

Explaining the Unexplained: A CClass-Enhanced Attentive Response (CLEAR) Approach to Understanding Deep Neural Networks

Devinder Kumar
University of Waterloo

devinder.kumar@uwaterloo.ca

Alexander Wong
University of Waterloo

a28wong@uwaterloo.ca

Graham W. Taylor
University of Guelph & CIFAR

gwtaylor@uoguelph.ca

Abstract

In this work, we propose **CClass-Enhanced Attentive Response (CLEAR)**: an approach to visualize and understand the decisions made by deep neural networks (DNNs) given a specific input. CLEAR facilitates the visualization of attentive regions and levels of interest of DNNs during the decision-making process. It also enables the visualization of the most dominant classes associated with these attentive regions of interest. As such, CLEAR can mitigate some of the shortcomings of heatmap-based methods associated with decision ambiguity, and allows for better insights into the decision-making process of DNNs. Quantitative and qualitative experiments across three different datasets demonstrate the efficacy of CLEAR for gaining a better understanding of the inner workings of DNNs during the decision-making process.

1. Introduction

In recent years, we have seen tremendous success in the field of artificial intelligence (AI). In particular, many of the recent advances have been related to one particular area of machine learning: deep neural networks (DNNs). DNNs have been shown to outperform previous machine learning techniques for a variety of tasks, such as fine-grained classification [18, 5], self-driving cars [4], captioning and answering questions about images [9, 1], and even defeating human champions at Go [11]. Although DNNs have demonstrated tremendous effectiveness at a wide range of tasks, when they fail, they often fail spectacularly, producing unexplainable and incoherent results that can leave one to wonder what caused the DNN to make such decisions. This lack of transparency and interpretability of DNNs during the decision-making process is largely due to the complex nature of DNNs, where individual neural responses, unlike other interpretable decision-making processes such as decision trees, provide very little insight as to what is actually going on.

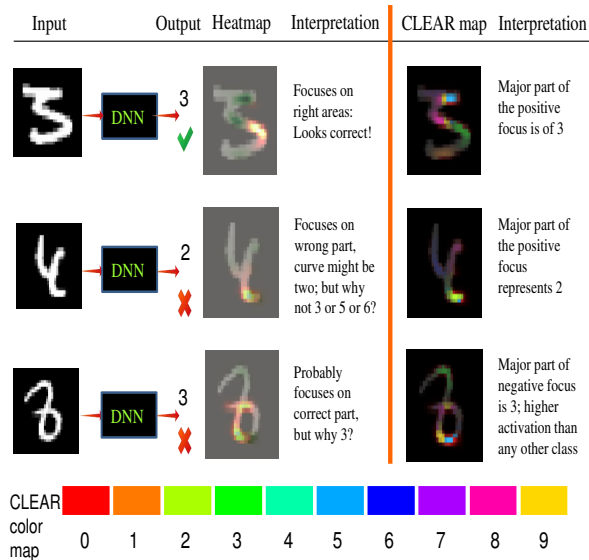


Figure 1. Examples of handwritten digits from MNIST are shown, along with: 1) the decision made by the DCNN, 2) heatmaps used in existing visualization methods, 3) the proposed **CClass-Enhanced Attentive Response (CLEAR)** maps, and 4) what can be interpreted based on the heatmaps and the proposed CLEAR maps. While the heatmaps used in existing approaches show which information in the image works for (positive focus: hot regions) or against (negative focus: green) a particular decision made, the proposed CLEAR map allows for the visualization of the attentive regions of interest, the corresponding attentive levels, as well as the dominant class for each attentive region of interest that the DCNN uses during the decision-making process. Each individual color in the CLEAR map represents the corresponding dominant attentive class at that location. Correspondence between colors and the dominant classes can be derived by the color map given at the bottom. In these examples, it is evident that the heatmaps are insufficient to fully interpret and explain the decision made by the DCNN, whereas the proposed CLEAR maps can explain the decision-making process more effectively through a multi-factor visualization approach.

The lack of transparency in the decision-making process of DNNs is a significant bottleneck in their widespread

adoption in industry, such as healthcare, defence, cybersecurity, etc., where the error tolerance is very low and the ability to interpret, understand, and trust decisions is critical. As such, a way to peer inside a DNN and see why it made a decision the way it did can have tremendous potential for pushing towards explainable AI, where a human expert gains the ability to understand, interpret and verify the decisions made.

Recently, a number of researchers have been exploring the understanding and interpretation of decisions made by DNNs, in particular by Deep Convolutional Neural Networks (DCNN), by asking the following question: *based on what information in the image is the DCNN making a decision?* To tackle this question, much recent work has focused on understanding the decision-making process of networks by leveraging heatmaps that provide information about which areas of the image is used by the DCNN to make a particular decision. These approaches have produced some promising results in revealing what is important to a decision made by a DCNN. More details regarding the relevant works are provided in Section 2. A common limitation with such heatmap-based approaches to understanding the decision-making process of DCNNs is that of **decision ambiguity**, where one can gain insight into **which** regions of interest are important for making decisions, but gives no insight as to **why** such regions of interest are important. As a result, these methods leave the “thought process” of the DCNN largely ambiguous.

In an attempt to mitigate the problem of decision ambiguity, we take a step towards “explaining the unexplained”, with regards to the decision-making process of DCNNs, through the introduction of **CLass-Enhanced Attentive Response (CLEAR)** maps that go beyond what existing heatmap-based approaches [19, 2, 10] can provide. The proposed CLEAR maps allow for the visualization of not only the attentive regions of interest and corresponding attentive levels of DCNNs during the decision-making process, but also the corresponding dominant classes associated with these attentive regions of interest. As such, compared to heatmaps, CLEAR maps are much more effective at conveying where and why certain regions of interest influence the decision-making process. An example of this is shown in Fig. 1. We further demonstrate the effectiveness of the proposed CLEAR maps, both quantitatively and qualitatively, by conducting a number of experiments using three different publicly available datasets.

2. Related Work

There has been a significant body of work in recent years in the domain of visualizing and understanding DCNNs. This literature can be broadly divided into two groups: first, approaches that focus on understanding the global structure of a trained network [3, 7, 15]; and second, approaches that

mainly focus on understanding the decision-making process of trained networks for a specific instance [14, 2, 18, 10, 12, 16, 19]. Our present work can be considered as belonging to the second category. Relevant work pertaining to each group is explained below:

Global Understanding-Based Methods: Many methods in this domain try to understand the decision-making process of the deep network by measuring its operating characteristics; for example, finding an input that maximizes the response of a particular neuron [6], measuring the network’s invariance to certain kinds of data augmentation [7], or determining global decision structure [3]. Other methods seek to find image instances from a database that maximally activate particular neurons or the posterior class probability of a given network [15].

Instance-Based Methods: These methods are based on interpreting individual decisions made by a DCNN for a particular image instance. One specific instance-based method was proposed by Simonyan et al. in [12], where using back-propagated partial derivatives of the class score with respect to pixel values were used to create class saliency maps. Zeiler & Fergus [16] proposed a deconvolution-based method to project the activations from feature space back to the input space (pixels) recursively. However, the method did not provide any meaning to the assignments other than that they should form a coherent set of interpretable pixels. Springenberg et al. [14] provided another gradient-based visualization method, which restricts the negative gradients from flowing backwards towards the input layer, leading to sharper visualization, still without attributing any meaning to the obtained visualization. However, the study strongly showed the efficacy of networks with global average pooling for image classification and visualization. To visually discern unique features for a particular category of image, Zhou et. al. [18] created a *class activation map* using DCNNs with global average pooling layers. This *class activation map* was also used for localising objects within the image. Bach et al. [2] and Montavon et al. [10] aimed at finding a general approach to visualize non-linear classifiers, leading to interesting heatmap generation. Recently, similar to the occlusion-based methodology for creating heatmaps in [16], Zintgraf et al. [19] proposed a method based on multivariate conditional sampling over image patches to visualize and interpret individual decisions of DCNNs as binary saliency maps to represent information that contributes for or against the decision.

In our work, instead of only obtaining feature maps, we attribute meaning to each pixel in the back-projected response in the input space using a class-based approach. Also, unlike [2, 10] or [19], that provide heatmaps or binary heatmaps for correctly classified samples, we create CLEAR maps that are more interpretable (Fig. 1) for both correctly or misclassified cases. Finally, compared to the

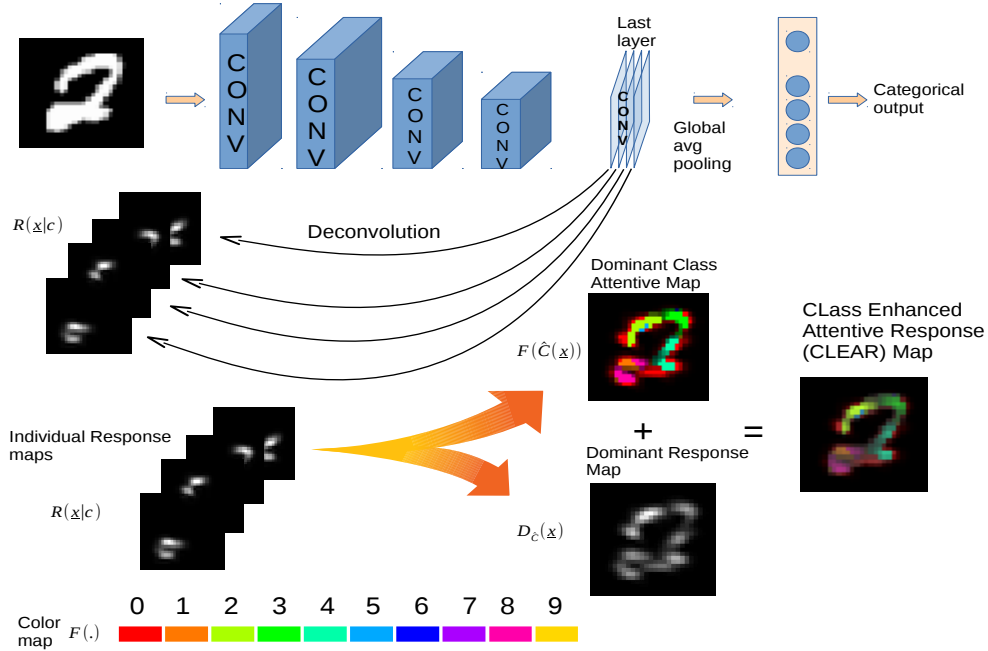


Figure 2. The procedure for generating **C**Class-Enhanced Attentive **R**esponse (CLEAR) maps. First, individual attentive response maps are computed for each class based on the last layer of the DCNN. Based on this set of attentive response maps, two different types of maps are computed: 1) a dominant attentive response map, which shows the dominant attentive level for each location in the image, and 2) a dominant class attentive map, which shows the dominant class involved in the decision-making process at each location. Finally, the dominant attentive response map and the dominant attentive class map are combined to produce the final CLEAR map for a given image.

per-class maps created in [18], CLEAR maps show multiple class-specific contributions at once.

3. Class Enhanced Attentive Response (CLEAR)

This section explains the procedure for generating the proposed **C**Class-Enhanced Attentive **R**esponse (CLEAR) maps. The main goal of CLEAR maps is to convey the following information: 1) the attentive regions of interest in the image responsible for the decision made by the DCNN; 2) the attentive levels at these regions of interest so that we understand their level of influence over the decision made by the DCNN; and 3) the dominant class associated with these attentive regions of interest so that we can better understand **why** a decision was made. The procedure for generating CLEAR maps can be summarized as follows (see Fig. 2). First, individual attentive response maps are computed for each kernel associated with a class by back-projecting activations from the output layer of the DCNN. Based on this set of attentive response maps, two different types of maps are computed: 1) a dominant attentive response map, which shows the dominant attentive level for each location in the image; and 2) a dominant class attentive map, which shows the dominant class involved in

the decision-making process at each location. Finally, the dominant attentive response map and the dominant attentive class map are combined visually by using color and intensity to produce the final CLEAR map for a given image.

Inspired by the effectiveness of the ALL-CNN [14] on different datasets, we leveraged a similar network architecture for building the DCNN used for classification in this paper. While, for clarity, we describe the procedure for computing individual attentive response maps based on the ALL-CNN architecture, the procedure will generalize to other DCNNs provided class-specific responses can be computed in input (pixel) space. ALL-CNNs are composed primarily of convolutional, ReLU, and max-pooling layers. Towards the output of the DCNN, the last convolutional layer contains a set of kernels equal to the number of classes, and then global averaging is performed before passing these energy values to the softmax output layer which represents categories. As such, each kernel can be thought of as being associated with a particular class.

The first step of CLEAR is to compute a set of individual attentive response maps, one for each of the classes learned by the DCNN, which we will denote as $\{R(\underline{x}|c)|1 \leq c \leq N\}$, where N is the number of classes. This is achieved in the current realization of CLEAR by back-propagating the responses of each kernel in the last

convolutional layer from feature space to the input space to form each attentive response map, thus extending upon the idea first introduced in [17]. To explain the formulation for the formation of CLEAR maps, first consider a single layer of a DCNN. Let \hat{h}_l be the deconvolved output response of the single layer l with K kernel weights w . The deconvolution output response at layer l then can be then obtained by convolving each of the feature maps z_l with kernels w_l and summing them as:

$$\hat{h}_l = \sum_{k=1}^K z_{k,l} * w_{k,l}. \quad (1)$$

Here $*$ represents the convolution operation. For notational brevity, we can combine the convolution and summation operation for layer l into a single convolution matrix G_l . Hence the above equation can be denoted as: $\hat{h}_l = G_l z_l$.

For multi-layered DCNNs, we can extend the above formulation by adding an additional un-pooling operation U as described in [17]. Thus, we can calculate the deconvolved output response from feature space to input space for any layer l in a multi-layer network as:

$$R_l = G_1 U_1 G_2 U_2 \dots G_{l-1} U_{l-1} G_l z_l \quad (2)$$

For CLEAR maps, we specifically calculate the output responses from individual kernels of the last layer of a network. Hence, given a network with last layer L containing $K = N$ kernels, we can calculate the attentive response map; $R(\underline{x}|c)$ (where \underline{x} denotes the response back-projected to the input layer, and thus an array the same size as the input) for any class-specific kernel c ($1 \leq c \leq N$) in the last layer as:

$$R(\underline{x}|c) = G_1 U_1 G_2 U_2 \dots G_{L-1} U_{L-1} G_L^c z_L. \quad (3)$$

Here G_L^c represents the convolution matrix operation in which the kernel weights w_L are all zero except that at the c^{th} location.

Given the set of individual attentive response maps, we then compute the dominant attentive class map, $\hat{C}(\underline{x})$, by finding the class at each pixel that maximizes the attentive response level, $R(\underline{x}|c)$, across all classes:

$$\hat{C}(\underline{x}) = \underset{c}{\operatorname{argmax}} R(\underline{x}|c). \quad (4)$$

Given the dominant attentive class map, $\hat{C}(\underline{x})$, we can now compute the dominant attentive response map, $D_{\hat{C}}(\underline{x})$, by selecting the attentive response level at each pixel based on the identified dominant class, which can be expressed as follows:

$$D_{\hat{C}}(\underline{x}) = R(\underline{x}|\hat{C}). \quad (5)$$

To form the final CLEAR map, we map the dominant class attentive map and the dominant attentive response map in the HSV color space as follows, then transform back into the RGB color space:

$$\begin{aligned} H &= F(\hat{C}(\underline{x})), \\ S &= 1, \\ V &= D_{\hat{C}}(\underline{x}). \end{aligned} \quad (6)$$

Here $F(\cdot)$ is the color map dictionary that assigns an individual color to each dominant attentive class, c . Fig. 2 shows an example of the CLEAR map overlaid on the image.

4. Experiments

In this section, we illustrate the efficacy of CLEAR maps for understanding and interpreting the decision-making of DCNNs. We conducted qualitative and quantitative experiments on three different datasets: the commonly used benchmarks MNIST and Street View House Numbers (SVHN), and the Stanford Dog dataset [8]. In the following section, we explain the experimental setup.

4.1. Setup

To conduct experiments on three different datasets, we trained three different DCNN architectures with all convolutional layers. For training on MNIST and SVHN, we set our network architecture similar to [14], as it has shown to perform very effectively for a variety of datasets. To train these networks, we used the default train and test split. We achieved an accuracy of 99.26% and 92.6% for the MNIST and SVHN datasets, respectively. For training on the Stanford dog dataset, we used a 16 layer VGG net [13], pre-trained on ImageNet. We modified the VGG net slightly by removing the two last fully-connected layers and augmenting with two convolutional layers at the end. We fine-tuned the last two layers using the Stanford dog dataset. As with MNIST and SVHN, the default train and test split was applied to this dataset, but instead, we only took 10 different classes for training. We made this decision, as it would be an arduous task to interpret from all 120 classes. For this fine-grained classification task, we achieved an accuracy of 58.74% for 10 classes, whereas the state-of-the art for this dataset with 120 classes is 68%.

In all three networks, as the last layer (convolutional layer) was linearly connected to the softmax activation function, each kernel can be considered to represent one separate class. It is important to note that the aim was to understand and interpret the decision of a trained network; hence we did not strive to achieve the best architecture and state-of-the art results for each dataset. Using the previously mentioned setup, we conducted the following experiments.

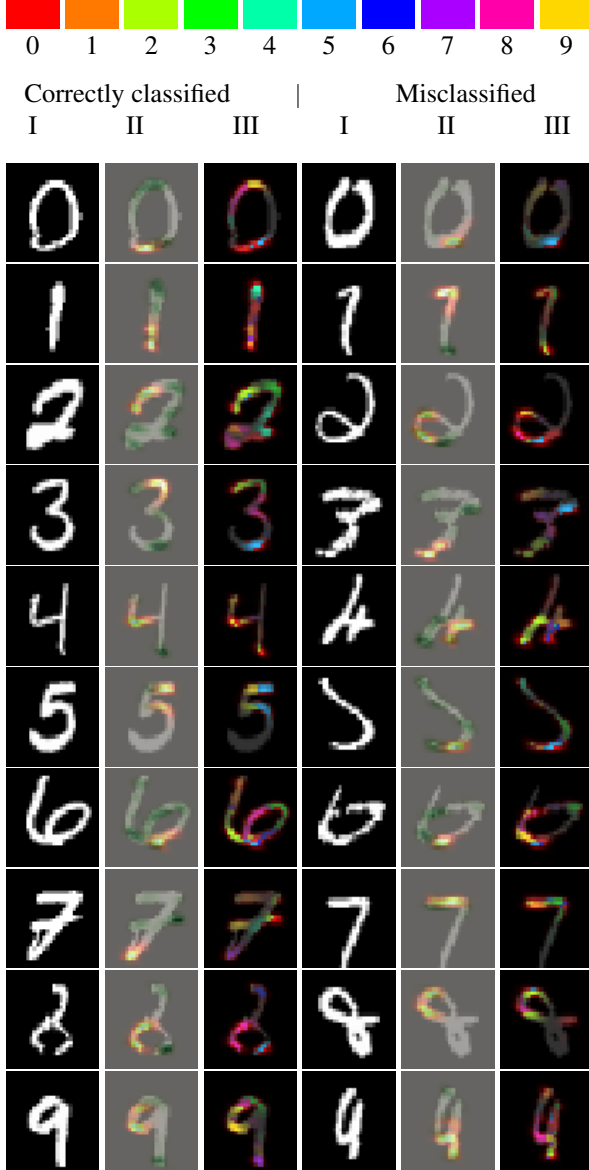


Figure 3. Example images from the MNIST dataset. Each row represents two sets of examples for digit 0-9: correctly classified example (left) and misclassified example (right). Each example set consists of the (I) original image, (II) heatmap results (where hot regions are focus of positive kernel and green represents dominant pixel results for the rest of kernels in the last layer) and (III) CLEAR maps. The color map on top shows the associations of different colors with their respective classes in the CLEAR map.

4.2. Qualitative Experiments: Understanding The Decision Making Process

In this set of experiments, we first create binary heatmaps and the proposed CLEAR maps for individual images in the three different datasets. The binary heatmaps represent which information in the image was used for or against the true class versus other image classes during classification.

The binary heatmaps were formed by overlaying the output response from the kernel representing the true class as “hot” regions and response of the rest of the kernels in the last layer, represented by green regions. The response for rest of the kernels is formed by performing max operation across the individual output responses. Thus, in the binary heatmaps, the hot regions and green regions represent the information for and against the actual class respectively, that was used for decision-making by the network. The binary heatmaps are constructed similarly to [19] and [10]. The CLEAR map formation is explained in Section 3 and Fig. 2.

For the SVHN and Stanford datasets, we also create an additional binary map. This map replaces the varying values in the binary heatmaps with a constant value. In the binary map, red and blue regions represent the information used for and against the class, respectively. We create these maps for visual clarity, as sometimes it is harder to visualize the green regions in the binary heatmaps.

MNIST: Some of the randomly chosen results for the MNIST dataset are shown in Fig. 3. This figure shows examples of correctly classified and misclassified examples by the network. From these results, observations that can be made are: 1) Looking at the example sets for digit 0, although positive support is contributed by the same bottom curved features in both examples, only in one case is the image correctly identified as zero. Looking at the CLEAR maps, we can see the dominant activations for the correctly classified example corresponds to class 0, whereas for the misclassified case they correspond to class 5. 2) Similarly, for digit 7 and 8 it is difficult to interpret the decision output of the DCNN, but looking at the CLEAR maps make them more interpretable.

SVHN: Presented similarly to the MNIST dataset, results obtained for the SVHN dataset are shown in Fig. 4. Some interesting observations are as follows: 1) For the misclassified 0 digit, the heatmap overwhelmingly focuses on the correct curves; but the network still misclassifies it. This is counterintuitive to human interpretation. But when observing the CLEAR maps, we see that almost all the strong activations are for classes other than 0. 2) For the digit 9, it is difficult to interpret the binary heatmaps, as the positive kernel focuses on the digit 1, but it still correctly classifies the digit as 9 with high confidence. Observing the CLEAR maps, we see that most of the dominant activation in the focus areas belong to digit 9, including the ones for digit 1.

Stanford Dog dataset: Results for Stanford Dogs are shown in Fig. 5–7. The key observations are the following: 1) Binary heatmaps can be used to find strong identifying features for different classes, as shown in Fig. 6. 2) In the same figure, the rightmost misclassified cases present an interesting observation. We can observe that the network misclassifies the Chihuahua breed as Shiz-Tzu and the Ridge-

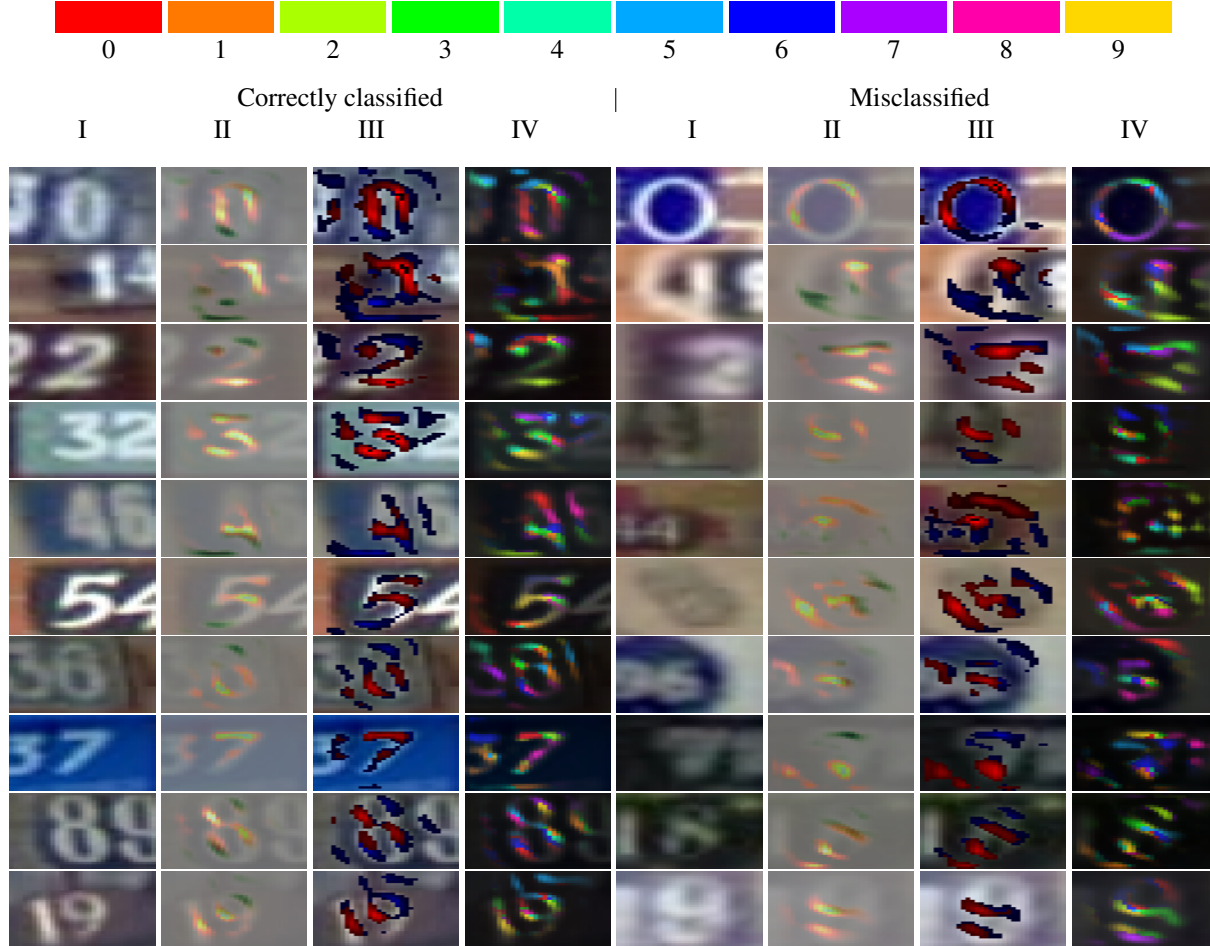


Figure 4. Correctly classified (left) and misclassified (right) images from the SVHN dataset. Each row represents two sets of examples for digit 0-9. Each example set consists of the (I) original image, (II) heatmap results (where hot regions are focus of positive kernels, and green regions for the rest of kernels), (III) binary map (red represents information for and blue represents information against the given image class) and (IV) CLEAR map respectively. The color map at the top shows the associations of different colors with their respective classes in the CLEAR map.

back breed as Afghan Hound. This happens even when the *positive* kernels associated with their respective true class focus on the strong discriminating features, as identified by correctly classified images on the left. In Fig. 7, the CLEAR maps show that for Chihuahuas, the strong activations are for the Shih-Tzu, whereas for the Ridgeback, the activations are stronger for Afghan Hounds.

Based on these results and observations, it is evident that binary maps are not enough for interpreting and explaining the individual decision outputs of a network. There is a strong motivation for class-based maps, such as CLEAR maps, that are more effective for understanding and interpreting the classification decisions made by a DCNN.

4.3. Quantitative Experiments

To re-validate our observations for the MNIST and SVHN datasets, we conducted two different quantitative ex-

periments. In the first experiment, we removed all parts of the image, except for regions responsible for the activations of the kernel associated with the class of the image (*positive kernel*). We call these regions *strong features* associated with the class. For the MNIST dataset, we replace the digit with the background and for the SVHN dataset, we replace the region with a gray patch. In the second experi-

Table 1. Evaluation to re-validate the effectiveness and contribution of identified strong features on accuracy.

Accuracy(%)	MNIST	SVHN
Full image	99.26	92.60
with only strong features	79.89	69.12
without strong features	43.45	54.46

ment, we do the opposite: we remove the regions responsible for the kernel associated with the true class of the input

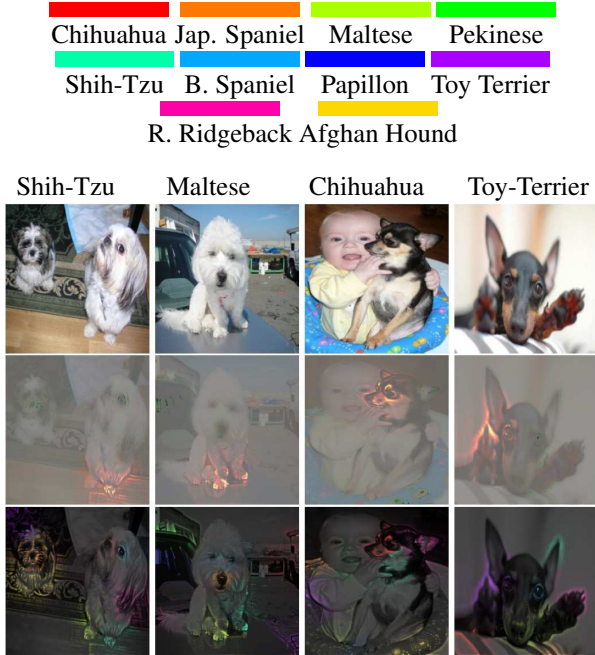


Figure 5. Multi-class maps for four different classes (Shih-Tzu, Maltese, Chihuahua and Toy-Terrier). In the last row, each color in the CLEAR maps represents a distinct class (breed) which can be identified by the color map at the top of the figure.

image and keep the rest of the image. Results are shown in Table 1, and demonstrate that the identified strong features are vital for correctly classifying a particular class. For the case where the network is still able to classify without the strong features, albeit with half of the accuracy in comparison to the above case, an argument can be made that for these cases, the network focuses again on similar or redundant features. An example is digit 3, where there are redundant strong curve features.

5. Discussion

This section discusses some general points associated with the CLEAR maps approach: 1) It is interesting to note that in Fig. 2, there is sparsity in the individual response maps from the last layer kernels. We observed the same pattern for all datasets considered. Evidence for classes tends to come from very specific localized regions. 2) For the datasets with a large number of classes, like the Stanford Dog dataset, we selected and created CLEAR maps with only 10 classes. We didn't strive to show the CLEAR maps for all 120 classes, as doing so would make it extremely difficult to interpret the decision outputs. For such cases, perhaps showing the top 10 most activated class or several different maps with N classes would be a better approach. 3) In the current realization of our approach, we use deconvolution responses with only fully convolutional networks. We would like to point out that even though end-to-end

learning in this case is only possible with Fully Convolutional Nets (FCN), our approach can be extended to be used with different network architectures with the use different response methods, such as Layer-wise Relevance Propagation (LRP) [2], Deep Taylor decomposition [10], or prediction differential analysis [19].

6. Conclusion

In this work, a novel approach to better understanding and visualizing the decision-making process of DNNs was introduced in the form of **CL**ass-**E**nhanced **A**ttentive **R**esponse (CLEAR) maps. CLEAR maps are designed to enable the visualization of not only the areas of interest that predominantly influence the decision-making process, but also the degree of influence as well as the dominant class of influence in these areas. This multi-faceted look at the decision-making process allows for a better understanding of not only where but why certain decisions are made by DCNNs compared to existing heatmap-based approaches. Experiments using three different publicly available datasets were performed and show the efficacy of CLEAR maps both quantitatively and qualitatively. Furthermore, we demonstrated that strong areas of interest identified with CLEAR maps play a pivotal role in the correct classification of the class. Future work will explore extending CLEAR to facilitate for scenarios characterized by a large number of classes (i.e. greater than 10), as well as exploring CLEAR with different network architectures.

Acknowledgment

This research has been supported by Canada Research Chairs programs, Natural Sciences Engineering Research Council of Canada (NSERC), and Canada Foundation for Innovation (CFI).

References

- [1] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. Lawrence Zitnick, and D. Parikh. Vqa: Visual question answering. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2425–2433, 2015. 1
- [2] 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*, 10(7):e0130140, 2015. 2, 7
- [3] D. Baehrens, T. Schroeter, S. Harmeling, M. Kawanabe, K. Hansen, and K.-R. Mäzler. How to explain individual classification decisions. *Journal of Machine Learning Research*, 11(Jun):1803–1831, 2010. 2
- [4] M. Bojarski, D. Del Testa, D. Dworakowski, B. Firner, B. Flepp, P. Goyal, L. D. Jackel, M. Monfort, U. Muller, J. Zhang, et al. End to end learning for self-driving cars. *arXiv preprint arXiv:1604.07316*, 2016. 1

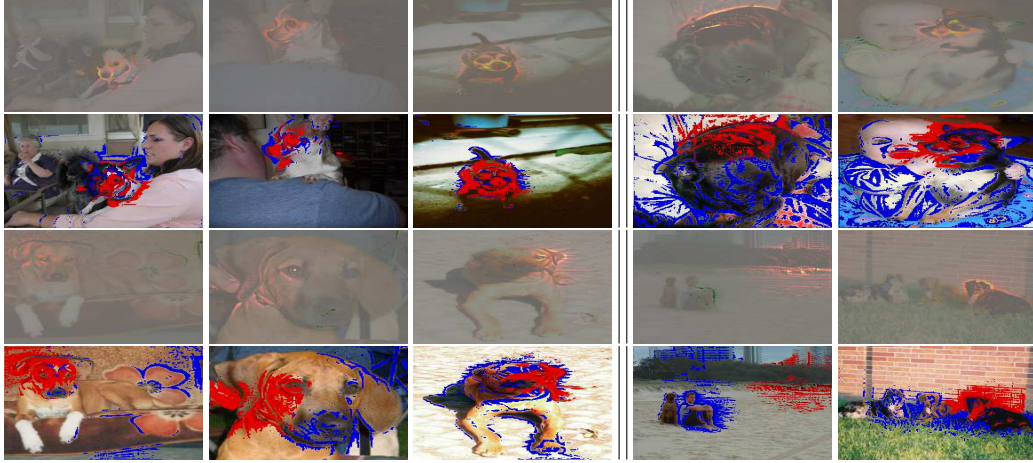


Figure 6. Three correctly classified examples (left), and two misclassified (right), for two different breeds of dogs. It can be observed that for the Chihuahua breed (top), face structure is used as a strong identifier. But we can also observe that for one of the examples (last image), even when the network focuses on correct features, it fails to classify correctly and thinks it is another breed, Shih-Tzu (for explanation, please see Fig 7). For the Rhodesian Ridgeback breed (bottom), the face, along with ears, are a strong identifier, but it fails to identify the breed while looking at the same features again in the rightmost image.

■ ■ ■ ■
 Chihuahua Jap. Spaniel Maltese Pekinese
■ ■ ■ ■
 Shih-Tzu B. Spaniel Papillon Toy Terrier
■ ■
 R. Ridgeback Afghan Hound



Figure 7. Heatmap, binary map, and CLEAR map for two misclassified images are shown. From the heatmap and binary map, it can be seen that the network correctly focuses on the strong identifying features of the Chihuahua and Ridgeback class, as shown in Fig 6, but still fails to classify the image correctly. The CLEAR maps in the third column show that activations for the correct Chihuahua class (red) is smaller than the activation for Shih-Tzu (light blueish), hence the network thinks that it is a Shih-Tzu instead of a Chihuahua. Similarly for the Ridgeback class, the activations of Afghan Hound are higher than the true class's activation.

[5] Y. Cui, F. Zhou, Y. Lin, and S. Belongie. Fine-grained categorization and dataset bootstrapping using deep metric learning with humans in the loop. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1153–1162, 2016. 1

[6] D. Erhan, A. Courville, and Y. Bengio. Understanding representations learned in deep architectures. *Department d'Informatique et Recherche Operationnelle, University of Mon-*

real, QC, Canada, Tech. Rep., 1355, 2010. 2

[7] I. Goodfellow, H. Lee, Q. V. Le, A. Saxe, and A. Y. Ng. Measuring invariances in deep networks. In *Advances in neural information processing systems*, pages 646–654, 2009. 2

[8] A. Khosla, N. Jayadevaprakash, B. Yao, and L. Fei-Fei. Novel dataset for fine-grained image categorization. In *First Workshop on Fine-Grained Visual Categorization, IEEE Conference on Computer Vision and Pattern Recognition*, Colorado Springs, CO, June 2011. 4

[9] M. Malinowski, M. Rohrbach, and M. Fritz. Ask your neurons: A neural-based approach to answering questions about images. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1–9, 2015. 1

[10] G. Montavon, S. Lapuschkin, A. Binder, W. Samek, and K.-R. Müller. Explaining nonlinear classification decisions with deep taylor decomposition. *Pattern Recognition*, 65:211–222, 2017. 2, 5, 7

[11] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *Nature*, 529(7587):484–489, 2016. 1

[12] K. Simonyan, A. Vedaldi, and A. Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2013. 2

[13] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 4

[14] J. T. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller. Striving for simplicity: The all convolutional net. *arXiv preprint arXiv:1412.6806*, 2014. 2, 3, 4

[15] J. Yosinski, J. Clune, A. Nguyen, T. Fuchs, and H. Lipson. Understanding neural networks through deep visualization. *arXiv preprint arXiv:1506.06579*, 2015. 2

- [16] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pages 818–833. Springer, 2014. [2](#)
- [17] M. D. Zeiler, D. Krishnan, G. W. Taylor, and R. Fergus. Deconvolutional networks. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 2528–2535. IEEE, 2010. [4](#)
- [18] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba. Learning deep features for discriminative localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2921–2929, 2016. [1](#), [2](#), [3](#)
- [19] L. M. Zintgraf, T. S. Cohen, T. Adel, and M. Welling. Visualizing deep neural network decisions: Prediction difference analysis. *ICLR 2017 (accepted)*, 2017. [2](#), [5](#), [7](#)