Contributions by Feature Layers in Two-Class Deep Learning Image Classification Decisions

Debanjali Banerjee
School of Computing and Informatics
University of Louisiana at Lafayette
U.S.A.
Email: debanjali.banerjee1@louisiana.edu

Chee-Hung Henry Chu
School of Computing and Informatics
University of Louisiana at Lafayette
U.S.A.
Email: chu@louisiana.edu

***Abstract*—Deep learning methods have excellent accuracy achievements in image classification but largely remains a black box method. Image classification is the core of many machine vision tasks, including object detection. Better understanding of how the classification decision is made will improve the understanding of such tasks as object detection. In this work, we train a deep learning network to classify between two classes. We compute the so-called SHapley Additive exPlanations (SHAP) values for the feature layers using input images against a population of other training images for the classification layer. The SHAP value is a special case of the Shapley value which explains the factors in a machine learning decision by measuring the output change due to change in each factor. The SHAP value is the Shapley value satisfying local accuracy, missingness, and consistency properties. Experimental results show the different responses from the lowest to the highest feature extraction layers.**

***Index Terms*—Deep Learning, Explainable Artificial Intelligence, Shapley Values, Image Classification.**

I. Introduction

Image classification was the task that sparked the research interest in deep learning for the past ten years. In image classification, an image is classified to one of several classes. Instead of using custom crafted feature vectors, a deep learning approach uses different, increasingly complex convolutional networks to extract images features as input to a classification layer. Image classification forms the core of more complex artificial intelligence systems such as object detection. As such, a better understanding of the image classification task might lead to better understanding of such tasks as object detection. Within three years of the publication of the AlexNet [1], deep learning systems were already shown to perform better than humans in image classification. Despite such advances, deep learning systems remain black boxes and little is known as to how they make decisions.

A number of recent works have been reported in developing methods that address how to explain complex machine learning systems. A recent survey [2] highlighted domain-dependent and context-specific methods for dealing with the interpretation of artificial intelligence systems, including the boundaries and gaps of recent advances. One approach is to remove a feature and then assessing the output change. The complications in computing are due to the large number of permutations involved in the remaining features that must be averaged to determine the Shapley values. In [3] the Shapley values were optimized to include the local accuracy, missingness, and consistency properties to form the so-called SHapley Additive exPlanations (SHAP) values. A different approach [4] is the synthesis of Local Interpretable Model-agnostic Explanations (LIME). Other methods that focus on the image space [5] often lead to solutions that overlap with image saliency research [6].

In our work, we are interested in determining the contributions of the features in the large hidden layers. We compute the SHAP values to understand the contributions of individual features not at the image level but at the hidden, feature levels. One reason for doing so is to validate that all features in a deep learning network make contributions to decisions.

In Section II, we describe the use of Shapley values to explain the contributions of individual features in a deep learning network. In Section III, we describe our experiments and the results. Finally, we draw our conclusion in Section IV.

II. Methodology

We assess the contributions of feature values in a deep learning network that has been trained to perform image classification. We describe our approach in assessing the contributions of the feature extraction stage of a deep learning network in Section II-A. We compute the SHAP values, which is a special case of the Shapley values. In Section II-B we present an overview of the Shapley values that we use for calculating the contributions.

A. Deep Learning Features

A convolutional neural network broadly speaking has an input layer, convolutional layers, max or average pooling layers, fully connected layers, and an output layer. The convolutional and max pooling layers are

 ISBN: 978-1-61208-953-9

typically in the feature extraction stage while the fully connected layers and an output layer are in the classification stage. A convolutional layer takes as input a block of feature values $F^l(i,j,k)$ where l is the layer index, $i = 0,\cdots,M_l-1$, $j = 0,\cdots,N_l-1$, and $k = 0,\cdots,K_l-1$, so that the block of feature values is of size $M_l \times N_l \times K_l$; typically $M_l = N_l$. The input feature block is transformed by a bank of convolutional kernels and associated nonlinearity to produce an output block $F^{l+1}(i,j,k)$ of size $M_{l+1} \times N_{l+1} \times K_{l+1}$ where M_{l+1}, N_{l+1} are controlled by the strides of the convolution step and K_{l+1} is controlled by the number of convolutional kernels used.

In machine learning, we use a training data set to determine the convolutional kernel coefficient values in a deep learning network. After training, when we apply an image as input, the feature values are calculated at all levels and an output decision is made. A question is: what is the contribution of a particular feature value in the decision? Our approach to answering this is to compute the SHAP value for the feature values $F^l(i,j,k)$ $i = 0,\cdots,M_l-1$, $j = 0,\cdots,N_l-1$, $k = 0,\cdots,K_l-1$, and l for the layers in the feature extraction stage.

B. Shapley and SHAP Values

Consider a simple example where there are four inputs x_0, x_1, x_2, and x_3 to a system f that provides an output $y = f(x_0,x_1,x_2,x_3)$. Suppose we would like to assess the contribution of x_0. An intuitive way to do so is to turn off x_0 and consider the change—either an increase or a decrease—to the output value. Let $y_0 = f(0,x_1,x_2,x_3)$ and $\delta_0 = y - y_0$. We can repeat this process to find δ_i for $i = 1,2,3$. The value δ_i would be an indicator of the contribution of x_i to the output y. When a training set is available, we feed the entire data set and average the output, so that $\overline{\delta_0} = \overline{y_0} - \bar{y}$, where $\overline{y_0}$ and $\bar{y}$ are the outputs with x_0 turned off and on, respectively, averaged over the data set. This intuitive example, however, does not consider the other cases, such as when other combinations of features are turned off.

A more comprehensive approach as illustrated by our example is as follows. Let the set of all features be F; in our example $F = \{x_0,x_1,x_2,x_3\}$. If we exclude an input, say x_0, from F, to study the contribution of x_0 to the output, then we need to consider all combinations of the remaining features in $F\backslash\{x_0\}$. The set of all such combinations is the power set of $F\backslash\{x_0\}$, $2^{F\backslash\{x_0\}} = \{\phi,\{x_1\},\{x_2\},\{x_3\},\{x_1,x_2\},\{x_1,x_3\},\{x_2,x_3\},\{x_1,x_2,x_3\}\}$. Let $S \in 2^{F\backslash\{x_0\}}$ so that S is a subset of features excluding x_0. Let f_S be the output f trained using the subset of features S. The difference $f_{S\cup\{x_0\}} - f_S$ is a measure of how much the output changes with the inclusion of x_0 relative to S, a subset of the features. We can then compute a weighted sum of the change in output over all possible subsets of $F\backslash\{x_0\}$. The weights are set such that they sum to 1 and that all weights associated with subsets with the same cardinality are equal. Let $C(n,k)$ denote the number of combinations in choosing k from n items. There are $C(|F|,|S\cup\{x_0\}|) = C(|F|,|S|+1)$ combinations of choosing subsets $S\cup\{x_0\}$ from F. Each such combination has $|S|+1$ terms so that the total number of terms in the sum is $(|S|+1)C(|F|,|S|+1)$. The change in output is then weighted by the reciprocal of the number of terms

$$\frac{1}{(|S|+1)C(|F|,|S|+1)}.$$

The weight expands to

$$\frac{(|S|+1)!(|F|-|S|-1)!}{(|S|+1)|F|!} = \frac{|S|!(|F|-|S|-1)!}{|F|!},$$

so that the weighted average is given by

$$\sum_{S\subset F\backslash\{x_0\}} \frac{|S|!(|F|-|S|-1)!}{|F|!}(f_{S\cup\{x_0\}} - f_S).$$

We note that to calculate the Shapley value for one of the $|F|$ features, we need to build $2^{|F|}$ different models.

The SHAP value is a special case of the Shapley value satisfying local accuracy, missingness, and consistency properties. It uses a number of so-called "Explainers" to possibly approximate the values for determining the contributions by feature values.

III. Experimental Results

We ran experiments to validate our methodology of determining the contributions of feature layers in a two-class deep learning image classifier. We describe our data set in Section III-A. We picked a standard image classifier, viz. the VGG 19 network, as described in Section III-B. We describe how we compute the SHAP values in Section III-C and show the results in Section III-D.

A. Data Set

We used the data set from the Kaggle cat-vs-dog competition [7]. The goal is to distinguish between a cat vs. a dog in the given input image. The images were of different aspect ratios, different sizes, and most images include background. Examples are shown in Figure 1. We used a balanced training set of 2,000 cats and 2,000 dogs to train the classifier layer. We used data augmentation that includes rotation and scaling.

B. The Classifier

We trained a VGG 19-based network [8] with the top classification layer replaced by a two-class classifier (Table I). Though the VGG network architecture is well-known, we include the summary here to refer to the layer levels in Section III. The classification layer takes the 512×7×7 tensor of the features and feeds

 ISBN: 978-1-61208-953-9

them to a layer of 64 hidden units and then to two output units, corresponding to the two classes. We used the ReLU nonlinearity in all neurons, except for the output units. We used transfer learning so that the lower, feature extraction layers with 20,024,384 weights were trained with the ImageNet data set. The top classification layers with 1,605,826 weights were trained with the application-specific data set of cats and dogs. We used dropout in training. The training accuracy was around 95% while the state-of-the-art was above 98%. Our goal here was not in achieving the ultimate in accuracy but to obtain a "good enough" architecture for analysis of the contributions of each of the feature layers.

TABLE I: SUMMARY OF THE VGG 19 NETWORK FOR CLASSIFICATION.

Layer Number	Type	Output Shape
1	InputLayer	[(None, 224, 224, 3)]
2	Conv2D	(None, 224, 224, 64)
3	Conv2D	(None, 224, 224, 64)
4	MaxPooling2D	(None, 112, 112, 64)
5	Conv2D	(None, 112, 112, 128)
6	Conv2D	(None, 112, 112, 128)
7	MaxPooling2D	(None, 56, 56, 128)
8	Conv2D	(None, 56, 56, 256)
9	Conv2D	(None, 56, 56, 256)
10	Conv2D	(None, 56, 56, 256)
11	Conv2D	(None, 56, 56, 256)
12	MaxPooling2D	(None, 28, 28, 256)
13	Conv2D	(None, 28, 28, 512)
14	Conv2D	(None, 28, 28, 512)
15	Conv2D	(None, 28, 28, 512)
16	Conv2D	(None, 28, 28, 512)
17	MaxPooling2D	(None, 14, 14, 512)
18	Conv2D	(None, 14, 14, 512)
19	Conv2D	(None, 14, 14, 512)
20	Conv2D	(None, 14, 14, 512)
21	Conv2D	(None, 14, 14, 512)
22	MaxPooling2D	(None, 7, 7, 512)
23	Flatten	(None, 25088)
24	Dense	(None, 64)
25	Dropout	(None, 64)
26	Dense	(None, 2)

C. SHAP Calculation

We used the publicly available SHAP package [9] in the Python environment. The SHAP package needs a sub-population to average over. We randomly selected 25 cat images and 25 dog images from the training set. We iteratively computed the SHAP values of the feature layer tensors from Layer 1 to Layer 22, which made up the feature extraction layers of the trained network. Each feature layer was a tensor with different spatial resolutions and depths that depended on the block. We visualized the SHAP values by scaling both the values and the spatial resolution to the input image. It was not practical to display the many different individual feature maps in a tensor; we summed the values across channels in a tensor to form a composite feature map.

D. Results

We used four input images for classification as shown in Figure 1. All four images were correctly classified by the deep learning classifier. Each image was fed as

Fig. 1: Input pictures for classification "cat3238" (top left); "cat3333" (top right); "dog3333" (lower left); "dog3399" (lower right).

input to the SHAP values calculation individually; i.e., they were not processed as a batch. Hence when, e.g., "cat3238" was used as the input, the other 3 images were not processed by the network. We loaded the network from its trained state for every round of SHAP values calculation.

We show the Shapley values for the feature map tensors at each layer when the input was "cat3333". In the following, the negative values are mapped to blue while the positive values are mapped to red. We overlay the SHAP values on a grayscale version of the input image.

In Figure 2, we show the feature maps when the spatial resolution were 224×224. In layers 2 and 3 we can see outlines of dogs in parts of the image that had very little color variation. Next we show the feature maps when the spatial resolution were 112×112 in Figure 3. The outlines of dogs can be seen in layers 4 and 5 as well.

(a) Layer 1 (b) Layer 2 (c) Layer 3

Fig. 2: SHAP values for layers 1, 2, and 3 when the input was "cat3333".

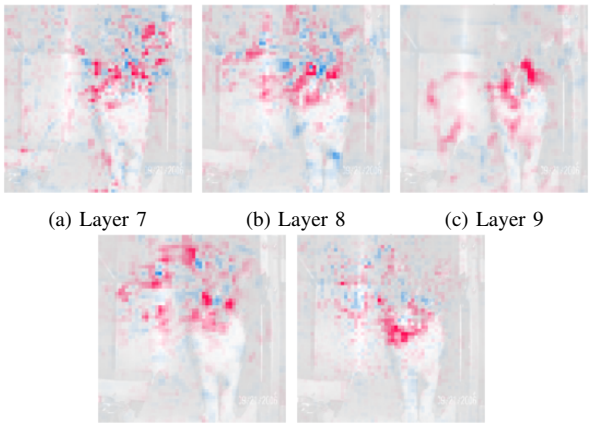
The feature maps when the spatial resolution were 56×56 in Figure 4. At this resolution, the high values

 ISBN: 978-1-61208-953-9

(a) Layer 4 (b) Layer 5 (c) Layer 6

Fig. 3: SHAP values for layers 4, 5, and 6 when the input was "cat3333".

were better localized to the cat in the input image, even though there were still responses to the area (left of the cat in the image) with low color variations. The feature maps when the spatial resolution were 28×28 and 14×14 in figures 5 and 6, respectively. We show the feature map at the top of the feature extraction layers when the spatial resolution was 7×7, in Figure 7. We can see that at the lower spatial resolution, the "shape" information is less evident and the focus was more on the face of the cat in the input image.



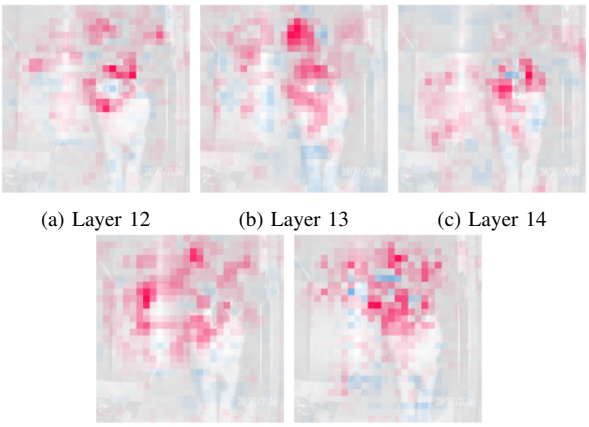
(a) Layer 7 (b) Layer 8 (c) Layer 9

(d) Layer 10 (e) Layer 11

Fig. 4: SHAP values for layers 7 to 11 when the input was "cat3333".

We repeated the calculations for the other 3 input images but do not show the full sets of results here in the interest of brevity. We wanted to explore the phenomenon of having "imagined" faces in the lower layers. In figures 8, 9, and 10, we show the responses at layers 2, 3, 5, and 6 for the other three input images. In all figures, we saw some outlines of other faces, some stronger (Figure 8) and some less (Figure 10).

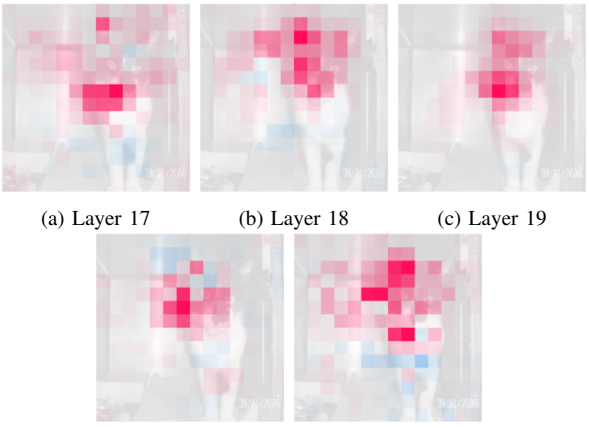
Given that we used transfer learning, we next compared the input images to images that were in the ImageNet data set that was used to train the feature extraction layers. We replaced the 50 cat and dog images by 50 images randomly selected from the ImageNet database in the SHAP value computations. In Figure 11,



(a) Layer 12 (b) Layer 13 (c) Layer 14

(d) Layer 15 (e) Layer 16

Fig. 5: SHAP values for layers 12 to 16 when the input was "cat3333".



(a) Layer 17 (b) Layer 18 (c) Layer 19

(d) Layer 20 (e) Layer 21

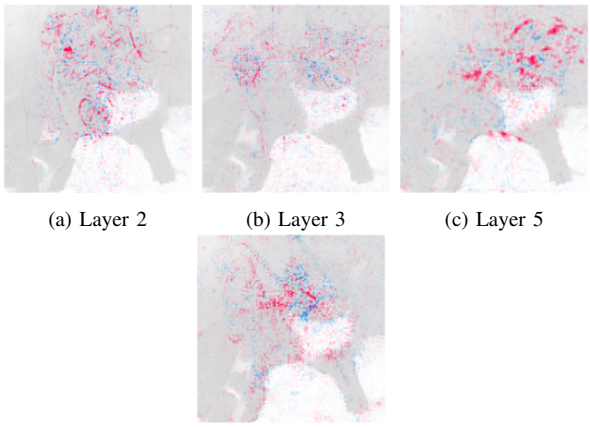
Fig. 6: SHAP values for layers 17 to 21 when the input was "cat3333".

Fig. 7: SHAP values for Layer 22 when the input was "cat3333".

we show the SHAP values at layers 2, 3, 5, and 6. While we saw some responses again in the background area with little color variation, we did not see the same face outline that we saw in Figure 2. or 3. In Figure 12, we show the SHAP values at the layers right before the spatial resolution was reduced in half.

We can see that by comparing the input image to the ImageNet images in computing the SHAP values, the

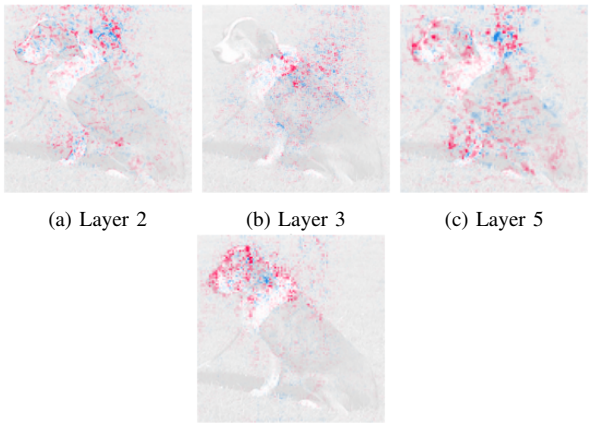
 ISBN: 978-1-61208-953-9



(a) Layer 2 (b) Layer 3 (c) Layer 5

(d) Layer 6

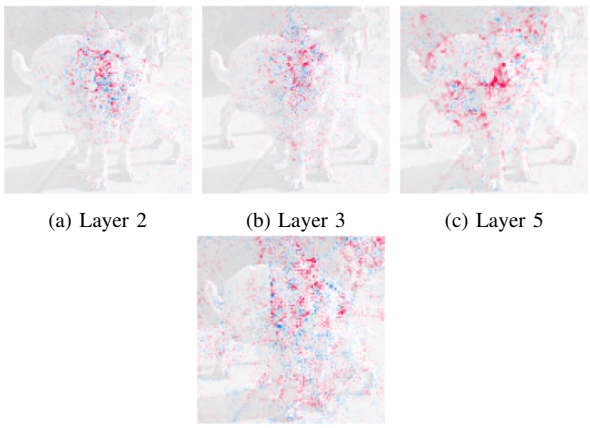
Fig. 8: SHAP values for layers layers 2, 3, 5, and 6 when the input was “cat3238”.



(a) Layer 2 (b) Layer 3 (c) Layer 5

(d) Layer 6

Fig. 9: SHAP values for layers layers 2, 3, 5, and 6 when the input was “dog3333”.

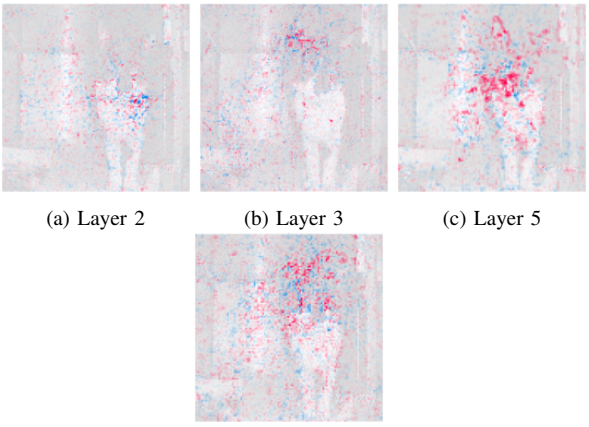


(a) Layer 2 (b) Layer 3 (c) Layer 5

(d) Layer 6

Fig. 10: SHAP values for layers 2, 3, 5, and 6 when the input was “dog3399”.

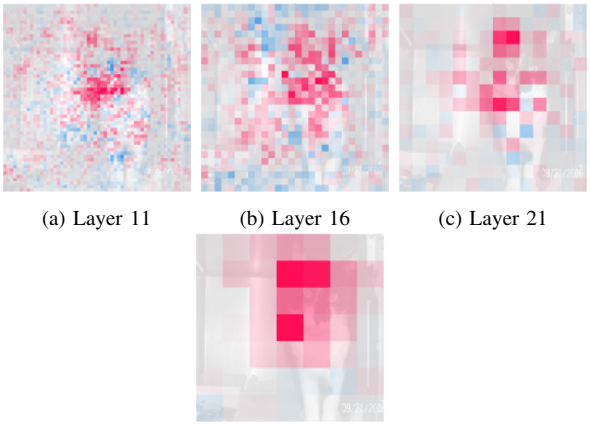
high values correspond more to the outline of the input image. The evidence is that the outline artifacts seen, e.g., in Figure 2 or 3 were influenced by the classification layers.



(a) Layer 2 (b) Layer 3 (c) Layer 5

(d) Layer 6

Fig. 11: SHAP values calculated using ImageNet images for comparison for layers 2, 3, 5, and 6 when the input was “cat3333”.



(a) Layer 11 (b) Layer 16 (c) Layer 21

(d) Layer 22

Fig. 12: SHAP values calculated using ImageNet images for comparison for layers 11, 16, 21 , and 22 when the input was “cat3333”.

IV. Conclusion and Future Work

We computed the SHAP values of the layered feature tensors in a deep learning network trained to distinguish between two classes. The SHAP values are a special case of the Shapley value that explains the factors in a machine learning decision by measuring the output change due to change in each factor. The SHAP value is the Shapley value satisfying local accuracy, missingness, and consistency properties. Our results showed that the lower layers exhibited shapes that fit other faces, some of

 ISBN: 978-1-61208-953-9

them not even from the same class. It appeared that the network worked on assembling the outlines of a shape much earlier in the layered architecture than expected, as early as Layer 2 which was immediately connected to the input layer.

There are a number of interesting directions of future work. In the present work, we examined cases in which the network made the correct classification decision. Given that the network accuracy is 95%, there are cases when the network misclassifies. Ongoing work using the same network architecture and the same data set includes running similar analyses for misclassified samples. We would also like to calculate the SHAP values for the top classification layer. When the network makes a decision, we would like to investigate creating a tree of high SHAP values to attempt to explain the decision. Along other directions, we would like to generalize our findings to different data sets and network architectures.

ACKNOWLEDGMENTS

The authors are grateful to the reviewers whose suggestions helped to improve the clarity of our paper. This work was supported in part by the U.S. National Science Foundation under grant number OIA-1946231 and the Louisiana Board of Regents for the Louisiana Materials Design Alliance (LAMDA).

REFERENCES

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems*, 2012.

[2] G. Vilone and L. Longo, "Notions of explainability and evaluation approaches for explainable artificial intelligence," *Information Fusion*, vol. 76, pp. 89–106, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1566253521001093

[3] S. Lundberg and S.-I. Lee. (2017) A unified approach to interpreting model predictions. Last accessed: 15 April 2022. [Online]. Available: https://arxiv.org/abs/1705.07874

[4] F. Stieler, F. Rabe, and B. Bauer, "Towards domain-specific explainable ai: Model interpretation of a skin image classifier using a human approach," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2021, pp. 1802–1809.

[5] M. Ghassemi, L. Oakden-Rayner, and A. L. Beam, "The false hope of current approaches to explainable artificial intelligence in health care," *The Lancet Digital Health*, vol. 3, pp. e745–e750, 2021.

[6] A. Singh, C. H. Chu, and M. A. Pratt, "Comparing color descriptors between image segments for saliency detection," in *Proceedings of the International Conference on Pattern Recognition Applications and Methods*, Rome, Italy, 2 2016, pp. 558–565.

[7] Kaggle. (2013) Dogs vs. cats: Create an algorithm to distinguish dogs from cats. Last accessed: 15 April 2022. [Online]. Available: https://www.kaggle.com/c/dogs-vs-cats

[8] K. Simonyan and A. Zisserman. (2014) Very deep convolutional networks for large-scale image recognition. Last accessed: 15 April 2022. [Online]. Available: https://arxiv.org/abs/1409.1556

[9] S. Lundberg. (2020) A game theoretic approach to explain the output of any machine learning model. Last accessed: 15 April 2022. [Online]. Available: https://github.com/slundberg/shap

 ISBN: 978-1-61208-953-9