Visual Interpretation of CNN Prediction Through Layerwise Sequential Selection of Discernible Neurons

MD. TAUHID BIN IQBAL1,2, (Member, IEEE), ABDUL MUQEET2, (Member, IEEE)
AND SUNG-HO BAE2, (Member, IEEE)

1Department of Computer Science and Engineering, Stamford University Bangladesh, Dhaka 1217, Bangladesh
2Department of Computer Science and Engineering, Kyung Hee University, Yongin 17104, South Korea
Corresponding author: Sung-Ho Bae (shbae@khu.ac.kr)

This work was supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF) by the Ministry of Science, Information and Communications Technology (ICT) and Future Planning under Grant 2018R1C1B3008159; and also supported in part by the Institute of Information and Communications Technology Planning and Evaluation (IITP) grant funded by the Korea Government (MSIT) (Artificial Intelligence Innovation Hub) under Grant 2021-0-02068.

ABSTRACT In recent years, researchers have been working to interpret the insights of deep neural networks in the pursuit of overcoming their opaqueness and so-called ‘black-box’ tag. In this work, we present a new posthoc visual interpretation technique that finds out discriminative image regions contributing highly towards networks’ prediction. We select most discernible set of neurons per layer and engineer the forward pass operation to gradually reach most discriminative image locations. While searching for discernible neurons, existing approaches either end up with low-resolution visualization maps, or suffer lack of neuron discriminativity in the way. Moreover, some methods concentrate only on current layer information overlooking meaningful information from adjacent layers, limiting the overall scope of selection. We address these issues by exploring the layer-to-layer connected structure of a neuron and obtaining contributions from its current layer along with its adjacent layers, e.g., succeeding and preceding layer. We introduce a score function where such contributions are assembled with appropriate priorities. Layerwise discernible neurons are then selected based on top scores, ensuring a reliable and credible selection. We validate our proposal through objective and subjective evaluations by examining its performance in terms of models’ faithfulness and human-trust, where we visualize its efficacy over other existing methods. We also perform sets of sanity check experiments on our method to show its overall reliability as a visualization map.

INDEX TERMS CNN prediction, DL interpretation, discernible neurons, explainable AI, layerwise selection, sanity checks.

I. INTRODUCTION
With the rise of unprecedented performance of deep learning methods in various computer vision tasks over last few years, network architectures also get complex to preserve diverse variations [1], [2]. Despite offering higher performance, these complex architectures provide less explanation of the question: ‘how it actually works?’. As a result, deep learning models have been tagged as black-box models, raising the necessity of eradicating its opaqueness while being more understandable and transparent for general use [3], [4].

To understand the inner representations of deep networks, it is important to see how and what a network learns in practice. One possible way is to look for salient image regions that contribute most for the networks’ prediction. Thus, we know how a network is making its prediction, as well as get an idea which portion of input data is guiding the network towards a misprediction. Such regions, in fact, can act as the visual explanations for a predicted label exhibiting class-specific patterns learned by a model. In this way, these posthoc visualizations aid interpreting the insights of CNN’s prediction.

A. PREVIOUS WORKS
The major works on this area utilize gradient information to visualize salient regions [5]–[13]; however, according to
the related comparative discussions at [14]–[21], we broadly divide the existing works into following categories.

1) PERTURBATION-BASED APPROACH
These approaches mainly observe the network behavior after perturbing specific input or neurons, aiming to understand its influence on later input or neurons. For example, Zeiler and Fergus [6] visualized the change in activations (of later layers) by occluding different parts of the input image. Zintgraf et al. [22] also analyzed the change in the prediction after marginalizing over each input patch. These approaches, however, often become practically inefficient due to the additional perturbation strategies that usually need different forward propagation through the networks. In this line of work, we also observe few works focusing on learning interpretable models on the model predictions [23], or perturbing the inputs to see the reaction at blackbox model [24], or both [25].

2) BACKPROPAGATION-BASED APPROACH
Unlike the above methods, backpropagation-based approaches propagate discernible signals from output node to input data through the layers in a single pass. The dominant group of works in this category are, in fact, gradient-based. For instance, Simonyan et al. [5] compute the sensitivity of classification score in terms of partial derivative of the classification score for a given class with respect to the changes in pixel values. Deconvolution-based works [6] take a similar approach to visualize salient feature concepts across different layers. Nonetheless, lack of neuron-discriminativity in these methods may affect the visualization, as pointed in [26]. Guided backprop [7] utilizes the gradient by modifying them for a better qualitative visual representation. Such gradient-based visualization maps often suffer from noise due to local variations in partial derivatives. Smilkov et al. [10] addresses this issue by SmoothGrad that smooths the gradients with a Gaussian kernel. Other works [11], [27] come up with gradient × input strategy, which is basically an element-wise product between the visualization maps (e.g., [5]) and the input. Gradients are also used after integrating them with respect to the series of interpolated images that attribute the prediction of a deep network to its inputs [12].

3) CAM BASED APPROACH
The works done based on the popular Class-specific Activation Map (CAM) [8], [28] concept are categorized in this group. CAM generates salient feature-maps by combining intermediate feature maps before global average pooling layer. Although this technique provides better flexibility than the prior approaches in terms of interpreting the prediction, its architecture-specific design limits its adaption scope to different architectures and applications, as also pointed in [29]. Improvements are, however, done over this method utilizing gradient information and named as Grad-CAM [9]. Selvaraju et al. [9] also perform element-wise product between the scores from Grad-CAM and Guided-backprop, aiming to come up with more fine-grained visualization, which they name as Guided Grad-CAM. Nevertheless, this strategy possesses the issues of Guided backprop caused by zeroing-out negative gradients during the back-propagation, as pointed in [11]. Moreover, Grad-CAM still uses low-resolution maps which often limits the performance of practical applications, e.g., weakly-supervised tasks [30]. Another line of works in this category are gradient-free CAMs [31]–[33], which basically capture the importance of each activation map by the target score in forward propagation. However, these methods are time-consuming and suffer parameter-dependency (e.g., Gaussian noise parameter). Use of ReLU non-linearity may lead some of these maps (e.g., [31]) to be noise-prone as well. As a result, concentration of the maps often spread-out, and create ambiguity while interpreting the miss-classified samples, as pointed in [32].

4) STRUCTURE-BASED APPROACH
We also observe group of works [22], [34]–[37] taking relevance score for every 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 feature while small changes do not affect the decision. Some other approaches [35] take probabilistic approaches to find contribution of each image patch (or pixel) to the classification detailing their understanding. Zhang et al. [38] compute marginal winning probabilities for neurons at each layer, where distinct attention map is computed as the sum of these probabilities across the feature maps. Huber et al. [39] utilizes only the most relevant neurons of each convolution layer and thus generates more selective saliency maps.

Mopuri et al. [29], in the same line, proposes CNN-Fixations that selects important neurons to trace salient image region in order to interpret the prediction. One important part of [29] is to select salient neurons at each layer, where authors suggest a naive approach of such selection by considering the neurons with high activations. During selection, this approach only concentrates on information from target layer, e.g., activations, and thus overlooks available meaningful information from other neighboring layers.

B. OUR MOTIVATION AND PROPOSAL
As discussed, structure-based approaches select and utilize discriminative neurons per layer to search for salient
image locations, and so far have shown their efficacy. We also observe in the former discussion that lack of neuron discriminativity affect visualization in some backpropagation-based methods. Hence, motivated by the importance of selecting discriminative neurons in a good visualization, we concentrate on coming up with a reliable selection strategy of such neurons per layer which will be sequentially propagated towards input layer to find most discriminative image locations. However, while selecting these neurons, some approaches, e.g., [29] only rely on current layer activations, ignoring other available significant information from adjacent layers which limits the overall scope and reliability of selection. Observed by the layer-to-layer connected structure of network where information at one layer contributes heavily to the outputs of its next layer, we infer that impact of a neuron can be better understood analyzing its influence on current and neighboring adjacent layers. We explore this layerwise connected structure of neuron with the notion that information from current layer together with information from its adjacent layers simultaneously contribute towards selecting a neuron as an important one. We also get motivation from one recent work on network pruning [40] that explores such layerwise structure, where authors prune a specific layer using weight tensors from adjacent layers along with its own layer-weights exhibiting superior results. However, we incorporate the idea of using adjacent layer information carefully, where we assemble each of their contributions with current layer’s contribution in a prioritized way. Contributions from all these layers are the key in gaining confidence for a neuron to be selected as an important one; nonetheless, priorities are also added to put appropriate attention to each of them. For a traditional neural net, the above way reliably ensures the selection of important neurons per layer, starting from last fully-connected (fc) layer to initial input data. As soon as we reach the input image, we generate visualization map by Gaussian blurring the salient neurons, i.e., image locations. One sample visualization of our method is provided in Fig. 1. Our contributions can be summarized as follows,

- We explore the layer-to-layer connected structure of a neuron to understand its impact through the contributions from its current and adjacent layers.
- We introduce a score-function to assemble the layerwise contributions with corresponding priorities, where a reliable way of selecting such priorities is also presented. Most impactful neurons are selected per layer based on top scores, and gradually propagated towards input data to locate and highlight desired discriminative regions.
- We conduct extensive experiments to examine the efficacy of proposed technique against other existing methods through objective and subjective evaluations. We also conduct sets of sanity check experiments to show its overall reliability as a visualization map.

II. METHODOLOGY

Our key purpose is to visually explain a CNN prediction by highlighting most important image pixels responsible for the prediction. To search for these pixels, we trace back corresponding prediction sequentially from last layer to input data by finding layerwise most discriminative neurons so that important neurons (pixels) of input image are identified.

In practice, our method takes a trained neural network, where we pass a test image to make prediction which is then traced back from final softmax layers. Neurons with top n-probabilities are considered as the neurons of interest at this layer, for which we try to find a set of neurons from its prior layer (e.g., last fc layer) contributing most to activate them. In this scenario, last fc layer is considered as our current (target) layer, softmax layer is the succeeding layer, and second-last fc layer (if exists, otherwise available conv/pool layer) is considered as the preceding layer (such a scenario is depicted in Fig. 2). That is, for a set of neurons at succeeding layer, we search for the neurons of current layer contributing most towards activating succeeding layers’ neurons of interest. The contribution of a neuron is defined through a proposed score function formulated from collective information from its current and adjacent layers. Important neurons at each layer are then carefully selected by exploring the score values. We iterate this selection process from softmax layer till input data to determine most discriminative image pixels, which are then highlighted to aid for interpretation purpose.

The proposal of contribution-score, mentioned above, is the key highlight of our method, through which we assess a neurons’ impact at a layer. For a neuron, its activation is typically considered as the key measure to estimate its impact [29]. We ask couple of questions in this regard: ‘why the activation gets high/low?’ and ‘how a specific activation contributes next?’. Regarding the former question, we observe that strong connections from preceding layer having high activation and weight basically cause current activation to be higher. We find this an important cue to analyze the generation of an impactful activation while having an essence how it may influence later. For the later question, we observe that a neuron with high activation may contribute firing multiple succeeding neurons, which also provides a discernible cue to estimate its impact. Thus, a comprehensive understanding on this issue can be gained exploring these aspects. Hence, while generating contribution-score for a neuron, we accumulate individual scores from its adjacent preceding and succeeding layers analyzing above aspects, and carefully aggregate them with current layer score.

A. FORMULATION

From now on, we denote current layer as $c$, and its adjacent layers, e.g., preceding and succeeding layer as $p$ and $s$-layer, respectively. A neuron from these layers will be denoted by putting a [ ] at the top as $\hat{c}$, $\hat{p}$, $\hat{s}$; and a selected discriminative neuron will be noted by putting an * at the top (e.g., $\hat{c}^*$, $\hat{p}^*$, $\hat{s}^*$). Now, lets assume there is a target layer $c$ for which we like to generate desired set of $\hat{c}^*$, and we have a number of selected $\hat{s}^*$ neurons from layer $s$. For each $\hat{s}^*$, we first calculate contribution-scores for all $\hat{c}$, and then, exploit these scores to get the set of $\hat{c}^*$ for each respective $\hat{s}^*$. 


Here, we discuss how we generate score for a neuron $c$. We describe this process elaborately in the following part of this subsection.

1) GENERATING CONTRIBUTION-SCORE OF NEURONS

Here, we discuss how we generate score for a neuron $\hat{c}$, for a selected $\hat{s}$. For this, we obtain individual contributions from its own layer $c$ along with its adjacent layers $p$ and $s$ within respective score-functions. At first, we derive a score for layer $c$ itself. For this, we see the strength of the connection between $\hat{c}$ and $\hat{s}$ since a stronger connection shows a higher impact of $\hat{c}$ on $\hat{s}$. We define this by multiplying its activation with connection-weight as,

$$Z_c = A_{\hat{c}} W_{\hat{c}}^s, \quad (1)$$

where, $Z_c$ is the score for layer $c$, $A_{\hat{c}}$ is the activation value of $\hat{c}$, and $W_{\hat{c}}^s$ is the connection weight between $\hat{c}$ and $\hat{s}$.

To obtain contribution from layer $p$, we scrutinize the impact of preceding neurons on $\hat{c}$’s activation in order to get an essence of its generation, strength and probable impact. Now, in a forward pass, signal is transferred from prior to later layer after a weighted sum of the activations of all prior neurons. Although there exists contributions from all neurons, all of them may not contribute significantly due to either low activations or low connection-weights. Hence, to understand the key point of the generation of $A_{\hat{c}}$, we only consider contributions from the most impactful neuron of prior layer. One may consider multiple such neurons; nonetheless, for simplicity, we stick to the most impactful one. This is because, a target $\hat{c}$ with most impactful $\hat{p}$ of high activation & connection-weight is likely to influence more than a $\hat{c}$ with such $\hat{p}$ of moderate/low activation & connection-weight. Thus, obtaining information of prior connection in the above way gives us an essence on the generation and strength of $A_{\hat{c}}$, and brings more explainability on the score function. To generate the score, we first observe individual connection-strengths of all $\hat{p}$ by multiplying their activations to the corresponding connection-weights (associated to target $\hat{c}$), and select the highest value among them as,

$$Z_p = \arg \max \{ A_{\hat{c}} W_{\hat{c}}^{\hat{s}} : \forall \hat{p} \}, \quad (2)$$

where, $Z_p$ is the score for layer $p$; $\arg \max(F)$ function returns highest value from operation $F$; for a $\hat{p}$, $A_{\hat{p}}$ denotes its activation value, and $W_{\hat{p}}^{\hat{s}}$ is the connection-weight between $\hat{p}$ and $\hat{c}$.

Lastly, we generate score for layer $s$ by analyzing the influence of target $\hat{c}$ on activating other neurons of succeeding layer. We observe this influence by calculating the number of times $\hat{c}$ possesses highest contribution in terms of $Z_c$ scores for all respective $\hat{s}$. By contributing to activate more neurons of later layer, a target $\hat{c}$ explicitly shows its impact at that layer, and becomes a strong candidate to be picked. Hence, we calculate the number of times $\hat{c}$ contributes most for $\hat{s}$ neurons, and consider this value as the score for layer $s$. We define it as,

$$Z_s = \sum_{\forall \hat{c}} G(\hat{c}, \hat{s}), \quad \text{where,}$$

$$G(\hat{c}, \hat{s}) = \begin{cases} 1, & \text{if } \arg \max \{ A_{\hat{c}} W_{\hat{c}}^{\hat{s}} : \forall \hat{c} \}, \\ 0, & \text{otherwise,} \end{cases} \quad (3)$$

where, $Z_s$ is the score for layer $s$; function $G(\cdot)$ takes input target $\hat{c}$ & one selected $\hat{s}$, and returns 1 if target $\hat{c}$ has the highest contributions for $\hat{s}$ (in terms of multiplication of its activation $A_{\hat{c}}$ and connected-weight $W_{\hat{c}}^{\hat{s}}$) among other existing $\hat{c}$. We iterate this process for all selected $\hat{s}$ through a counter, and assign final value to $Z_s$. A lower default value is assigned to $Z_s$ in case no such contribution is found.

After we obtain individual scores from all the layers, we aggregate them to generate final score function. There are multiple ways to aggregate them; nonetheless, we opt for a weighted sum approach. The ‘weight’ here denote the priorities or importance we want to give to a layers’ score.
Algorithm 1: Selection of Discriminative Set of Neurons for a Layer

**Input:** Discriminative neurons from succeeding layer (set of $\hat{s}^*$), Layer-wise pre-learned priorities ($\alpha_c$, $\alpha_s$, $\alpha_p$).

**Output:** Discriminative set of neurons for layer $c$, $\mathbb{H}(\hat{c}^*)$

1. for each neuron $\hat{s}^*$ do
2. for each neuron $\hat{c}$ do
3. Calculate score $\Omega_\gamma(\hat{c}) = \alpha_c Z_c(\alpha_s Z_s + \alpha_p Z_p)$, where $Z_c$, $Z_s$, and $Z_p$ are the calculated contribution-scores from succeeding, current and previous layers, respectively (details are in section II-A)
4. end for
5. Select $k$-discriminative neurons with highest scores, and define this set as, $\mathbb{H}_\gamma(\hat{c}^*) = \arg\max_k \{\Omega_\gamma(\hat{c}) : \forall \hat{c}\}$
6. end for
7. Select the final set of discriminative neurons for layer $c$ by taking union of all $\mathbb{H}_\gamma(\hat{c}^*)$ as, $\mathbb{H}(\hat{c}^*) = \bigcup_{\gamma} \mathbb{H}_\gamma(\hat{c}^*)$.
8. return $\mathbb{H}(\hat{c}^*)$

Now, note that due to the disparate type of contributions from the layers, we need to aggregate them carefully. Specifically, $Z_c$ and $Z_p$ are devised from the corresponding activation values, and hence we can sum them easily. Besides, $Z_c$ is a numerical measurement exhibiting number of times $\hat{c}$ contributes most for selected $\hat{s}^*$ neurons. Hence, we use $Z_c$ as a multiplier to the weighted sum of $Z_c$ and $Z_p$ in order to amplify the importance of target $\hat{c}$. Formally, we devise it as,

$$\Omega_\gamma(\hat{c}) = \alpha_c Z_c(\alpha_s Z_s + \alpha_p Z_p), \quad (4)$$

where, $\Omega_\gamma(\hat{c})$ denote the final score of $\hat{c}$ for a selected $\hat{s}^*$; $\alpha$ with subscript $s$, $c$, $p$ denote priorities for $s$, $c$, and $p$ layer, respectively. Values of $\alpha$ are learned beforehand, as discussed in Section II-B2.

2) SELECTION OF DISCERNIBLE NEURONS

For a selected $\hat{s}^*$, we calculate score (Eq. 4) for all $\hat{c}$ in the above way, and select most influential $k$-number of $\hat{c}$ based on their highest score values. We define this set of selected neurons as,

$$H_\gamma(\hat{c}^*) = \arg\max_k \{\Omega_\gamma(\hat{c}) : \forall \hat{c}\}, \quad (5)$$

where, $\arg\max$ operator returns $k$-top neurons based on the highest $\Omega$ values; $H_\gamma(\hat{c}^*)$ is the selected set of $\hat{c}^*$ for a particular $\hat{s}^*$.

In this way, we obtain respective sets for all $\hat{s}^*$. We now take the union of all these sets, and remove repetition of same neurons to generate desired final set of discriminative neurons for $c$ layer. We define it as,

$$H(\hat{c}^*) = \bigcup_{\gamma} H_\gamma(\hat{c}^*), \quad (6)$$

where, $\bigcup$ operation combines neurons from all $\mathbb{H}_\gamma(\hat{c}^*)$ sets into the final set, $\mathbb{H}(\hat{c}^*)$, after removing repetition of same neurons.

In the above way, we sequentially select the set of most discernible neurons at each layer. Finally, we reach to image and obtain most contributing neurons of it (i.e., image pixels). As soon as we reach to these pixels, we generate visualization map by Gaussian blurring the pixel locations, as depicted in Fig. 1. The Gaussian parameters are found empirically.

B. DISCUSSION ON METHOD-ATTRIBUTES

In this part, we elaborate the descriptions of different method-attributes, such as, selection process for conv/pooling layer; selection of the priority values (denoted in Eq. 4); and visualization of different method variants.

1) SELECTION PROCESS IN CONVOLUTION/POOLING LAYER

In conv-layer, pixel within a receptive field is convolved with kernel weights to generate activation in the same spatial location at next layer. One $\hat{s}^*$ in conv-layer is basically a pixel location in one of the channels of $s$ layer (as denoted by purple colored pixel in Fig. 2). For such a $\hat{s}^*$, we find desired set of neurons $\hat{c}^*$ (pixel locations) by analyzing the locations at same $(x, y)$ spatial position among all the channels of layer $c$. That is, scores are calculated along the channels of layer $c$ at the same $(x, y)$ locations those in fact aid activating $\hat{s}^*$. Lastly, most contributing locations are selected. This selection is done using similar procedure as mentioned above in Section II-A, except some modifications in the score calculations due to the convolution process which we elaborate here.

Among the scores, $Z_c$ is calculated through the multiplication of activations of neuron $\hat{c}$ and respective weights between $\hat{c}$ and $\hat{s}^*$. In conv-layer, we regard activations of a location $\hat{c}$ through its surrounding pixel values within a receptive field, as depicted in conv-layer part in Fig. 2, and weights are its associated kernel weights. Both the activations and weights appear as 3D-blobs having spatial locations $(x, y)$ along with channel information. To replicate the multiplication, we compute hadamard product between them, which also results in a 3D-blob of same size as the activation blob. For each channel in this output blob, we sum up respective values across $(x, y)$ spatial region to process each of them with a single numeric value. These values are basically the respective $Z_c$ values for candidate locations. Note that $Z_p$ for such a candidate location, $\hat{c}$, is calculated in the similar way, but for the layer $p$, while picking the maximum value from the channels. Besides, $Z_s$ is simply the numeric count to assess the effect of $\hat{c}$ in later layer, hence we do not have to bring any modification for this. Other procedures, such as the selection of most contributing locations are same as detailed in Section II-A2.

In pooling layer, an activation in layer $s$ is basically the maximum value present in the corresponding receptive field in the layer $c$. Hence, while backtracking an activation across

81992

VOLUME 10, 2022
such a layer, we locate the maximum activation of layer $c$ within its receptive field. That is, for pooling layers, we first extract 2D receptive fields of the corresponding neuron in layer $c$, and then find the location having highest activation. We choose to use maximum activation 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 given in Fig. 2, and the general algorithm is provided in Alg. II-A1.

### 2) SELECTION OF PRIORITY-VALUES

In this part, we discuss the optimal selection of the priority values ($\alpha$) that are multiplied with each $Z$-scores in the score-function (Eq. 4). Assigning such layerwise priorities is important since different layers contribute differently, and hence priorities favor us providing appropriate importance to each layers’ information while bringing more generality to the score function (Eq. 4). Note that among the $Z$ scores, $Z_c$ is a multiplier that amplifies a neuron’s chance to be selected as an important one. Putting $\alpha_c$ also as a multiplier (to $Z_c$) do not contribute much in this context. For this reason, we simply omit $\alpha_c$ from the equation.

Now, there can be multiple ways of selecting the optimal values of $\alpha_p$ and $\alpha_c$. One way is to search for their best experimental outcomes in a sample validation-set based on an evaluation metric, such as IoU (Intersection over Union) score. To generate IoU score, bounding box is first generated in the predicted image (using visualization map is one way of generating such box), and it is compared with ground truth bounding box in terms of a ratio-value between their overlap area and the unified area. This strategy shows how accurately model identifies class-specific object regions by rewarding predicted bounding boxes for heavily overlapping with the ground-truth. Our idea is that a map with good parameter-choice would successfully trace back the discriminant locations in test image, and thus significantly aids the generation of accurate bounding box and respective IoU score. Hence, we first generate maps for different $\alpha_p$ & $\alpha_c$ values, and then calculate IoU scores using the bounding boxes generated from each of such maps. We now find out the optimal combination of priority values providing best IoU scores.

For the experiment, we randomly select 1000 images from ImageNet Validation dataset [41] and calculate average IoU (Intersection over Union) scores for them using our method for a range of pre-defined $\alpha_p$ and $\alpha_c$ values ([0.1 ~ 1.0]). For a test image, after reaching its discriminative regions using proposed method, we generate bounding box around the target object simply by tracking the largest connecting components. However, the average IoU scores for different $\alpha_p$ and $\alpha_c$ are given in Fig. 3. We observe that IoU scores start increasing as we increase $\alpha_c$ value and IoU scores decrease as we decrease $\alpha_c$. For $\alpha_p$, we observe the opposite case; that is, as we increase $\alpha_p$ value, IoU decreases, and vice-verse. These results lead us to the fact that performance depends a lot on the correct combination of priority values. In fact, the linear relationship between $\alpha_c$ and IoU scores indicates that performance mainly depends on the contribution from current layer while being partially dependent on the contribution from proceeding layer. The highest score is obtained at $\alpha_c = 0.9$ and $\alpha_p = 0.3$, which also supports the prior observation. Hence, we select these values as the optimal values for $\alpha_c$ and $\alpha_p$.

Note that we keep the priority-values same for all the layers from top to bottom. Considering all these layers producing diverse information, one may prioritize them differently, e.g., using different priority values for top, middle and bottom layers. One may also consider deriving priority values for other datasets to see whether those values differ for other images. We leave these issues here as potential future research direction.

![Figure 3: Average IoU scores (normalized) for different $\alpha_c$ and $\alpha_p$ priority-values generated using 1000 randomly selected ImageNet samples. Both the values range within 0.1 ~ 1.0, and denoted by subscript c and p respectively.](image-url)
3) METHOD-VARIANTS

To this part, we explain how the inclusion of adjacent layer information benefits proposed visualization. Note that in prior discussions (Section I-B and preambles of Section II), we discuss primary motivation and idea of including such information. Here, we explicitly illustrate how such information aids current layer information generating accurate, credible and aesthetic visualization. For this, we assess visualization maps with and without adjacent layers’ information after modifying the score function (Eq. 4). We modify it for different cases as,

- Case A: with layer $c$ only
  \[ \hat{\mathbf{s}}(\tilde{c}) = \alpha_c \mathbf{Z}_c \]

- Case B: with layer $c$ and layer $p$
  \[ \hat{\mathbf{s}}(\tilde{c}) = \alpha_c \mathbf{Z}_c + \alpha_p \mathbf{Z}_p \]

- Case C: with layer $c$ and layer $s$
  \[ \hat{\mathbf{s}}(\tilde{c}) = \alpha_s \mathbf{Z}_s \alpha_c \mathbf{Z}_c \]

- Case D (proposed): with layer $c$ along with both layer $p$ and $s$
  \[ \hat{\mathbf{s}}(\tilde{c}) = \alpha_s \mathbf{Z}_s (\alpha_c \mathbf{Z}_c + \alpha_p \mathbf{Z}_p) \]

- Case E: without layer $c$
  \[ \hat{\mathbf{s}}(\tilde{c}) = \alpha_s \mathbf{Z}_s \alpha_p \mathbf{Z}_p \]

We provide visual maps for each of these cases for different objects in Fig. 4. We observe, compared to Case A where we use layer $c$ information only, target objects are covered more accurately as we add information from layer $p$ (Case B) and layer $s$ (Case C), respectively. In case of using both these layers along with layer $c$ (Case D), maps produce visually most aesthetic and accurate maps than other given cases. We also derive Case E where we do not use layer $c$, rather use information from layer $s$ and $p$ only. Despite neglecting information from the most important layer, objects are still covered through a somewhat scattered visualization in this case.

This is a key case in our proposal showing that adjacent layers possess meaningful knowledge of current activation, and thus, can significantly aid current layers’ information in our gradual search for important neurons.

III. EXPERIMENTAL RESULTS

One such visualization work must show its consistency against model’s prediction (faithfulness) while gaining human trust. In this regard, we conduct objective and subjective evaluations to compare our method against other existing methods in terms of faithfulness and human trust. We also conduct some sanity checks (e.g., model randomization test and class-sensitivity analysis) to demonstrate its overall reliability as a visualization map.

A. OBJECTIVE EVALUATION

For objective evaluation, we evaluate faithfulness of our method with respect to the model prediction. This can be understood analyzing a maps’ ability to accurately explain the function learned by a model, as pointed in [9] and [43]. Ribeiro et al. [25] also suggested the explanations to be locally accurate in the vicinity of predicted object in order to be faithful to the model. It would be interesting to see how tweaking the explanation map region would question the locality of class-object and influence the prediction confidence. This analysis can be a significant cue to understand consistency of a map to the model, which we aim to observe in this part. In this regard, we evaluate the performance of our
method against others on two different metrics: (a) average drop % and (b) % increase in confidence- each of which is described below.

1) AVERAGE DROP %
In the first experiment, we compare the prediction confidence of an input image against the confidence of its corresponding explanation map when the map is used as input. A good map always highlights the respective class-specific parts, and thus it would keep most important such regions intact in that image. Hence, when an explanation map is used as input, most relevant object parts are expected to be occluded to some extent. Since a network mainly searches important object patterns to make a decision, occluding such parts would mostly lower the confidence of the model as compared to the confidence of original image for the particular class. We expect this drop in confidence to be sufficiently low in practice. We define the metric (%drop) as the ratio of confidence score between explanation map and original input image, and average result is reported for the samples under our considerations. A lower value, in this regard, indicates better consistency of the method.

The idea of this experiment comes from [43], where the results are generated using PASCAL VOC 2007 [44] and ImageNet [41] validation-images. Similar to [43], 2510 validation images of PASCAL VOC 2007 dataset have been used for the experiment. In ImageNet, we randomly select three subset of 2510 validation images, and average out the results in order to maintain consistency with the PASCAL VOC 2007 setting. VGG-16 is chosen to generate results. In ImageNet, we take the implement of [45] to generate visualization maps for Grad-CAM, Integrated Gradients, Gradient × Input, and Guided backprop. Others are generated using the given code repositories of corresponding papers. Visualization maps of all these methods for some sample-objects are also provided in Fig. 5 in order to compare them qualitatively.

Note that for this experiment, we use the implementation of [45] to generate visualization maps for Grad-CAM, Integrated Gradients, Gradient × Input, and Guided backprop. Others are generated using the given code repositories of corresponding papers. Visualization maps of all these methods for some sample-objects are also provided in Fig. 5 in order to compare them qualitatively.

2) % INCREASE IN CONFIDENCE
Since CNN mainly looks at most relevant object region for respective prediction, at times the entirety of such region could be highlighted by the map. In this case, providing only the explanation map region as input would increase the confidence in prediction compared to the confidence of original image due to ignoring insignificant image parts. We measure this increase in confidence by the number of times such improvement observed for the particular class while providing only explanation map regions as input. We report the result as percentage in Table 2. For this experiment, we keep the settings same as prior experiments, and generate results for other methods under our considerations. Results in Table 2 shows that proposed approach gains boost in confidence for the most time than other methods in both the datasets, which again shows its efficacy in discriminative localization of target object.

Superior results in above experiments demonstrate that proposed method shows better confidence in generating faithful explanations than other methods demonstrating its consistency with the models’ prediction. In both experiments, CNN-Fixations achieves closer performance to the ours while variants of CAM-based methods, e.g., Grad-CAM++ and Guided Grad-CAM also achieve competitive results.

B. SUBJECTIVE EVALUATION
So far, we have explored the faithfulness property of our method in above experiments. We now conduct couple of subjective evaluation experiments to analyze its interpretability in terms of class-discrimination and human-trust.

1) CLASS-DISCRIMINATION EVALUATION
In this experiment, we measure visual discriminativity of visualization methods in representing predicted object by

### TABLE 1. Comparative objective evaluation results in terms of average drop% for different visualization maps on PASCAL VOC 2007 and ImageNet dataset. Results with lower values are better in this table. Results with [*] are from [43].

| Methods            | PASCAL VOC 2007 | ImageNet |
|--------------------|-----------------|----------|
| Integrated Gradients | 69.67          | 52.98    |
| Gradient × Input    | 68.40          | 52.03    |
| Guided Back-propagation | 65.81       | 48.78    |
| Grad-CAM           | 61.22*         | 46.43*   |
| Guided Grad-CAM     | 56.55          | 45.12    |
| Grad-CAM++         | 57.02*         | 43.31*   |
| CNN Fixations       | 47.75          | 36.55    |
| Proposed           | 45.98          | 34.14    |

### TABLE 2. Comparative objective evaluation results in terms of % increase in confidence for different maps on PASCAL VOC 2007 and ImageNet dataset. Results with higher values are better in this case. Results with [*] are from [43].

| Methods            | PASCAL VOC 2007 | ImageNet |
|--------------------|-----------------|----------|
| Integrated Gradients | 8.92           | 9.95     |
| Gradient × Input    | 9.88           | 10.22    |
| Guided Back-propagation | 10.49        | 11.41    |
| Grad-CAM           | 13.63*         | 13.67*   |
| Guided Grad-CAM     | 15.24          | 13.98    |
| Grad-CAM++         | 14.98*         | 15.44*   |
| CNN Fixations       | 19.20          | 20.67    |
| Proposed           | 21.63          | 22.06    |
examine which of them best represent the class-specific features from human visual perspective. For the experiment, we select 10 different classes from ImageNet validation set, and for each class we randomly select 5 different images. For the selected images, we generate maps for different visualization methods with VGG-16 network. We now provide these maps to 20 attendees along with their original input-image and corresponding class-label after hiding the methods’ identity, and ask them ‘Which map best describes the class-object present in the input image?’ In this way, we aim to obtain human perspective about the best visualization map exhibiting class-specific object properly while providing an aesthetic visualization.

To keep this evaluation simple for the attendees, we decide to provide them a limited set of maps, instead of providing many of them as used in Section III-A. Observed by the dominant performance from the prior objective evaluation, we consider Grad-CAM++, Guided Grad-CAM, CNN-Fixations and the proposed one, for this test. That is, for each given image, we provide 4-visualization maps to each attendee along with original image and corresponding class label. Attendees are requested to answer through a google-form, where the answers were given in a checkbox manner. If an attendee chooses single option (visualization map) for a question, score 1 is added to that option. In case of multiple answers provided, the scores are normalized to 1. Thus, for each question, we obtain normalized scores for the given options. These scores are then added, ending up to a total achievable score of 50. As we observe the final scores given at Table 3, proposed method achieves a score of 26.41, which is by far better than other methods, e.g., Guided Grad-CAM (7.00), Grad-CAM++ (7.42) and CNN-Fixations (9.17). Hence, according to these human evaluations, our method seems to provide most aesthetic and class-specific visualization maps than other given methods.

2) HUMAN TRUST EVALUATION

In this experiment, we examine if there are multiple maps for different networks produced by a visualization method, which map would seem more trustworthy in human perspective. Basically, we examine whether this human perception (about a dominating map of a network among others) complies with precalculated machine accuracy of respective network; and should they comply each other, it would help place trust in the explanation. We compare our visual maps produced for AlexNet and VGG-16 with the knowledge that VGG-16 shows better accuracy than AlexNet on image classification tasks [9], [46]. To check whether proposed visualization comply with this result, we select 58 ImageNet samples, where both networks make correct prediction, and then we generate maps for them using our method. For each image, we provide corresponding maps for AlexNet and VGG-16 to 12 attendees, and ask them ‘Which map best describes the object present in the image?’ Surprisingly, all attendees voted for VGG-16 as producing a better map than AlexNet. We also ask them to rate the reliability for both maps by requesting them to provide score between 1 ~ 5 (worst to best) within radio-button option. Obtained scores are then normalized, where we found VGG-16 (38.08) has a higher

### TABLE 3. Class-discrimination evaluation: given scores are generated from 20 attendee’s answer to the question ‘Which map best describes the class-object present in the input image?’. 5 different ImageNet samples from 10 different classes are used for this experiment.

| Methods         | Guided Grad-CAM | Grad-CAM++ | CNN-Fixations | Proposed |
|-----------------|-----------------|------------|---------------|----------|
| Scores          | 7.00            | 7.42       | 9.17          | 26.41    |
score than AlexNet (36.07). The result complies with the previous finding that VGG-16 shows superior performance against AlexNet for object classification. In this way, based on the consistent explanation from human prediction, our visualization method can help users place trust in a models’ prediction.

Superior results in above experiments show that proposed method enables discriminative visualization of predicted classes (class-discrimination) and helps picking more reliable models (human-trust) in practice. Note that in the Appendix of this paper, we provide screenshots of sample questionnaire for both the above-mentioned subjective evaluation experiments.

C. SANITY CHECKS

Adebayo et al. [47] points out that visualization methods may not always show its dependency towards corresponding model and data. Hence, sole reliance on visual inspection can often be misleading. To assess the reliability of a method, authors stress to conduct sanity checks before deploying it in practice. Following the recommendation, we conduct couple of sanity check experiments on our method, which we describe in this part.

1) MODEL RANDOMIZATION TEST

In this test, we check sensitivity of our visualization map against model properties, e.g., model parameters. The assumption is that if a visualization method depends on the learned parameters of a model, we should expect a substantially different map in case the model parameter changes. If the output remains similar, this can be regarded as insensitive to model properties, and hence can be inferred as unreliable for interpretation.

To conduct this test, we compare visual map of a target image for a trained network against its map for the same network but having randomly initialized parameters (untrained). For this, we randomize network-weights in a cascading fashion; that is, starting randomizing from top layer and continue up to the bottom layer. While randomization up to a specific layer, we check the difference of corresponding map against original map. We first calculate this difference in terms of a popular correlation measure, i.e., Spearman’s rank correlation [48]. Spearman’s rank correlation (denoted as $\rho$ or $r_s$ in the literature) measures the strength of association between two ranked variables. A $r_s$ value varies between $-1$ and $+1$, where a value of $\pm 1$ indicates a perfect degree of either positive or negative association between two variables. In this
FIGURE 7. Visualization maps for AlexNet after randomizing the layer-parameters in a cascading-style.

(a) Input (junco) and its AlexNet visualization map.

Cascading randomization of parameters $\rightarrow$

fc8    fc7    fc6    conv5

conv4    conv3    conv2    conv1

(b) AlexNet maps after randomizing layer parameters.

FIGURE 8. Visualization maps for VGG-16 after randomizing the layer-parameters in a cascading-style.

(a) Input (junco) and its VGG-16 visualization map.

Cascading randomization of parameters $\rightarrow$

fc8    fc7    fc6    conv5_3    conv5_2

conv5_1    conv4_3    conv4_2    conv4_1    conv3_3

conv3_2    conv3_1    conv2_2    conv2_1    conv1_2

(b) VGG-16 maps after randomizing layer parameters.

FIGURE 9. Visualization maps for some input images and their adversarial pairs. Adversarial pairs are generated for VGG-16 using Deepfool [52]. Corresponding labels are given at the top of each image.

Using proposed method, Spearman’s correlation values are generated for 1000 randomly selected ImageNet samples, and layerwise (i.e., randomization from top layer to target layer) average results for AlexNet and VGG-16 are shown at Fig. 6a and Fig. 6b, respectively. As evident from the figures, while continuing weight-randomization for both networks, we observe gradually decreasing correlation values, indicating substantial dissimilarities in respective maps. Now, for visual inspection, we provide qualitative results for a sample image (junco) for both AlexNet (Fig. 7) and VGG-16 (Fig. 8). Here, we explicitly observe how visualization get scattered as we advance from top to bottom layers after destroying their learned weights.

For a comparative evaluation, we have also generated results within same experimental settings for other existing visualization methods, including Integrated Gradients, Gradient $\times$ Input, Guided Backprop, Guided Grad-CAM, Grad-CAM, Grad-CAM++ and CNN-Fixations. Results are provided in Fig. 6a and Fig. 6b, respectively. We observe that Guided Grad-CAM and Guided Back-propagation show insensitivity to higher layer weights although they get better as they move towards lower layers. Gradient $\times$ Input and Integrated Gradients show consistently higher $r_s$-values, indicating correspondence between original map and distorted map throughout the way. On the contrary, Grad-CAM, Grad-CAM++ and CNN-Fixations show low $r_s$ values than the above-mentioned methods. Note that similar behavior is also observed for the respective methods in the sanity checks conducted in [47]. However, proposed method achieves lowest $r_s$ values in all the distorted layers, demonstrating its strong sensitivity towards the model compared to other methods.
To assess the consistency of above findings, we use another correlation measure, e.g., Structural Similarity Index (known as SSIM) [49] to observe the difference of the maps. SSIM actually measures the perceptual difference between two images. The SSIM values ranges between 0 to 1; 1 means perfect match between the images being compared. Our results (with SSIM) for AlexNet and VGG-16 are given at Fig. 6c and Fig. 6d respectively, where we also observe as we continue weight randomization from top to bottom layers, SSIM values decrease slowly, demonstrating the consistent changes in the maps. Similar as prior experiment, we compare the performance against other methods in this setting, and provide results in Fig. 6c and Fig. 6d. Guided backprop and Guided Grad-CAM show very high SSIM values while Grad-CAM, Grad-CAM++, CNN-Fixations show moderately low SSIM values. Proposed one achieves lowest SSIM values in all the distorted layers. We observe that results are in line with the prior experiment. Thus, comparative quantitative measures (Fig. 6) together with individual qualitative results (Fig. 7 and Fig. 8) demonstrate that as model parameter changes, our method consistently follows respective changes by shifting its focus, showing its dependency and sensitivity towards model parameters.

### TABLE 4. Average correlation scores (in terms of Spearman’s rank correlation ($\rho_s$-values) and SSIM) between different visualization maps of 1000 ImageNet samples and their adversarial pairs. Results are generated for AlexNet and VGG-16 networks.

| Methods          | $\rho_s$-values | SSIM  |
|------------------|-----------------|-------|
|                  | AlexNet  | VGG-16 | AlexNet  | VGG-16 |
| Integrated Grads  | 0.53     | 0.48   | 0.68     | 0.63   |
| Gradient $\times$ Input | 0.55     | 0.53   | 0.70     | 0.66   |
| Guided Backprop   | 0.79     | 0.78   | 0.85     | 0.82   |
| Grad-CAM          | 0.35     | 0.38   | 0.50     | 0.51   |
| Guided Grad CAM   | 0.77     | 0.69   | 0.85     | 0.84   |
| Grad-CAM++        | 0.32     | 0.34   | 0.52     | 0.50   |
| CNN-Fixations     | 0.24     | 0.20   | 0.41     | 0.37   |
| Proposed          | **0.22** | **0.17** | **0.36** | **0.32** |

#### 2) CLASS-SENSITIVITY ANALYSIS

We conduct another sanity check on our method in terms of its sensitivity towards predicted-class. Similar experiments are observed in the literature [50], [51], where the purpose is to check whether visualization map focuses on the representative class-object in a correct and reliable manner. For instance, if current focus of the image is distracted intentionally, an unreliable visualization map would discontinue to follow the changes, and hence, can be regarded as class-insensitive. Should the map be class-sensitive, its focus must shift substantially with respect to the change of the image.

To check this aspect, we generate adversarial pair of an input image, and generate visualization maps for both of them to examine their difference of focus. An adversarial pair is generated through adversarial attack which brings subtle changes to the image leading the network predict a different class. In Fig. 9, we provide visualization maps of some sample images and their adversarial pairs. As we observe, in the adversarial images, main concentration of their visualization map shifts to other areas; for example, bassinet to triangle-like tripod shape (top-left example), and so on. Furthermore, for quantitative analysis, we collect 1000 ImageNet samples, and calculate difference between the visual map of original and their adversarial pairs. Similar as before, difference is calculated using Spearman’s rank correlation ($\rho_s$) and SSIM values, and average results are reported at Table 4 for our method along with other existing methods. In comparison, our method achieves lowest dissimilarity values than the other methods for both AlexNet and VGG-16 networks. Such low $\rho_s$ and SSIM values suggest considerable amount of dissimilarities between original and adversarial map, pointing substantial shifts of concentration in adversarial maps. Thus, above results show that proposed visualization map is not biased towards image itself, rather as the concentration changes in image, it tracks the changes quite successfully exhibiting its consistency towards respective image-class.
From the experimental results reported above (Section III-A~III-C), we observe that some methods, e.g., Guided Grad-CAM and Guided backprop show poor results in both the sanity check experiments although they show moderate performance in objective evaluation. We found Integrated Gradients, Gradient×Input performing poorly while Grad-CAM, Grad-CAM++ and CNN-Fixations show comparably better results than the above-mentioned methods in objective evaluation and sanity check experiments. However, proposed method achieves superior results in objective and subjective evaluation; as well as shows strong model-dependency in the sanity-check experiments.

IV. CONCLUSION AND FUTURE DIRECTION

In this work, we present a new visual interpretation method that sequentially selects discernible neurons per layer and gradually trace back the prediction up to input image to find discriminative image locations responsible for the prediction. While selecting such neurons per layer, we consider contributions from adjacent layers along with its current layer after adding layer-specific priority-values. In the proposal, we present reliable ways of extracting such layerwise contributions and layer-specific priority-values, and show their consistency in our experiments. We conduct comparative subjective & objective evaluations and couple of sanity-checks to evaluate the proposal’s overall reliability as a visualization map, where we visualize its superior performance over other existing methods.

However, there is still enough room to further explore the attribute and applicability of our method. For instance, in the score function (Eq. 4), currently we consider only the first-placed adjacent layer information. How information
from sequentially appeared multiple adjacent layers would affect the visualization is an interesting research direction. In Section II-B2), we show an IOU-based approach for layerwise priority-value selection. We acknowledge that such priorities may vary in diverse cases, and hence, a more generalized solution can be formulated to facilitate the cases where necessary boundary information for IOU calculation is unavailable. Moreover, in the current work, we showed the applicability of our idea to traditional architectures, like AlexNet and VGG-16. It is worthwhile to mention that some minor tweaking in the design should be addressed for recent modules, like inception, skip-connection and LSTM units. Lastly, such a visualization map can be utilized in various applications, such as, object localization, image captioning, visual question answering, knowledge distillation etc. (as also done in [9],[20], [29] and [43]), where we are yet to test the utility of our proposal. We leave these issues here as our future endeavor.

APPENDIX

Here we provide screenshots of example questionnaires for two subjective evaluation experiments, detailed in Section III-B. Since the attendees are not subject-expert, at first we briefly explain them about good and bad representation of visualizations with some examples. One such example slide is given in Fig. 10. In Fig. 11, we present a representation of visualizations with some examples. One such example is given in Fig. 11. In Fig. 11, we provide screenshots of example questionnaires for two subjective evaluation experiments, detailed in Section III-B1, and Subjective Evaluation II denotes human-trust evaluation done in Section III-B2. We hide the exact experiment title in order to restrict any sort of bias towards the experiment.

REFERENCES

[1] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, “Going deeper with convolutions,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2015, pp. 1–9.
[2] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2016, pp. 770–778.
[3] F. Doshi-Velez and B. Kim, “Towards a rigorous science of interpretable machine learning,” 2017, arXiv:1702.08608.
[4] X. Li, H. Xiong, X. Li, X. Wu, X. Zhang, J. Liu, J. Bian, and D. Dou, “Interpretable deep learning: Interpretation, interpretability, trustworthiness, and beyond,” 2021, arXiv:2103.10689.
[5] K. Simonany, A. Vedaldi, and A. Zisserman, “Deep inside convolutional networks: Visualising image classification models and saliency maps,” 2013, arXiv:1312.6034.
[6] M. D. Zeiler and R. Fergus, “Visualizing and understanding convolutional networks,” in Proc. Eur. Conf. Comput. Vis. Cham, Switzerland: Springer, 2014, pp. 818–833.
[7] J. T. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller, “Striving for simplicity: The all convolutional net,” 2014, arXiv:1412.6806.
[8] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, “Learning deep features for discriminative localization,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2016, pp. 2921–2929.
[9] 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 Proc. IEEE Int. Conf. Comput. Vis. (ICCV), Oct. 2017, pp. 618–626.
[10] D. Smilkov, N. Thorat, B. Kim, F. Viégas, and M. Wattenberg, “Smooth-Grad: Removing noise by adding noise,” 2017, arXiv:1706.03825.
[11] A. Shrikumar, P. Greenside, and A. Kundaje, “Learning important features through propagating activation differences,” in Proc. 34th Int. Conf. Mach. Learn. vol. 70, 2017, pp. 3145–3153.
[12] M. Sundararajan, A. Taly, and Q. Yan, “Axiation attribute for deep networks,” in Proc. 34th Int. Conf. Mach. Learn., vol. 70, 2017, pp. 3319–3328.
[13] M. B. Muhammad and M. Yeasin, “Eigen-CAM: Class activation map using principal components,” in Proc. Int. Joint Conf. Neural Netw. (IJCNN), Jul. 2020, pp. 1–7.
[14] P. Linardatos, V. Papastefanopoulos, and S. Kotsiantis, “Explainable AI: A review of machine learning interpretability methods,” Entropy, vol. 23, no. 18, p. 18, Dec. 2020.
[15] Q.-S. Zhang and S.-C. Zhu, “Visual interpretability for deep learning: A survey,” Frontiers Inf. Technol. Electron. Eng., vol. 19, no. 1, pp. 27–39, 2018.
[16] R. Guidotti, A. Monreale, S. Ruggieri, F. Turini, F. Giannotti, and D. Pedreschi, “A survey of methods for explaining black box models,” ACM Comput. Surv., vol. 51, no. 3, pp. 3:1–3:42, Aug. 2018.
[17] E. Montavon, W. Samek, and K.-R. Müller, “Methods for interpreting and understanding deep neural networks,” Digit. Signal Process., vol. 73, pp. 1–15, Feb. 2018.
[18] W. Samek, Explainable AI: Interpreting, Explaining and Visualizing Deep Learning, vol. 11700, Cham, Switzerland: Springer, 2019.
[19] E. Tjoo and C. Guan, “A survey on explainable artificial intelligence (XAI): Toward medical XAI,” IEEE Trans. Neural Netw. Learn. Syst., vol. 32, no. 11, pp. 4799–4819, 2021.
[20] G. Joshi, R. Walambe, and K. Kotecha, “A review on explainability in multimodal deep neural nets,” IEEE Access, vol. 9, pp. 59800–59821, 2021.
[21] A. B. Arrieta, N. Díaz-Rodríguez, J. D. Ser, A. Bennetot, S. Tabik, A. Barbado, S. Garcia, S. Gil-Lopez, D. Molina, R. Benjamins, R. Chatila, and F. Herrera, “Explainable artificial intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI.” Inf. Fusion, vol. 58, pp. 82–115, Jun. 2020.
[22] L. M. Zintgraf, T. S. Cohen, T. Adel, and M. Welling, “Visualizing deep neural network decisions: Prediction difference analysis,” 2017, arXiv:1702.04595.
[23] D. Baehrens, T. Schroeter, S. Harmeling, M. Kawanabe, K. Hansen, and K.-R. Müller, “How to explain individual classification decisions,” J. Mach. Learn. Res., vol. 11, no. 6, pp. 1803–1831, 2010.
[24] J. M. Zitngraf, T. S. Cohen, T. Adel, and M. Welling, “Visualizing deep neural network decisions: Prediction difference analysis,” 2017, arXiv:1702.04595.
[25] M. T. Ribeiro, S. Singh, and C. Guestrin, “Why should I trust you?: Explaining the predictions of any classifier,” in Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining, Aug. 2016, pp. 1135–1144.
[26] A. Mahendran and A. Vedaldi, “Salient deconvolutional networks,” in Proc. Eur. Conf. Comput. Vis. Cham, Switzerland: Springer, 2016, pp. 120–135.
[27] P.-J. Kindermans, K. Schütt, K.-R. Müller, and S. Dahne, “Investigating the influence of noise and distractors on the interpretation of neural networks,” 2016, arXiv:1611.07270.
[28] R. R Selvaraju, A. Das, R. Vedantam, M. Cogswell, D. Parikh, and D. Batra, “Grad-CAM: Why did you say that?” 2016, arXiv:1611.07450.
[29] R. K. Morariu, U. J. Garg, and R. V. Babu, “CNN Teacher: An untravelling approach to visualize the discriminative image regions,” IEEE Trans. Image Process., vol. 28, no. 5, pp. 2116–2125, May 2019.
[30] P.-T. Jiang, C.-B. Zhang, Q. Hou, M.-M. Cheng, and Y. Wei, “Score-CAM: Score-weighted visual explanations for convolutional neural networks,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW), Jun. 2020, pp. 24–25.
[31] H. Wang, Z. Wang, M. Du, F. Yang, Z. Zhang, S. Ding, P. Mardziel, and X. Hu, “Score-CAM: Score-weighted visual explanations for convolutional neural networks,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW), Jun. 2020, pp. 24–25.
[32] S. Desai and H. G. Ramaswamy, “Ablation-CAM: Visual explanations for deep convolutional network via gradient-free localization,” in Proc. IEEE Int. Conf. Appl. Comput. Vis. (WACV), Mar. 2020, pp. 983–991.
[34] S. Bach, A. Binder, G. Montavon, F. Klauschen, K.-R. Müller, and W. Samek, “On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation,” PLoS ONE, vol. 10, no. 7, Jul. 2015, Art. no. e0130140.

[35] H. Cholakkal, J. Johnson, and D. Rajan, “Backtracking ScSPM image classifier for weakly supervised top-down saliency,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2016, pp. 5278–5287.

[36] M. Robnik-Šikonja and I. Kononenko, “Explaining classifications for individual instances,” IEEE Trans. Knowl. Data Eng., vol. 20, no. 5, pp. 589–600, May 2008.

[37] G. Montavon, A. Binder, S. Lapuschkin, W. Samek, and K.-R. Müller, “Layer-wise relevance propagation: An overview,” in Explainable AI: Interpreting, Explaining and Visualizing Deep Learning, Cham, Switzerland: Springer, 2019, pp. 193–209.

[38] J. Zhang, S. A. Bargal, Z. Lin, J. Brandt, X. Shen, and S. Sclaroff, “Top-down neural attention by excitation backprop,” Int. J. Comput. Vis., vol. 126, no. 10, pp. 1084–1102, 2016.

[39] T. Huber, D. Schiller, and E. André, “Enhancing explainability of deep reinforcement learning through selective layer-wise relevance propagation,” in Proc. Joint German/Austrian Conf. Artif. Intell. (Künstliche Intelligenz), Cham, Switzerland: Springer, 2019, pp. 188–202.

[40] S. Park, J. Lee, S. Mo, and J. Shin, “Lookahead: A far-sighted alternative of magnitude-based pruning,” in Proc. Int. Conf. Learn. Represent., 2020. [Online]. Available: https://openreview.net/forum?id=r1y3gHYDB

[41] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, “ImageNet large scale visual recognition challenge,” Int. J. Comput. Vis., vol. 115, no. 3, pp. 211–252, Dec. 2015.

[42] Find the Bounding-Box of an Object. Accessed: Sep. 22, 2020. [Online]. Available: https://scipy-lectures.org/advanced/image_processing/auto_examples/plot滕_find_object.html#find-the-bounding-box-of-an-object

[43] A. Chattopadhay, A. Sarkar, P. Howlader, and V. N. Balasubramanian, “Grad-CAM++: Generalized gradient-based visual explanations for deep convolutional networks,” in Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV), Mar. 2018, pp. 839–847.

[44] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. (2007) The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results. [Online]. Available: http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html

[45] U. Ozbulak. (2019). Pytorch CNN Visualizations. [Online]. Available: https://github.com/utkuozbulak/pytorch-cnn-visualizations

[46] Y. He. Models Accuracy on ImageNet 2012 Val. Accessed: Aug. 28, 2021. [Online]. Available: https://github.com/BVLC/caffe/wiki/Models-accuracy-on-ImageNet-2012-val

[47] J. Adedayo, J. Gilmer, M. Muellly, I. Goodfellow, M. Hardt, and B. Kim, “Sanity checks for saliency maps,” in Proc. Adv. Neural Inf. Process. Syst., 2018, pp. 9505–9515.

[48] J. H. Zar, “Spearman rank correlation,” in Encyclopedia of Biostatistics, vol. 7, Wiley, Jul. 2005, doi: 10.1002/0470011815.b2a15150.

[49] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, “Image quality assessment: From error visibility to structural similarity,” IEEE Trans. Image Process., vol. 13, no. 4, pp. 600–612, Apr. 2004.

[50] W. Nie, Y. Zhang, and A. Patel, “A theoretical explanation for perplexing behaviors of backpropagation-based visualizations,” 2018, arXiv:1805.07039.

[51] B. Kim, J. Seo, S. Jeon, J. Koo, J. Choe, and T. Jeon, “Why are saliency maps noisy? Cause of and solution to noisy saliency maps,” 2019, arXiv:1902.04893.

[52] S.-M. Moosavi-Dezfooli, A. Fawzi, and P. Frossard, “DeepFool: A simple and accurate method to fool deep neural networks,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2016, pp. 2574–2582.

MD. TAUHID BIN IQBAL (Member, IEEE) received the bachelor’s degree in information technology from the University of Dhaka, Bangladesh, in 2012, and the Ph.D. degree in computer science and engineering from Kyung Hee University, South Korea, in 2019. He has conducted the postdoctoral research at the Machine Learning and Visual Computing (MLVC) Laboratory, Kyung Hee University. He is currently working as an Assistant Professor with the Department of Computer Science and Engineering, Stamford University Bangladesh. His current research interests include explainable AI, deep learning interpretation, adversarial attack/defense, GAN, facial image analysis, and generation and its recognition.

ABDUL MUQUEET received the bachelor’s degree from the University of Karachi, Karachi, Pakistan, in 2014, and the M.S. degree from Kyung Hee University, South Korea. His research interests include deep learning interpretation and efficient deep learning models.

SUNG-HO BAE (Member, IEEE) received the B.S. degree from Kyung Hee University, South Korea, in 2011, and the M.S. and Ph.D. degrees from the Korea Advanced Institute of Science and Technology (KAIST), South Korea, in 2012 and 2016, respectively. From 2016 to 2017, he was a Postdoctoral Associate at the Computer Science and Artificial Intelligence Laboratory (CSAIL), Massachusetts Institute of Technology (MIT), MA, USA. Since 2017, he has been an Assistant Professor with the Department of Computer Science and Engineering (CSE), Kyung Hee University. His current research interests include model compression, interpretation, adversarial attack/defense, and neural architecture search for deep neural networks.

* * *