Layer-wise Relevance Propagation for Deep Neural Network Architectures

Alexander Binder[1], Sebastian Bach[2], Gregoire Montavon[3], Klaus-Robert Müller[3], and Wojciech Samek[2]

[1] ISTD Pillar, Singapore University of Technology and Design
8 Somapah Road, 487372 Singapore
alexander_binder@sutd.edu.sg
[2] Machine Learning Group, Fraunhofer HHI
Einsteinufer 37, 10587 Berlin, Germany
wojciech.samek@hhi.fraunhofer.de
[3] Machine Learning Group, TU Berlin
Marchstr. 23, 10587 Berlin, Germany

Abstract. We present the application of layer-wise relevance propagation to several deep neural networks such as the BVLC reference neural net and googlenet trained on ImageNet and MIT Places datasets. Layer-wise relevance propagation is a method to compute scores for image pixels and image regions denoting the impact of the particular image region on the prediction of the classifier for one particular test image. We demonstrate the impact of different parameter settings on the resulting explanation.

Keywords: Deep Neural Networks, Non-linear Explanations

1 Introduction

Deep neural networks are well-known to excel in many fields including image recognition [14, 6, 23], among other fields such as natural language processing [7, 22] or speech recognition [26]. While their results are impressive, the understanding of what makes a neural network arrive at a particular decision is still an open problem. Figure 1 gives an example. To the left, the figure shows a correctly classified image for class *motor scooter* by the BVLC reference classifier of the caffe package [13]. However it is not clear whether the recognition is due to parts of the scooters or due to the image composition as a street scene, due to the typical sitting position of the people or due to other properties. A heatmap computed by layer-wise relevance propagation given in the right side of Figure 1 shows that the most contributing parts are the wheels and backside views of a few scooters.

In certain fields such as security, selection of results or medical applications an interpretation of a decision is as important as the decision itself. Interpretation of non-linear models has recently gained much interest. Several works have been dedicated to the understanding of general non-linear estimators [4, 3, 12]. The

Fig. 1: An image of motor scooters in a complex background and an explanation computed by layer-wise relevance propagation [1] for a classification achieved by the BVLC reference classifier of the caffe package [13].

success of deep neural networks has sparked research into the interpretation of the predictions of deep neural networks. One outcome in this field is layer-wise relevance propagation [1, 2]. In this paper we will present results of applying layer-wise relevance propagation to various deep neural networks and show the impact of parameter choices.

2 Related Work

Recently several methods have been proposed for analyzing what a deep neural network has learned [9, 17, 18]. A large body of great ideas deals with the interpretation what a neuron or a layer of neurons has learned, see for example [9, 15, 25]. Other approaches deal with inverting neural network representations [8, 16] or are about finding images with unusual properties and wrong classification outcomes [19, 11, 24].

Much of the above work can be rephrased as the problem of understanding what do neurons encode and which neurons are most important for the prediction of an image. Here we focus on a different question, namely on the interpretation which pixels of an image are most important for the prediction of an image. In this direction, norms of gradients [21] as well as deconvolution [27] have been proposed for marking the relevance of an image region.

3 Layer-wise Relevance Propagation

A deep neural network is an feed-forward graph of elementary computational units (neurons), each of them realizing a simple function of type

$$x_j^{(l+1)} = g\Big(0, \sum_i x_i^{(l)} w_{ij}^{(l,l+1)} + b_j^{(l+1)}\Big), \text{ e.g. } g(z) = \max(0, z) \tag{1}$$

where j indexes a neuron at a particular layer $l+1$, where $\sum_i$ runs over all lower-layer neurons connected to neuron j, and where $w_{ij}^{(l,l+1)}, b_j^{(l+1)}$ are parameters specific to pairs of adjacent neurons and learned from the data. A deep network derives its complexity from the interconnection of a large number of these elementary units, and from the availability of an efficient algorithm for learning the model (error backpropagation). The output of a deep neural network is obtained by evaluating these neurons in a feed-forward pass. Conversely, [1] have shown that the same graph structure can be used to redistribute the relevance $f(\boldsymbol{x})$ at the output of the network onto pixel-wise relevance scores $\{R_p^{(1)}\}$, by using a local redistribution rule

$$R_i^{(l)} = \sum_j \frac{z_{ij}}{\sum_{i'} z_{i'j}} R_j^{(l+1)} \quad \text{with} \quad \mathrm{z_{ij}} = \mathrm{x_i^{(l)} w_{ij}^{(l,l+1)}} \tag{2}$$

where i indexes a neuron at a particular layer l, and where $\sum_j$ runs over all upper-layer neurons to which neuron i contributes. Application of this rule in a backward pass produces a relevance map (heatmap) that satisfies the desired conservation property $\sum_p R_p^{(1)} = f(\boldsymbol{x})$. This decomposition algorithm is termed Layer-wise Relevance Propagation (LRP). See Fig. 2 for an overview.

In addition to the naive propagation rule in Eq. 2 we evaluate two other LRP algorithms in this paper, namely the ϵ-variant and the β-variant. The first rule is given by:

$$R_i^{(l)} = \sum_j \frac{z_{ij}}{\sum_{i'} z_{i'j} + \epsilon\,\mathrm{sign}(\sum_{i'} z_{i'j})} R_j^{(l+1)} \tag{3}$$

Here for $\epsilon > 0$ the conservation idea is relaxated in order to gain better numerical properties. The second formula is given by:

$$R_i^{(l)} = \sum_j \left(\alpha \cdot \frac{z_{ij}^+}{\sum_{i'} z_{i'j}^+} + \beta \cdot \frac{z_{ij}^-}{\sum_{i'} z_{i'j}^-} \right) R_j^{(l+1)}. \tag{4}$$

Here, z_{ij}^+ and z_{ij}^- denote the positive and negative part of z_{ij} respectively, such that $z_{ij}^+ + z_{ij}^- = z_{ij}$. We enforce $\alpha + \beta = 1$, $\alpha > 0$, $\beta \leq 0$ in order for the relevance propagation equations to be conservative layer-wise. A comparison of the properties of the three decomposition rules is given in Table 1.

	naive ($\epsilon = 0$)	ϵ-variant	β-variant
numerically stable	no	yes	yes
consistent with linear mapping	yes	yes	no
conserves relevance	yes	no	yes

Table 1: Comparison of LRP variants.

4 Experimental Results

First approaches to *quantitatively* compare several methods [20] for computing region-wise scores show that LRP performs well compared to methods such as [21, 27] which are related to norms of gradients.

Figure 3 shows that LRP is not a mere gradient detector. All the images in Figure 3 have relatively strong background gradients. A gradient detector would have assigned high scores to the background of the bunny, to the edges of the yellow flowers in the rooster picture and to the strong gradient from mountain to sky in the cat pic. Unlike a gradient detector, LRP picks up only a few edges which are mostly relevant to the object to be predicted. As for the edge artifacts, note that the receptive field of the deep neural networks is quadratic, thus we resorted to padding an image with the nearest pixel in order to process non-quadratic images.

4.1 Impact of Parameter Settings

The two types of formulas presented in the preceding section have two major parameters, ϵ and β. Here we will explore their effects on examples for two classifier for the 1000 classes of the ImageNet dataset. The classifiers are the BVLC reference and the googlenet model from Caffe [13]. For $\epsilon = 0$ the relevance is perfectly conserved from a neuron to its inputs, however due to canceling out positive and negative inputs, the denominator can become small. In such a case, each of the inputs receives a high weight due to the small size of the denominator. As a consequence, the explanation can become sensitive to noise. Figure 4 shows this effect for a relatively small ϵ set to 0.01. Comparing this against a larger stabilizer such as $\epsilon = 100$ shows an explanation which appears denoised for the larger choice $\epsilon = 100$. This effect holds for various tested classifiers, however the size of ϵ yielding a good description is varying. For BVLC reference and for VGG CNN S model from [5] $\epsilon = 100$ is a good choice, whereas for the googlenet model from [13] $\epsilon = 100$ is too sparse. Note that the googlenet model has substantially more layers, so that a smaller ϵ makes sense to avoid too strong dampening.

The effect of parameter β is demonstrated in Figure 5. β controls how much fraction of the relevance a neuron is assigned to distributed onto inputs with negative weighted activations. In a neural network such inputs can be interpreted as inhibitors. One can see from Figure 5 that increasing the value of β and thus putting more weight onto inhibitors reduces the amount of positive evidence and keeps only the strongest regions. Figure 5 reveals that the optimal value of β depends on the classifier. For the less deep-layered VGG CNN S classifier used in Figure 5, a higher value of β yields better explanations, while for googlenet a smaller value of beta resulting in less suppression retains more structure, consistent with the large number of layers in the googlenet classifier.

4.2 Examples on other datasets - Pascal VOC 2012

For this experiment a neural network was retrained starting from the BVLC reference classifier. The main difference to previous approaches is that PASCAL

VOC2012 [10] is a multi-label dataset and objects of multiple classes can be present in one image. Consequently, we performed a multi-label training, using a hinge-loss summed over all classes, resulting in 20 binary classifiers. Figure 6 shows that the resulting explanations depend on which class is used.

4.3 Examples on other datasets - MIT Places

Here we used the classifier provided by [28]. Note that in Figure 7 the explanation for the images showing abbeys picks up the characteristic peaked shape of gothic bows and not the road in the foreground of the second pic. Similarly, for the windmill the explanation identifies the wind blades but not strong gradients from the horizon or the road.

5 Conclusion

In this paper we analyzed the impact of network architecture and different parameter setting on the resulting explanation. Although the noisiness of a heatmap can be easily controlled by changing parameters ϵ or β, for different architectures these parameters may have different effects. In future work we will investigate how to chose those parameters in order to obtain an optimal trade-off between numerical stability of the decomposition and sparsity / meaningfullness of the heatmap.

References

1. Bach, S., Binder, A., Montavon, G., Klauschen, F., Müller, K.R., Samek, W.: On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. PLOS ONE 10(7), e0130140 (2015)
2. Bach, S., Binder, A., Montavon, G., Müller, K., Samek, W.: Analyzing classifiers: Fisher vectors and deep neural networks. CoRR abs/1512.00172 (2015), `http://arxiv.org/abs/1512.00172`
3. Baehrens, D., Schroeter, T., Harmeling, S., Kawanabe, M., Hansen, K., Müller, K.R.: How to explain individual classification decisions. Journal of Machine Learning Research 11, 1803–1831 (2010)
4. Braun, M.L., Buhmann, J.M., Müller, K.: On relevant dimensions in kernel feature spaces. Journal of Machine Learning Research 9, 1875–1908 (2008), `http://doi.acm.org/10.1145/1390681.1442795`
5. Chatfield, K., Simonyan, K., Vedaldi, A., Zisserman, A.: Return of the devil in the details: Delving deep into convolutional nets. In: British Machine Vision Conference, BMVC (2014)
6. Ciresan, D.C., Giusti, A., Gambardella, L.M., Schmidhuber, J.: Deep neural networks segment neuronal membranes in electron microscopy images. In: Adv. in NIPS. pp. 2852–2860 (2012)
7. Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., Kuksa, P.P.: Natural language processing (almost) from scratch. Journal of Machine Learning Research 12, 2493–2537 (2011)

8. Dosovitskiy, A., Brox, T.: Inverting convolutional networks with convolutional networks. CoRR abs/1506.02753 (2015)
9. Erhan, D., Bengio, Y., Courville, A., Vincent, P.: Visualizing higher-layer features of a deep network. Tech. Rep. 1341, University of Montreal (Jun 2009)
10. Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results
11. Goodfellow, I.J., Shlens, J., Szegedy, C.: Explaining and harnessing adversarial examples. CoRR abs/1412.6572 (2014)
12. Hansen, K., Baehrens, D., Schroeter, T., Rupp, M., Müller, K.R.: Visual interpretation of kernel-based prediction models. Molecular Informatics 30(9), 817–826 (2011)
13. Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., Darrell, T.: Caffe: Convolutional architecture for fast feature embedding. arXiv preprint arXiv:1408.5093 (2014)
14. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Adv. in NIPS 25. pp. 1106–1114 (2012)
15. Le, Q.V.: Building high-level features using large scale unsupervised learning. In: ICASSP. pp. 8595–8598 (2013)
16. Mahendran, A., Vedaldi, A.: Understanding deep image representations by inverting them. In: Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) (2015)
17. Montavon, G., Braun, M., Müller, K.R.: Kernel analysis of deep networks. Journal of Machine Learning Research 12, 2563–2581 (2011)
18. Montavon, G., Braun, M.L., Krueger, T., Müller, K.R.: Analyzing local structure in kernel-based learning: Explanation, complexity and reliability assessment. Signal Processing Magazine, IEEE 30(4), 62–74 (2013)
19. Nguyen, A.M., Yosinski, J., Clune, J.: Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. CoRR abs/1412.1897 (2014)
20. Samek, W., Binder, A., Montavon, G., Bach, S., Müller, K.R.: Evaluating the visualization of what a deep neural network has learned. CoRR abs/1509.06321 (Sep 2015), http://http://arxiv.org/abs/1509.06321.pdf
21. Simonyan, K., Vedaldi, A., Zisserman, A.: Deep inside convolutional networks: Visualising image classification models and saliency maps. CoRR abs/1312.6034 (2013)
22. Socher, R., Perelygin, A., Wu, J., Chuang, J., Manning, C.D., Ng, A.Y., Potts, C.: Recursive deep models for semantic compositionality over a sentiment treebank. In: Proc. of EMNLP. pp. 1631–1642 (2013)
23. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. CoRR abs/1409.4842 (2014)
24. Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I.J., Fergus, R.: Intriguing properties of neural networks. CoRR abs/1312.6199 (2013)
25. Yosinski, J., Clune, J., Nguyen, A.M., Fuchs, T., Lipson, H.: Understanding neural networks through deep visualization. CoRR abs/1506.06579 (2015)
26. Yu, D., Deng, L.: Automatic Speech Recognition - A Deep Learning Approach. Springer (October 2014), http://research.microsoft.com/apps/pubs/default.aspx?id=230891
27. Zeiler, M.D., Fergus, R.: Visualizing and understanding convolutional networks. In: ECCV. pp. 818–833 (2014)
28. Zhou, B., Lapedriza, A., Xiao, J., Torralba, A., Oliva, A.: Learning deep features for scene recognition using places database. In: Adv. in NIPS. pp. 487–495 (2014)

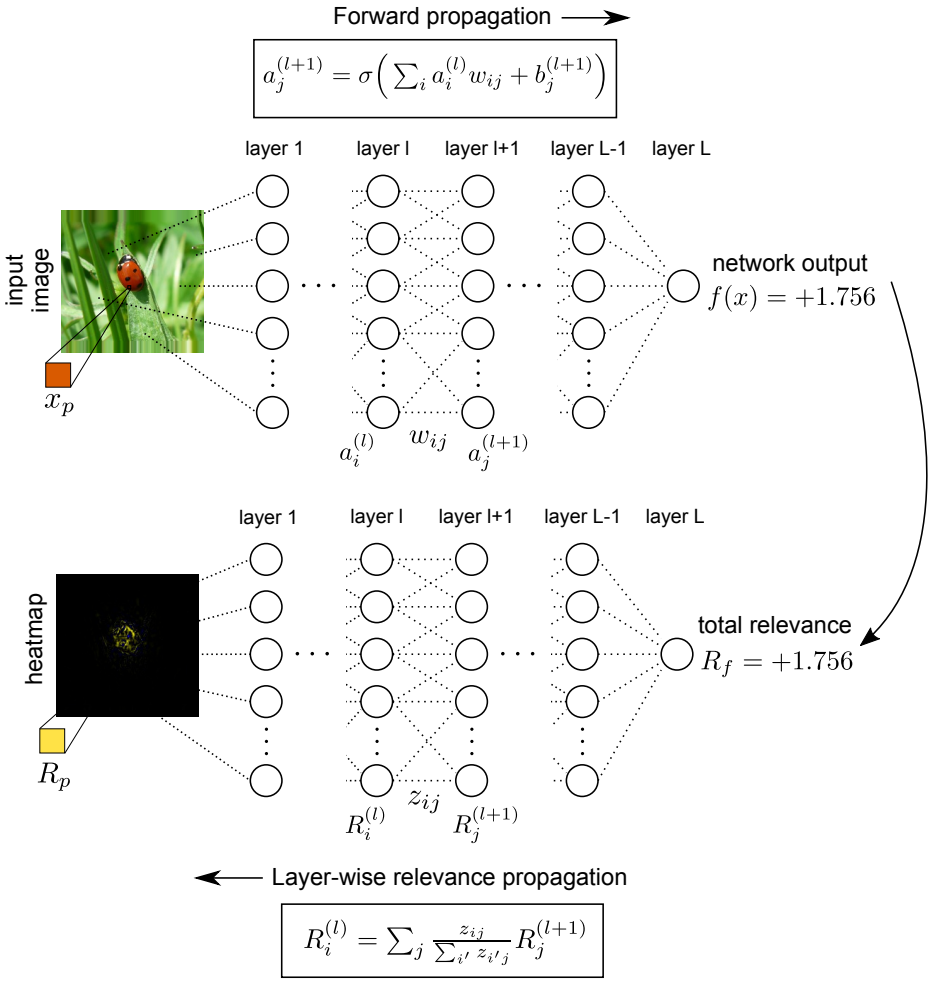

Fig. 2: Overview of LRP. First, the input image is processed by the network and a network output is computed. Then, the output value is backprojected layer to layer onto the pixel by using the LRP formula in the box. The pixel relevances $\{R_p\}$ are visualized as heatmap.

Fig. 3: Layer-wise Relevance propagation does not fall for strong irrelevant gradients: heatmaps are computed for backgrounds with strong gradients. Explanations are computed by layer-wise relevance propagation [1] for a classification achieved by the BVLC reference classifier of the caffe package [13].

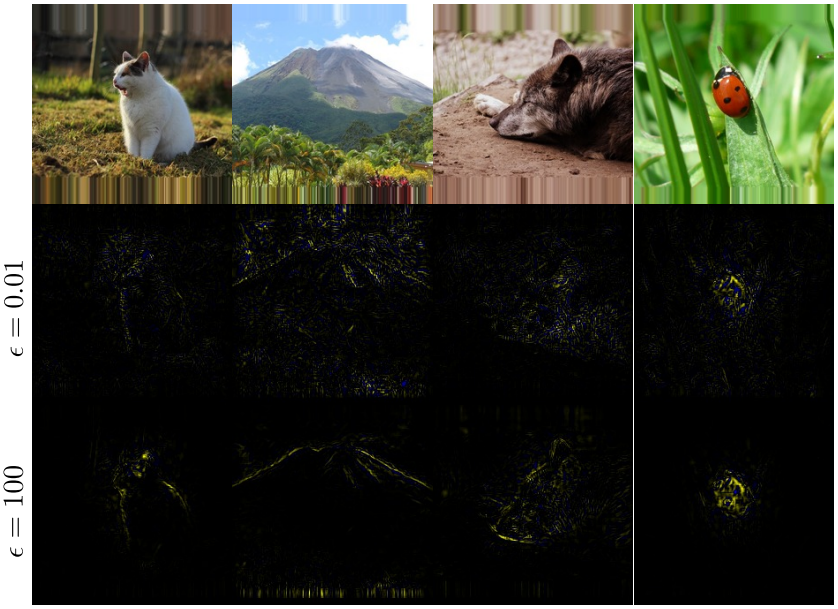

Fig. 4: The impact of small (middle row) and large (bottom row) stabilizer ϵ. Here set to $\epsilon = 0.01$ and $\epsilon = 100$. Explanations are computed by layer-wise relevance propagation [1] for a classification achieved by the BVLC reference classifier of the caffe package [13].

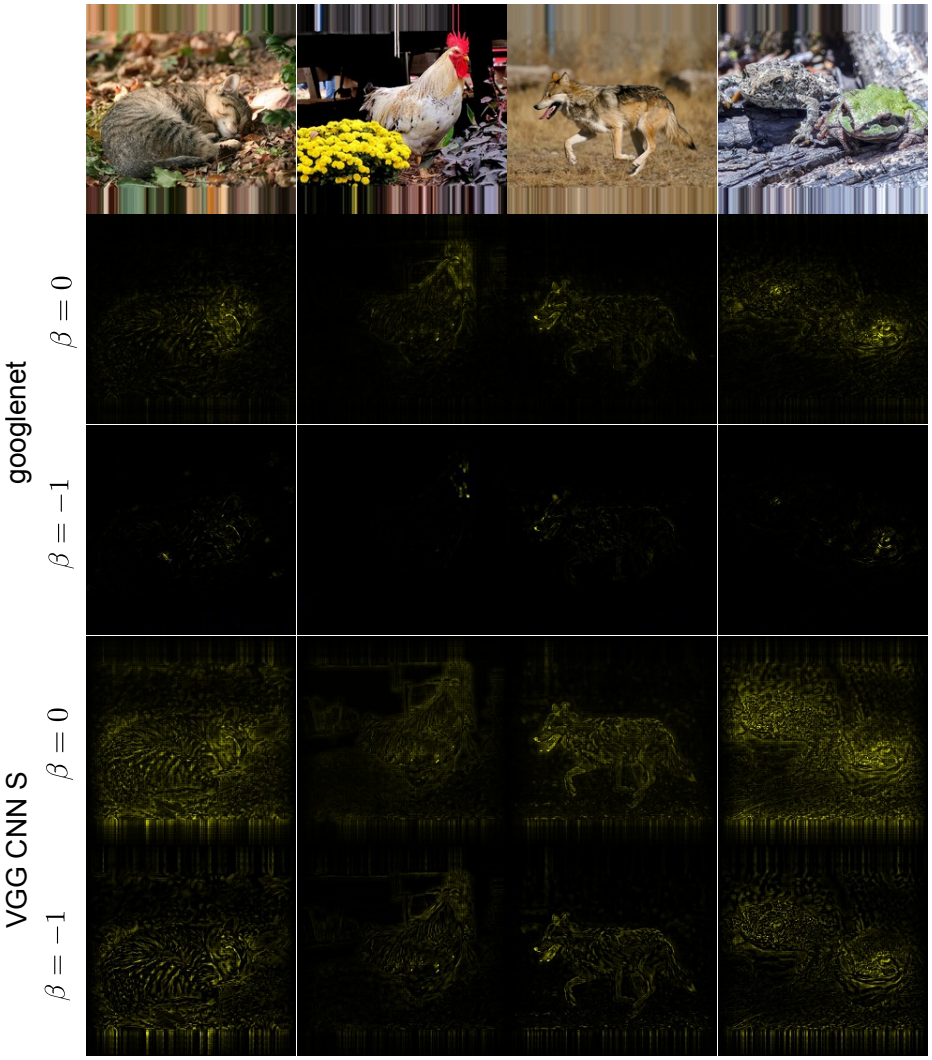

Fig. 5: The impact of small and large suppression control β. Here set to $\beta = 0$ and $\beta = -1$. Explanations are computed by layer-wise relevance propagation [1] for a classification achieved by the googlenet classifier of the caffe package [13] and by the VGG CNN S classifier in [5].

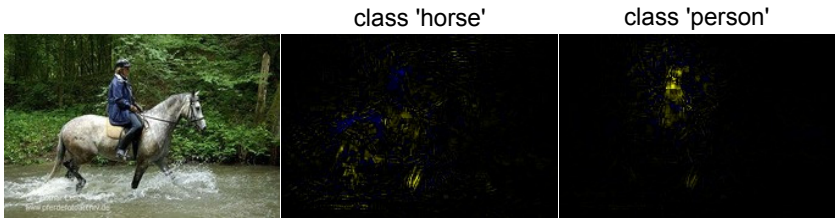

Fig. 6: Explanations are computed by layer-wise relevance propagation [1] for a classification achieved by a classifier trained for multi-label recognition on PASCAL VOC2012. Middle column shows explanation for class horse. Right column shows explanation for class person. Note the shift of focus depending on which object class is to be explained.

Fig. 7: Explanations are computed by layer-wise relevance propagation [1] for a classification achieved by a classifier trained the MIT Places dataset [28].