Object Boundary Detection and Classification with Image-level Labels

Jing Yu Koh[1], Wojciech Samek[2], Klaus-Robert Müller[3,4], Alexander Binder[1]

[1]ISTD Pillar, Singapore University of Technology and Design, Singapore
[2]Department of Video Coding & Analytics, Fraunhofer Heinrich Hertz Institute, Berlin, Germany
[3]Department of Computer Science, TU Berlin, Germany
[4]Department of Brain and Cognitive Engineering, Korea University, Seoul, Republic of Korea

Abstract. Semantic boundary and edge detection aims at simultaneously detecting object edge pixels in images and assigning class labels to them. Systematic training of predictors for this task requires the labeling of edges in images which is a particularly tedious task. We propose a novel strategy for solving this task, when pixel-level annotations are not available, performing it in an almost zero-shot manner by relying on conventional whole image neural net classifiers that were trained using large bounding boxes. Our method performs the following two steps at test time. Firstly it predicts the class labels by applying the trained whole image network to the test images. Secondly, it computes pixel-wise scores from the obtained predictions by applying backprop gradients as well as recent visualization algorithms such as deconvolution and layer-wise relevance propagation. We show that high pixel-wise scores are indicative for the location of semantic boundaries, which suggests that the semantic boundary problem can be approached without using edge labels during the training phase.

1 Introduction

Neural net based predictors achieve excellent results in many data-driven tasks, examples among the newer being [6, 15, 14, 10, 17], while others such as video detection or machine translation [21, 8] are equally impressive. Rather than extending neural networks to a new application, we focus here on the question whether a neural network can solve problems which are *harder* than the one for which the network was trained. In particular, we consider the task of semantic boundary detection which we aim to solve without appropriately fine-grained training labels.

The problem of semantic boundary detection (SBD) [5] can be defined as the simultaneous detection of object edge pixels and assignment of class labels to such edge pixels in images. Recently, the work of [3, 16, 24, 25] showed substantial improvement using neural nets, however, the approach relied on end-to-end training with a dataset for which semantic boundary labels were available.

When trying to build a predictor for SBD, practitioners face the problem that the classical inductive machine learning paradigm requires to create a dataset with semantic boundary labels, that is, for each image a subset of pixels in images corresponding to object edges is labeled with class indices. Creating such labelling is a particularly tedious task, unlike labelling whole images or drawing bounding boxes, both of which can be done very quickly. The best proof for this difficulty is the fact that we are aware of only one truly semantic boundary dataset [5]. Note that SBD is different from contour detection tasks [23] which aim at finding contours of objects without assigning class labels to them. In that sense the scope of our proposed work is different from unsupervised contour detection as in [13].

The main question in this paper is to what extent it is possible to solve the semantic boundary or edge detection task without having appropriately fine-grained labels, i.e., pixel-level ground truth, which are required for classical training paradigms ? We do not intend to replace the usage of pixel-wise boundary labels when they are available. We aim at use cases in which pixel-wise boundary labels are not available during the training phase. One example of using weaker annotations for semantic boundary detection is [9] where bounding box labels are used to learn semantic boundaries. We propose a novel strategy to tackle a problem requiring fine-grained labels, namely semantic boundary detection, with a classifier trained for image classification using only image-wise labels. For that we use neural nets that classify an image, and apply existing visualization methods that are able to assign class-specific-scores to single pixels. These class-specific pixel scores can then be used to define semantic boundary predictions.

The contribution of this paper is as follows. We demonstrate that classifier visualization methods are useful beyond producing nice-to-look-at images, namely for approaching prediction tasks on the pixel-level in the absence of appropriately fine-grained training labels. As an example, we apply and evaluate the performance of classifier visualization methods to the SBD task. We show that these visualization methods can be used for producing quantifiably meaningful predictions at a higher spatial resolution than the labels, which were the basis for training the classifiers. We discuss the shortcomings of such approaches when compared to the proper training paradigm that makes use of pixel-level labels. We do not expect such methods to beat baselines that employ the proper training paradigm and thus use pixel-level labels during training, but rather aim at the practitioner's case in which fine-grained training data is too costly in terms of money or time.

2 Obtaining Pixel-level Scores from Image-wise Predictions

In the following we introduce the methods that we will use for producing pixel-level scores without pixel-level labels during training time. It is common to all these methods that they take a classifier prediction $f_c(x)$ on an image x

and produce scores $s_c(p)$ for pixels $p \in x$. Suppose we have classifiers $f_c(x)$ for multiple classes c. Then we can tackle the SBD problem by (1) classifying an image, i.e., determine those classes that are present in the image, and (2) computing pixel-wise scores for those classes using one of the following methods.

2.1 Gradient

Probably the most obvious idea to tackle the SBD problem is to run a forward prediction with a classifier, and compute the gradient for each pixel. Let x be an input image, $f_1, \ldots, f_C$ be C outputs of a multi-class classifier and x_p be the p-th pixel. Computing pixel-wise scores for a class c and pixel p can be achieved using

$$s(p) = \left\| \frac{\partial f_c}{\partial x_p}(x) \right\|_2 \tag{1}$$

The norm runs here over the partial derivatives for the (r, g, b)-subpixels of a pixel p. Alternatively one can sum up the subpixel scores in order to have a pixel-score. Using gradients for visualizing sensitivities of neural networks has been shown in [20]. A high score in this sense indicates that the output f_c has high sensitivity under small changes of the input x_p, i.e. there exists a direction in the tangent space located at x for which the slope of the classifier f_c is very high.

In order to see the impact of partial derivatives, consider the case of a simple linear mapping that takes subpixels $x_{p,s}$ of pixel p as input.

$$f(x) = \sum_p \sum_{s \in \{r,g,b\}} w_{p,s} x_{p,s} \tag{2}$$

In this case backpropagation combined with an ℓ_2-norm yields:

$$s(p) = (w_{p,r}^2 + w_{p,g}^2 + w_{p,b}^2)^{1/2} \tag{3}$$

Note that the input $x_{p,s}$, and in particular its sign plays no role in a visualization achieved by backpropagation, although obviously the sign of $x_{p,s}$ does matter for deciding whether to detect an object ($f(x) > 0$) or not ($f(x) < 0$). This is a limiting factor, when one aims to explain what pixels are relevant for the prediction $f(x) > 0$.

2.2 Deconvolution

Deconvolution [26] is an alternative method to compute pixel-wise scores. Starting with scores given at the top of a convolutional layer, it applies the transposed filter weights to compute scores at the bottom of the same layer. Another important feature is used in max-pooling layers, where scores from the top are distributed down to the input that yielded the maximum value in the max pooling. Consider the linear mapping case again. Then deconvolution in the sense of

multiplying the transposed weights w (as it is for example implemented in the Caffe package) yields for subpixel s of channel p

$$s(p,s) = f_c(x) w_{p,s} \tag{4}$$

This score can be summed across subpixels, or one can take again an ℓ_p-norm. When using summation across subpixels, then deconvolution is proportional to the prediction $f_c(x)$, in particular it expresses the dominating terms $w_{p,s}x_{p,s} \approx f_c(x)$ correctly which contribute most to the prediction $f(x)$.

2.3 Layer-wise Relevance Propagation

Layer-wise Relevance Propagation (LRP) [2] is a principled method for explaining neural network predictions in terms of pixel-wise scores. LRP reversely propagates a numerical quantity, named relevance, in a way that preserves the sum of the total relevance at each layer. The relevance is initialized at the output as the prediction score $f_c(x)$ and propagated down to the inputs (i.e., pixels), so that the relevance conservation property holds at each layer

$$f_c(x) = \ldots = \sum_j R_j^{(l+1)} \ldots = \sum_i R_i^{(l)} = \ldots = \sum_p R_p^{(1)} \tag{5}$$

where $\{R_j^{(l+1)}\}$ and $\{R_i^{(l)}\}$ denote the relevance at layer $l+1$ and l, respectively, and $\{R_p^{(1)}\}$ represents the pixel-wise relevance scores.

Let us consider the neural network as an feed-forward graph of elementary computational units (neurons), each of them realizing a simple function of type

$$x_j^{(l+1)} = g\Big(0, \sum_i x_i^{(l)} w_{ij}^{(l,l+1)} + b_j^{(l+1)}\Big) \quad \text{e.g. } g(z) = \max(0,z) \tag{6}$$

where j denotes a neuron at a particular layer $l+1$, and, where $\sum_i$ runs over all lower-layer neurons connected to neuron j. $w_{ij}^{(l,l+1)}, b_j^{(l+1)}$ are parameters of a neuron. The prediction of a deep neural network is obtained by computing these neurons in a feed-forward pass. Conversely, [2] have shown that the same graph structure can be used to redistribute the relevance $f(x)$ at the output of the network onto pixel-wise relevance scores $\{R_p^{(1)}\}$, by using a local redistribution rule

$$R_i^{(l)} = \sum_j \frac{z_{ij}}{\sum_{i'} z_{i'j}} R_j^{(l+1)} \quad \text{with} \quad \mathrm{z_{ij}} = \mathrm{x_i^{(l)} w_{ij}^{(l,l+1)}} \tag{7}$$

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.

We consider two other LRP algorithms introduced in [2], namely the ϵ-variant and the β-variant. The first rule is given by:

$$R_i^{(l)} = \sum_j \frac{z_{ij}}{\sum_{i'} z_{i'j} + \epsilon \operatorname{sign}(\sum_{i'} z_{i'j})} R_j^{(l+1)} \tag{8}$$

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

$$R_i^{(l)} = \sum_j \left(\alpha \cdot \frac{z_{ij}^+}{\sum_{i'} z_{i'j}^+} + \beta \cdot \frac{z_{ij}^-}{\sum_{i'} z_{i'j}^-} \right) R_j^{(l+1)}. \tag{9}$$

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. Note that for $\alpha = 1$ this redistribution rule is equivalent (for ReLU nonlinearities g) to the z^+-rule by [18].

In contrast to the gradient, LRP recovers the natural decomposition of a linear mapping

$$f(x) = \sum_{i=1}^{D} w_i x_i \tag{10}$$

i.e., the pixel-level score

$$R_i = w_i x_i \tag{11}$$

not only depends on whether the classifier reacts to this input dimension ($w_i > 0$), but also if that feature is actually present ($x_i > 0$). An implementation of LRP can be found in [12].

3 Experiments

We perform the experiments on the SBD dataset with a Pascal VOC multilabel classifier from [11] that is available in the BVLC model zoo of the Caffe [7] package. This classifier was trained using the 4 edge crops and the center crops of the ground truth bounding boxes of the Pascal VOC dataset [4]. We do not use pixel labels at training time, however, for evaluation at test time we use the pixel-wise ground truth, in order to be able to compare all methods quantitatively. Same as [5] we report the maximal F-score and the average precision on the pixel-level of an image. We stick to the same convention regarding counting true positives in a neighborhood, as introduced in [5].

3.1 Performance on the SBD task

Table 1 shows the average precision (AP) scores for all methods. We can see

Table 1. Average precision (AP) and maximal F-scores (MF) scores of various methods to compute pixel-wise scores from whole image classifiers without pixel-labels at training time, compared against the original method *InverseDetectors* [5] and Boundary detection using Neural nets *HFL* [3]. Only the last two both use pixel-labels at training time. All other use no pixel-level labels during training. Grad denotes Gradient, Deconv denotes [26], ϵ and β refer to LRP variants given in equations (8) and (9) taken from [2].

training phase:	image-level labels						pixel-level labels	
Method:	Gradient	Deconv	$\beta = 0$	$\beta = -1$	$\epsilon = 1$	$\epsilon = 0.01$	InvDet [5]	HFL [3]
AP	22.5	25.0	28.4	27.3	**31.4**	31.2	19.9	**54.6**
MF	31.0	33.3	35.1	34.1	38.0	**38.1**	28.0	**62.5**

from the table that the neural-network based method [3] which uses pixel-level ground truth at training time performs best by a large margin. Methods that do not employ pixel-level labels at training time perform far worse. However, we can see a certain surprise: all the methods perform better than the method [5] on Semantic Boundary Detection that was the best baseline before the work of [3] replaced it. Note that [5] as well as [3] relies on pixel-wise labels during training, whereas the proposed methods require only image-wise labels. This result gives a realistic comparison of how good methods on pixel-wise prediction without pixel-labels in the training phase can perform.

The pixel-wise scores for LRP are computed by summing over subpixels. For Gradient and Deconvolution using the negative sum over subpixels performed better than using the sum or the ℓ_2-norm. For both cases negative pixel scores were set to zero. This follows our experience with Deconvolution and LRP that wave-like low-level image filters, which are typically present in deep neural nets, receive equally wave-like scores with positive and negative half-waves. Removing the negative half waves improves the prediction quality. Table 2 shows the

Table 2. Comparison of various ways to combine subpixel scores into a pixel-wise score.

subpixel aggregation method	sum	sum of negative scores	ℓ_2-norm
Gradient AP	22.0	22.5	18.8
Deconvolution AP	22.9	25.0	21.9

comparison of AP scores for various methods to compute a pixel-wise score from subpixel scores. Note that we do not show the ℓ_2-norm, or the summed negative scores for the LRP methods, as LRP does preserve the sign of the prediction and thus using the sum of negative scores or ℓ_2-norm has no meaningful interpretation for LRP.

3.2 Shortcomings of visualization methods

Semantic boundaries are not the most relevant regions for the decision of above mentioned classifiers trained on images of natural scenes. This does not devaluate models trained on shapes. It merely says that, given RGB images of natural scenes as input, above object class predictors put considerable weight on internal edges and textures rather than outer boundaries, an effect which can be observed in the heatmaps in Figures 1 and 2. This is the primary hypothesis why the visualization methods above are partially mismatching the semantic boundaries. We demonstrate this hypothesis quantitatively by an experiment. For this we need to introduce a measure of relevance of a set of pixels which is independent of the computed visualizations.

Perturbation analysis We can measure the relevance of a set of pixels $\mathcal{S} \subset x$ of an image x for the prediction of a classifier by replacing the pixels in this set by some values, and comparing the prediction on the modified image $\tilde{x}_{\mathcal{S}}$ against the prediction score $f(x)$ for the original image [19] (similar approach has been applied for text in [1]). This idea follows the intuition that most random perturbations in a region that is important for classification will lead to a decline of the prediction score for the image as a whole: $f(\tilde{x}_{\mathcal{S}}) < f(x)$. It is clear that there exist perturbations of a region yield an increase of the prediction score: for example a change that follows the gradient direction locally. Thus we will draw many perturbations of the set $\mathcal{S}$ from a random distribution P and measure an approximation the expected decline of the prediction

$$m = f(x) - \mathbb{E}_{\mathcal{S}\sim P}[f(\tilde{x}_{\mathcal{S}})] \tag{12}$$

We intend to measure the expected decrease for the set $\mathcal{S}$ being the ground truth pixels for the SBD task, and compare it against the set of highest scoring pixels. For a fair comparison the set of highest scoring pixels will be limited to have the same size as the number of ground truth pixels. Highest scoring pixels will be defined by the pixel-wise scores from the above methods. We will show that the expected decrease is higher for the pixel-wise scores, which indicates that ground truth pixels representing semantic boundaries are not the most relevant subset for the classifier prediction.

The experiment to demonstrate this will be designed as follows. For each test image and each ground truth class we will take the set of ground truth pixels, and randomly perturb them. For a (r, g, b)-pixel we will draw the values from a uniform distribution in $[0, 1]^3 \subset \mathbb{R}^3$. For each image and present class of semantic boundary task ground truth we repeat 200 random perturbations of the set in order to compute an approximation to Equation (12). We compute the average over all images to obtain the average decrease on ground truth pixels m_{GT}. m_{GT} is an average measure of relevance of the ground truth pixels. m_{GT} is to be compared against the analogous quantity m_V derived from the top-scoring pixels of a visualization method. For a given visualization method $V \in \{Gradient, Deconv, LRP\text{-}\beta, LRP\text{-}\epsilon\}$, we define the set of pixels to be perturbed as the pixels with the highest pixel-wise scores computed from the visualization

method. The set size for this set will be the same as the number of ground truth pixels of the semantic boundary task of the same image and class. Running the same perturbation idea according to Equation (12) on this set yields a measure m_V of average decrease of classifier prediction that is specific to the most relevant pixels of the given visualization method.

Table 3. Comparison of the averaged prediction scores. $f(x)$ denotes the average prediction for the unperturbed images for all ground truth classes. m_{GT} denotes the average prediction for images with perturbed ground truth pixels. m_{Deconv} and $m_{LRP,\epsilon=1}$ denotes the average prediction for images with perturbed highest scoring pixels having the same cardinality as the ground truth pixels, using Deconvolution and LRP.

$f_c(x)$	m_{GT}	m_{Deconv}	$m_{LRP,\epsilon=1}$
10.20 ± 0	7.73 ± 0.36	5.68 ± 0.38	1.73 ± 0.34

Table 3 shows the results of the comparison. Note that we take the ground truth in the image that has been resized to match the receptive field of the neural net (227×227), and apply one step of classical morphological thickening. This thickened ground truth set will be used. The standard deviation was computed for the 200 random perturbations and averaged over images and classes. We can see from the table that the decrease is stronger for the visualization methods compared to the ground truth pixels. This holds for Deconvolution as well as for LRP. The pixels highlighted by these methods are more relevant for the classifier prediction, even though they disagree with boundary pixel labels. In summary this supports our initially stated hypothesis that boundary pixels are not the most relevant for classification, and our explanation why these methods are partially mismatching the set of boundary ground truth labels.

We can support this numerical observation also by example images. We can observe two error cases. Firstly, the pixel-wise predictions may miss semantic boundaries that are deemed to be less discriminative for the classification task. This adds to false negatives. Secondly, the pixel-wise predictions may assign high scores to pixels that are relevant for the classification of an object and lie inside the object. Figure 1 shows some examples. We can clearly see false negatives and false positives in these examples, for example for the car and LRP-$\epsilon = 1$ where the window regions are deemed to be highly relevant for the classifier decision, but the outer boundary on the car top is considered irrelevant which is a bad result with respect to boundary detection. For the cat most of the methods focus on its face rather than the cat boundaries. The bird is an example where deconvolution gives a good result. For the people with the boats the heatmap is shown for the people class. In this example LRP-$\epsilon = 1$ focuses correctly most selectively on the people, same as for the tiny car example.

We can observe from these figures a common sparsity of the pixel-wise prediction methods. This motivates why we did not aim at solving segmentation tasks with these methods. Finally we remark that this sparsity is not an artefact of the particular deep neural network from [11] tuned for PASCAL VOC.

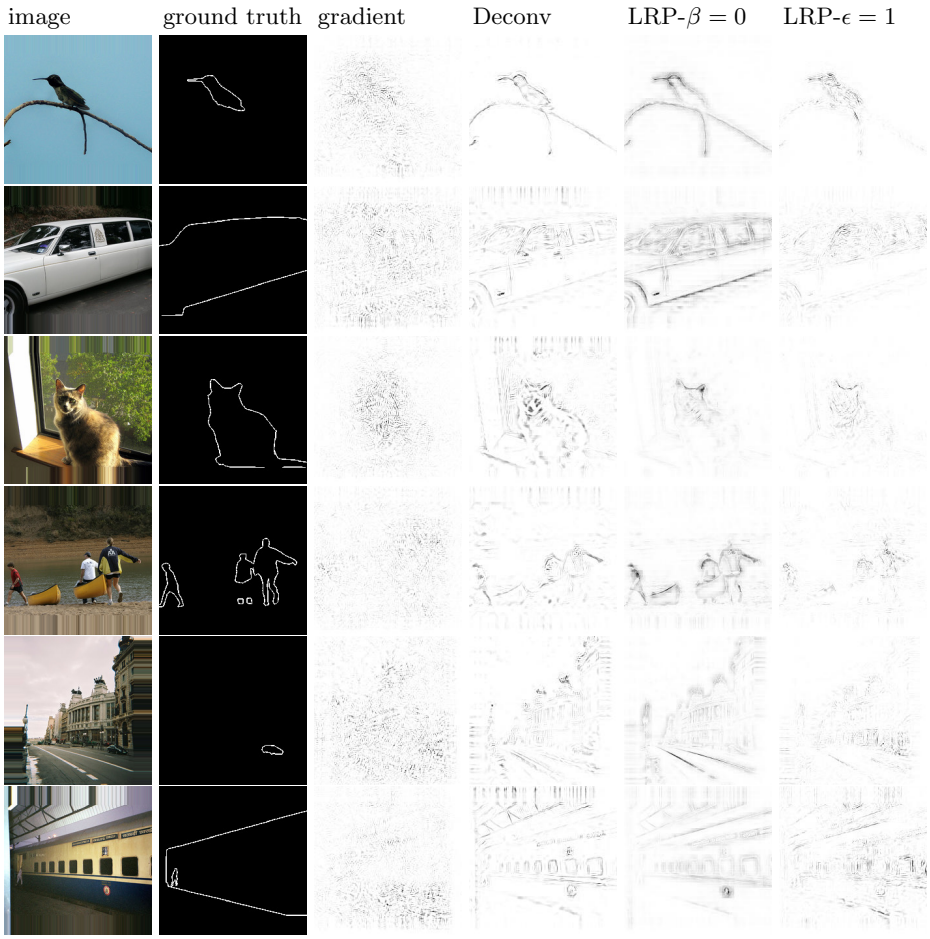

Fig. 1. Heatmaps of pixel-wise scores compared against the groundtruth. From left to right: original image, pixel-level ground truth, gradient (negative scores), Deconvolution (negative scores), LRP with $\beta = 0$ and with $\epsilon = 1$.

Figure 2 shows the same effect for the GoogleNet Reference Classifier [22] of the Caffe Package [7]. As an example, for the wolf, parts of the body in the right have missing boundaries. Indeed this part is not very discriminative. A similar interpretation can be made for the lower right side of the dog which has a strong image gradient but not much dog-specific evidence.

Fig. 2. Heatmaps of pixel-wise scores computed for the GoogleNet Reference Classifier of the Caffe Package show the sparsity of pixel-wise prediction methods. The used classifiers were: Timber wolf, Bernese mountain Dog and Ram. Left Column: image as it enters the deep neural net. Middle: pixel-wise scores computed by LRP with $\epsilon = 1$. Right: pixel-wise scores computed by LRP with $\beta = 0$.

4 Conclusion

We presented here several methods for zero-shot learning for semantic boundary detection and evaluated them quantitatively. These methods are useful when pixel-level labels are unavailable at training time. These methods perform reasonably against previous state of the art. It would be interesting to evaluate these methods on other datasets with class-specifically labeled edges, if they would become available in the future. Furthermore we have shown that classifier visualization methods [20, 26, 2] have applications beside pure visualization due to their property of computing predictions at a finer scale.

References

1. Arras, L., Horn, F., Montavon, G., Müller, K.R., Samek, W.: Explaining predictions of non-linear classifiers in nlp. In: Proc. of the 1st Workshop on Representation Learning for NLP. pp. 1–7. Association for Computational Linguistics (2016)
2. 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)
3. Bertasius, G., Shi, J., Torresani, L.: High-for-low and low-for-high: Efficient boundary detection from deep object features and its applications to high-level vision. In: IEEE ICCV. pp. 504–512 (2015)
4. Everingham, M., Van Gool, L., Williams, C., Winn, J., Zisserman, A.: The pascal visual object classes (voc) challenge. IJCV 88(2), 303–338 (2010)
5. Hariharan, B., Arbelaez, P., Bourdev, L.D., Maji, S., Malik, J.: Semantic contours from inverse detectors. In: IEEE ICCV. pp. 991–998 (2011)
6. Hariharan, B., Arbeláez, P.A., Girshick, R.B., Malik, J.: Hypercolumns for object segmentation and fine-grained localization. In: IEEE CVPR. pp. 447–456 (2015)
7. Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R.B., Guadarrama, S., Darrell, T.: Caffe: Convolutional architecture for fast feature embedding. In: Proc. of the ACM Int. Conf. on Multimedia. pp. 675–678 (2014)
8. Karpathy, A., Toderici, G., Shetty, S., Leung, T., Sukthankar, R., Li, F.: Large-scale video classification with convolutional neural networks. In: IEEE CVPR. pp. 1725–1732 (2014)
9. Khoreva, A., Benenson, R., Omran, M., Hein, M., Schiele, B.: Weakly supervised object boundaries. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2016)
10. Koutník, J., Cuccu, G., Schmidhuber, J., Gomez, F.J.: Evolving large-scale neural networks for vision-based reinforcement learning. In: GECCO. pp. 1061–1068 (2013)
11. Lapuschkin, S., Binder, A., Montavon, G., Müller, K.R., Samek, W.: Analyzing classifiers: Fisher vectors and deep neural networks. In: IEEE CVPR. pp. 2912–2920 (2016)
12. Lapuschkin, S., Binder, A., Montavon, G., Müller, K.R., Samek, W.: The layer-wise relevance propagation toolbox for artificial neural networks. Journal of Machine Learning Research 17(114), 1–5 (2016)
13. Li, Y., Paluri, M., Rehg, J.M., Dollar, P.: Unsupervised learning of edges. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2016)
14. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: IEEE CVPR. pp. 3431–3440 (2015)
15. Malinowski, M., Rohrbach, M., Fritz, M.: Ask your neurons: A neural-based approach to answering questions about images. In: IEEE ICCV. pp. 1–9 (2015)
16. Maninis, K.K., Pont-Tuset, J., Arbelaez, P., Gool, L.V.: Convolutional oriented boundaries: From image segmentation to high-level tasks. IEEE Transactions on Pattern Analysis and Machine Intelligence PP(99), 1–1 (2017)
17. Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A.A., Veness, J., Bellemare, M.G., Graves, A., Riedmiller, M., Fidjeland, A.K., Ostrovski, G., Petersen, S., Beattie, C., Sadik, A., Antonoglou, I., King, H., Kumaran, D., Wierstra, D., Legg, S., Hassabis, D.: Human-level control through deep reinforcement learning. Nature 518(7540), 529–533 (02 2015)

18. Montavon, G., Bach, S., Binder, A., Samek, W., Müller, K.R.: Explaining nonlinear classification decisions with deep taylor decomposition. Pattern Recognition 65, 211–222 (2017)
19. Samek, W., Binder, A., Montavon, G., Lapuschkin, S., Müller, K.R.: Evaluating the visualization of what a deep neural network has learned. IEEE Transactions on Neural Networks and Learning Systems (2016)
20. Simonyan, K., Vedaldi, A., Zisserman, A.: Deep inside convolutional networks: Visualising image classification models and saliency maps. CoRR abs/1312.6034 (2013)
21. Sutskever, I., Vinyals, O., Le, Q.V.: Sequence to sequence learning with neural networks. In: Adv. in NIPS. pp. 3104–3112 (2014)
22. 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)
23. Xie, S., Tu, Z.: Holistically-nested edge detection. International Journal of Computer Vision (2017), http://dx.doi.org/10.1007/s11263-017-1004-z
24. Yang, J., Price, B., Cohen, S., Lee, H., Yang, M.H.: Object contour detection with a fully convolutional encoder-decoder network. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2016)
25. Yu, Z., Feng, C., Liu, M.Y., Ramalingam, S.: CASENet: Deep Category-Aware Semantic Edge Detection. ArXiv e-prints (May 2017)
26. Zeiler, M.D., Fergus, R.: Visualizing and understanding convolutional networks. In: ECCV. pp. 818–833 (2014)