 doi:10.20944/preprints202101.0579.v1

Interpreting and Comparing Convolutional Neural Networks: A Quantitative Approach

Mohammad Mohaiminul Islam* and Zahid Hassan Tushar*

College of Sciences and Technologies

University of Bordeaux, France

{mohammad-mohaiminul.islam, zahid-hassan.tushar}@etu.u-bordeaux.fr

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. We also ask the question, "Do different input classes have a relationship or are they unrelated?" For instance, could there be an overlapping set of highly active neurons to identify different classes? Could there be a set of neurons that are useful for one input class whereas misleading for a different one? Intuition answers these questions positively, implying the existence of a structured set of neurons inclined to a particular class. Knowing this structure has significant values; it provides a principled way for identifying redundancies across the classes. Here the interpretability profiling has been done by evaluating the correspondence between individual hidden neurons and a set of human-understandable visual semantic concepts. We also propose an integrated gradient-based class-specific relevance mapping approach that takes the spatial position of the region of interest in the input image. Our relevance score verifies the interpretability scores in terms of neurons tuned to a particular concept/class. Further, we perform network ablation and measure the performance of the network based on our approach.

Key Words: Network Interpretation,Image Classification, Convolutional Neural Network, Integrated Gradient.

1 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

* These two authors contributed equally

Code is available at : https://github.com/niazoys/ICNN

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 28 January 2021 doi:10.20944/preprints202101.0579.v1

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 [3], segmentation [4], and object detection [5].

The performance of CNNs is mainly driven by the discriminative power of its units. Furthermore, CNNs have also been recently measured by their interpretability. 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 [5][6]. Similarly, hidden units of a network that has been trained to detect objects learn part detection as intermediate latent variables [7]. However, these spontaneous decompositions are not linearly interpretable. It is widely understood that this kind of networks can learn an efficient representation of underlying hidden variables to distinguish between the states, but also that these representations are generally not well understood. To solve this problem, Bau et al. [8] proposed a general analytic framework, *Network Dissection*, for interpreting deep visual representations and for quantifying their interpretability. Using Broden [8], a broadly and densely labeled data set, the framework tries to identify the semantics of the hidden units of a CNN and then aligns them with concepts that are understandable and interpretable by humans. They have also evaluated their method on various CNNs (AlexNet [3], ResNet [9],VGG [10] or GoogLeNet [11] trained on tasks such as object and scene recognition [8].

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 [8]. 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 [12] 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.

2 Background

This section introduces the fundamentals of Integrated Gradient briefly, also describes the libraries and tools that have been used in the project.

2.1 Integrated Gradient

Integrated gradients is a method originally proposed in Sundararajan et al., “Axiomatic Attribution for Deep Networks” 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: 28 January 2021 doi:10.20944/preprints202101.0579.v1

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

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

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

2.2 Tools & Libraries

In this project, primarily all the generated models and related experiments have been conducted using Python as the programming language and PyTorch as the Deep learning framework. We have used the AlexNet[3], ResNet18[9] and VGG11[10] architecture that is trained on ImageNet [13] dataset provided by the PyTorch for this study. To experiment with Integrated Gradient we have used a model interpretability framework called *Captum* which is design to work with PyTorch. Besides we have used other computer vision and machine learning packages like OpenCV, Scikit-learn , SciKit-image, matplotlib, seaborn for miscellaneous work.

3 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 be obtained by sampling the image patches that maximize the activation probability of each hidden unit of a CNN [14][5], or they can also be obtained by using variants of backpropagation to generate salient features [14][15][16]. On the other hand, 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 [17][18][19]. 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 maps feature activitions of a convolutional unit back to the input pixel space [20]. Authors in [8] 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. In a different study, authors have suggested

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 28 January 2021 doi:10.20944/preprints202101.0579.v1

testing the internal layers by training linear probs, which scrutinizes the information flow dynamics between layers and their impact on prediction [21].

On the other hand, 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[22], Layer wise relevance propagation[23] and nonlinear classification decisions with deep Taylor decomposition[24]. Integrated Gradients (IG) proposed by Sundararajan et al.[12] as an attribution method for deep neural networks, which unlike some of the previously mentioned approach ([22],[23] [24]) 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 back-propagation. Since in our work we are concerned with off the shelf network without altering its architectures , we use Integrated gradient.

4 Dataset

For this study, we have chosen a heterogeneous dataset called Broden [8]. This dataset has been created by fusing several densely labeled image datasets: Pascal-Context [25], Pascal-Part [25], ADE [26], Open- Surfaces [27] and the Describable Textures Dataset [28]. 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 [29]. 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.

Table 1: Statistics of each label type included in the data set[8].

Category	Classes	Sources	Avg. sample
part	234	ADE [25], Pascal-Part [21]	854
material	32	OpenSurfaces [25]	1,703
object	584	ADE[25],Pascal-Context [19]	491
color	11	Generated	59,250

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. Broaden includes only those images having at least ten samples per label. See an example of the images and annotations from the Broaden dataset in Figure 1. Also, Table. 1 shows the statistics of each category, source, and classes under them with average image samples per label.

The Borden data set have been used for assigning a visual concept to the neurons and calculating the Integrated Gradient the network at each convolutional layers.However we need some closely

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 28 January 2021 doi:10.20944/preprints202101.0579.v1

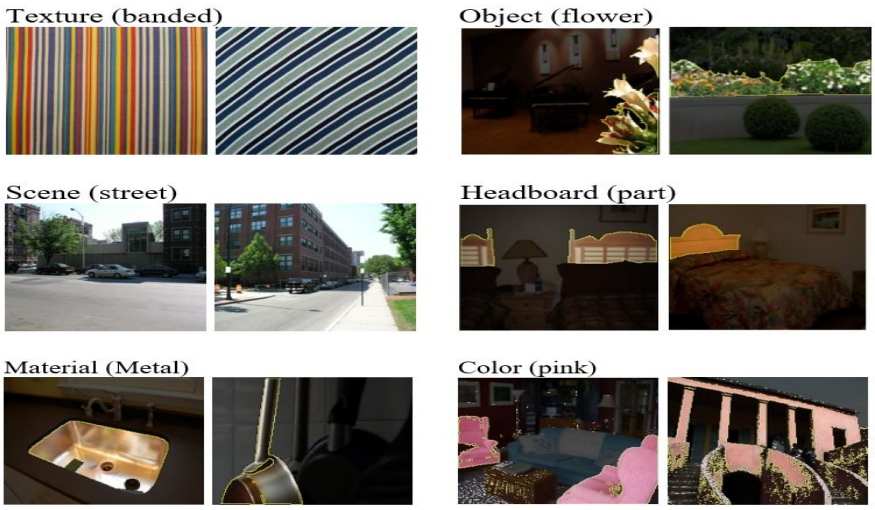


Figure 1: Samples from the Broden Dataset with pixel-wise dense annotation

resembling data to ImageNet[13] for controlled performance testing network since they have been trained on this particular dataset. To this aim, we take 1200 images each class of 10 specific classes namely *bench, sea, boat, refrigerator, lamp, lighthouse, bus, bottle, vase & cat* from ImageNet to measure the class specific performance of the model before and after the network ablation.

5 Proposed Approach

We divide this work into two main parts, namely *interpretability profiling* and *class specific relevance profiling*. The idea of interpretable representation depends on the fact of whether humans are able to interpret it in a meaningful way. Therefore, the evaluation of quantifying the interpretability is defined in terms of linking with human-interpretable visual concepts. Our measurement for interpretability works by gathering the response of each hidden unit in certain Convolutional layers and by quantifying the match between these responses and the visual concepts annotated in the Broaden dataset. In this work, we have replicated and adapted a previous approach [8], however, as improvement , we include here our development of the interpretability measure along with some previously undiscovered facts and visualization. In addition to our previous work, further we have developed a new mechanism to quantify the relevance of each neurons of the network with the respect to each input class. Here we extract the neuron wise integrated gradient for the entire network and aggregate the integrated gradient map into a single score taking into consideration of spatial location of the object of interest in the input images. Which allowed us to perform systematic network ablation to reveal some interesting findings.

5.1 Interprability Profiling

We evaluate the interpretability of every unit of the Convolutional Neural Network as a solution to a binary segmentation task to every visual concept in Broden by calculating the IoU (Intersection over Union) score (see equation 2) between the unit response to an image and the annotated concept masks in the dataset.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 28 January 2021 doi:10.20944/preprints202101.0579.v1

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 activation responses considering 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 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:

$$IoU_{i,c} = \frac{\sum_x |M_i(x) \cap L_c(x)|}{\sum_x |M_i(x) \cup L_c(x)|} \tag{2}$$

As the data set 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 3 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 and can be provided as a scalar for a unit or as a matrix for a layer or a network. Hence this score allows us to compare networks in terms of interpretability.

It should be noted here that the robustness of this framework depends on the dataset: if a unit has a match with a human-understandable concept that is absent from the dataset, then it will not score well for interpretability of that concept.

5.2 Class Specific Relevance Extraction

Here we extract the integrated gradient map using the theoretical background described in subsection 2.1. 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

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

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

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 28 January 2021 doi:10.20944/preprints202101.0579.v1

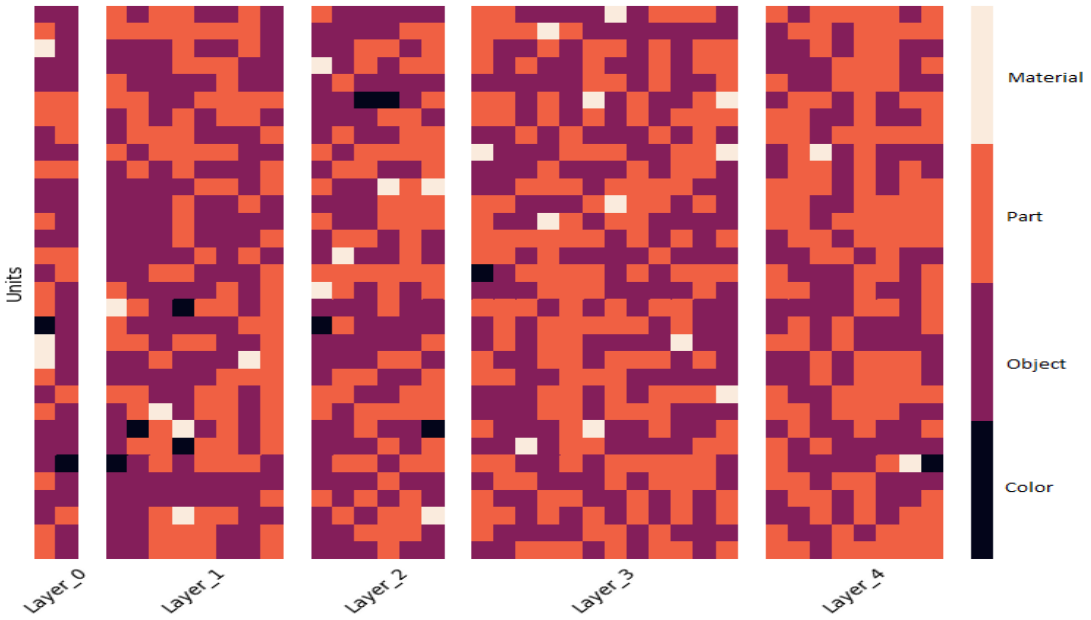


Figure 2: QI score map of Alexnet.

the map to avoid signs. *bilinearUpsample(·)* upsamples the gradient maps using bilinear interpolation to have the same spatial size as the annotation mask. The entire process can be can be found in Appendix-A.

6 Experimental Study, Results & Discussion

This section presents the setup and generated results with a detailed analysis of the study that has been done on this project. This section of the report is divided into two parts, the first one describing experiments on the interpretability profiling and the second one on the relevance score computation per neuron based on the Integrated Gradient method with relevant experiment and results .

In order to evaluate the proposed approach we have experimented on AlexNet[3], ResNet [9], VGG [10] architecture trained on ImageNet [30]. To illustrate the interpretability profiling of the network first, we have calculated the intersection over union (IoU) score, and we present a grid visualization for each unit of the entire network. We also show the layer and network wise distribution of each type of concept. Next we compute the relevance score of the network and show the relation between the two methods. Finally we perform systematic network ablation depending on relevance score and present the results.

6.1 Interpretability Evaluation & Discussion

Interpretability scores Q_I has been calculated for three network architectures Alexnet, Resnet18 and VGG11. Results are summarized in terms of associated concept type in Figure 2, 5 and 7 respectively. Each dot representing a single convolutional unit of the network, and each of the different color codes represent a particular type of concepts e.g., object, part, materials, color. For the sake of simplicity,

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 28 January 2021 doi:10.20944/preprints202101.0579.v1

we show the representation for top 1 concepts. This images helps us to get an overview of the entire networks in a single glance. From the visualization, the distribution of each type of concept with spatial position of each unit within a layer of the network can be realized.

We will dive into alexnet architecture at first and develop an understanding of our findings. Next, we will examine other network architectures such as Resnet18 and VGG11 to find consistency in our approach.

AlexNet: Using our approach, we analyze and compare the interpretability within all the convolutional layers of AlexNet trained for image classification tasks on ImageNet. Results are shown in the Figure 3. Also Figure 3 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 [8].

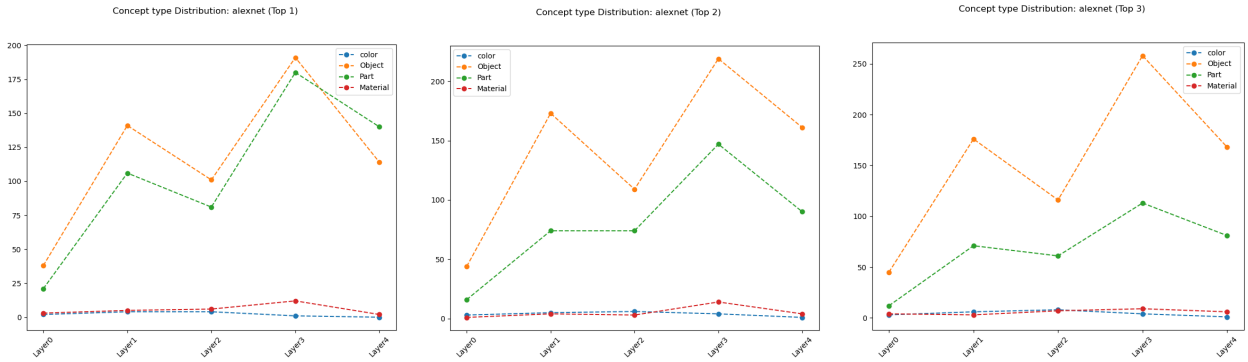


Figure 3: Top 3 Layer wise distribution of each type of concepts for AlexNet.

In general, we expect that at shallow layers, we will get more low-level cues e.g. color detectors. Differently, we are getting 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 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, but they are actually being activated by the colors of that object or part.

We can better understand the phenomenon 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.

Figure 4 Shows the activation response of unit 41 for a set of images that represents the object *Water*. From the top left sub-image of Figure 4 it can be clearly 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.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 28 January 2021 doi:10.20944/preprints202101.0579.v1

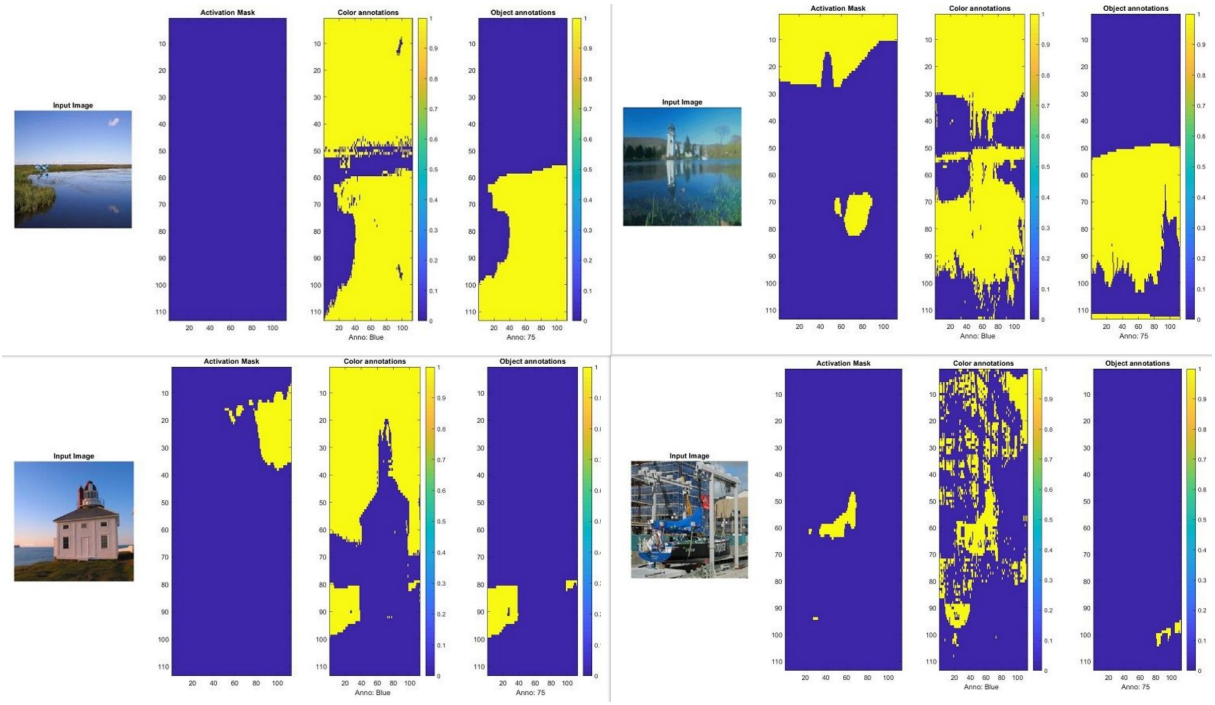


Figure 4: Activation response map for Unit 41 in *conv1* of AlexNet. Activation accumulated for different input images from a subset of images containing Water Object annotation. In each sub image from right to left figure shows input image, Activation map, color annotation map and object annotation map.

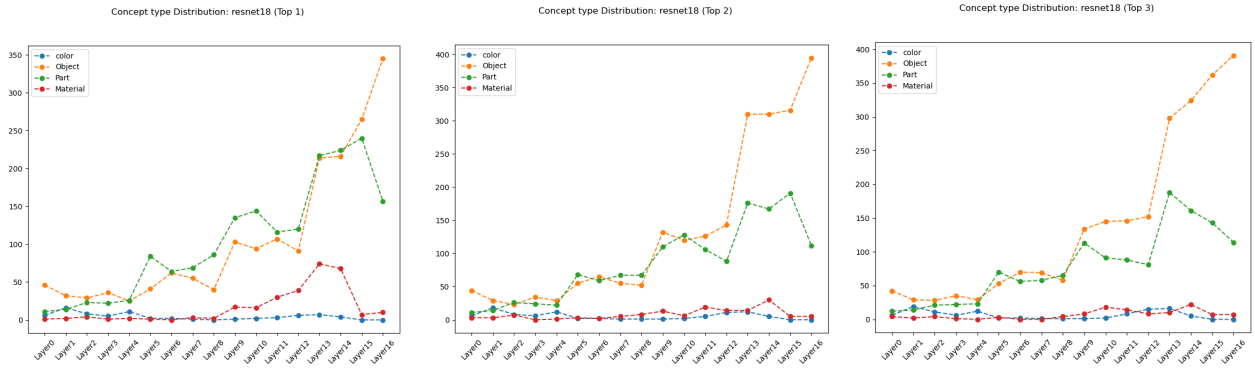


Figure 6: Top 3 Layer wise distribution of each type of concepts for Resnet18.

Resnet18: In figure 5 the Q_I score map of the resnet18 architecture is illustrated. From a glance at this figure, we realize that most of the units are learning *part* and *object* type concepts. There are only a few *color* and *material* type concept detector neurons. This response is analogous to our Q_I score map of alexnet architecture in figure 2 indicating the similar representation learning strategy of CNNs.

Now if we dig dipper into this network, using our approach we can list the quantitative distribution of each concept type in different layers as shown in figure 6. We can see that in the final two convolutional layers, there is a surge in the concept detectors. It is due to the fact the final two layers of resnet contains almost double neurons than the previous layer. Also, in the final layer the emergence

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 28 January 2021 doi:10.20944/preprints202101.0579.v1

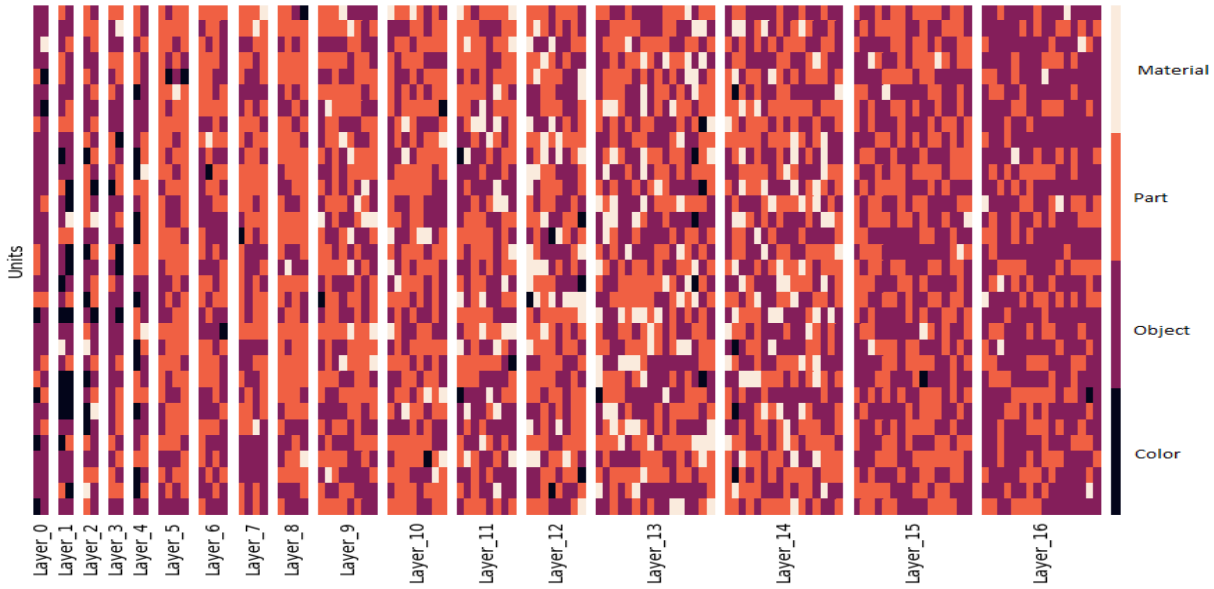


Figure 5: Q_I score map of Resnet18.

of object type concept detector is noticeable. It might indicate the fact that resnet 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.

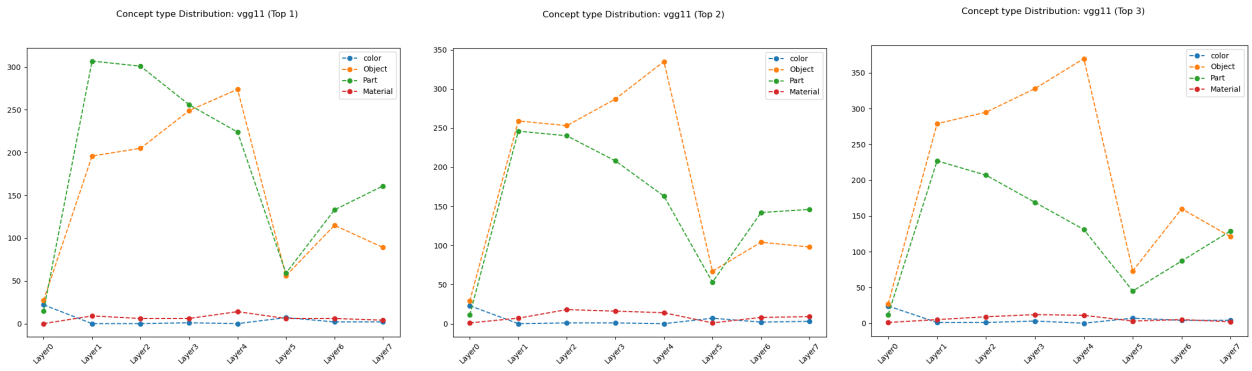


Figure 8: Top 3 Layer wise distribution of each type of concepts for VGG11.

VGG11: In this section we depict the response of VGG11 architecture in our interpretability framework. The figure 7 provides the QI score map of the VGG11 architecture. 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 Broaden dataset, the method was unable to capture this information.

Now let us investigate the summary of various concept type for different layers of the VGG11 ar-

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 28 January 2021 doi:10.20944/preprints202101.0579.v1

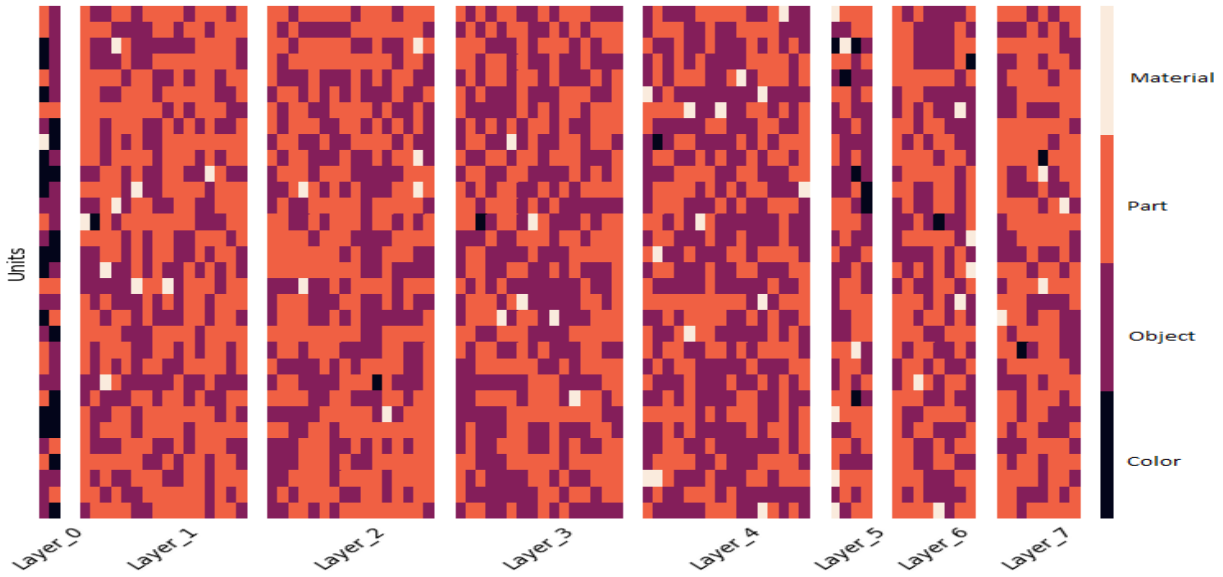


Figure 7: QI score map of vgg11.

chitecture as given in figure 8 which is computed for the Top 3 concepts per unit.We can notice that for the top three concepts per unit, they are mostly *object* and *part* type detectors. 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 turned out 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.

6.2 Interpretability Score & Relevance Score

For the sake of simplicity we select 10 classes that are exactly available both on the Broden and ImageNet dataset since we are not changing the architecture of the models nor we retrain the models in this experiment. Next we compute integrated gradient for each class and and each of the unit in the whole network using the approach described in subsection 5.2. 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 2. 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 approach identified almost same set of neurons as important for a given class.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 28 January 2021 doi:10.20944/preprints202101.0579.v1

Table 2: Error percentage between IoU score and integrated gradient score across all layer and 10 classes

Networks	Mean error	Layer Error Deviation
Alexnet	7.56 %	$\pm 2.68\%$
VGG11	6.21 %	$\pm 1.96\%$
ResNet18	9.58 %	$\pm 5.11\%$

6.3 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 figure 9 , 10 & 11 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.

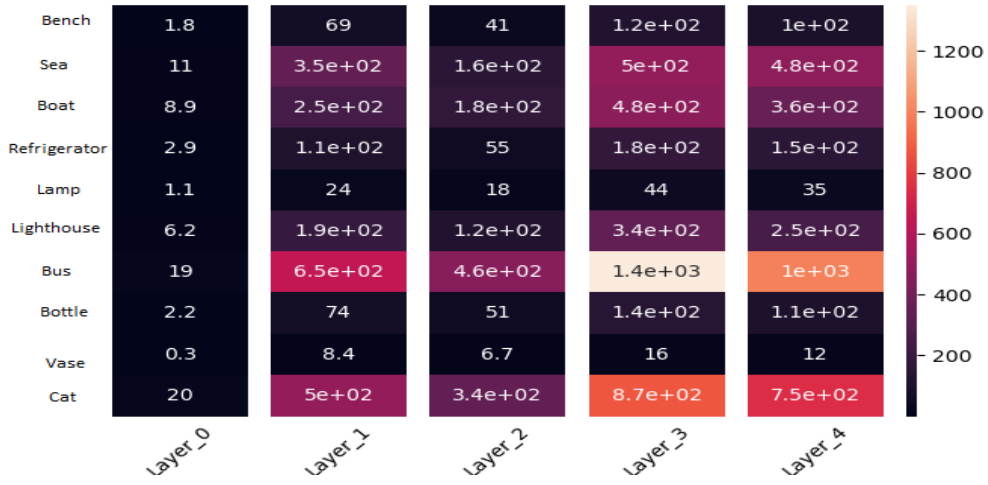


Figure 9: Alexnet Layer wise Relevance score

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 28 January 2021 doi:10.20944/preprints202101.0579.v1

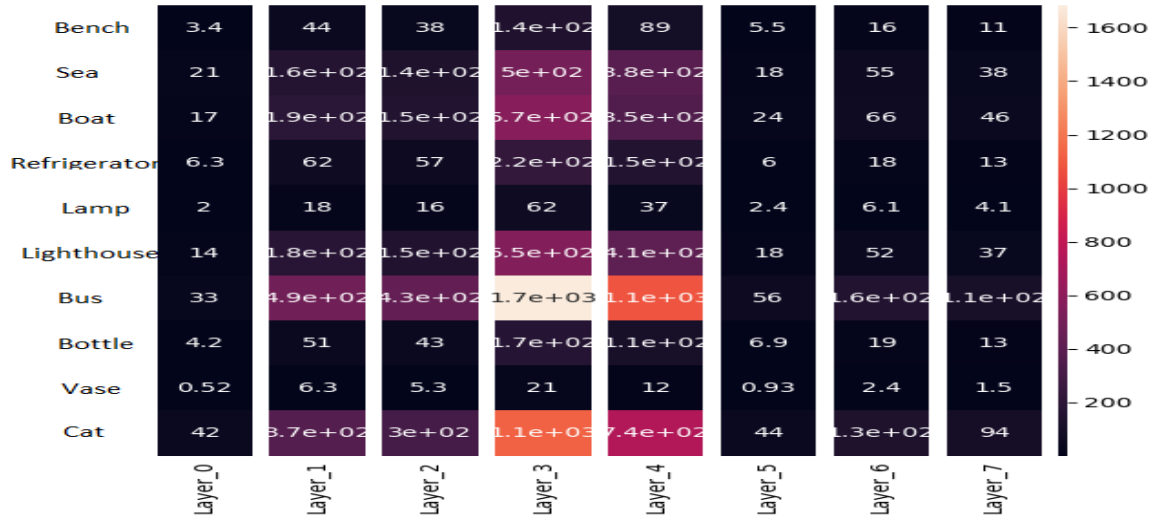


Figure 10: VGG Layer wise Relevance score

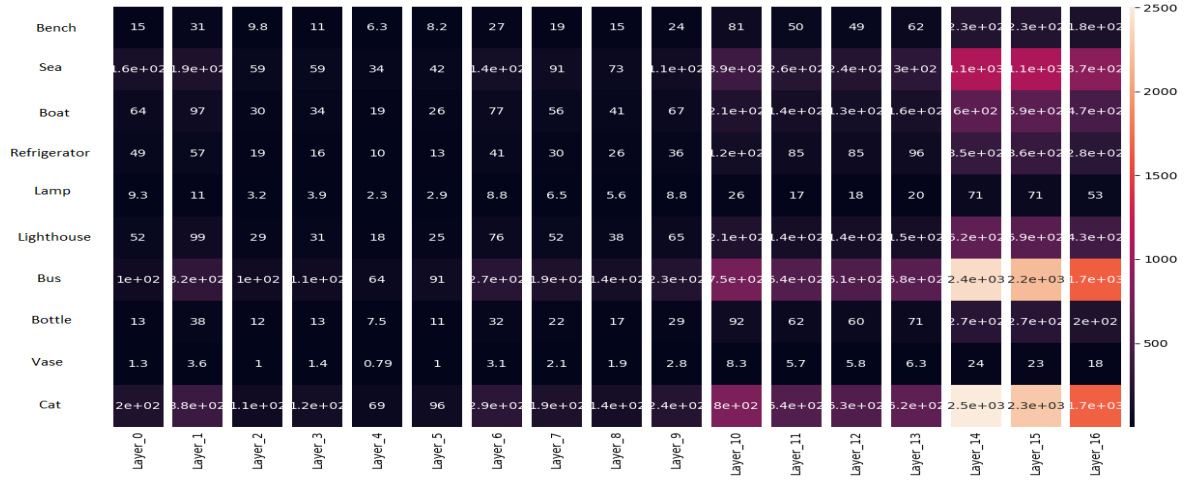


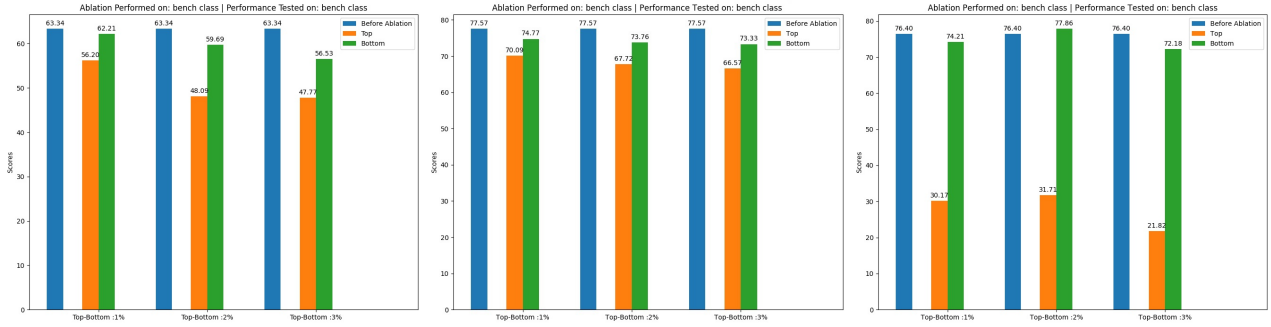
Figure 11: ResNet Layer wise Relevance score

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 % of the highest and lowest scores in each convolutional layer except the first and last convolution layer. The reason behind not ablating the neurons for the 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 (except the first & last layer). 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 figure 12 & 13. Result shows that as we turn off the top neurons the performance decreases significantly for

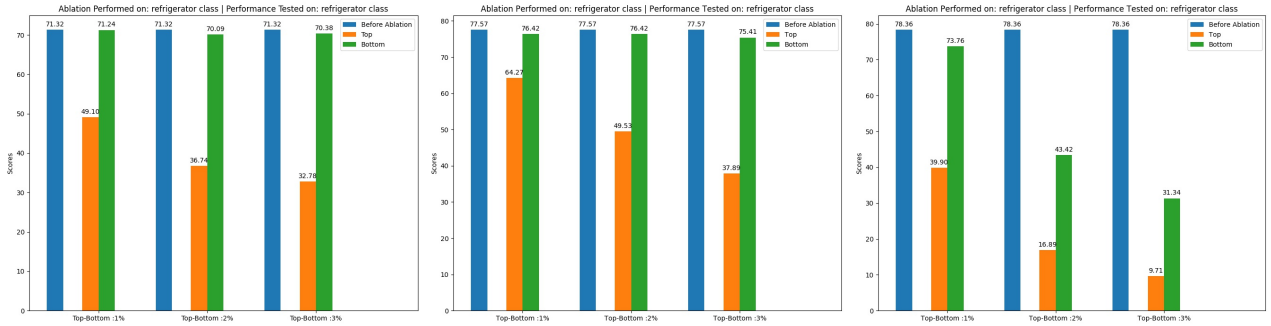
Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 28 January 2021 doi:10.20944/preprints202101.0579.v1

both of the classes and for all three network. However the amount of decrease in performance varies network to network, as it can be seen in figure 12c & 13c, 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.



(a) Alexnet (b) VGG (c) ResNet

Figure 12: Performance measurement on $bench$ input class



(a) Alexnet (b) VGG (c) ResNet

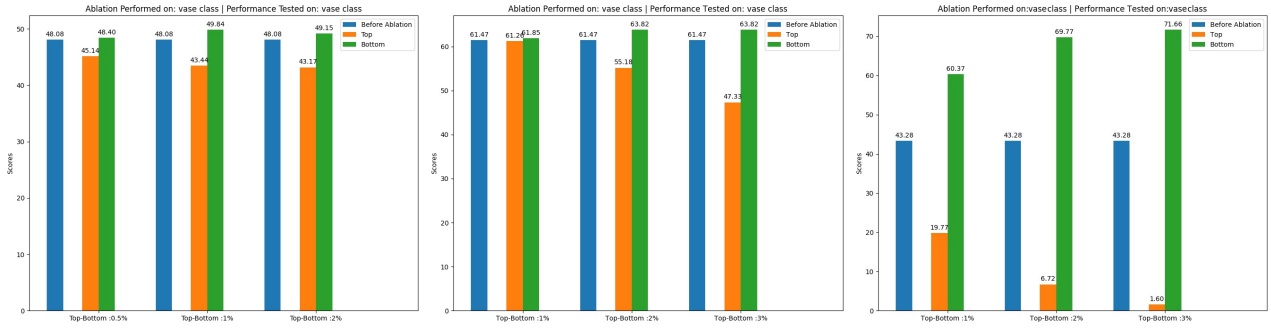
Figure 13: Performance measurement on $Refrigerator$ input class

Now as we turned off the same amount of neurons with least relevance score, the performance decrease is almost non existing or negligible. However, we have noticed few interesting cases where the turning off the least relevance score neurons yields a better performance than the original network. Figure 14 shows such a case for $vase$ input class where the performance increased for all three networks. This indicates that the neurons with least relevance score are not only unimportant for the classification but had somewhat negative effects. This actually provides us a set of definite neurons with their spatial position in each layer that are not much of important or not needed at all for classification of a particular input class. This can be helpful when we want to develop very specialized model to detect or classify single or only handful of object from more general off the shelf network. The results for all the classes can be found in the Appendix-B section.

Next we investigate the result of performing ablation on one class and measuring performance on a different class. This leads us to few interesting findings. Here for the sake of simplicity we show

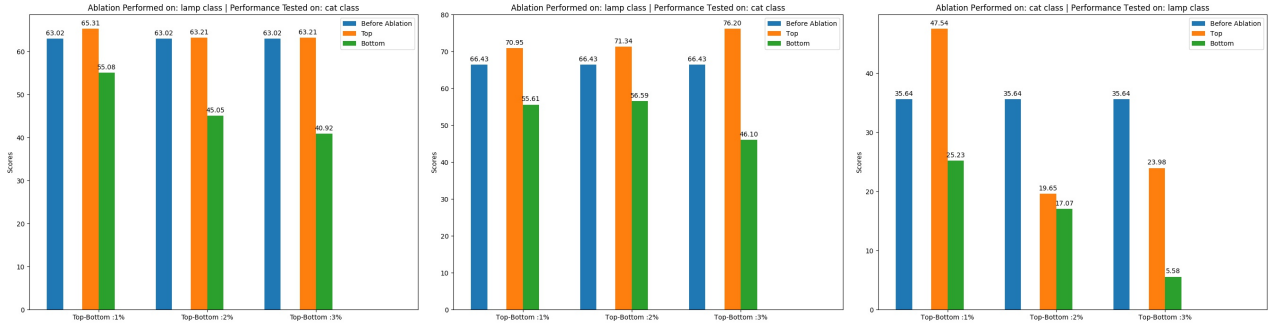
Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 28 January 2021 doi:10.20944/preprints202101.0579.v1

the results for 2 different cases however in general this is true for many other cases. In Figure 15 the cross performance measurement of $lamp$ and cat class is shown. It can be seen that there are not any obvious pattern between in the performance variance between these classes. Again in Figure 16 cross performance measurement between bus and $boat$ input classes shows a very consistent result as we have seen in self-class performance measurement. This might be due to the fact that some of these classes have been sharing some common features in latent feature representation space and whereas some of these classes are not.



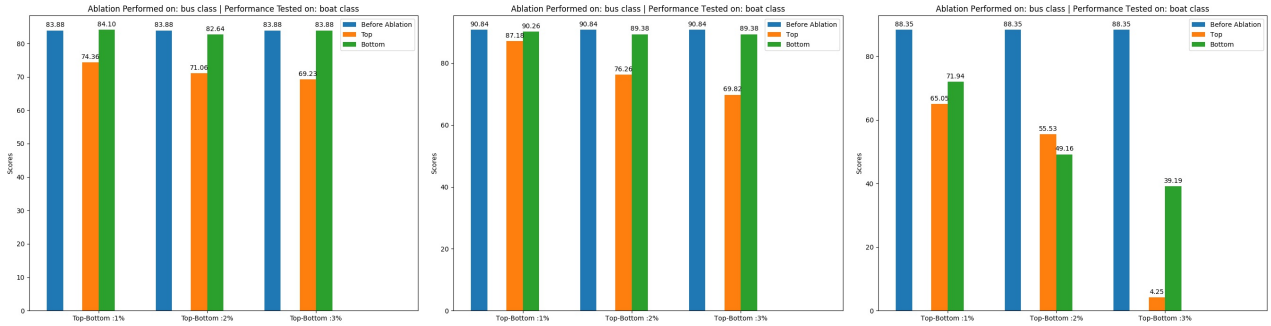
(a) Alexnet (b) VGG (c) ResNet

Figure 14: Performance measurement on $Vase$ input class



(a) Alexnet (b) VGG (c) ResNet

Figure 15: Cross performance measurement on $Lamp$ & Cat input class



(a) Alexnet (b) VGG (c) ResNet

Figure 16: Cross performance measurement on Bus & $Boat$ input class

 doi:10.20944/preprints202101.0579.v1

7 Limitation

Potential concept bias in the interpretation. A few potential predispositions may be affecting to our strategy as follows: 1) Our technique won't be able to distinguish units that identify concepts that don't show up in the Broden dataset, including some hard to-name ideas, for example, 'the corner of a room' or 'open surface of a chair'; 2) Some units may identify a concept at a finer level, for example, a wooden stool chair leg, which is more explicit than ideas in Broden; hence, yielding a low IoU score for the 'chair' category. This is similar to the discovered detection bias in case of color concept explained in 6.1; 3) Our approach gauges the level of alignment between singular unit activation and a visual concept, so it won't distinguish a group of units that might jointly represent a single concept; 4) as the deep networks have more convolutional units our approach might be biased when comparing between different network profiles.

Visual content sensitivity & Annotation dependency. Our current relevance score computation approach is sensitive to the relative size proportion of the object of interest in the input images. As covering a larger portion of a input image with object we want to classify tend to create overall larger relevance score skewing the distribution for the particular input class. Also our method need the object annotation mask to compensate the spatial variance of the of the object in the input images making its applicability to limited amount of dataset.

8 Conclusions

In this project, we have reviewed different approaches and adapted network dissection [8] with our own modification to measure the interpretability of convolutional neural networks. Besides, we have leveraged an integrated gradient based class specific relevance mapping approach to verify our interpretability framework. We applied our method to investigate how detectors for different level visual concepts arise in various layers of a network. 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 when classifying a different class. Further, we verified our methods by ablating the significant units of the network identified by our framework and observing a fall in the performance of the network.

Although effort has been made in this project to shed light on the internal behavior of the convolutional neural network, a lot of open research questions like we have discussed in section 7 still remains open to be addressed .

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] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, pp. 1097–1105, 2012.

[4] S. Gidaris and N. Komodakis, "Object detection via a multi-region and semantic segmentation-aware cnn model," in *Proceedings of the IEEE international conference on computer vision*, pp. 1134–1142, 2015.

[5] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Object detectors emerge in deep scene cnns," *arXiv preprint arXiv:1412.6856*, 2014.

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

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

[8] D. Bau, B. Zhou, A. Khosla, A. Oliva, and A. Torralba, "Network dissection: Quantifying interpretability of deep visual representations," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 6541–6549, 2017.

[9] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

[10] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[11] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1–9, 2015.

[12] M. Sundararajan, A. Taly, and Q. Yan, "Axiomatic attribution for deep networks," 2017.

[13] 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," *International Journal of Computer Vision (IJCV)*, vol. 115, no. 3, pp. 211–252, 2015.

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

[15] A. Mahendran and A. Vedaldi, "Understanding deep image representations by inverting them," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5188–5196, 2015.

[16] K. Simonyan, A. Vedaldi, and A. Zisserman, "Deep inside convolutional networks: Visualising image classification models and saliency maps," *arXiv preprint arXiv:1312.6034*, 2013.

[17] 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.

[18] A. Sharif Razavian, H. Azizpour, J. Sullivan, and S. Carlsson, "Cnn features off-the-shelf: an astounding baseline for recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pp. 806–813, 2014.

[19] 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.

[20] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *European conference on computer vision*, pp. 818–833, Springer, 2014.

[21] G. Alain and Y. Bengio, "Understanding intermediate layers using linear classifier probes," *arXiv preprint arXiv:1610.01644*, 2016.

[22] 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.

[23] A. Binder, G. Montavon, S. Bach, K.-R. Müller, and W. Samek, "Layer-wise relevance propagation for neural networks with local renormalization layers," 2016.

[24] G. Montavon, S. Lapuschkin, A. Binder, W. Samek, and K.-R. Müller, "Explaining nonlinear classification decisions with deep taylor decomposition," *Pattern Recognition*, vol. 65, pp. 211–222, 2017.

[25] R. Mottaghi, X. Chen, X. Liu, N.-G. Cho, S.-W. Lee, S. Fidler, R. Urtasun, and A. Yuille, "The role of context for object detection and semantic segmentation in the wild," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 891–898, 2014.

[26] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, and A. Torralba, "Scene parsing through ade20k dataset," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 633–641, 2017.

[27] S. Bell, K. Bala, and N. Snavely, "Intrinsic images in the wild," *ACM Transactions on Graphics (TOG)*, vol. 33, no. 4, pp. 1–12, 2014.

[28] M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, and A. Vedaldi, "Describing textures in the wild," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3606–3613, 2014.

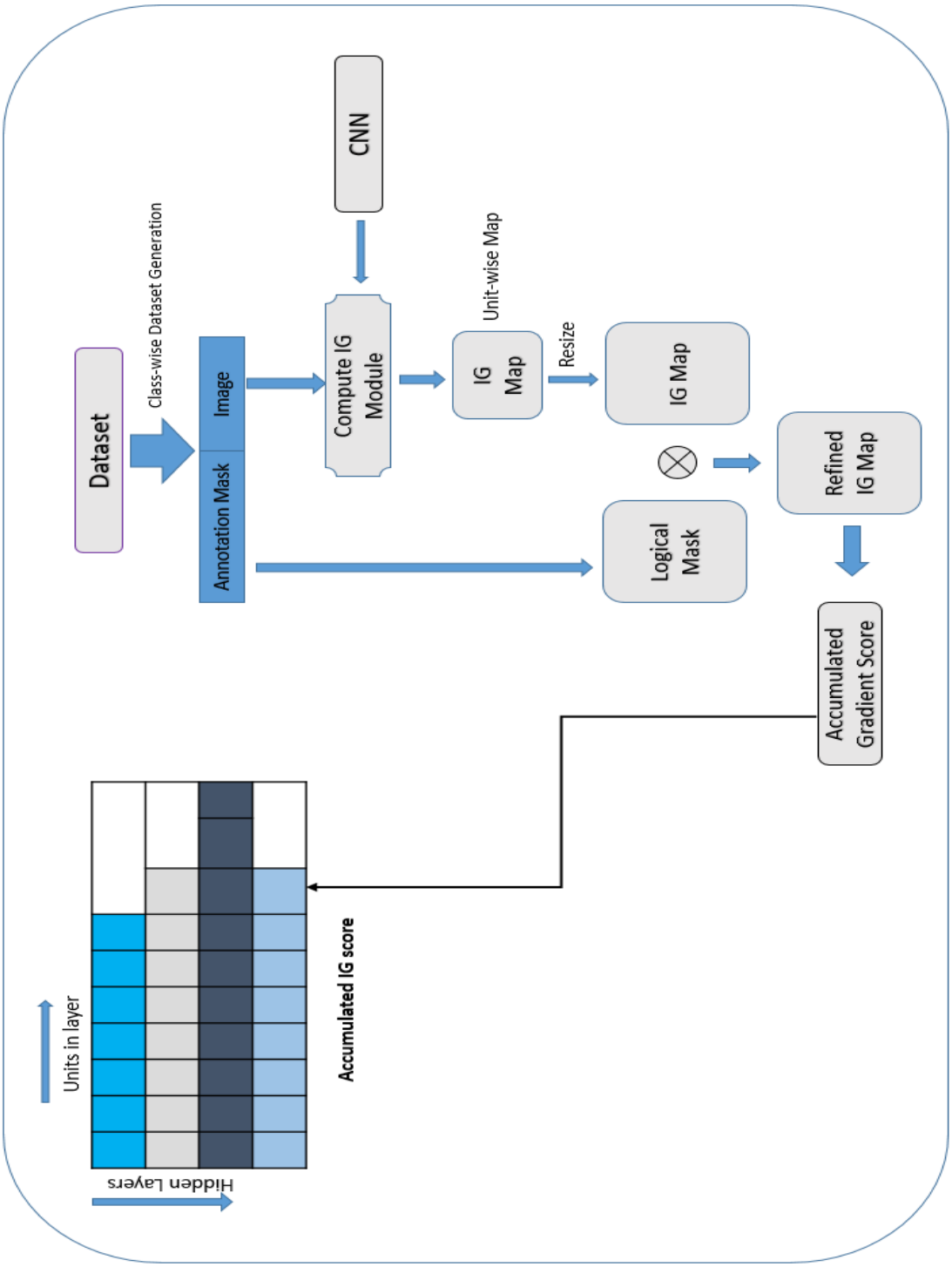
[29] J. Van De Weijer, C. Schmid, J. Verbeek, and D. Larlus, "Learning color names for real-world applications," *IEEE Transactions on Image Processing*, vol. 18, no. 7, pp. 1512–1523, 2009.

[30] V. Badrinarayanan, A. Kendall, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 12, pp. 2481–2495, 2017.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 28 January 2021 doi:10.20944/preprints202101.0579.v1

Appendices

Appendix-A : IG Computation Process



Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 28 January 2021 doi:10.20944/preprints202101.0579.v1

Appendix-B : Network Performance on 10 classes

Class	Networks											
	Alexnet				Resnet				VGG			
Bench		**Top**	**Bot**	**No Ab***		**Top**	**Bot**	**No Ab***		**Top**	**Bot**	**No Ab***
	1%	56.20	62.21		**1%**	30.17	74.21		**1%**	70.09	74.77	
	2%	48.09	59.69	63.34	**2%**	31.71	77.86	76.40	**2%**	67.72	73.76	77.57
	3%	47.77	56.53		**3%**	21.82	72.18		**3%**	66.57	73.33	
Boat	**1%**	77.44	83.52		**1%**	24.54	73.55		**1%**	86.23	90.70	
	2%	73.92	82.71	83.88	**2%**	26.52	56.34	88.35	**2%**	80.29	89.52	90.84
	3%	68.21	84.69		**3%**	7.84	43.52		**3%**	77.51	89.23	
Bottle	**1%**	58.97	59.19		**1%**	57.45	66.41		**1%**	70.44	67.48	
	2%	51.90	57.67	63.22	**2%**	41.26	66.72	72.49	**2%**	69.76	64.29	73.02
	3%	50.61	55.70		**3%**	35.64	62.77		**3%**	59.35	63.07	
Bus	**1%**	87.57	88.41		**1%**	90.87	87.72		**1%**	89.41	93.02	
	2%	84.27	84.57	89.18	**2%**	94.40	90.48	91.40	**2%**	83.65	92.40	93.09
	3%	82.19	84.65		**3%**	77.28	89.26		**3%**	71.68	91.94	
Cat	**1%**	60.66	60.85		**1%**	22.49	45.57		**1%**	67.21	66.75	
	2%	60.59	61.18	63.02	**2%**	0.85	29.64	71.87	**2%**	71.54	61.51	66.43
	3%	57.51	59.74		**3%**	0.85	28.85		**3%**	66.16	58.62	
Lamp	**1%**	24.23	24.40		**1%**	50.54	26.31		**1%**	31.06	33.22	
	2%	15.99	24.81	24.31	**2%**	62.53	24.15	35.64	**2%**	27.31	31.89	33.64
	3%	13.82	24.98		**3%**	60.28	17.99		**3%**	27.23	31.72	
Light	**1%**	72.67	76.45		**1%**	54.65	84.06		**1%**	74.37	87.40	
	2%	64.18	76.34	76.18	**2%**	43.59	82.69	85.76	**2%**	66.54	86.64	87.84
	3%	63.64	74.86		**3%**	34.17	70.54		**3%**	59.58	84.88	
Regfr	**1%**	49.10	71.24		**1%**	39.90	73.76		**1%**	64.27	76.42	
	2%	36.74	70.09	71.32	**2%**	16.89	43.42	78.36	**2%**	49.53	76.42	77.57
	3%	32.78	70.38		**3%**	9.71	31.34		**3%**	37.89	75.41	
Sea	**1%**	42.23	42.07		**1%**	19.77	50.37		**1%**	25.52	41.31	
	2%	18.47	40.34	43.03	**2%**	10.72	69.77	53.28	**2%**	15.62	40.09	41.81
	3%	15.28	38.33		**3%**	25.60	41.66		**3%**	6.63	40.22	
Vase	**1%**	45.14	48.40		**1%**	19.77	60.37		**1%**	61.26	61.85	
	2%	43.44	49.84	48.08	**2%**	6.72	69.77	43.28	**2%**	55.18	63.82	61.47
	3%	43.17	49.15		**3%**	1.60	71.66		**3%**	47.33	63.82	