{ij}}$ has been calculated by a popular machine learning framework called *Keras*, which implements automatic differentiation and is computationally efficient. Using the exponential function allows to express the higher order derivatives in terms of the first order derivative, thus improving the computational efficiency.

$$\frac{\partial^2 Y^c}{(\partial A^k_{ij})^2} = \exp(S^c)[(\frac{\partial S^c}{\partial A^k_{ij}})^2 + \frac{\partial^2 S^c}{(\partial A^k_{ij})^2}] \tag{9}$$

Assuming the $Relu()$ activation function $f(x) = max(x, 0)$ the higher order derivatives can be determined as follows :

$$\frac{\partial f}{\partial x} = 1 \; x \geq 0$$
$$= 0 \tag{10}$$

The second derivative is

$$\frac{\partial^2 f}{\partial x^2} = 0 \tag{11}$$

Substituting Eq. 11 into 9 we get

$$\frac{\partial^2 Y^c}{(\partial A_{ij}^k)^2} = \exp(S^c)[(\frac{\partial S^c}{\partial A_{ij}^k})^2] \tag{12}$$

Similarly the third derivative is given by 13

$$\frac{\partial^3 Y^c}{(\partial A_{ij}^k)^3} = \exp(S^c)[(\frac{\partial S^c}{\partial A_{ij}^k})^3] \tag{13}$$

Substituting results from Eq. 12 and 13 into Eq. 6, we obtain the weights α_{ij}^{kc} as :

$$\alpha_{ij}^{kc} = \frac{(\frac{\partial S^c}{\partial A_{ij}^k})^2}{2(\frac{\partial S^c}{\partial A_{ij}^k})^2 + \sum_a \sum_b A_{ab}^k (\frac{\partial S^c}{\partial A_{ij}^k})^3} \tag{14}$$

From Eq. 14 we observe that the α_{ij}^{kc} can be obtained from a close form solution which is a function of the first order derivatives, these derivatives can be easily obtained from a machine learning framework. The resulting α_{ij}^{kc} would be used to obtain the weights given by Eq. 2 and further used to determine the saliency map give by Eq. 7.

3.2 GRAD CAM

CAM is one of the earlier approaches to this problem, despite its appreciable class discriminative efficiency (i.e. it can localize objects without positional supervision) it had some drawbacks such as the architecture had to be modified and it placed restrictive constrains over the changed. Furthermore, the model may trade-off accuracy for interpretability and it required training of a final linear classifier which would increase computational load.

GRAD CAM was proposed to overcome the issues faced in CAM, which required no re-training or architectural modification. It aims to provide a closed form solution to the weights w_k^c (where w_k^c is the weight that connects the k^{th} feature map with the c^{th} class) in Eq. 15.

$$Y^c = \sum_k w_k^c \frac{1}{Z} \sum_i \sum_j A_{ij}^k \tag{15}$$

Let the *Global Average Pooling(GAP)* output be denoted by F^k. This implies Eq. 15 can be rewritten as

$$Y^c = \sum_k w_k^c F^k \tag{16}$$

Taking the derivative with respect to the feature map F^k we get

$$\frac{\partial Y^c}{\partial F^k} = \frac{\frac{\partial Y^c}{\partial A_{ij}^k}}{\frac{\partial F^k}{\partial A_{ij}^k}} \tag{17}$$

From Eq. 15 we can observe that $\frac{\partial F^k}{\partial A_{ij}^k} = \frac{1}{Z}$, substitute this result in Eq. 17 we get

$$\frac{\partial Y^c}{\partial F^k} = Z \frac{\partial Y^c}{\partial A_{ij}^k} \tag{18}$$

From Eq. 16 we observe that $\frac{\partial Y^c}{\partial F^k} = w_k^c$, substituting this in Eq. 18 we get

$$w_k^c = Z \frac{\partial Y^c}{\partial A_{ij}^k} \tag{19}$$

Sum the result of Eq. 19 with respect to all the pixels in layer k i.e.

$$\sum_i \sum_j w_k^c = \sum_i \sum_j Z \frac{\partial Y^c}{\partial A_{ij}^k} \tag{20}$$

The value $\sum_i \sum_j = Z$ (total number of pixels) substitute this in Eq. 20 to get the final result for weights w_k^c given by Eq. 21.

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

$$w_k^c = \sum_i \sum_j \frac{\partial Y^c}{\partial A_{ij}^k} \tag{21}$$

The resulting weights w_k^c is used to generate the localization maps using Eq. 22.

$$L_c = Relu(\sum_k w_k^c A^k) \tag{22}$$

3.3 CAM

CAM is one of the first to explore CNN interpretability and attempted to visualize the reasoning behind decision making of a CNN. Being the first few, it lacked generalizability and could only be applied to CNN architectures having Global Average Pooling layer in between.

In CAM, the authors propose that a CNN with a Global Average Pooling (GAP) layer shows localization capabilities despite not being explicitly trained to do so. In a CNN with GAP, the final classification 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. The output of GAP layer can be given as:

$$GAP(A^k) = \sum_i \sum_j A_{ij} \tag{23}$$

$$Y^c = \sum_k w_k^c \sum_i \sum_j A_{ij} \tag{24}$$

The class-specific saliency map, denoted by L^c, can be calculated for each spatial location (i, j) as:

$$L_{ij}^c = \sum_k w_k^c \cdot A_{ij}^k \tag{25}$$

L_{ij}^c directly correlates with the importance of a particular spatial location for a particular class c and thus functions as a visual explanation of the class predicted by the network. The weights w_k^c is obtained by training a linear classifier for each class c using an activation map of the last convolutional layer generated generated for a given Image. The issue with CAM is that it required re-training of the classifier if GAP is not the penultimate layer.

4 Experiments and Results

We conducted experiments to study the correlation between the visual explanations and the resulting model prediction, the experiments result in both a visual and quantitative evaluation and benchmarking of the three presented algorithms.

4.1 Qualitative Evaluation

This section presents the observations of experiments that were conducted to evaluate the resulting visual explanations of the algorithms. Firstly, we evaluate the faithfulness of the generated explanations of each of the algorithms, for this a class conditional explanation map E^c is generated by point-wise multiplication of the up-sampled (upto image resolution) class-conditional saliency maps with the original image. i.e.

$$E^c = L_c \circ I \tag{26}$$

Where L_c is the localization map that was obtained from Eq 22 (in case of GRAD CAM) or Eq 7 (in case of GRAD CAM++) and I is the input image and $\circ$ is the *Hadamard product*. The generated explanation maps and heatmaps (localization maps) are used for the visual evaluation of the algorithms.

Two experiments were conducted to benchmark the performance of the algorithms. Firstly, we check if the generalizability of the given method. The

second experiment provides qualitative evaluation of the resulting explanations particularly in the presence of multiple objects in an image.

4.1.1 Benchmark between CNN architectures

For a given test image, if the object classifier predicts the correct class of the object, it is expected that the underlying CNN has observed and based its predictions according to the region of the image that belongs to the object. Therefore, we can expect that the resulting explanation map to highlight the same region of the input image independent of the underlying CNN model. This experiment aims to determine the ability of the explanation algorithms to provide consistent result across multiple CNN models.

Fig. 1h, demonstrates the superiority of the predictions of GRAD CAM++ in comparison to GRAD CAM. Every row of Fig. 1h uses a different neural network (VGG16 , ResNet50, ResNet101, EfficientNet from row 1 to 4). We can observe that the predictions of GRAD CAM++ not only covers the entire space of the object in the image but also is more consistent across CNN models in prediction.

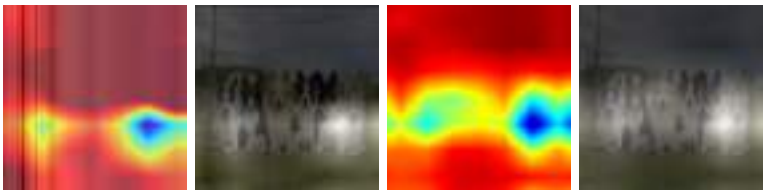
4.1.2 Benchmark between Algorithms

This experiment aims to evaluate the performance of the across various images, particularly with multiple objects of interest placed in a single image. Since the gradients (equally weighted) are dependent on the patial footprint of the object in the scene GRAD CAM fails to localize multiple occurrences of the same object in the image. On the other-hand using a weighted average as in case of GRAD CAM++ allows for better salient features and improves the accuracy of the resulting explanation. Fig. 2d depicts the resulting explanation maps, it can be observed that GRAD CAM++ is able to accurately localize the position of multiple objects in the image. The failure of GRAD CAM in presence of multiple objects can also be seen in Fig. 1h.

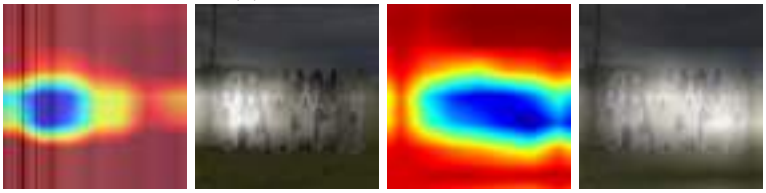
4.2 Quantitative Evaluations

To study the performance of these algorithms [1] proposes three metrics namely (i) Average drop % , (ii) % increase in confidence and (iii) win %. Currently we have evaluated the performance of GRAD CAM and GRAD CAM++ on the first two metrics and their descriptions are given below and the quantitative results is presented in Table 1. The evaluations we performed using a custom test dataset which primarily contained three classes cat, dog and people.

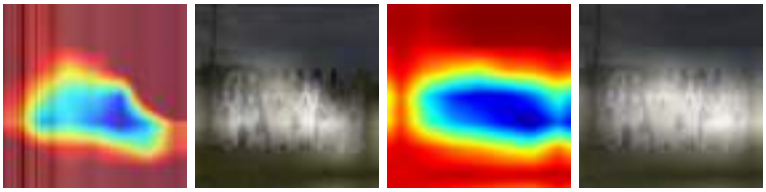
1. **Average drop %** It is expected that the removal of parts of the image will decrease the confidence of prediction as compared to when the full image is available. This property has been used to study the efficacy of the explanation maps, the full image is expected to have a higher confidence in predictions as compared to the confidence when only the explanation maps are provided as input. This change in confidence is noted, although



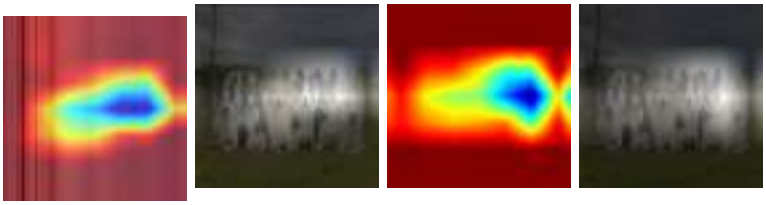
(a) Explanations from VGG16



(b) Explanations from ResNet50

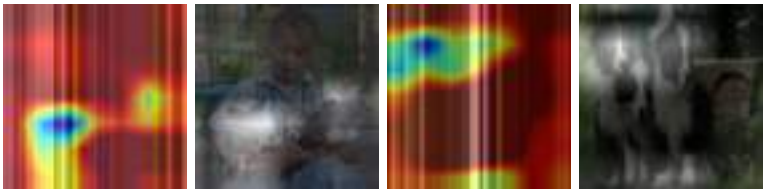


(c) Explanations from ResNet101

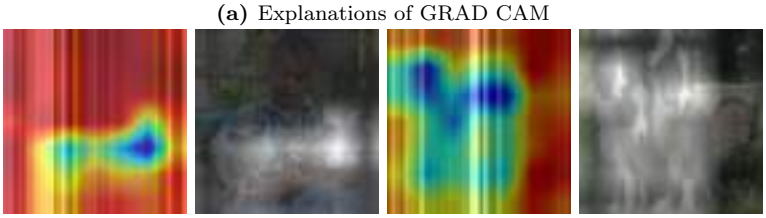


(d) GRAD CAM Heat Map L_c **(e)** GRAD CAM Explanation map E^c **(f)** GRAD CAM++ Heat Map L_c **(g)** GRAD CAM++ Explanation map E^c

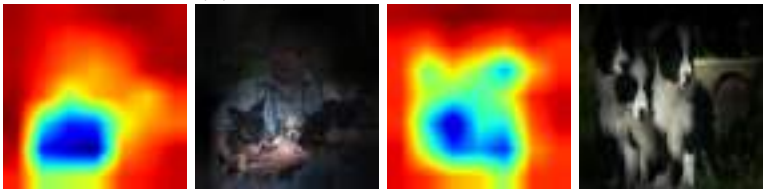
(h) The columns (d)(e) demonstrate the results of GRAD CAM and the columns (f)(g) display the results of GRAD CAM++. Every row corresponds to a different CNN, it can be observed that GRAD CAM++ is more consistent in explaining predictions in comparison to GRAD CAM.



(a) Explanations of GRAD CAM



(b) Explanations of GRAD CAM++



(c) Explanations of CAM

(d) Depiction of the localization and explanation maps from GRAD CAM, GRAD CAM ++ and CAM. The maps are generated for two input images a first image consists of multiple objects of interest (a lady with a few puppies in the same image, but here we choose to observe explanations only from the dog class) and the the second image is a pair of dogs very close to each other. It can be seen that GRAD CAM++ covers the entire area of the object (including multiple objects) and its explanations are better that GRAD CAM. Compared others, it can be observed that CAM is only focused on one of the two instances.

Table 1: Quantitative Results

Metric	GRAD CAM++	GRAD CAM
Average Drop % (Lower the better)	**35.606**	55.60
% Increase in Confidence (Higher the better)	**21.15**	11.53

both GRAD CAM and GRAD CAM++ are expected to have a drop in confidence [1] hypothesizes that GRAD CAM++ maintains a higher confidence in the predicted label. This implies that explanation maps from GRAD CAM++ includes more relevant features for decision making.

The metric is computed as the average drop in model confidence and is given by

$$Averagedrop\% = \frac{100}{N} \times \sum_{i=1}^{N} \frac{\max(0, Y_i^c - O^c)}{Y_i^c}$$

. Where the prediction confidence with full image is given by Y^c and O^c is the confidence with only explanation maps. a max function is used to handle cases when $O^c > Y^c$.

2. **% Increase in Confidence** in this metric we evaluate the number of times the confidence of prediction has increased when only the explanation map was given as the input. This metric is complementary to the previous one.

$$\%Increase = \sum_{i=1}^{N} \frac{R(Y_i^c, O_i^c)}{N}$$

where $R(.)$ is an indicator function that returns 1 if $Y_i^c < O_i^c$ else returns 0.

5 Conclusion and Future Work

Three algorithms have been implemented from scratch namely CAM, GRAD CAM, GRAD CAM++ and the performance of the methods have been qualitatively and quantitatively analyzed and benchmarked. All the main objectives of the paper have been implemented and we are currently placed in an appropriate position to go slightly beyond the expected deliverables. The future scope of our work is as follows:

1. Benchmarking and evaluation: Further test these algorithms on multiple datasets and perform quantitative evaluation using other metrics such as win % and IOU.

2. Testing with custom CNN architectures: Build and train simple CNN models and interpret its performance using multiple CAM methods.

3. Sequential Networks: Attempt to extend GRAD CAM++ to sequential networks for tasks such as image captioning. Specifically, we intend to extend it to ConvLSTMs.

References

[1] 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.

[2] 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.

[3] 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.