INTERPRETING CNN PREDICTION THROUGH LAYER-WISE SELECTED DISCERNIBLE NEURONS

Anonymous authors
Paper under double-blind review

ABSTRACT

In recent years, researchers have seen working on interpreting the insights of deep networks in the pursuit of overcoming their opaqueness and so-called 'black-box' tag from them. In this work, we present a new visual interpretation technique that finds out discriminative image locations contributing highly towards networks' prediction. We select the most contributing set of neurons per layer and engineer the forward pass operation to gradually reach to the important locations of the input image. We explore the connectivity structure of the neuron and obtain support from succeeding and preceding layer along with its evidence from current layer to advocate for a neuron's importance. While conducting this operation, we also add priorities to the supports from neighboring layers, which, in practice, provides a reliable way of selecting the discriminative set of neurons for the target layer. We conduct both the objective and subjective evaluations to examine the performance of our method in terms of model's *faithfulness* and *human-trust*, where we visualize its efficacy over other existing methods.

1 INTRODUCTION

With the rise of unprecedented performance of deep leaning methods in various computer vision tasks over last few years, the network architecture also gets complex (Szegedy et al., 2015; He et al., 2016) to preserve such diverse variations. Such complex architecture although facilitates with higher recognition performance, but provides less understanding of the fact that 'how it actually works!'. As a result, people now and then tag it as a black-box model, raising the necessity of eradicating its opaqueness while being more understandable and transparent for general use.

To understand the inner representations of deep networks, it is important to see how and what the network learns in practice. One possible way is to look for the salient image regions that contribute the most for the networks' prediction. In this way, we not only get to know how a network is making its prediction, but also we will have the idea which portion is guiding the network towards a miss-prediction. In fact, with the use of such visualization, researchers can better interpret the insights of CNN's prediction. The straight-forward way that these techniques take is to finding out discriminative image regions that supplement the label predictions, acting as the visual explanations for the predicted label making us understand the class specific patterns learned by the models. Majority of the work on this area utilize gradient information to visualize the salient regions contributing towards predicting input label (Simonyan et al., 2013; Zeiler & Fergus, 2014; Springenberg et al., 2014; Zhou et al., 2016; Selvaraju et al., 2017) , among which some are simply constrained to specific network architecture while some other outputs low-resolution visualizations, limiting the overall understandability of the method.

In this work, we present a visual interpretation technique that finds out the most discriminative image locations contributing to the network prediction. For this, we select discriminative set of neurons per layer and engineer the forward pass operation to gradually reach to the important locations of the input image. To be specific, for a given network's prediction, we start from the softmax layer to find the discriminative neurons for each layer through a novel proposed way and gradually traceback to the input image for the most important locations. While finding out such set of neurons, we explore the connectivity structure of the neuron and obtain evidence support from current layer along with supports from the neighboring layers, i.e., succeeding and preceding layer. While doing this, we also provide priorities of the information obtained from each neighboring layer. In this way, we come up

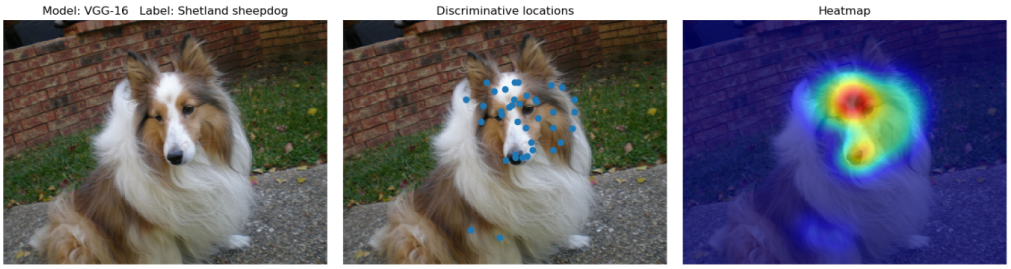


Figure 1: Input image and corresponding discriminative locations found by our method. Third column shows the heatmap generated from the discriminative locations.

with a reliable approach of finding out the discriminative neurons per layer to be propagated for the important image locations at the end. Our method is advantageous with a generic approach that can be adopted to any network, where it can produce high-resolution pixel-wise localization of important pixels. We conduct set of experiments to visualize the efficacy of proposed technique against other existing methods, through both objective and subjective evaluations. One sampel visualization is provided in Fig. 1.

2 Related Works

In recent years, there have been numerous efforts on interpreting the deep network in terms of visualizing the network's performance. The dominant group of researches have been conducted on the visualization based on gradient based approach. Simonyan et al. (2013) compute the sensitivity of classification score in terms of the partial derivative of the classification score for a given class with respect to the pixel value changes. Deconvolution-based works (Zeiler & Fergus, 2014) take a similar approach to visualize the salient feature concepts across different layers. Guided backprop (Springenberg et al., 2014) is another method that utilizes the gradient by modifying them for a better qualitative visual representation.

Class-specific activation maps (CAM) (Zhou et al., 2016) generate salient feature-maps by combining the intermediate feature maps before global average pooling layer. Although such techniques provide better flexibility than the prior approaches in terms of interpreting the prediction, they are disadvantageous with their architecture-specific design. Improvements are done over this method by utilizing gradient information, as in (Selvaraju et al., 2017). Nevertheless, Selvaraju et al. (2017) still use low-resolution maps which perhaps are disadvantageous for better interpretation.

We also observe group of work (Bach et al., 2015; Cholakkal et al., 2016; Robnik-Šikonja & Kononenko, 2008; Zintgraf et al., 2017) taking relevance score for each feature with respect to a class and estimate whether the prediction change in absence of that feature. Large changes in prediction indicate the importance of the feature while small changes indicate the opposite. Some other approaches (Cholakkal et al., 2016) take probabilistic approaches to find the contribution of each image patch (or pixel) to the classification detailing their understanding. Zhang et al. (2018) computes marginal winning probabilities for neurons at each layer, where distinct attention map is computed as the sum of these probabilities across the feature maps.

Recently, Mopuri et al. (2018) proposes CNN-Fixations based method that selects important neurons to trace the salient image region interpreting the network's prediction. One important part of (Mopuri et al., 2018) is to select the salient neurons for each layer, where authors suggest a naive approach of looking at the best activation values. However, this approach only concentrates on the current layers activation values to select the best set of neurons. Considering the connected structures of each neuron to the other neurons from preceding and succeeding layers, we argue that information from neighboring layers, e.g., the preceding and succeeding layers also contribute towards its excitation. While selecting the important neuron, we consider information from succeeding, current and preceding layer along with providing priorities to each layer information. In this way, we come up with a more reliable way of selecting the salient neurons in order to be propagated towards the salient image regions. As soon as we reach the input image, with the discriminative location, we

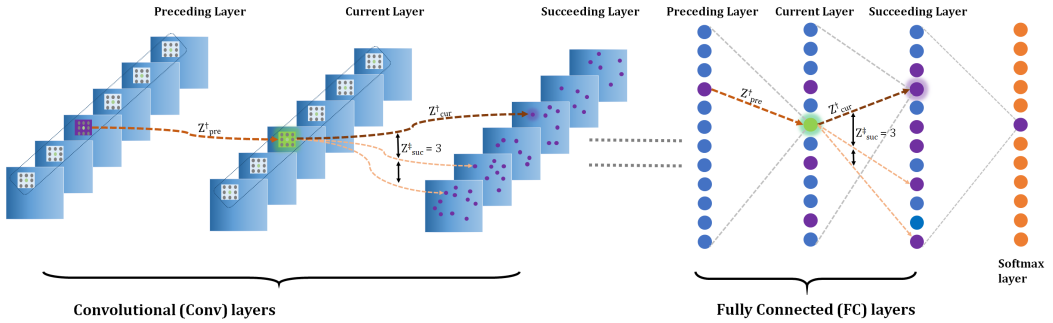


Figure 2: Example of a fc-layer and conv layer is shown, where we find the discriminative neurons. Purple neurons denote the set of important neurons for a layer. For a such neuron (purple with glow) from the succeeding layer, we calculate score for a target neuron (green with glow). Supports from the succeeding, preceding and current layers are denoted by corresponding $\mathcal{Z}$ notations. Details of calculating those $\mathcal{Z}$-values can be found in section 3.1.

generate heatmap by Gaussian blurring those neurons. We require no hyper-parameters or heuristics in the entire process of back-tracking the evidence and our method can easily be adopted to any network architecture.

3 METHODOLOGY

We describe the basic methodology of our approach in this section. The main goal is to identify the discriminative image locations that contribute most for the CNN prediction, providing an explainable interpretation of the CNN. Similar to the other recent works (Mopuri et al., 2018; Huber et al., 2019) that select important neurons to trace-back the salient input pixels, we also engineer the forward-pass operation to discover the discriminative salient image pixels. Such strategy typically starts from the network's prediction neuron and sequentially selects the most active set of neurons per layer to get through to the input image. Considering only the activation values can perhaps be considered as one of the naive ways of selecting the discriminative set of neurons for each layer, as also observed in (Mopuri et al., 2018). Nevertheless, since the neurons are connected to the preceding and succeeding layer, we argue that information from such neighboring layers is also informative in selecting the best set of neurons per layer. Moreover, while utilizing information from the neighboring layers, we assign learned weights to those layers which also specify the importance of the neighboring layers in selecting such neurons.

3.1 FORMULATION

In this part, we describe the general methodology of our approach. We start with a neural network with L layers. After a forwards pass is completed during the inference, we begin with the last fc layer as the current layer, where the softmax layer is considered as the succeeding layer and previous fc layer (if exists, otherwise other conv/pool layer) will be considered as the preceding layer. Neurons with top n-probabilities from softmax layer are considered as the discriminative neurons for that layer, for which we try to find a set of most contributing neurons for the last fc layer (current layer). For selecting such set of discriminative neurons, we consider the sequential connectivity structure of a network where the contribution of a neuron is related and dependent on the connectivity of neighboring layers. In our approach, to select a discriminative neuron, we consider the evidence from the current layer along with support from the preceding and succeeding layer. Formally, for each of the discriminative neuron ($\ominus$) from the succeeding layer, we calculate a score, $\Omega^{\odot}_{\ominus}$, for each of the neuron ($\odot$) at current layer,

$$\Omega^{\odot}_{\ominus} = \alpha_{suc}\mathcal{Z}^{\ddagger\odot}_{suc}(\alpha_{cur}\mathcal{Z}^{\dagger\odot}_{cur} + \alpha_{pre}\mathcal{Z}^{\dagger\odot}_{pre}), \tag{1}$$

where, $\mathcal{Z}$ and α with subscript cur, pre and suc denote calculated support and corresponding weight values for the current, preceding and succeeding layers, respectively. This equation indicates that for

each neuron, we calculate three support values from current, the succeeding and preceding layers. The corresponding weight values for each layer are learned beforehand, which denote how much we want to prioritize the support information from each layers.

Two superscript ‡ and † denote different support metrics. Support $\mathcal{Z}$ with superscript † denotes that the support is devised by multiplying its activation value with its weight connected to selected neuron from another layer. For the current layer, we calculate it by multiplying the current activation $\mathcal{A}^{\odot}_{cur}$ with its connected weight $\mathcal{W}^{\odot}_{cur}$ with the selected neuron from the succeeding layer. Formally, we define it as,

$$\mathcal{Z}^{\dagger\odot}_{cur} = \mathcal{A}^{\odot}_{cur} \cdot \mathcal{W}^{\odot}_{cur}. \tag{2}$$

For the preceding layer, we obtain the support from the most contributing neuron of the preceding layer that is connected with the target neuron from current layer. To get such support, we first observe the individual supports for all the neurons of the preceding layer that are connected with the target neuron (from current layer) by multiplying their activations with corresponding weight value. From all the supports, we select the most contributing one by picking up the highest value. We define $\mathcal{Z}^{\dagger\odot}_{pre}$ as,

$$\mathcal{Z}^{\dagger\odot}_{pre} = \arg\max\{[\mathcal{A}^{\diamond}_{pre} \cdot \mathcal{W}^{\diamond}_{pre}] : \diamond \in \mathcal{N}_{pre}\}, \tag{3}$$

where, $\mathcal{N}_{pre}$ denotes the set of neurons from preceding layer. We $\mathcal{A}^{\diamond}_{pre}$ and $\mathcal{W}^{\diamond}_{pre}$ denote the activation and corresponding weight value for a neuron ($\diamond$) from the set $\mathcal{N}_{pre}$.

We also observe the influence of current neuron ($\odot$) in the succeeding layer in the sense that for how many cases (selected neurons from succeeding layer), it provides the best support. To be specific, we specify a counter to check for how many times the current neuron possess the highest contribution in terms of its $\mathcal{Z}^{\odot}_{cur}$ value for different selected neurons from succeeding layer. Default value 1 is used in case the above condition is not satisfied. The counter value is used as $\mathcal{Z}^{\ddagger\odot}_{suc}$ in Eq. 1. Basically, if the current neuron provides best support for different selected neurons from succeeding layer, we consider current neuron ($\odot$) as one of the influential neurons for current layer, which is why we multiply this counter value in the Eq. 1 to provide more support for the current neuron.

We calculate $\Omega^{\odot}_{\ominus}$ values for the set of all neurons ($\mathcal{N}_{cur}$) at current layer, for a selected neuron ($\ominus$) from the succeeding layer. Finally we consider top *k*-neurons based on highest $\Omega^{\odot}_{\ominus}$ values, which we define as the most contributing neurons, for the selected neuron ($\ominus$) from succeeding layer. The selected set of neurons $\mathcal{N}^{*}_{cur^{\ominus}}$, is derived as,

$$\mathcal{N}^{*}_{cur^{\ominus}} = \arg\max_{k}\{\Omega^{\odot}_{\ominus} : \odot \in \mathcal{N}_{cur}\}. \tag{4}$$

Note that we use a superscript $*$ as a sign of being selected set of neurons. However, in the above way, for all the individual discriminative neurons from succeeding layer, we select the set of discriminative neurons for current layer. We take the union of all such sets and obtain the final set of discriminative neurons for current layer. We define it as,

$$\mathcal{N}^{*}_{cur} = \bigcup_{\ominus \in \mathcal{N}^{*}_{succ}} \mathcal{N}^{*}_{cur^{\ominus}}, \tag{5}$$

where, $\mathcal{N}^{*}_{suc}$ denotes the set of discriminative neurons for successive layer. In this way, we sequentially select the set of most contributing neurons for each of the layers. Finally, we reach to the input image and get the most distinctive image pixels.

While generating discriminative set of neurons in conv layers, we also use the same strategy (Eq 1) as taken in fc layer, except computing support value $Z^{\dagger}$ slightly differently. Since the neurons at conv layer consist of 3D fields having channel info along with spatial location info (x, y), the neurons appear as 3D spatial blob. For each selected neuron, we first extract the corresponding activations within a receptive field, as denoted in green rectangle in conv-layer part in Fig. 2. Recall that in the fc layer, while dealing with any of the $Z^{\dagger}$ values, we multiply the activation value with corresponding weight value. Now the change we do in conv layer while dealing this issue is that we compute the Hadamard product between the above-mentioned receptive activations and associated filter weights of the specified neuron. We note that result of this Hadamard product is also a 3D spatial blob of same size. Therefore, for any further calculation, *i.e.,* obtaining the highest value,

Algorithm 1 Selection of discriminative set of neurons for a layer

Input: Discriminative set of neurons from successive layer $\mathcal{N}^*_{suc}$,
Layer-wise pre-learned weights $(\alpha_{cur}, \alpha_{suc}, \alpha_{pre})$.
Output: Discriminative set of neurons for current layer $\mathcal{N}^*_{cur}$

1: **for** each neuron $\ominus \in \mathcal{N}^*_{suc}$ **do**
2: **for** each neuron $\diamond \in \mathcal{N}_{cur}$ **do**
3: Calculate score $\Omega^{\odot}_{\ominus} = \alpha_{suc}\mathcal{Z}^{\ddagger^{\odot}}_{suc}(\alpha_{cur}\mathcal{Z}^{\dagger^{\odot}}_{cur} + \alpha_{pre}\mathcal{Z}^{\dagger^{\odot}}_{pre})$, where $\mathcal{Z}^{\ddagger^{\odot}}_{suc}$, $\mathcal{Z}^{\dagger^{\odot}}_{cur}$, and $\mathcal{Z}^{\dagger^{\odot}}_{pre}$ are the calculated supports from successive, current and previous layers, respectively (details are in section 3.1)
4: **end for**
5: Select k-discriminative neurons with highest scores, $\mathcal{N}^*_{cur^{\ominus}} = \arg\max_k\{\Omega^{\odot}_{\ominus} : \odot \in \mathcal{N}_{cur}\}$.
6: **end for**
7: Select the final set of discriminative neurons by taking union of all $\mathcal{N}^*_{cur^{\ominus}}$, $\mathcal{N}^*_{cur} = \bigcup_{\ominus \in \mathcal{N}^*_{succ}} \mathcal{N}^*_{cur^{\ominus}}$
8: **return** $\mathcal{N}^*_{cur}$

we sum up the output values across the (x, y) spatial region to numerically process each channel information with single value. In this way, we simply trace back the (x, y) location of the succeeding layer onto the strongest contributing locations (channels) of the current layer. For pooling layers, we simply extract 2D receptive fields of the corresponding neuron and find the locations (or neurons) having the highest activations since most of the models normally use max-pooling to sub-sample the feature maps. Sample illustration for fc and conv layer in extracting the important neurons is provided in Fig. 2 and the general algorithm is provided in Alg. 1.

3.2 Layer-wise Weight selection

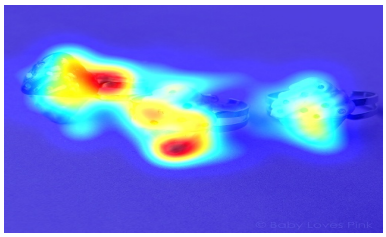
In this section, we describe a strategy to learn the layer-wise weights $(\alpha_{cur}, \alpha_{suc}, \alpha_{pre})$ defined in Eq. 1. We adopt one of the popular reinforcement learning techniques to learn these weights.

First, let us define the above weight selection problem as a tuple of action, A and corresponding rewards, R. Such action and corresponding reward value can be generated in many different ways. However, in our approach, we consider Intersection-Over-Union (IOU) score for an input image to define tuple $< A, R >$. For the experiment, we randomly select 1000 images from Imagenet Validation dataset (Russakovsky et al., 2015) and calculate IOU score for each of them within the pre-defined range of values for the weights defined above (i.e, α_{cur}). We consider the IOU score as action A and if its IOU score passes a threshold (0.5), then we put a binary reward $R = 1$, otherwise no reward is given.

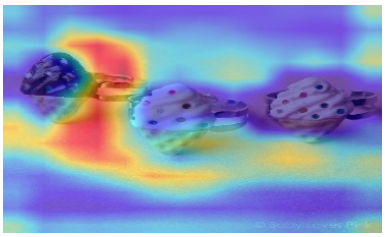
To elaborate it, after tracing back for the discriminative regions to the input image, we generate bounding box (details of generating bounding box are provided in section 4.1.2) around the target object. Afterwards, we compare this with the ground-truth to get the IOU score. The rationale behind considering IOU score is that it shows how accurately a method identifies the important image regions. However, the IOU score for each image is calculated separately for each of the weight values for respective predefined value range. Now we can define the above scenario as a Bernoulli multi-armed bandit problem (Auer et al., 2002) as a tuple of $< A, R >$, where,

- For each individual target weight, (*i.e.,* α_{cur}), we have K-predefined discrete values with reward probabilities $\theta_1, ..., \theta_K$.
- At each trial t, we take an action a that is to generate IOU score at an input image for the target weight within its any of the K-values, and receive a reward r.
- The value of action a is the expected reward, $Q(a) = E[r|a] = \theta$. If action a_t at the trial t is on the i^{th} value of the target weight, then $Q(a_t) = \theta_i$.
- R is a reward function. In the case of Bernoulli bandit, we observe a reward r in a stochastic fashion. At the each trial t, $r_t = R(a_t)$ may return reward 1 with a probability $Q(a_t)$ or 0 otherwise. The goal is to maximize the cumulative reward $\sum_{t=1}^{T} r_t$.

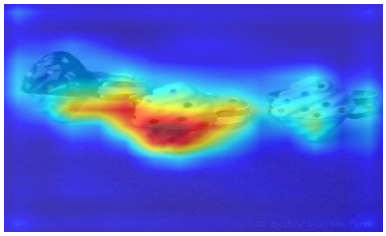
(a) Original Image



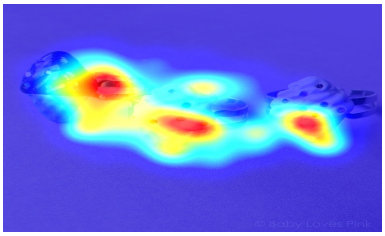
(b) CNN-Fixations



(c) Grad-CAM



(d) Grad-CAM++



(e) Proposed

Figure 3: Visual comparison among different methods.

We solve the above problem using standard UCB solver (Agrawal, 1995), where the optimal action a^* (that is the optimal value for each weights) is selected based on optimal probability θ^* defined as, $\theta^* = Q(a^*) = \max Q(a)_{q \in A} = \max \theta_{i1<=i<=K}$. Note that the above strategy is applied separately for the three weights mentioned above within their respective range of values, and the optimal values for each of the weights are selected separately.

4 Experimental Result

Of course, one such interpretation work must show its consistency with the model's prediction (faithfulness) and should be good enough to gain human trust. To show these things, we conduct both the objective and subjective evaluations to compare our method against other existing methods in terms of *faithfullness* and *human trust*. We also evaluate the robustness of our method in presence of adversial noise. In addition, we analyze the model's prediction from their heatmaps in case there is a miss-classification.

4.1 Objective Evaluation

For the objective evaluation, we judge the consistency of the methods with respect to the prediction of the model. We conduct couple of experiments for this purpose. The details are given below.

4.1.1 Pointing game Evaluation

The pointing game technique was introduced in (Zhang et al., 2018), which examines the discriminativeness of different attention maps for the sake of object-localization purpose. If the maximum

Table 1: Comparative objective evaluations for different methods. Note*: results are taken from (Chattopadhay et al., 2018).

Methods	Pointing-Game Scores (%)	Increase-in-performance (%)
Grad-CAM	40.67	2.94*
Grad-CAM++	50.57	17.65*
CNN Fixations	53.40	22.36
Proposed	**55.25**	**24.84**

(a) Original Image (b) Discriminative locations found by our method (c) Image with occlusion in non-object area.

Figure 4: Generating occlusion in the image.

point in the attention map falls within the ground truth bounding box, one $\#hit$ is counted, otherwise it is considered as one $\#miss$. This technique simply asks for pointing at an object of specific category of the image and it does not require any mechanism to highlight full object range . In this technique, the final score $\mathcal{PG}$ is calculated as a ratio of number of $\#hits$ with respect to total samples $(\#hits + \#misses)$.

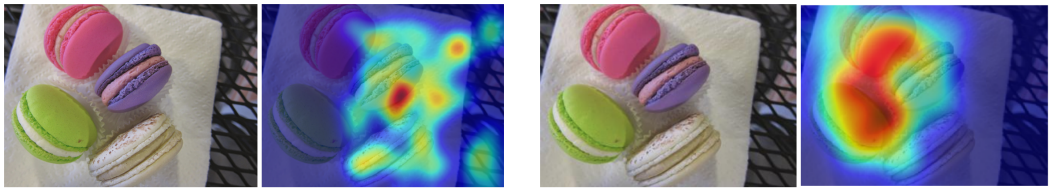
$$\mathcal{PG} = \frac{\#hits}{\#hits + \#misses}. \tag{6}$$

In our approach, we randomly collect 2501 images from ImageNet (ILSVRC2012) validation set, and calculate $\#hit$ or $\#miss$ for each image. The final $\mathcal{PG}$ score is then calculated based on the above equation. The experiment is conducted 3 times and the average results are reported at Table 1. We conduct this experiment for the other existing methods, namely Grad-CAM (Selvaraju et al., 2017), Grad-CAM++ (Chattopadhay et al., 2018), CNN-Fixations (Mopuri et al., 2018), using the same strategy described above. As we see from the table, proposed method achieves the highest $\mathcal{PG}$ score than other methods, demonstrating its ability to localize the designated objects in a better way than the other methods.

4.1.2 CHANGE-IN-CONFIDENCE EVALUATION

In this experiment, we examine the efficacy of the model's visual explanation in the the ovreall decision process. To conduct this experiment, we occlude the specific parts of the image based on the visual map generated by the designated method, and check the change in classifier's confidence due to the forced occlusion. At first, we generate bounding box around the salient regions. For the generation of bounding box, we first discard the outlier points. We define a point as an outlier if there is a absence of sufficient neighboring points within a given circle. The number of outlier points and the radius of the circle are found empirically for the method. However, as soon as we remove the outliers, we generate the best fitting bounding box covering the remaining points.

For our experiment, we keep the object part within the bounding box and occlude other region, as shown in Fig. 4. The rationale of this experiment is that for the prediction, CNN mainly looks for the important region, and as a result, occluding the unimportant region may enhance the confidence score of the prediction. To observe the change in performance, we occlude the other regions apart from the bounding box and examine the increase in performance. To be specific, we consider how many times the model gain an increase in performance, and report the average results in the Table 1. The results are generated similarly as before on randomly selected 2501 images from ImageNet. As



Label: Bath towel | Heatmap | Perturbed Label: Ice cream | Heatmap

Figure 5: Performance on perturbed adversial images.

we observe from the results, by occluding the non-object region, we gain the confidence boost for the maximum time than other method, which clearly shows its efficacy in correct localization of the designated object.

For both the above experiments, we achieve expected results that exhibit a better confidence in generating explanation that are more faithful to the deep network, as compared to other methods. Some of the visual comparative results are also provided in Fig. 3. Note also that for both the experiments, we have used 2501 images from ImageNet validation set. We do this in order to keep similarity in the experimental settings, as in (Chattopadhay et al., 2018).

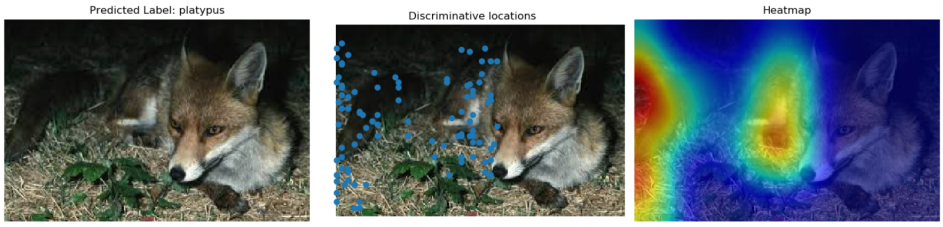
4.2 Subjective Evaluation: Human Trust

In the aforementioned set of experiments for objective evaluations, we explored the faithfulness property of the method; on the contrary, in this section, we conduct subjective evaluation experiment to evaluate its interpretibility in terms of human trust.

To elaborate more, if the visual explanation for different methods are given, we want to evaluate which one seems more trustworthy. The purpose of this experiment is to check whether the human perception of the visual output (heatmaps) of our method complies with machine accuracy. For this purpose, we compare the visual maps for AlexNet and VGG-16 generated by our method. We know from the past researches that VGG-16 shows better results than AlexNet on image classification tasks [79.09 mAP (vs. 69.20 mAP) on PASCAL dataset (Everingham et al., 2010)]. In order to comply with this result, we take 58 images where both the AlexNet and VGG-16 predicts the object correctly and generate visual map for our method. We then provide the maps to 12 users, first ask them *'Which map best describes the object present in the image?'*. Surprisingly, all the users voted for VGG-16 as producing a better map than ALexNet. Later, we also ask the users that *'How reliable the map of each model?'* We provide reliability scores for both the models between $1 \sim 5$ with the radio-button option. The obtained scores are then normalized, where we found that VGG-16 (38.08) has a higher score than AlexNet (36.07), which also complies with the previous finding that VGG-16 achieves higher accuracy than AlexNet for object classification. In this way, based on the explanation from the human prediction, our visualization method can help users place trust in a model that can generalize better.

4.3 Robustness Evaluation

Recent work show that deep networks shows vulnerability to adversial attacks (Goodfellow et al., 2014). This finding actually provides a great chance to test the robustness of a method's performance in practice. Adversial attacks perturb the images such a way that fools the network to miss-classify the existing object with a high probability. We generate images for Imagenet trained on VGG-16 using Deepfool (Moosavi-Dezfooli et al., 2016) and observe the visualizations for our method. As we can see from Fig. 5 that the adversial attack changes the input label from bath-towel to ice-cream. In the perturbed image, the heatmap concentration shifts to round-shape objects. However, portion of heatmap also covers the towel portion, which is still a visually present object in the image. Therefore, in case of adversial attack, our visualization map not only shows shifted concentration of object, but also covers the original object that is stil present in the image. In this way, we observe the robustness of our visualization map.



(a) Discrminative location and Map for AlexNet

(b) Discrminative location and Map for VGG-16

Figure 6: Visualization map for AlexNet and VGG-16 on a *grey-fox* object, where VGG-16 classifies it correctly and AlexNet missclassifies it.

4.4 Analyzing miss-classified Image

One of the best purpose of the visualization methods is the analysis of *'why the model performs like this?'*. Analyzing the miss-classified images for different models is one of the examples of such usages since if the visualization method can offer a proper explanation for their predictions, it is possible to improve the modalities of the architecture, as well as various aspects of training and performance. Proposed method can act as a tool to aid analyzing such aspect. We exhibit this by analyzing a miss-classified image for object recognition purpose, as shown in Fig. 6. As we see from the figure that the image is wrongly classified by Alexnet, but classified correctly by VGG-16. If we see in detail, AlexNet actually look for totally different regions (i.e., backgrounds, tail of the fox) that actually leads the model towards wrong classification. In such way, we can analyze the model's performance in practice and may find out the rooms for improving different aspect of the model through the visual analysis.

5 Conclusion

In this work, we present a new visual interpretation method that sequentially selects the discernible neurons for each layer considering neighboring layers information, and gradually trace back to the input image to find the salient part contributing mostly to the classifier's prediction. Proposed approach calculates the visual map after the forward-pass and demonstrates better visualization against other existing methods. We conduct both the subjective and objective experiments to show the superiority of our method, and as well show the robustness of our method in presence of adversial noise. Moreover, in cases of miss-classifications, our approach can be beneficial to offer visual explanations to aid making the CNN more transparent, offering improvement scopes for various training and architecture aspect of the model.

References

Rajeev Agrawal. Sample mean based index policies by o (log n) regret for the multi-armed bandit problem. *Advances in Applied Probability*, 27(4):1054–1078, 1995.

Peter Auer, Nicolo Cesa-Bianchi, and Paul Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine learning*, 47(2-3):235–256, 2002.

Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PloS one*, 10(7):e0130140, 2015.

Aditya Chattopadhay, 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)*, pp. 839–847. IEEE, 2018.

Hisham Cholakkal, Jubin Johnson, and Deepu Rajan. Backtracking scspm image classifier for weakly supervised top-down saliency. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5278–5287, 2016.

Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2): 303–338, 2010.

Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Tobias Huber, Dominik Schiller, and Elisabeth André. Enhancing explainability of deep reinforcement learning through selective layer-wise relevance propagation. In *Joint German/Austrian Conference on Artificial Intelligence (Künstliche Intelligenz)*, pp. 188–202. Springer, 2019.

Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: a simple and accurate method to fool deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2574–2582, 2016.

Konda Reddy Mopuri, Utsav Garg, and R Venkatesh Babu. Cnn fixations: an unraveling approach to visualize the discriminative image regions. *IEEE Transactions on Image Processing*, 28(5): 2116–2125, 2018.

Marko Robnik-Šikonja and Igor Kononenko. Explaining classifications for individual instances. *IEEE Transactions on Knowledge and Data Engineering*, 20(5):589–600, 2008.

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*, 115(3):211–252, 2015.

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*, pp. 618–626, 2017.

Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2013.

Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, and Martin Riedmiller. Striving for simplicity: The all convolutional net. *arXiv preprint arXiv:1412.6806*, 2014.

Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1–9, 2015.

Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pp. 818–833. Springer, 2014.

Jianming Zhang, Sarah Adel Bargal, Zhe Lin, Jonathan Brandt, Xiaohui Shen, and Stan Sclaroff. Top-down neural attention by excitation backprop. *International Journal of Computer Vision*, 126 (10):1084–1102, 2018.

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*, pp. 2921–2929, 2016.

Luisa M Zintgraf, Taco S Cohen, Tameem Adel, and Max Welling. Visualizing deep neural network decisions: Prediction difference analysis. *arXiv preprint arXiv:1702.04595*, 2017.