Military Institute of Science and Technology (MIST), Dhaka-1216, Bangladesh

Interpreting and Comparing Convolutional Neural Networks: A Quantitative Approach

Mohammad Mohaiminul Islam[§]
College of Science and Technology
University of Bordeaux, France
Bordeaux, France
mohammad-mohaiminul.islam@etu.u-bordeaux.fr

Zahid Hassan Tushar[§]
Department of Electrical and Computer Engineering
University of Hawaii at Manoa
Hawaii, USA
zhtushar@hawaii.edu

***Abstract*—A convolutional neural network (CNN) is sometimes understood as a black box in the sense that while it can approximate any function, studying its structure will not give us any insights into the nature of the function being approximated. In other terms, the discriminative ability does not reveal much about the latent representation of a network. This research aims to establish a framework for interpreting the CNNs by profiling them in terms of interpretable visual concepts and verifying them by means of Integrated Gradient. The interpretability profiling has been done by evaluating the correspondence between individual hidden neurons and a set of human-understandable visual semantic concepts. An integrated gradient-based class-specific relevance mapping approach is proposed that verifies interpretability profiling. Moreover, it is insightful to examine the correlation between the different input classes in terms of an overlapping set of highly active neurons. The result suggests the existence of a structured set of neurons inclined to a particular class. Finally, network ablation is performed to illustrate the performance of the network based on our approach.**

***Index Terms*—Network Interpretation, Image Classification, Convolutional Neural Networks, Integrated Gradient.**

I. Introduction

Convolutional neural networks (CNNs) constitute a subset of artificial neural networks. CNNs are inspired by the structure of the animal visual cortex system, unlike the ordinary neural networks which are designed after the operation of neurons (nodes) in the animal brain [1] [2]. A CNN is composed of several convolutional layers in conjunction with other layers, mainly providing non-linear capabilities (activation), data reduction (pooling), and classification (fully connected). CNNs are primarily employed for image processing and have proven to provide higher performances than other methods for tasks as image classification, segmentation, and object detection.

The performance of CNNs is mainly driven by the discriminative power of its units. Understanding the internal working mechanisms of these computational methods has a high impact on the field of Computer Vision as it enables training mechanisms with a higher semantic level. However, it is very hard to intuitively understand the prediction or inference process of deep CNNs— i.e. how do these networks reach a particular decision for a specific input, due to the non-linear layers and their deep and complex internal architectures.

Experiments studying the hidden units of CNNs revealed that these units learn intermediate latent variables that are interpretable by humans. For instance, hidden units of a network that has been trained to detect scene learn object detection as intermediate latent variables [3]. Similarly, hidden units of a network that has been trained to detect objects learn part detection as intermediate latent variables [4]. However, these spontaneous decompositions are not linearly interpretable.

In this study, our aim is to build a complete profile of a certain CNN expressed in terms of interpretability using modified Network Dissection framework [5]. Our interpretability profiling is based on correspondence between individual hidden units and four categories of human interpretable concepts: objects, object parts, materials, and colors. Then we use Integrated Gradient [6] to calculate per neuron basis relevance score which corresponds to the interpretability score with a very little error margin. Further we perform network ablation based on the relevance score to show inter-class entanglement of the networks.

II. Related work

A large number of methods have been employed by the researchers to shed light on the internal representation of CNNs; most of them are based on visualization. The internal representations can either be obtained by sampling the image patches that maximize the activation probability of each hidden unit of a CNN [7] or using variants of backpropagation to generate salient features [7] [8] [9]. Interestingly, some other techniques have been used to understand the discriminative power of networks: isolating parts of the network, transferring them to combine with other networks or limiting them up to certain layers, and checking their ability on different problems [10] [11] [12]. This kind of approach tries to boil down the working mechanism of CNNs into visual images interpretable by humans. Here our work aims to match the internal representation of CNNs with labeled and interpretable images.

Most relevant to our approach is investigating the behavior of each individual hidden unit in a layer of the networks. Zeiler and R. Fergus proposed a visualization technique that

[§]All authors contributed equally

978-1-6654-9522-6/21/$31.00 ©2021 IEEE

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 22 November 2021 doi:10.20944/preprints202101.0579.v2

maps feature activations of a convolutional unit back to the input pixel space [13]. Authors in [5] proposed an analytic framework based on the behavior of individual hidden units that has been adopted in this study. Also—as aforementioned, in [5], human evaluation has been used to show that units are behaving as object detectors in a network that was trained to detect scenes.

Conversely, with the aim to better understand the complex input-to output behavior of a deep neural network, a number of approach have been proposed based on attribution problem. Attributions measure the contribution of the model's output explained in terms of its input variables. For example, an attribution method image classification assigns a relevance score to each pixel of the given input image that tries to explain the model's predicted class. Some of the recent approaches of attribution are GradCAM [14], Layer wise relevance propagation [15] and nonlinear classification decisions with deep Taylor decomposition. Integrated Gradients (IG) proposed by Sundararajan et al. [6] as an attribution method for deep neural networks, which unlike some of the previously mentioned approach [14], [15] is fully independent of the structure of the model's architecture, and can be easily implemented with access to just the input's gradients after backpropagation.

III. Dataset

For this study, we have chosen a heterogeneous dataset called Broden [5]. This dataset contains a wide range of images of materials, colors, textures, parts, objects, and scenes. Except for the textures and scenes, most of the examples are annotated at pixel level. However, in our experiment, we have excluded the image-wise annotated categories, i.e., textures and scenes, as they tend to bias the overall interpretability process. Also, each pixel of the image is annotated with eleven color names as defined by Weijer [16]. The main goal of this dataset is to provide a baseline for visual concepts through which we want to express the interpretability of the evaluated CNN. All the concepts in this dataset are merged and normalized from their original dataset.

Furthermore, every class of this dataset corresponds to an English word. Labels are created without the consideration of the positional distinction, such as 'right' and 'bottom'; also some overly general synonyms have been avoided. Broden includes only those images having at least ten samples per label. See an example of the images and annotations from the Broden dataset in Fig. 1.

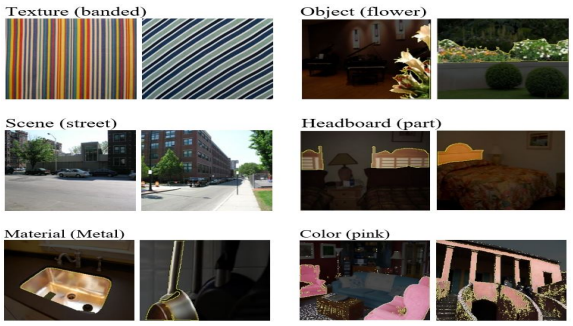


Fig. 1: Samples from the Broden Dataset with pixel-wise dense annotation

IV. Methodology

We divide this work into two main parts, namely *interpretability profiling* and *class specific relevance profiling*.

A. Interprability Profiling

We evaluate the interpretability of every unit of the CNN as a solution to a binary segmentation task to every visual concept in Broden by calculating the Intersection over Union (IoU) score (see equation 1) between the unit response to input and the annotated concept masks in the dataset. The activation map $A_i(x)$ has been calculated for every image x in the dataset and for all the units i of a CNN. Then for each unit i we compute the distribution a_i of its activations for all the images of the dataset. Now, to account only for relevant activation responses, we keep a small portion of the top activation values. To this aim, we compute a top quantile level T_i in such manner that $P(a_i > T_i) = 0.005$ over every spatial location of the activation map.

In order to compare activation maps and annotated masks, we first resize the activation maps by scaling $A_i(x)$ up to the masks resolution using bilinear interpolation. We name the upscaled version of $A_i(x)$ as $S_i(x)$. $S_i(x)$ is then converted into a binary mask $M_i(x) \equiv A_i(x) \geq T_i(x)$, by selecting specific regions whose activation exceeds the threshold $T_i(x)$. These segmented masks are evaluated against every concept c in the dataset by computing the intersection over union score between $M_i(x)$ and $L_c(x)$ — the annotation mask for image x and concept c, for each pair i, c as follows:

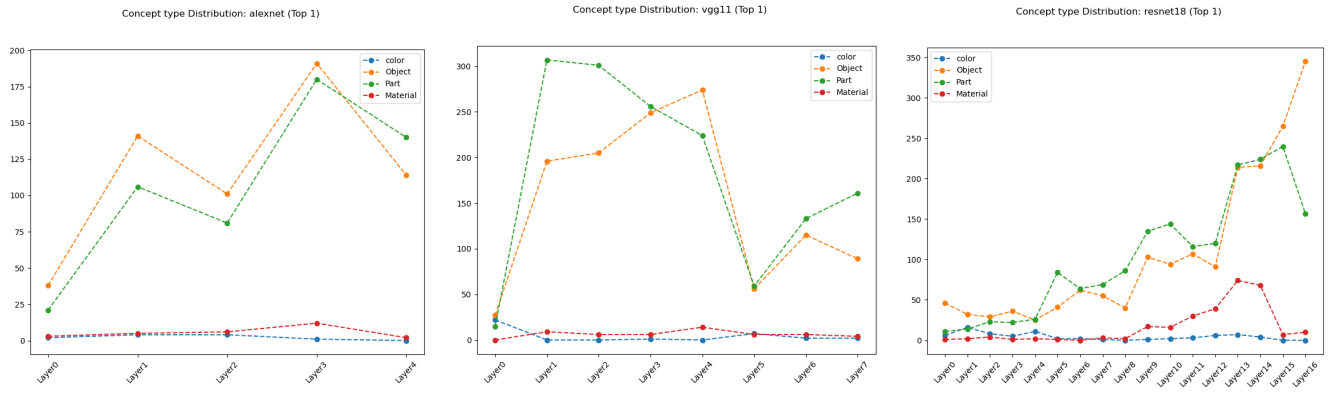
$$IoU_{i,c} = \frac{\sum_x |M_i(x) \cap L_c(x)|}{\sum_x |M_i(x) \cup L_c(x)|} \tag{1}$$

As the dataset contains some labels which are not present on every subset of inputs, the sums are calculated only on the subset of images that have at least one labeled concept of the c category. The $IoU_{i,c}$ score aims to indicate how much a unit i is aligned with the concept c. Note that one unit might detect multiple concepts; for the sake of analysis, we choose to report only the top 1 concepts for each individual unit i. The obtained top IoU scores: $Q_I^t(i, c), t = [1, 2, 3]$ can be interpreted as confidence scores for the interpretability of networks. Hence this score allows us to compare networks in terms of interpretability.

B. Class Specific Relevance Extraction

Integrated Gradient is a method originally proposed in [6] that aims to attribute an importance value to each input feature of a machine learning model based on the gradients of the model output with respect to the input. In particular, integrated gradients defines an attribution value for each feature by considering the integral of the gradients taken along a straight path from a baseline instance x' to the input instance x .

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 22 November 2021 doi:10.20944/preprints202101.0579.v2



(a) AlexNet (b) VGG (c) ResNet

Fig. 2: Layer wise distribution of each type of concepts (Vertical and horizontal axis denote No. of detectors and convolutional layer in each network respectively).

Formally, suppose we have a function $F : \mathcal{R}^n \rightarrow [0, 1]$ that represents a deep network, and an input $x = (x_1, \ldots, x_n) \in \mathcal{R}^n$. An attribution of the prediction at input x relative to a baseline input $x\,'$ is a vector $A_F(x, x\,') = (a_1, \ldots, a_n) \in \mathcal{R}^n$ where a_i is the contribution of x_i to the function $F(x)$

We consider an input instance x, a baseline instance $x\,'$ and a model $P : X \rightarrow Y$ which acts on the feature space X and produces an output y in the output space Y. In this case the function F is defined as $F(x) = P(x)$ if the model output is a scalar and $F(x) = P_j(x)$ if the model output is a vector, with the index k denoting the j-th element of $P(x)$. $P_j(x)$ is the probability of class j, which could be the true class corresponding to x . The attributions $A_i(x, x\,')$ for each feature x_i with respect to the corresponding feature $x\,'_i$ in the baseline are calculated as

$$A_i(x, x\,') = (x - x') \times \int_{\alpha=0}^{1} \frac{\partial F(x' + \alpha \times (x - x'))}{\partial x_i} d\alpha \quad (2)$$

where $\frac{\partial F}{\partial x_i}$ is the gradient of F along the i_{th} dimension at x .

We calculate the integrated gradient map $G_{i,c}(x_n)$ for every image x_n in the class c and for all the neurons i at each layer of the network. To take into account the spatial location of our object of interest in the input x_n we perform element wise multiplication operation with the binary annotation mask M_{x_n} corresponding to our input x_n. This yields a masked integrated gradient map containing only the integrated gradient values of the location of object of interest which we denote as $MG_{i,c}(x_n)$. Now let p run across all the pixels P in the in a gradient map $MG_{i,c}(x_n)$ and n across all the samples N in class c then the relevance score for each unit i and each class c is $R_{i,c}$ given by

$$R_{i,c} = \frac{1}{N} \sum_{n=1}^{N} \sum_{p=0}^{P} M_{x_n} \odot \psi(G_{i,c}(x_n)) \quad (3)$$

Here, $\psi(.)$ is the pre-processing operator. For this work, we choose $\psi(.)$ function as $\psi(.) = bilinearUpsample((abs(.)))$, where abs(·) is absolute value operation to take only the magnitude of importance while ignoring the sign. One can also use separately the positive and negative parts of the map to avoid signs. *bilinearUpsample(·)* upsamples the gradient maps using bilinear interpolation to have the same spatial size as the annotation mask.

V. Experimental Results & Discussion

A. Interpretability Evaluation & Discussion

Experiments start with AlexNet architecture as it is one of the first widely successful deep model . Next, VGG is considered as it has lower number of parameters compared to AlexNet as well as demonstrated comparable performance for applications. Finally, Resnet has a different architecture from the other two networks as it has skip connections to avoid gradient vanishing problem. Those networks are extensively used in the deep learning community which motivated us to examine our approach with these architectures. Although our approach is tested on the basic version of these architecture to compensate for the computational resource constraints, it is possible to extend to other architectures as well.

AlexNet: Results of the interpretability profiling for all the layers of AlexNet trained on image classification tasks on ImageNet are shown in the Fig. 2a. Also Fig. 2a shows that there are more object and part detectors than color concept detectors. This seems to violate the assumption that, at shallower convolutional layer low-level cues e.g. colors arise and more complex detectors arise at deeper convolutional layers [5].

In general shallow layers contains more low-level cues e.g., color detectors. Differently, we found more objects and part detectors than color ones. This may be due to the fact that the colors are divided into 11 fixed categories. If inside an object or a part class, all the exemplars share similar colors or different sheds of the same color that not defined within

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 22 November 2021 doi:10.20944/preprints202101.0579.v2

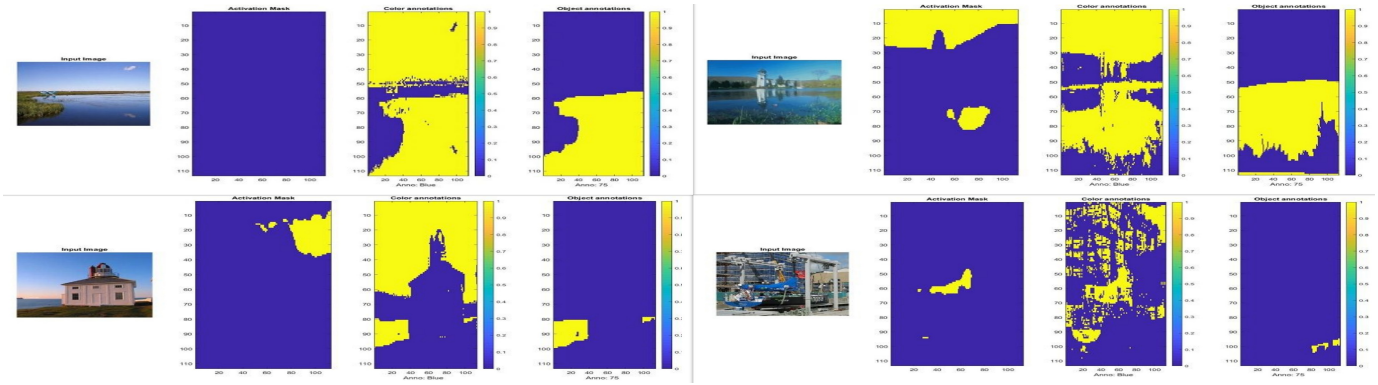


Fig. 3: Architecture of the treatment Response Assessment and Prediction pipeline

these 11 categories, our approach has no way to calculate IoU for undefined color or sheds of color. As a result, IoU score of parts and objects are higher than color in such case even though the units are in essence responding to a color concept (the object's or part's one). Hence some of the units are having a higher score for parts and objects despite of being activated by the colors of that object or part.

The phenomenon can be better understood by looking at annotations from the dataset and activation map of a particular unit. One such unit is 41 in *conv1* layer of AlexNet, which seems tuned to *Water* according to the IoU score but actually is responding to the specific shed of blue color. Let us illustrate this through a set of qualitative examples.

Fig. 3 shows the activation response of unit 41 for a set of images that represents the object *Water*. From the top left sub-image of Fig. 3 it can clearly be seen that the input image contains *Water* object, and the annotation map shows the unit's activation to the image. The unit is not responding to the *Water* nor the *blue* concepts. In top right sub-image we can notice that some spatial location of *Water* and *Sky* has been activated which are of *blue* color particularly. Finally, in the bottom right and left sub-images, although there is a lot of blue-colored region in the input, the unit tends to respond only to some specific spatial locations. This indicates that the unit under investigation is tuned to a specific shed of *blue* color, not to any object or in general to all the sheds of *blue* color.

Resnet18: In Fig. 2c the unit distribution of the Resnet18 architecture is illustrated. At a glance, most of the units are learning *part* and *object* type concepts. There are only a few *color* and *material* type concept detector neurons. A closer inspection using our approach reveals the quantitative distribution of each concept type in different layers as shown in Fig. 2c.There is a surge in the concept detectors in the final two convolutional layers. It is due to the fact the final two layers of Resnet18 contains almost double neurons than the previous layer. Also, in the final layer the emergence of object type concept detector is noticeable. It might indicate the fact that this network captures the object structure as a whole very well rather than focusing on parts of the object. This behaviour can also be justified given the depth of Resnet18 compared to other two networks. As with more depth the network has more ability to create the higher level of abstraction.

VGG11: A summary of various concept type for different layers of the VGG11 architecture is given in Fig. 2b. Apparently we have less *color* and *material* type concept detectors in regards to the ample amount of *part* and *object* type concepts. The explanation for this unusual response of the network is similar to the one given for AlexNet architecture. The units are learning color contents of the image in the shallower layers but due to the availability of the few *color* types in our Broden dataset, the method was unable to capture this information. The logical explanation for this specific phenomenon is the same as the previously examined network architectures.

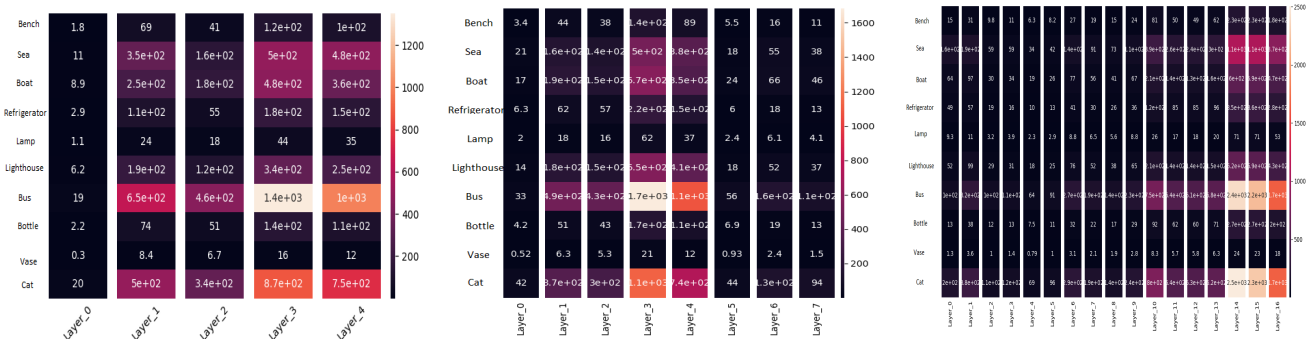
Figure and discussion suggest the following ideas regarding the operation of convolutional units:

- Units are tuned to a color despite having a higher IoU score for an object type concept. It appears that these units represent the colors of the object not the object itself.
- Units tend to represent a specific shed of a color. On the other hand, dataset annotations are the same for all the sheds of same color. Consequently, the union score gets much higher than the intersection score. Hence IoU for color concepts gets a small value. Similarly, it is also true that, if we had used denser color sampling beyond the 11 categories, results may have been different.

B. Interpretability Score & Relevance Score

For the sake of simplicity we select 10 classes that are exactly available both on the Broden and ImageNet dataset. Next we compute integrated gradient for each class and and each of the unit in the whole network using the approach described in subsection IV-B. We ask the question that if the neuron marked as important by IoU score for a specific class/concept is also marked important by the integrated gradient approach. In order to answer the question we compare the layer wise spatial location of the neurons yielded by both approach and report them in the table I. We report the mean error for all the layer in a particular network and across 10 classes under investigation with the error deviation. Number shows that indeed both of the

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 22 November 2021 doi:10.20944/preprints202101.0579.v2



(a) AlexNet (b) VGG (c) ResNet

Fig. 4: Layer wise Relevance score (Vertical and horizontal axis denote class type and convolutional layer in each network respectively).

approach identified almost same set of neurons as important for a given class.

TABLE I: Error percentage between IoU score and integrated gradient score across all layer and 10 classes

Networks	Mean error	Layer Error Deviation
AlexNet	7.56 %	±2.68%
VGG11	6.21 %	±1.96%
Resnet18	9.58 %	±5.11%

C. Relevance Distribution & Network Ablation

Relevance Distribution. To analyze the layer wise distribution of the relevance score to each class we aggregate the scores of individual neurons within a layer. For all the networks under investigation for such aggregated scores we can now visualize per-layer scores shown in Fig. 4 for three networks. The plots represent a heat map of attributions across all layers and 10 classes. It is interesting to observe that for different input classes different layer or combination of layers gains increasingly high relevance. For example cat class has a very high gradient for all the layers overall but in layer 3 and layer 4 there is a very large value in the AlexNet whereas for some other classes like $vase$ & $lamp$ has a very negligible value. Again when we compare between the networks for same class this can be observed that, it does not follow a specific pattern in terms of relevance score distribution. In case of both cat & bus the later layers like layer 3 & 4 are gaining a high relevance score in AlexNet; the layer at the middle of the network are getting the highest relevance for VGG; finally for the ResNet again it is near to the end of the network. In other words not all the layers are equally important for classification of all the input classes. This in turn might suggest that the neurons that learns to pick up or detect the necessary features for a particular type of input object classification may arise at different depth of a network.

Network Ablation. Depending on the relevance score that we have computed, we perform systematic network ablation by turning off some neurons in the network to understand the class wise effects and importance of the turned off neurons. Here we turn off the 1, 2 and 3 % neurons with the highest and lowest scores in each convolutional layer except the first and last convolution layer. The reason behind not ablating the neurons in first layer is that it might lead us to complete blindness along some channels. On the other hand, we do not prune the last layer as it contains the most high-level features and doing so might damage the entire performance of the network.

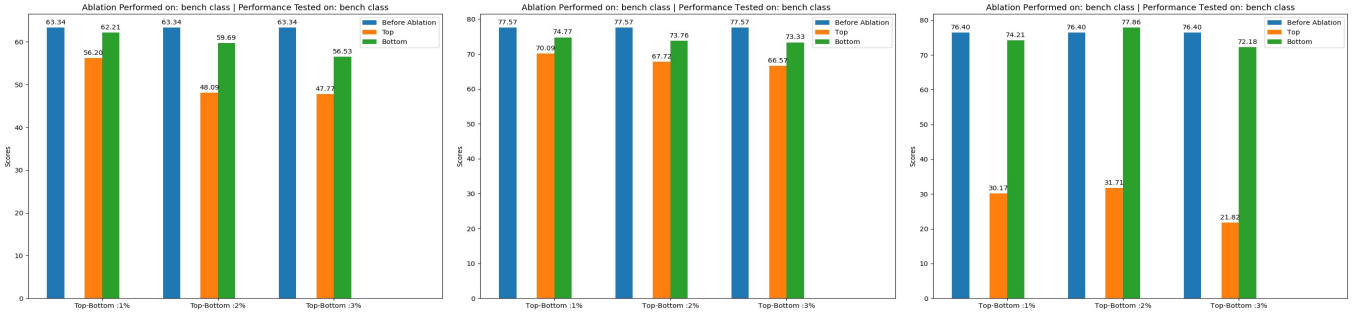
First we measure the performance of the original network on a certain input class. Then we turn off the top and bottom 1, 2 & 3% percent of the neurons of each layer convolutional layer depending of relevance score for that class. Next we measure the performance of of the ablated network on the class we have performed the ablation as well as 9 other classes. To demonstrate we only show the performance on two classes $bench$ & $refrigerator$ for all the networks on Fig. 5 & 6. Result shows that turning off the top neurons decreases performance significantly for both of the classes. However the amount of decrease in performance varies network to network, as it can be seen in Fig. 5c & 6c, that the performance decrease for *ResNet* is much heavier than other two network. The reason can be two fold; first as *ResNet* is a deeper network compared to other networks the amount of neuron being turned off is much larger than other networks; second the architecture of the *ResNet* again is more complex with different residual block and short skip connections for which the neuron inter dependencies or entangled condition of the neurons in feature representation are more dominant.

Now turning off the same amount of neurons with least relevance score, the performance decrease is almost non existing or negligible. This actually provides us a set of definite neurons with their spatial position in each layer that are not important or not needed at all for classification of a particular class. This can be helpful when we want to develop very specialized model to detect or classify single or only handful of objects.

VI. Conclusions

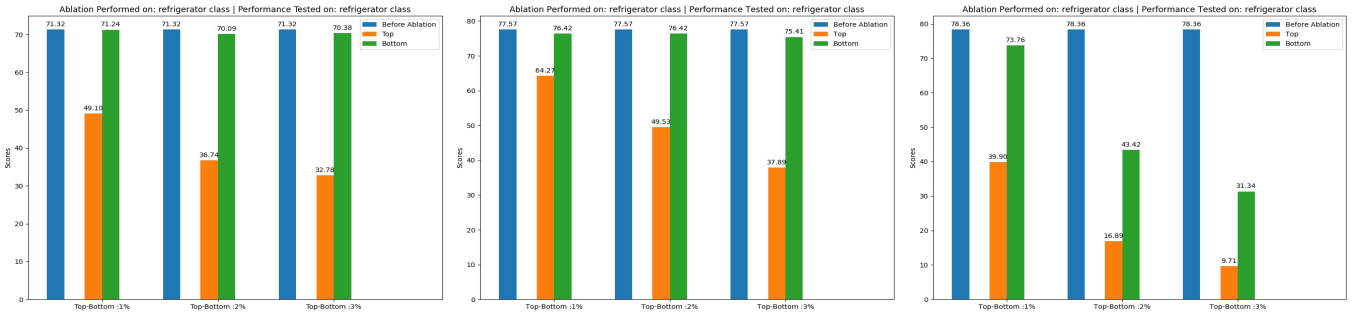
We applied our method to investigate how detectors for different visual concepts arise in various layers of a network.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 22 November 2021 doi:10.20944/preprints202101.0579.v2



(a) AlexNet (b) VGG (c) ResNet

Fig. 5: Performance measurement on *bench* input class



(a) AlexNet (b) VGG (c) ResNet

Fig. 6: Performance measurement on $Refrigerator$ input class

Results suggest that the shallower layers represent more low-level visual cues like colors, whereas deeper layers represent a more complex ones. IG mapping approach were able to identify neurons of the network that are crucial for the correct recognition of a particular concept. We found positive correlation between the neurons extracted by both modality. Also, we were able to identify a set of neurons that influence multiple classes to be recognised while there are other set of neurons that helps to classify one class but confuses the network classifying a different class. Further, network ablation verified our method as ablating significant units led to a fall in the network's performance while removal of insignificant units had not much effect on the performance.

REFERENCES

[1] K. Fukushima, "A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position," Biol. Cybern., vol. 36, pp. 193–202, 1980.

[2] D. H. Hubel and T. N. Wiesel, "Receptive fields and functional architecture of monkey striate cortex," The Journal of physiology, vol. 195, no. 1, pp. 215–243, 1968.

[3] L. Bazzani, A. Bergamo, D. Anguelov, and L. Torresani, "Self-taught object localization with deep networks," in IEEE winter conference on applications of computer vision (WACV), pp. 1–9, IEEE, 2016.

[4] E. Crawford and J. Pineau, "Spatially invariant unsupervised object detection with convolutional neural networks," in the AAAI Conference on Artificial Intelligence, vol. 33, pp. 3412–3420, 2019.

[5] D. Bau, B. Zhou, A. Khosla, A. Oliva, and A. Torralba, "Network dissection: Quantifying inter-pretability of deep visual representations," in IEEE conference on Conference on Computer Vision and Pattern Recognition, pp. 6541–6549, 2017.

[6] M. Sundararajan, A. Taly, and Q. Yan, "Axiomatic attribution for deep networks," in Proceedings of the 34th International Conference on Machine Learning, Sydney, Australia, PMLR 70, 2017.

[7] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3d convolutional networks," in IEEE international conference on computer vision, pp. 4489–4497, 2015.

[8] A. Mahendran and A. Vedaldi, "Understanding deep image representations by inverting them," in IEEE conference on Conference on Computer Vision and Pattern Recognition, pp. 5188–5196, 2015.

[9] K. Simonyan, A. Vedaldi, and A. Zisserman, "Deep inside convolutional networks: Visualising image classification models and saliency maps," in International Conference on Learning Representations, 2014.

[10] P. Agrawal, R. Girshick, and J. Malik, "Analyzing the performance of multilayer neural networks for object recognition," in European Conference on Computer Vision, pp. 329–344, Springer, 2014.

[11] A. Sharif Razavian, H. Azizpour, J. Sullivan, and S. Carlsson, "Cnn features off-the-shelf: an astounding baseline for recognition," in the IEEE conference on Conference on Computer Vision and Pattern Recognition, pp. 806–813, 2014.

[12] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?," in Advances in neural information processing systems, pp. 3320–3328, 2014.

[13] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in ECCV, pp. 818–833, Springer, 2014.

[14] 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," International Journal of Computer Vision, vol. 128, p. 336–359, Oct 2019

[15] A. Binder, G. Montavon, S. Bach, K.-R. Müller, and W. Samek, "Layer-Wise relevance propagation for neural networks with local renormalization layers," in Proceedings of International Conference on Artificial Neural Networks (ICANN), 2016.

[16] J. Van De Weijer, C. Schmid, J. Verbeek, and D. Larlus, "Learning color names for real-world applications," in IEEE Transactions on Image Processing, vol. 18, no. 7, pp. 1512–1523, 2009.