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 benchmarking, 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. Furthermore, we harness the ability of the resulting explanation maps to perform object localization.

2 Overview of Our Work

The completed work 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] .

4. Object localization using the explanation maps of GRAD CAM++ (image segmentation).

5. An appropriate attempt to expand the scope of work and test the working of GRAD CAM on sequential models.

Having accomplished all the objectives we go slightly beyond the expected deliverable and test the working of GRAD CAM++ on custom trained and built CNN architectures.

3 An Intuitive Introduction

Development of methods that would improve interpretability of the internal workings of a CNN model has received attention recently. GRAD CAM++ is a generalization of its predecessors GRAD-CAM and CAM, the key insight is that the average provides an equal weight to all the pixels in the image independent of the spatial footprint of the object. If the image consists of multiple instances of the same object (hence same class) and one of these objects occupies a smaller spatial footprint the corresponding gradient diminishes and object cannot be localized.

GRAD CAM++ overcomes the issue by using a weighted combination of the positive partial derivatives of the last CNN layer to generate visual explanations of the classification model. If a given object occupies a smaller footprint the corresponding weight would be proportionately higher hence providing a stable gradient.

Fig 1 illustrates the working of GRAD CAM and GRAD CAM++ through a hypothetical example, where the assigned task is binary classification. Consider the saliency map L^c and a binary object classification task that returns 0 if object is absent and 1 if present. A^k_{ij} represents the (i, j) spatial location at the k^{th} feature map (A). If a visual pattern is detected $A^k_{ij} = 1$ else 0, in Fig. 1 the dark regions correspond to 1. The partial derivative $\frac{\partial y^c}{\partial A^k_{ij}} = 1$ if $A^k_{ij} = 1$ else 0.

The weights w^c for GRAD CAM is given by the Eq. 21, substituting these values into the equation we get $w^1_c = \frac{15}{80}$, $w^2_c = \frac{4}{80}$ and $w^2_c = \frac{2}{80}$. It can be clearly

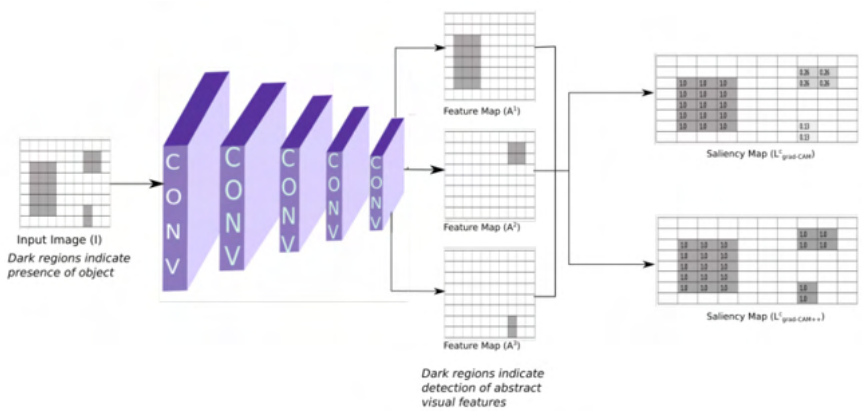


Figure 1: A Hypothetical example to illustrate the intuition behind GRAD CAM++, the CNN is performing binary object classification. It can be observed that in GRAD CAM (which uses an equally weighted average) provides a poor saliency map where the gradients of objects with a less spatial footprint has diminished. Whereas GRAD CAM ++ which uses a weighted mean, provides for improved salient features i.e. all relevant regions of the image are equally highlighted.

observed that the gradients are proportional to the spatial footprint of the object (in case of equal weight averaging). For an image with multiple instances of the same object each with different orientations and size, the feature maps may be activated with different footprints and the gradients from the smaller footprint would diminish.

GRAD CAM++ solves the problem by deriving a weighted average of pixel-wise gradients. The resulting weights map from GRAD CAM++ is given by 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*. In the above example by selecting appropriate values of α_{ij}^{kc} (as explained in section 4.1) we get uniform weights all the instances in the image. The novelty of GRAD CAM++ is the pixel-wise weighting of gradients, and the availability of closed form solutions for these weights

4 Methodology

4.1 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 $\mathbf{c}$ 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.

$$\frac{\partial Y^c}{\partial A_{ij}^k} = \sum_a \sum_b A_{ab}^k \alpha_{ij}^{kc} \frac{\partial^2 Y^c}{(\partial A_{ij}^k)^2} + (\sum_a \sum_b \alpha_{ab}^{kc} (\frac{\partial Y^c}{\partial A_{ab}^k})) \tag{4}$$

$$\frac{\partial^2 Y^c}{(\partial A_{ij}^k)^2} = 2\alpha_{ij}^{kc} \frac{\partial^2 Y^c}{(\partial A_{ij}^k)^2} + \sum_a \sum_b A_{ab}^k (\alpha_{ij}^{kc} \frac{\partial^3 Y^c}{(\partial A_{ij}^k)^3}) \tag{5}$$

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

$$\alpha_{ij}^{kc} = \frac{\frac{\partial^2 Y^c}{(\partial A_{ij}^k)^2}}{2\frac{\partial^2 Y^c}{(\partial A_{ij}^k)^2} + \sum_a \sum_b A_{ab}^k (\frac{\partial^3 Y^c}{(\partial A_{ij}^k)^3})} \tag{6}$$

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

$$L_{ij}^c = Relu(\sum_k w_k^c A_{ij}^k) \tag{7}$$

4.1.1 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

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

The quantity $\frac{\partial S^c}{\partial A_{ij}^k}$ 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.

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

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

$$\begin{aligned} \frac{\partial f}{\partial x} &= 1 \ x \geq 0 \\ &= 0 \end{aligned} \tag{10}$$

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.

4.2 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.

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

4.3 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

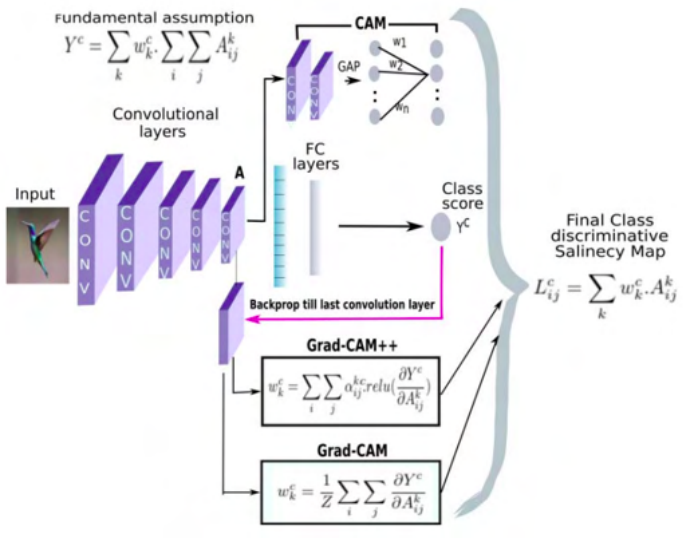


Figure 2: An overview of GRAD CAM++, GRAD CAM and CAM with their computational expressions

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.

The working methodology of the three implemented algorithms namely GRAD CAM++, GRAD CAM and CAM can be best summarized as shown in Fig. 6.

5 Experiments and Results

We conducted experiments to study and perform quantitative and qualitative evaluation of the visual explanations that are obtained from the three presented algorithms.

5.1 Qualitative Evaluations

This section presents the observations of experiments that were conducted to evaluate the resulting visual explanations of the algorithms qualitatively. 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.

5.1.1 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. 3h, demonstrates the superiority of the predictions of GRAD CAM++ in comparison to GRAD CAM. Every row of Fig. 3h 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.

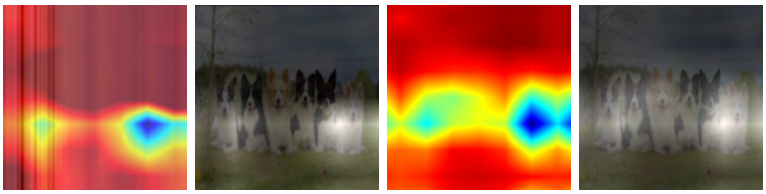
5.1.2 Benchmark between Algorithms

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. 5d 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. 3h.

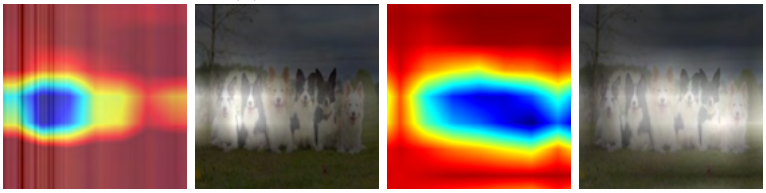
5.2 Quantitative Evaluations

To study the performance of these algorithms [1] proposes three metrics namely (i) Average drop % , (ii) % increase in confidence and (iii) win % , 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 fruits, animals such as cats, dog, elephant and frog.

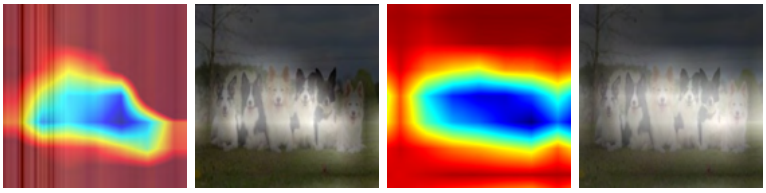
1. **Average drop %** It is expected that the removal of parts of the image will decrease the confidence of prediction as compared to when the full image is available. This property has been used to study the efficacy of the explanation maps, the full image is expected to have a higher confidence in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



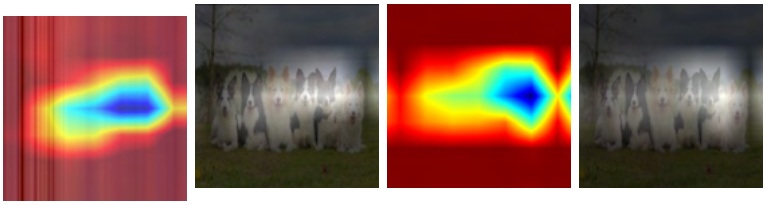
(a) Explanations from VGG16



(b) Explanations from ResNet50

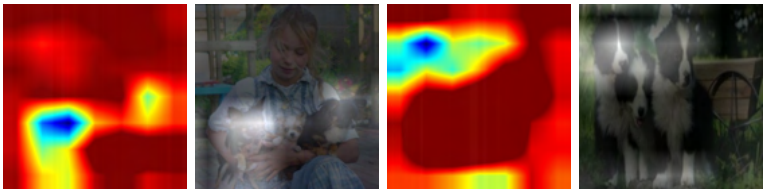


(c) Explanations from ResNet101



(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.

maps are provided as input. This change in confidence is noted, although 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.

3. **% Win** For a given set of images the win % measures the number of times the fall in the model's confidence for an explanation map generated by Grad-CAM++ is less than that by generated using GRAD CAM.

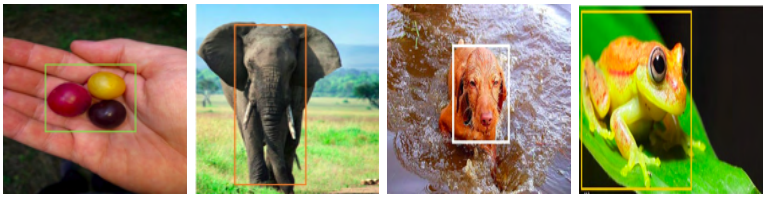
5.3 Harnessing Explanations for Object Localization

In this section we evaluate the effectiveness of GRAD CAM++ and GRAD CAM in performing class conditional object localization. We use *EfficientNet* as the backbone CNN architecture and evaluate the mode performance on our test dataset described above.

For a given image I and a prediction class c a corresponding explanation map $E_I^c(\delta)$ is generated using Eq. 7 and Eq. 22. Here the localization maps L_c are min-max normalized and threshold by a value δ (all intensities value above δ is clamped to 1 during implementation we set $\delta = 0$). The corresponding *Intersection Over Union*(IOU) metric is given by Eq. 27.

$$Loc_I^c(\delta) = \frac{Area(InternalPixels)}{Area(boundingbox) + Area(ExternalPixels)} \tag{27}$$

For this experiment we have created bounding-box annotations for 10 images, the average IOU(Higher the better) for GRAD CAM++ was **0.3835** and that of GRAD CAM was 0.311.



(a) Actual Image and boxes represent the ground truth annotations

(b) Object Localization using GRAD CAM++

(c) Object Localization using GRAD CAM

(d) Object Localization capabilities of GRAD CAM++ and GRAD CAM

Table 1: Quantitative Results

Metric	GRAD CAM++	GRAD CAM
Average Drop % (Lower the better)	**35.606**	53.316
% Increase in Confidence (Higher the better)	**21.16**	11.538
% Win (Higher the better)	**67.30**	32.692

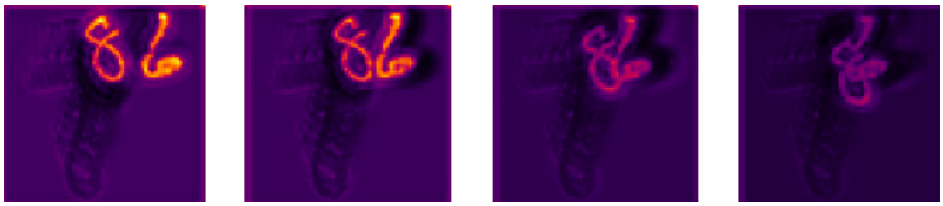


Figure 6: GRAD CAM explanation maps on ConvLSTM observe the tracks that are in the neighbourhood of each number this represents the previous trajectory.

5.4 Application of GRAD CAM to Sequential Models

Sequential models are machine learning models which take sequences as input and can also output data sequences. Sequences of data can be in any form ranging from video and audio streams to Time Series weather data and point clouds.

Neural Networks model time series data using a feedback loop within the module to retain contextual information. However, it suffers from exploding/vanishing gradient problem. LSTMs, a variant of RNNs, have proven to learn solve this issue and are able to learn long term dependencies. We use a variant of LSTMs, called ConvLSTM, to extend the visual capability of Grad-CAM. Extending CAM like techniques requires that intermediate feature representations remain structurally similar to the input. ConvLSTM is just like LSTM, but the internal matrix multiplications are exchanged with convolution operations. This makes sure that hidden state of CovLSTM remain 2 dimensional and could be used for visualization.

In this application we use ConvLSTM to predict the next frame state i.e. the process of predicting the next frame given the current frame and the past history of motions. We use Moving MNIST data to train a next frame prediction model. LSTMs store memory of its previous states in hidden state and cell state. We use GRAD CAM to visualize the the hidden state at each time step. As seen in Fig. 6 motion footprints of the digits can be observed and a general idea of trajectory can be inferred.

The visible footprints of motion in a single hidden state vector shows that LSTMs propagate considerable previous information in its hidden state and learn changes across time step. Based on these trajectories and hidden state visualization, we can also evaluate the temporal capacity of any ConvLSTM model and compare its performance with other models.

5.5 Ablation Studies

It is known that well constructed and appropriate CNN models lead to accurate predictions. In this section we aim to study and explain the relation between model complexity (complexity of a model includes the number of convolution blocks, activation and dropout layers) and the resulting predictions through the explanation maps of GRAD CAM++. We demonstrate that with the increase in appropriate complexity, the model learns to focus on regions that are relevant

for prediction.

The experiment consists of building and training of custom CNNs and increasing the complexity (or growing the CNN model by adding convolution blocks, activation and dropout layers) at every trial by adding further CNN blocks. In the below detailed trails the CNN blocks are followed by a *Dense* layer and a *SoftMax* classification layer, the model was trained on the *CIFAR100 Dataset* for 50 epochs using the *ADAM* optimizer.

The results of the experiment is given in Fig. 7, **Trial A** consists of only a single convolution block (i.e. a CNN layer, followed by *Relu* activation and a *MaxPooling* layer), we can observe from the resulting heat and explanation map that the model has not learnt to classify the object i.e the model cannot focus on the appropriate region in the image.

A second *Convolution Block* with a higher number of filters is added to the first in **Trial B** we observe a slight improvement in the performance, the model has started to identify the region of interest that are relevant for classification this indicates that increasing the model complexity is the right direction to improve performance.

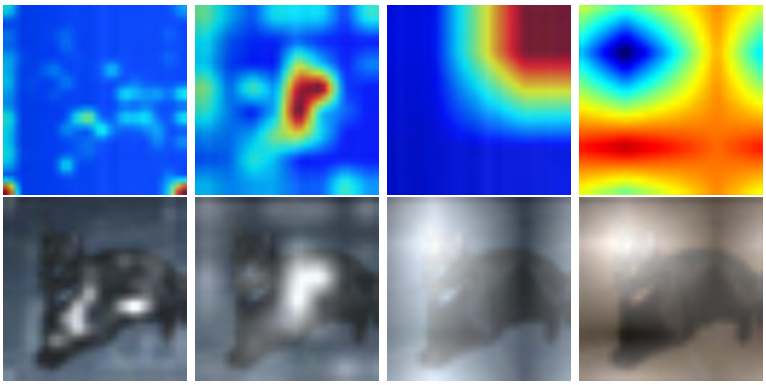
In **Trial C** we further deepen the network by adding a third Convolution blocks, it can observe that explanation maps are providing more clarity into that the regions that the model has observed. In order to prevent over-fitting a *Dropout* layer is included in this trial.

In **Trial D** a deep CNN model has been built with five Convolution blocks and *Dropout* layers that are placed in appropriate positions such that it leads to maximum accuracy. It can be observed from the resulting heat and explanation maps that the model has now learnt to focus on the necessary regions of the image and is able to localize objects correctly.

Through this set of experiment it can demonstrated that by appropriate increase in model complexity a CNN will be able to identify and localize the objects of interest and hence perform accurate classification.

6 Conclusion

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. We survey the performance of these algorithms on multiple backbone CNN architectures such as ResNet50, VGG16, ResNet101, EfficientNet. Furthermore, we harness the ability of the resulting explanations to perform object localization. Moreover, we enhance the scope of the project by implementing GRAD CAM to sequential models such as ConvLSTM. Lastly, ablation studies have been conducted where the resulting localization maps are used as a tool to demonstrate the correlation between the improved performance of the model and the increased complexity.



(a) Trial **A** **(b)** Trial **B** **(c)** Trial **C** **(d)** Trial **D**

Figure 7: Heat-map and explanation map results from GRAD CAM++ of multiple models that trained with varied levels of model complexity

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.