

# SALIENCY-DRIVEN CLASS IMPRESSIONS FOR FEATURE VISUALIZATION OF DEEP NEURAL NETWORKS

Sravanti Addepalli <sup>\*‡</sup>      Dipesh Tamboli <sup>\*†</sup>      R. Venkatesh Babu <sup>‡</sup>      Biplab Banerjee <sup>†</sup>

<sup>‡</sup> Department of Computational and Data Sciences, Indian Institute of Science, Bengaluru, India  
<sup>†</sup> Indian Institute of Technology Bombay, Mumbai, India

## ABSTRACT

In this paper, we propose a data-free method of extracting *Impressions* of each class from the classifier’s memory. The Deep Learning regime empowers classifiers to extract distinct patterns (or features) of a given class from training data, which is the basis on which they generalize to unseen data. Before deploying these models on critical applications, it is very useful to visualize the features considered to be important for classification. Existing visualization methods develop high confidence images consisting of both background and foreground features. This makes it hard to judge what the important features of a given class are. In this work, we propose a saliency-driven approach to visualize discriminative features that are considered most important for a given task. Another drawback of existing methods is that, confidence of the generated visualizations is increased by creating multiple instances of the given class. We restrict the algorithm to develop a single object per image, which helps further in extracting features of high confidence, and also results in better visualizations. We further demonstrate the generation of *negative images* as naturally fused images of two or more classes. Our code is available at: <https://github.com/val-iisc/Saliency-driven-Class-Impressions>.

**Index Terms**— Visualization, Class Impressions, Saliency Maps.

## 1. INTRODUCTION

Deep Learning has resulted in unprecedented progress in many of the computer vision applications such as classification [1], segmentation [2] and object recognition [3]. In terms of performance metrics such as classification accuracy, deep learning has outperformed the best of classical methods by a large margin [1]. However, one of the key issues with Deep Neural Networks is the explicability of the model. In traditional image processing algorithms, features are usually handcrafted using methods such as SIFT [4] and HoG [5], which are very intuitive to understand, visualize and explain. However, in a deep learning framework, features are learned

by the model, and these are generated using complex non-linear mappings from pixel space [6]. This makes it hard to understand features that are important for a given task. Explainable models are very important in applications such as autonomous navigation, medical diagnosis and surveillance systems. Explainability is necessary for legal compliance, identifying biases in the developed model and to improve accountability of failure cases. In order to address these issues, there have been several works [7, 8, 9] on the visualization of various aspects of the Deep Convolutional Networks. This includes visualization of filters, activation maps, image-specific saliency maps and visualization of the important features of a trained model [8, 10]. Visualizing the important features of a trained model is useful to understand the inherent patterns or features that the model uses to make an inference. Such visualizations can help in validating the model and ensuring that the model does not overfit to some features that may be very specific to the domain in hand. This method can be used to test the generalizability of the model to unseen data.

In this work, we propose a novel method of generating useful visualizations from a network, which we term as Saliency Driven Class Impressions (SCI). The proposed method uses saliency maps [8] to generate highly discriminative features of a given class, while suppressing weak features. Prior data-free visualization works [8, 11] focus on generating features that maximize activations of a given class. This results in the generation of weak features also, such as background, which may belong to multiple classes. Another issue with the visualizations developed by prior art is that they maximize activations by generating multiple instances of the same object. This prevents the network from developing more robust features. We use the combination of a novel saliency-driven update rule, region-growing approach and Total Variation loss [12] to develop single instances of objects. This results in the generation of more confident features, and also more aesthetically pleasing and natural-looking visualizations of objects in a given class.

Our contribution in this work has been summarized here:

- We propose a data-free method of generating visualizations containing highly discriminative features learned by the network.

<sup>\*</sup>Equal contribution author

- We demonstrate the generation of single-object images from the classifier’s memory
- We generate natural looking *negative images*, which could be used to train more robust classifiers

The organization of this paper is briefly described here. In the following section, we present related literature in the field of Visualization of Deep Networks. Section-3 explains our proposed method in detail. We conclude the paper with our analysis in Section-4.

## 2. RELATED LITERATURE

### 2.1. Review of Visualization methods

The first layer filters of Convolutional Neural Networks can be directly visualized in pixel level. However, we need more sophisticated methods to understand the dynamics of the remaining layers. One popular line of work in this area involves generation of images by maximizing activations at any given intermediate layer, or at the final output of the network [7, 13, 8, 11].

### 2.2. Image-Specific Class Saliency Maps

A method for class saliency visualization was proposed by Simonyan et al. [8]. The authors define Saliency maps as the *spatial support* of a class in a given image. Saliency maps highlight regions that contribute to the classification of a given image to its respective class. This is very useful to visually understand the features that are important for classifying the object. Saliency maps are computed by taking derivative of the class scores with respect to the input image. The magnitude of the derivative indicates the pixels that need to be modified the least to affect the class score the most. Hence, saliency maps can be used to judge the importance of each pixel. Saliency based methods such as CAM [14], Grad-CAM [15] and Grad-CAM++ [16] have emerged as popular tools to highlight features in a given input image. In this work, we use saliency maps to define an adaptive learning rate for updating each pixel in the generated visualizations.

### 2.3. Class Impressions

The method of generating visualizations from a trained classifier was first proposed by Erhan et al. [7], which was applied on Convolutional Networks later by Simonyan et al. [8]. There are several works on the addition of regularizers to improve the quality of generated images [17]. We use the recent methodology proposed by Mopuri et al. [11] as a starting point for this work. They initially start with a noise image and iteratively update this using gradient ascent to maximize the logits of a given class. A set of transformations such as random rotation, scaling, RGB jittering and random cropping

between iterations ensures that the generated images are robust to these transformations; a feature that natural images typically possess. The generated images are termed as *Class Impressions* (Fig.1(a)). Some of the drawbacks of this approach are as follows: Class Impressions lack the texture and smoothness of natural images, logits are maximized by creating multiple instances of the same object, and this approach does not necessarily generate only the discriminative features of a given class. As seen in Fig.1(a), generated images contain background and foreground features. Background features are clearly not discriminative features, as they may be common to many other classes as well.

We address these issues in the proposed approach, which is described in detail in the following section.

## 3. PROPOSED METHOD

In this section, we describe each of the improvements proposed in this paper, followed by the combined impact of all. We further discuss the methodology developed for generation of single-object images, followed by a brief note on the hyper-parameters used in the loss function. Finally, we demonstrate generation of natural-looking *negative images*.

The notation used in the remainder of this paper is as follows:  $SCI_c$  denotes the Saliency-driven Class Impressions generated for class  $c$ .  $L_c(I)$  denotes the pre-softmax output corresponding to class  $c$  for an image  $I$ .  $I_{i,j}$  denotes the intensity of the pixel at the location  $(i, j)$  in the image  $I$ .

A VGG-F network [18] pre-trained on ILSVRC [19] dataset is used for all the experiments described in this paper. The generated Class Impressions are of dimension  $224 \times 224 \times 3$ , which is the same as the dimension of images in the ILSVRC dataset. We start with an initial mean image, and add a noise image with each pixel sampled independently from  $U[0, 255]$ . The image is updated iteratively for a fixed number of iterations using gradient ascent to maximize a pre-defined loss, described later in this section. A set of random transformations [11] (Section-2.3) are applied after every iteration. We also control the gradient flow to constraint the Class Impressions (CIs) to develop a single object only.

### 3.1. Total Variation Loss as a Natural Image prior

One of the key statistical properties that characterize images, specifically natural images, is their spatial smoothness. In order to enforce this prior, the objective function of generating Class Impressions [11] is modified to include the Total Variation (TV) loss [12], as shown below:

$$SCI_c = \operatorname{argmax}_I L_c(I) - \lambda_1 \sum_{i,j} |I_{i+1,j} - I_{i,j}| + |I_{i,j+1} - I_{i,j}| \quad (1)$$

Here,  $\lambda_1$  is the weightage given to the TV loss component. The images generated using this loss are shown in Fig.1(b)



**Fig. 1.** Class *Tench*: (a) CI (b, c) CI with TV loss



**Fig. 2.** *Madagascar Cat*: Generated CIs after (a) 100, (b) 200, and (c) 500 iterations

and (c). This loss penalizes the high frequency components, thereby producing locally smooth images. After using TV loss, texture of the fish resembles that of the original image better. It must be noted that this process is not equivalent to applying a low pass filter (LPF) on the output image. In the case of applying LPF to the output image, important edge information is also lost. However, when TV loss is used during training, the edge information that is crucial for classification is preserved, as it helps maximize activations of that class. For example, eye of the fish is more predominant in Fig.1(b) and (c) when compared to Fig.1(a). The optimization of each component of loss function in Eq.1 is done individually. The TV loss is applied only once in every  $k$  iterations. For Fig.1(b),  $k = 2$ , and for Fig.1(c),  $k = 1$ .

### 3.2. Saliency-driven Class Impressions

Traditionally, Class Impressions (CIs) [11] are generated by maximization of the pre-softmax value of a given class. This can be achieved in two ways:

- By improving the confidence of generated features in a small subset of pixels
- By maximizing the number of pixels that have the required features

A study of the CIs generated over iterations shows that the optimization follows a mix of both approaches. This results in images that have repeated structures. Fig.2 shows the generated class impressions for the class, *Madagascar Cat*. After 100 iterations there are 2 cats clearly visible in the image. This number increases to 3 after 200 iterations and 4 after 500 iterations. Contrary to this, we expect the initial features to become more discriminative with progress in the number

of iterations. This could potentially be achieved by adding a loss that penalizes the number of pixels that contribute to the correct classification of an image. However, the drawback of this approach is that a threshold needs to be selected to count only some of the pixels as *strong contributors*. This threshold needs to be image-specific, as the magnitude of gradient and number of contributing pixels can be different for different images, based on their scale and texture.

To address this issue, we propose to control the learning rate of each pixel adaptively, based on the Saliency maps computed at each step. This ensures that the pixels that maximally contribute to the objective function develop at a faster rate when compared to the non-contributing pixels. Generation of Saliency maps does not add to the computational complexity, as the derivative of the loss function with respect to the input is also required to be computed for the generation of CIs. We start with a uniform learning rate of 0.1 for all pixels. The learning rate update rule is illustrated here. We first calculate  $grad_{cum,i}$ , which is the cumulative weighted gradient upto the  $i^{th}$  iteration as shown below:

$$grad_{cum,i} = \begin{cases} grad_{cum,i-1} \cdot iter + C_1 \cdot \frac{grad_i}{\|grad_i\|}, & \text{if } iter < t \\ grad_{cum,i-1} \cdot iter + C_2 \cdot \frac{grad_i}{\|grad_i\|}, & \text{otherwise} \end{cases} \quad (2)$$

The weightage given to the direction of current gradient  $grad_i$  is  $C_1$  upto the  $t^{th}$  iteration and  $C_2$  beyond this. The value of  $grad_{cum,0}$  is considered to be a uniform value of 0.1 for all pixels. We use the value of  $C_1$  as shown below:

$$C_1 = C_2 \cdot \frac{max_{iter}}{t} \quad (3)$$

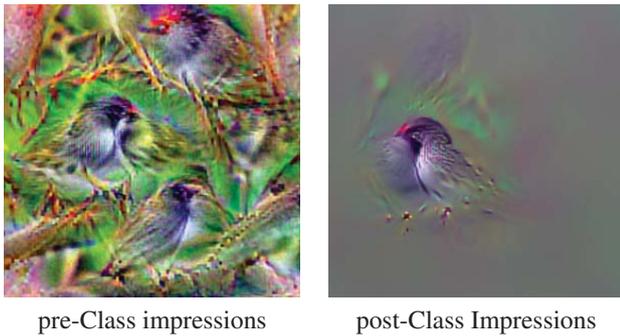
The value of  $C_2$  is chosen as 4 and  $t$  is set to 150 in all our experiments. The maximum number of iterations is represented by  $max_{iter}$ . The cumulative weighted gradient,  $grad_{cum,i}$  is used as a weighting factor to the saliency map  $grad_i$ , to obtain the final learning rate map  $lr_{map}$  for the image.

$$lr_{map} = grad_{cum,i-1} \cdot grad_i \quad (4)$$

### 3.3. Methodology for Single object generation

The Saliency driven approach described in the above section helps in developing the important regions of an image faster than other regions. In order to develop a single object for a given image, it is also important to ensure that the salient regions are connected. In this section, we describe the methodology used to achieve this. We first develop initial *Pre-Class Impressions (pre-CI)* by setting a very high learning rate for all pixels. With this setting, Class Impressions develop within a very few iterations. Next, we select the most activated region of the CIs and develop this region again, with a lower learning rate, as shown in Fig. 3.

The methodology used for selecting the most activated region is described here. Let  $circ(x, y, r)$  denote an image of



pre-Class impressions      post-Class Impressions  
**Fig. 3.** Pre-CI and the corresponding post-CI

the same size as that of the pre-CI, with a value of 1 within a circle of radius  $r$  centered at  $(x, y)$ , and a value of 0 otherwise. We define the region around  $(x, y)$  as the most activated region, if the circle of radius  $r$  around this pixel has the highest magnitude of cumulative activation value as shown below:

$$(x, y)_{max} = \underset{(x,y)}{\operatorname{argmax}} \sum_i \sum_j |preCI \cdot circ(x, y, r)| \quad (5)$$

A circular mask,  $circ((x, y)_{max}, r)$  is constructed to be centered at the most activated region  $(x, y)_{max}$ . This is multiplied with the  $lr_{map}$  in Eq.4 to generate the final learning rate map for each pixel. The mask helps in developing the most activated region first, and further expanding this into a full object. The radius of this circular mask starts with 1 and increases linearly over 150 iterations to a value of 150, beyond which it is kept constant.

The inclusion of Total Variation loss ensures that the region around the object fades out, and only the high confidence features sustain.

#### 4. RESULTS AND CONCLUSIONS

We show a comparison of our results with that of the baseline by Mopuri et al. [11] in Fig. 4. It can be seen that the proposed method generates images of significantly improved quality, containing single instances of objects, unlike the corresponding results of Ask, Acquire and Attack (AAA) [11], where multiple instances are developed.

##### 4.1. Images developed by fusion of two classes

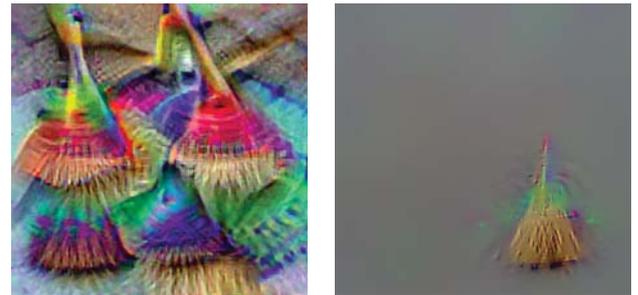
We use the proposed method to generate negative class images by fusing class impressions of 2 or more classes. Here, we start from a single pixel and develop CIs of one class around this pixel for a few iterations. Next, the second class is developed from a different region for a few iterations. This generates a natural fusion between the class impressions of both classes, as can be seen in Fig.5. These negative class images do not belong to either of the two classes confidently,



Watch-Tower, AAA [11]      Watch-Tower, Ours

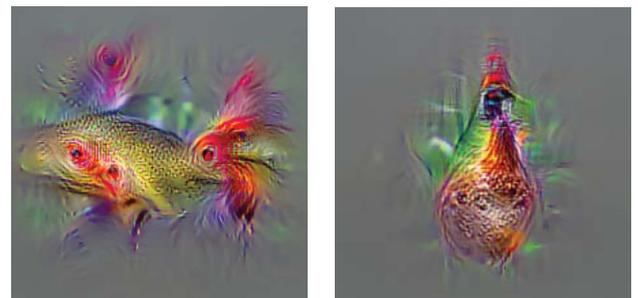


Fox, AAA [11]      Fox, Ours



Broom, AAA [11]      Broom, Ours

**Fig. 4.** Comparison with existing methods



Fusion of fish and hen      Fusion of bottle and hen

**Fig. 5.** Negative Images

as they contain features that belong to the other class. Such images can be used to train robust classifiers, improve the confidence predictions of a classifier and also for developing a reject option for out-of-distribution samples.

## 5. REFERENCES

- [1] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton, “Imagenet classification with deep convolutional neural networks,” in *Advances in neural information processing systems*, 2012, pp. 1097–1105. [1](#)
- [2] Jonathan Long, Evan Shelhamer, and Trevor Darrell, “Fully convolutional networks for semantic segmentation,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3431–3440. [1](#)
- [3] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi, “You only look once: Unified, real-time object detection,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 779–788. [1](#)
- [4] David G Lowe, “Distinctive image features from scale-invariant keypoints,” *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004. [1](#)
- [5] Navneet Dalal and Bill Triggs, “Histograms of oriented gradients for human detection,” 2005. [1](#)
- [6] Yann LeCun, Léon Bottou, Yoshua Bengio, Patrick Haffner, et al., “Gradient-based learning applied to document recognition,” *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998. [1](#)
- [7] Dumitru Erhan, Yoshua Bengio, Aaron Courville, and Pascal Vincent, “Visualizing higher-layer features of a deep network,” *University of Montreal*, vol. 1341, no. 3, pp. 1, 2009. [1](#), [2](#)
- [8] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman, “Deep inside convolutional networks: Visualising image classification models and saliency maps,” *arXiv preprint arXiv:1312.6034*, 2013. [1](#), [2](#)
- [9] Matthew D Zeiler and Rob Fergus, “Visualizing and understanding convolutional networks,” in *European conference on computer vision*. Springer, 2014, pp. 818–833. [1](#)
- [10] Jason Yosinski, Jeff Clune, Anh Nguyen, Thomas Fuchs, and Hod Lipson, “Understanding neural networks through deep visualization,” *arXiv preprint arXiv:1506.06579*, 2015. [1](#)
- [11] Konda Reddy Mopuri, Phani Krishna, and R. Venkatesh Babu, “Ask, acquire, and attack: Data-free uap generation using class impressions,” in *European Conference on Computer Vision*. Springer, 2018, pp. 20–35. [1](#), [2](#), [3](#), [4](#)
- [12] Leonid I Rudin, Stanley Osher, and Emad Fatemi, “Nonlinear total variation based noise removal algorithms,” *Physica D: nonlinear phenomena*, vol. 60, no. 1-4, pp. 259–268, 1992. [1](#), [2](#)
- [13] Quoc V. Le, Marc’Aurelio Ranzato, Rajat Monga, Matthieu Devin, Kai Chen, Greg S. Corrado, Jeff Dean, and Andrew Y. Ng, “Building high-level features using large scale unsupervised learning,” in *Proceedings of the 29th International Conference on International Conference on Machine Learning*, USA, 2012, ICML’12, pp. 507–514, Omnipress. [2](#)
- [14] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba, “Learning deep features for discriminative localization,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2921–2929. [2](#)
- [15] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv 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. [2](#)
- [16] Aditya Chattopadhyay, Anirban Sarkar, Prantik Howlader, and Vineeth 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](#)
- [17] Chris Olah, Alexander Mordvintsev, and Ludwig Schubert, “Feature visualization,” *Distill*, 2017, <https://distill.pub/2017/feature-visualization>. [2](#)
- [18] Ken Chatfield, Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman, “Return of the devil in the details: Delving deep into convolutional nets,” *arXiv preprint arXiv:1405.3531*, 2014. [2](#)
- [19] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al., “Imagenet large scale visual recognition challenge,” *International journal of computer vision*, vol. 115, no. 3, pp. 211–252, 2015. [2](#)