Towards Interpretable Object Detection by Unfolding Latent Structures

Tianfu Wu[†] and Xi Song [*]
[†]Department of ECE and the Visual Narrative Initiative, NC State University
tianfu_wu@ncsu.edu, xsong.lhi@gmail.com

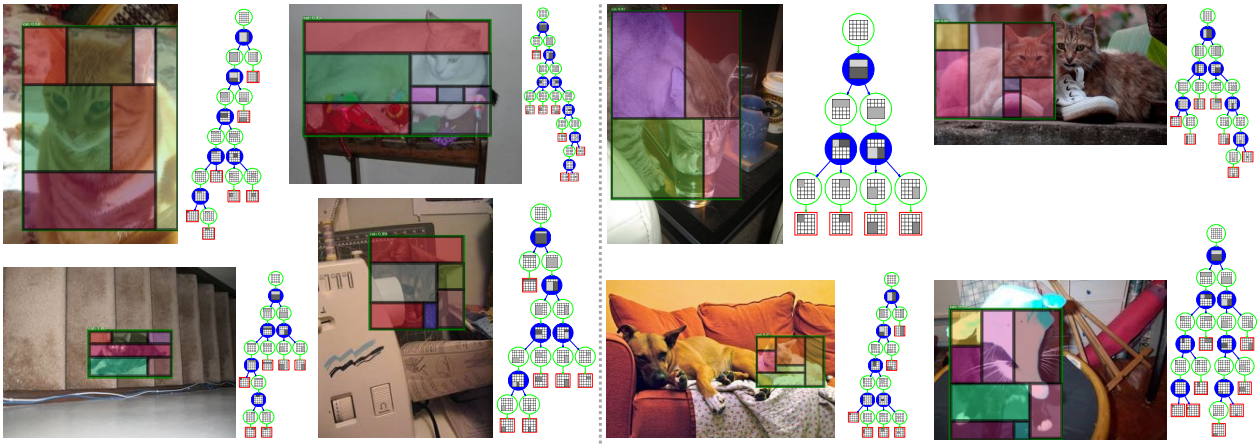

Figure 1. *Why are those bound boxes detected as cats?* Illustration of qualitatively interpreting model interpretability via unfolding latent structures end-to-end without any supervision used in training. Latent structures are represented by parse trees (shown to the right of each image) computed on-the-fly and object layouts/configurations (superposed on the bounding boxes) are collapsed from the parse trees. The left four images are from PASCAL VOC2007 test dataset and the right four ones from the COCO val2017 dataset. For clarity, only one detected object instance is shown. See text for details. Best viewed in color and magnification.

Abstract

This paper first proposes a method of formulating model interpretability in visual understanding tasks based on the idea of unfolding latent structures. It then presents a case study in object detection using popular two-stage region-based convolutional network (i.e., R-CNN) detection systems [19, 50, 7, 23]. The proposed method focuses on weakly-supervised extractive rationale generation, that is learning to unfold latent discriminative part configurations of object instances automatically and simultaneously in detection without using any supervision for part configurations. It utilizes a top-down hierarchical and compositional grammar model embedded in a directed acyclic AND-OR Graph (AOG) to explore and unfold the space of latent part configurations of regions of interest (RoIs). It presents an AOGParsing operator that seamlessly integrates with the RoIPooling [19]/RoIAlign [23] operator widely used in R-CNN and is trained end-to-end. In object detection, a bounding box is interpreted by the best parse tree derived from the AOG on-the-fly, which is treated as the qualitatively extractive rationale generated for interpreting detection. In experiments, Faster R-CNN [50] is used to test the proposed method on the PASCAL VOC 2007 [13] and the COCO 2017 [40] object detection datasets. The experimental results show that the proposed method can compute promising latent structures without hurting the performance. The code and pretrained models are available at https://github.com/iVMCL/iRCNN.

1. Introduction

1.1. Motivation and Objective

Recently, deep neural networks [37, 32] have improved prediction accuracy significantly in many vision tasks, and even outperform humans in image classification tasks [24, 58]. In the literature of object detection, there has been a critical shift from more explicit representation and mod-

[*]X. Song is an independent researcher.

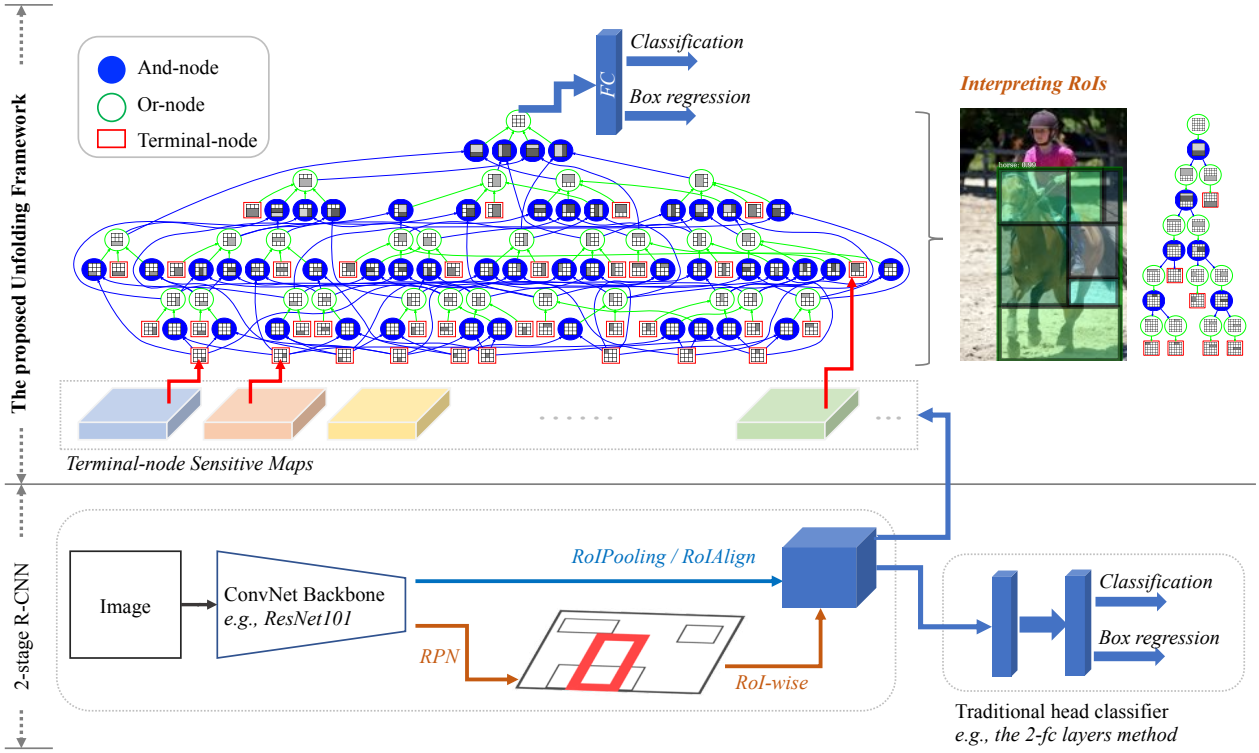

Figure 2. Illustration of the proposed method using Faster R-CNN [50] as the baseline system. Unfolding latent structures of Region-of-Interest (RoI) is realized by a generic top-down grammar model represented by a directed acyclic AND-OR Graph (AOG). *The AOG can be treated as the counterpart of explicit part representations for the implicit (black-box) flatten and fully-connected layer in the traditional head classifier.* For clarity, we show an AOG constructed for a 3×3 grid using the method proposed in [56]. The AOG unfolds the space of all possible latent part configurations. See text for details. (Best viewed in color and magnification)

els such as the mixture of deformable part-based models (DPMs) [16] and its many variants, and hierarchical and compositional AND-OR graphs (AOGs) models [56, 69, 59, 60], to less transparent but much more accurate ConvNet based approaches [50, 7, 49, 42, 23, 8]. Meanwhile, it has been shown that deep neural networks can be easily fooled by so-called adversarial attacks which utilize visually imperceptible, carefully-crafted perturbations to cause networks to misclassify inputs in arbitrarily chosen ways [47, 2], even with one-pixel attack [57]. And, it has also been shown that deep learning can easily fit random labels [66]. It is difficult to analyze why state-of-the-art deep neural networks work or fail due to the lack of theoretical underpinnings at present [1]. From cognitive science perspective, state-of-the-art deep neural networks might not learn and think like people who know and can explain "why" [34]. Nevertheless, there are more and more applications in which prediction results of computer vision and machine learning modules based on deep neural networks have been used in making decisions with potentially critical consequences (e.g., security video surveillance and autonomous driving).

It has become a common recognition that prediction without interpretable justification will have limited applicability eventually. It is a crucial issue of addressing machine's inability to explain its predicted decisions and actions (e.g., eXplainable AI or XAI proposed in the DARPA grant solicitation [10]), that is to improve accuracy and transparency jointly: Not only is an interpretable model capable of computing correct predictions of a random example with very high probability, but also rationalizing its predictions, preferably in a way explainable to end users. Generally speaking, learning interpretable models is to let machines make sense to humans, which usually consists of many challenging aspects. So there has not been a universally accepted definition of the notion of model interpretability. Especially, it remains a long-standing open problem to measure interpretability in a principled quantitative way.

To address the interpretability challenge, many work have proposed to visualize the internal filter kernels or to generate attentive activation maps, which reveal a lot of insights of what DNNs have learned in a post-hoc way. Complementary to those methods, **this paper focuses on how to unfold the latent structures for addressing model interpretability in learning and inference end-to-end** (see some examples in Figure 1). We first propose a method of formulating model interpretability, centered on the idea

of unfolding meaningful latent structures in a weakly-supervised way. We then present a case study in object detection. Our goal is to investigate the feasibility of integrating top-down grammar models with bottom-up ConvNet backbones end-to-end. The former are introduced to represent the space of latent structures hierarchically and compositionally and thus define the unfolding operations. We also aim to qualitatively rationalize the popular two-stage region-based ConvNets detection system, i.e., R-CNN [19, 50, 7] without hurting the detection performance. Jointly improving the performance and transparency is out of the scope of this paper, which is left for future work.

1.2. Method Overview

Figure 2 illustrates the proposed method for object detection. We adopt two-stage R-CNN as the baseline system. We focus on weakly-supervised extractive rationale generation in the RoI prediction component in R-CNN (e.g., the widely used 2-fc layers implementation), that is learning to unfold latent discriminative part configurations of RoIs automatically and simultaneously in detection without using any supervision for part configurations. We address the following two challenges.

i) **Moving from traditional flat structure representations of RoIs to hierarchical and compositional structure representations, and thus enabling from "uninformative" fully-connected exploration and exploitation of RoI features to grammatically-guided exploration and exploitation.** The popular RoIPooling/RoIAlign layers usually use predefined flat grid quantization such as 7×7 cells of input RoIs whose sizes vary. In the 2-fc layers implementation of RoI prediction component, the 7×7 cells is flatten, followed by two FC layers (see the bottom of Figure 2). In terms of prediction, this implementation is of highly discriminative power, seeking the most discriminative linear combinations in the high-dimensional RoI feature space. To enable interpretable object detection with respect to inferring latent object layouts/configurations, the flat structure of RoIs needs to be enriched, similar in spirit to how the spatial pyramid representation [35] was developed to enrich the bag-of-feature representation in scene classification tasks. *In this paper, we utilize a generic top-down hierarchical and compositional grammar model embedded in a directed acyclic AND-OR Graph (AOG) [56, 60] to explore and unfold the space of latent part configurations of RoIs* (see an example in the top of Figure 2). There are three types of nodes in an AOG: an *AND-node* represents binary decomposition of a large part into two smaller ones, an *OR-node* represents alternative ways of decomposition, and a *Terminal-node* represents a part instance. The AOG is consistent with the general image grammar framework [18, 70, 15, 69].

ii) **Distilling and inducing meaningful latent structures in weakly-supervised discriminative tasks.** According to the observations in network dissection [3], model interpretability and performance do not have strong correlations in discriminative tasks. Intuitively, since the objective function usually cares about performance only subject to generic model regularization, the model will pick up whatever features that are useful for minimizing the loss on the training dataset. So, even with the hierarchical and compositional representation introduced for RoIs, we are facing the difficulty of distilling and inducing the underlying meaningful latent structures in a weakly-supervised manner. In this paper, we first introduce *Terminal-node sensitive feature maps* in computing features using the AOG (see the top of Figure 2), similar in spirit to the position-sensitive feature maps used in the R-FCN [7]. Each Terminal-node feature map is low-dimensional (e.g., 20). We then introduce *a value sub-network* that computes the figure of merits (attention weights) of different Terminal-nodes which will be informative for bottom-up and top-down parsing RoIs with the AOG, similar in spirit to value sub-networks in deep reinforcement learning, e.g. in the AlphaGo [54]. We call the **AOGParsing operator** for the proposed component. We compare three ways of applying the value network.

- *The vanilla reweighing method* that re-calibrates the Terminal-node feature maps using the output of the value network.
- *A sparsity-inducing method* that only keeps the Top-k Terminal-nodes for each RoI individually, where k can be the grid size of RoIs (e.g., 49 of a 7×7 RoI).
- *An adversarial attack method* that is the opposite of the sparsity-inducing method, and removes the Top-k Terminal-nodes in terms of the output of the value network. With the discrminative object function, the sparsity-inducing method may be trapped in the subspace of unmeaningful yet discriminatively powerful latent structures. The adversarial attack method encourages exploration in the entire space of latent structures.

We note that defining interpretability-sensitive loss functions w.r.t. the AOG is a complementary direction to be studied in future work.

In experiments, we apply the proposed method using Faster R-CNN [50] as baseline system with the residual net [24] pretrained on the ImageNet [52]. We test our method on the PASCAL VOC 2007 [13] and the COCO 2017 [40] datasets with qualitatively meaningful latent structures learned and comparable performance retained.

2. Related Work

In general, model interpretability is very difficult to characterize.Efforts in addressing model interpretability w.r.t.

DNNs can be roughly categorized into the following two lines of work.

Interpret post-hoc interpretability of deep neural networks by associating explanatory semantic information with nodes in a deep neural network. There are a variety of methods including identifying high-scoring image patches [20, 43] or over-segmented atomic regions [51] directly, visualizing the layers of convolutional networks using deconvolutional networks to understand what contents are emphasized in the high-scoring input image patches [65], identifying items in a visual scene and recount multimedia events [64, 17], generating synthesized images by maximizing the response of a given node in the network [12, 36, 55] or by developing a top-down generative convolutional networks [45, 62], and analyzing and visualizing state activation in recurrent networks [26, 29, 38, 14] to link word vectors to semantic lexicons or word properties. On the other hand, Hendricks et al [25] extended the approaches used to generate image captions [30, 46] to train a second deep network to generate explanations without explicitly identifying the semantic features of the original network. Most of these methods are not model-agnostic except for [51]. More recently, the Grad-CAM work [53], built on top of the CAM work [68], can produce a coarse localization map highlighting the important regions in the image used by deep neural networks for predicting the concept. In similar spirit, the excitation back-propagation method [67] can generate task-specific attention map. The latest network dissection work [3] reported empirically that interpretable units are found in representations of the major deep learning architectures [32, 4, 24] for vision, and interpretable units also emerge under different training conditions. On the other hand, they also found that interpretability is neither an inevitable result of discriminative power, nor is it a prerequisite to discriminative power. Most of these methods are not model-agnostic except for [51, 31]. In [31], a classic technique in statistics, influence function, is used to understand the black-box prediction in terms of training sample, rather than extractive rationale justification.

Learn interpretable models directly. Following the analysis-by-synthesis principle, generative image modeling using deep neural networks has obtained significant progress with very vivid and sharp images synthesized since the breakthrough work, generative adversarial network [21], was proposed [11, 22, 6, 62, 48]. Apart from deep neural networks, Lake et al [33] proposed a probabilistic program induction model for handwritten characters that learns in a similar fashion to what people learn and works better than deep learning algorithms. The model classifies, parses, and recreates handwritten characters, and can generate new letters of the alphabet that look right as judged by Turing-like tests of the model's output in comparison to what real humans produce. There are a variety of interpretable models based on image grammar [70, 15, 41, 69], which can offer intuitive and deep explanation, but often are suffered from difficulties in learning model structures and recently being outperformed in terms of accuracy by deep neural networks significantly.

Spatial attention-like mechanism has been widely studied in deep neural network based systems, including, but not limited to, the seminal spatial transform network [27] which warps the feature map via a global parametric transformation such as affine transformation, the exploration of global average pooling and class specific activation maps for weakly-supervised discriminative localization [68], the deformable convolution network [9] and active convolution [28], and more explicit attention based work in image caption and visual question answering (VQA) such as the show-attend-tell work [63] and the hierarchical co-attention in VQA [44]. Attention based work unfold the localization power of filter kernels in deep neural networks. *The proposed end-to-end integration of the top-down full structure grammar and bottom-up deep neural networks attempts to harness the power from both methodologies in visual recognition, which can be treated as hierarchical and compositional structure based spatial attention mechanism.*

Our Contributions. This paper makes three main contributions to the emerging field of learning interpretable models as follows: (i) It presents a method of integrating a generic top-down grammar model, embedded in an AOG, and bottom-up ConvNets end-to-end to learn qualitatively interpretable models in object detection. (ii) It presents an AOGParsing operator which can seamlessly integrate with the RoIPooling/RoIAlign operators widely used in R-CNN based detection systems. (iii) It shows detection performance comparable to state-of-the-art R-CNN systems, thus shedding light on addressing accuracy and transparency jointly in learning deep models for object detection.

3. Interpreting Model Interpretability

In this section, we present a generic formulation of model interpretability in visual understanding tasks which accounts for unfolding well-defined latent structures in a weakly-supervised way.

Intuitively, we would expect that an interpretable model could learn and capture latent semantic structures automatically which are not annotated in training data. For example, if we consider the basic image classification task with only image labels available in training as commonly used, to compare which classification models are more interpretable or explainable, one principled way is to show the capability of extracting the latent localization of object of interest w.r.t. the ground-truth label. Similarly, a person detector is more interpretable if it is learned using person bounding box annotations only, but capable of interpreting a person detection with the latent semantic structure explained, ide-

ally the kinetic pose. So, *our intuitive idea is that model interpretability can be posed as the capability of exploring the latent space of a higher level task* (e.g., localization vs classification and pose recovery vs detection) in a principled way, and of capturing the sufficient statistics in the latent space. The more a model can explore and capture the latent tasks at higher level, the better the model interpretability is.

To that end, we first consider an underlying task hierarchy, e.g., from image classification, to object localization and detection, to object part recovery (object parsing), and all the way to full image parsing (i.e., all image pixels are explained-away in a mathematically sound way). Then, for a task at hand (e.g., object detection), we seek a principled way of defining and exploring the latent space of the task of object part-based parsing, and then compute extractive rationale for the task at hand.

Let Λ be the domain on which the latent structures are defined such as the image lattice in image classification or the RoI in object detection. Our formulation is a straightforward top-down method consisting of two components:

- *A domain parser* that unfolds the latent structures of the domain Λ in an effective and compact way. The parser can be built either in a greedy pursuit way as done in the classic deformable part-based models (DPMs) [16] or in a top-down fashion such as the classic quad-tree method or more generally as done in the AND-OR Tree (AOT) models [56, 60]. We use the latter in this paper. Denote by Ω_Λ the space of latent structures computed by a domain Parser.
- *A data-driven parsing algorithm* that seeks the optimal latent structure in Ω_Λ for a given sample x defined on Λ. Thanks to the DAG structure of the AOG used in this paper, it is straightforward to implement the parsing algorithm in two phases: a bottom-up phase following the depth-first search (DFS) order to compute the figure of merits of all nodes in the AOG, and a top-down phase following the breadth-first search (BFS) order to retrieve the optimal latent structure by making decisions at each encountered OR-nodes.

4. A Case Study: Interpretable R-CNN

In this section, we first briefly present background on R-CNN and the construction of the top-down AOG [56, 60] to be self-contained. Then, we present the end-to-end integration of AOG and R-CNN.

4.1. Background

The R-CNN Framework. The R-CNN framework consists of three components: (i) A ConvNet backbone such as the Residual Net [24] for feature extraction, parameterized by Θ_0 and shared between the region-proposal network (RPN) and the RoI prediction network. (ii) The RPN network for objectness detection (i.e., category-agnostic detection through binary classification between foreground objects and background) and bounding box regression, parameterized by Θ_1. Denote by B a RoI (i.e., a foreground bounding box proposal) computed by the RPN. (iii) The RoI prediction network for classifying a RoI B and refining it, parameterized by Θ_2, which utilizes the RoIPooling operator and usually use one or two fully connected layer(s) as the head classifier and regressor. The parameters $\Theta = (\Theta_0, \Theta_1, \Theta_2)$ are trained end-to-end.

The AOG as the domain parser. In the R-CNN framework, a RoI is interpreted as a predefined flat configuration. To learn interpretable models, we need to explore the space of latent part configurations defined in a RoI. To that end, a RoI is first divided into a grid of cells as done in the RoIPooling operator (e.g., 3×3 or 7×7). Denote by $S_{x,y,w,h}$ and $t_{x,y,w,h}$ a non-terminal symbol and a terminal symbol respectively, both representing the sub-grid with left-top (x, y) and width and height (w, h) in the RoI. We only utilize binary decomposition, either Horizontal cut or Vertical cut, when interpreting a non-terminal symbol. We have four rules,

$$S_{x,y,w,h} \xrightarrow{Termination} t_{x,y,w,h} \tag{1}$$

$$S_{x,y,w,h}(l; \leftrightarrow) \xrightarrow{Ver.Cut} S_{x,y,l,h} \cdot S_{x+l,y,w-l,h} \tag{2}$$

$$S_{x,y,w,h}(l; \updownarrow) \xrightarrow{Hor.Cut} S_{x,y,w,l} \cdot S_{x,y+l,w,h-l} \tag{3}$$

$$\begin{aligned} S_{x,y,w,h} \rightarrow\ & t_{x,y,w,h} | S_{x,y,w,h}(l_{min}; \leftrightarrow) | \cdots | \\ & S_{x,y,w,h}(w - l_{min}; \leftrightarrow) | S_{x,y,w,h}(l_{min}; \updownarrow) | \cdots | \\ & S_{x,y,w,h}(h - l_{min}; \updownarrow), \end{aligned} \tag{4}$$

where l_{min} represents the minimum side length of a valid sub-grid allowed in the decomposition (e.g., $l_{min} = 1$). When instantiated, the first rule will be represented by *Terminal-nodes*, both the second and the third by *AND-nodes*, and the fourth by *OR-nodes*.

The top-down AOG is constructed by applying the four rules in a recursive way [56, 60]. Denote an AOG by $\mathcal{G} = (V, E)$ where $V = V_{And} \cup V_{Or} \cup V_T$ and V_{And}, V_{Or} and V_T represent a set of AND-nodes, OR-nodes and Terminal-nodes respectively, and E a set of edges. We start with $V = \emptyset$ and $E = \emptyset$, and a first-in-first-out queue $Q = \emptyset$. It unfolds all possible latent configurations. Figure 2 shows the AOG constructed for a 3×3 grid.

A parse tree is an instantiation of the AOG, which follows the breadth-first-search (BFS) order of nodes in the AOG, selects the best child node for each encountered OR-nodes, keeps both child nodes for each encountered AND-node, and terminates at each encountered Terminal-node. *A configuration* is generated by collapsing all the Terminal-nodes of a parse tree onto the image domain.

4.2. The AOGParsing Operator in R-CNN

We now present a simple end-to-end integration of the top-down AOG in R-CNN as illustrated in Figure 2. Consider an AOG $\mathcal{G}_{h,w,l_{min}}$ with the grid size being $h \times w$ and the minimum side length l_{min} allowed for nodes (e.g., $\mathcal{G}_{3,3,1}$ in Figure 2).

Terminal-node sensitive feature maps. Denote by F_t the Terminal-node sensitive feature map for a Terminal-node $t \in V_T$ in the AOG, $\mathcal{G}_{h,w,l_{min}}$. All F_t's have the same dimensions, $C \times H \times W$, where the height H and the width W are the same as those of outputs of RoIPooling/RoIAlign (e.g., 7×7), and the channel C the number of channels which is relatively small, especially for big AOGs (e.g., $C = 20$). Let F_{RoI} be the output feature maps of RoIPooling or RoIAlign (see Figure 2). F_t's are usually computed through either 1×1 or 3×3 convolution.

Denote by f_t the C-dimension feature of a Terminal-node t. f_t is computed via either channel-wise average-pooling or max-pooling in sub-domain occupied by $t_{x,y,w,h}$ in the feature map F_t. Denote by f_{V_T} the $C \times |V_T|$-dimension feature vector concatenated from all the Terminal-nodes.

Computing Terminal-node value. We use a simple 2-layer FC sub-network (e.g., FC+ReLu+FC+Sigmoid) which takes f_{V_T} as the input and outputs $|V_T|$ scores for Terminal-nodes value. Let s_t be the value score of a Terminal-node t. Let s_{V_T} be the slice repeated Terminal-nodes value vector which is of $C \times |V_t|$-dimension. Based on the three policies of applying the value network, we have $s^{base}_{V_T}$ as the baseline weight vector, $s^{k}_{V_T}$ the Top-k sparsity-inducing one, and the $s^{adv}_{V_T}$ the adversarial attack one. Without loss of generality, denote by f^p_t as the re-calibrated feature vector for a Terminal-node t according to a given policy $p \in \{base, k, adv\}$. Similarly, $f^p_{V_T}$ is the concatenated feature vector.

Computing features and values for AND- and OR-nodes. For simplicity, we use MEAN and MAX operations for AND-nodes and OR-nodes respectively. We follow the DFS order. For an AND-node, both its feature and value are the average of its child nodes. For an OR-node, its value is the maximum of values of its child nodes and its feature is then the one from the child node with the maximum value.

Computing the optimal parse tree for each sample. The parse tree can be retrieved in a straightforward way following the BFS order of nodes in the AOG. Starting from the root node, each encountered OR-node selects its best child and each encountered AND-node keeps all the child nodes. The latent structure is then defined by the Terminal-nodes in the retrieved parse tree. Each sample is then represented by a $C \times |V_T|$-dimension feature with Terminal-nodes in the inferred latent structure kept only and others zeroed-out.

As illustrated in Figure 2, another FC layer can be further used to fuse the information of the inferred latent structures, which is shared by the classification and box regression branches.

Method	mAP (VOC)	Box AP (COCO)
Faster R-CNN [50]*	82.1	38.5
Faster R-CNN-D [71]*	82.2	-
Ours $\text{AOG}_{3,3,1} + base$	81.9	-
Ours $\text{AOG}_{3,3,1} + k$	81.2	-
Ours $\text{AOG}_{3,3,1} + adv$	81.4	-
Ours $\text{AOG}_{5,5,1} + base$	82.1	38.2
Ours $\text{AOG}_{5,5,1} + k$	80.4	37.0
Ours $\text{AOG}_{5,5,1} + adv$	81.4	38.0
Ours $\text{AOG}_{7,7,1} + base$	81.7	-
Ours $\text{AOG}_{7,7,1} + k$	81.2	-
Ours $\text{AOG}_{7,7,1} + adv$	81.7	-

Table 1. Performance comparisons using Average Precision (AP) at the intersection over union (IoU) threshold 0.5 (AP@0.5) in the PASCAL VOC2007 test dataset (using the protocol, competition "comp4" trained using both 2007 and 2012 trainval datasets) and the *coco_val2017* dataset. * reported by retraining the models provided in MMDetection for fair comparisons.

Latent structure oriented feature normalization. Different sample in a min-batch may use very different latent structures of varied number of Terminal-nodes selected. To reduce the fluctuations for the following FC layers, we can normalize the features of a latent structure by dividing the number of selected Terminal-nodes.

Thanks to its DAG structure, the integration of AOG will not affect the end-to-end training. However the training efficiency is usually affected by the bottom-up phase and the top-down phase of the AOGParsing operation due to their serial nature.

4.3. The Folding-Unfolding Learning

Since the Terminal-node sensitive feature maps and values are computed with randomly initialized parameters, it is not reasonable to compute good node values and make good decisions on selecting the best child for each OR-node at the beginning in the forward step. All nodes not retrieved by the parse trees will not get gradient update in the backward step. So, we resort to a folding-unfolding learning strategy. In the folding stage, we directly use $f^p_{V_T}$, so all Terminal-nodes and the value sub-network are trained in a fair fashion. After a few epochs, we then switch to the unfolding stage of learning following the entire recipe in Section 4.2.

5. Experiments

In this section, we present experimental results on the PASCAL VOC 2007 [13] and the COCO 2017 [40]. We implement the proposed method in the latest MMDetection [1]

[1] https://github.com/open-mmlab/mmdetection

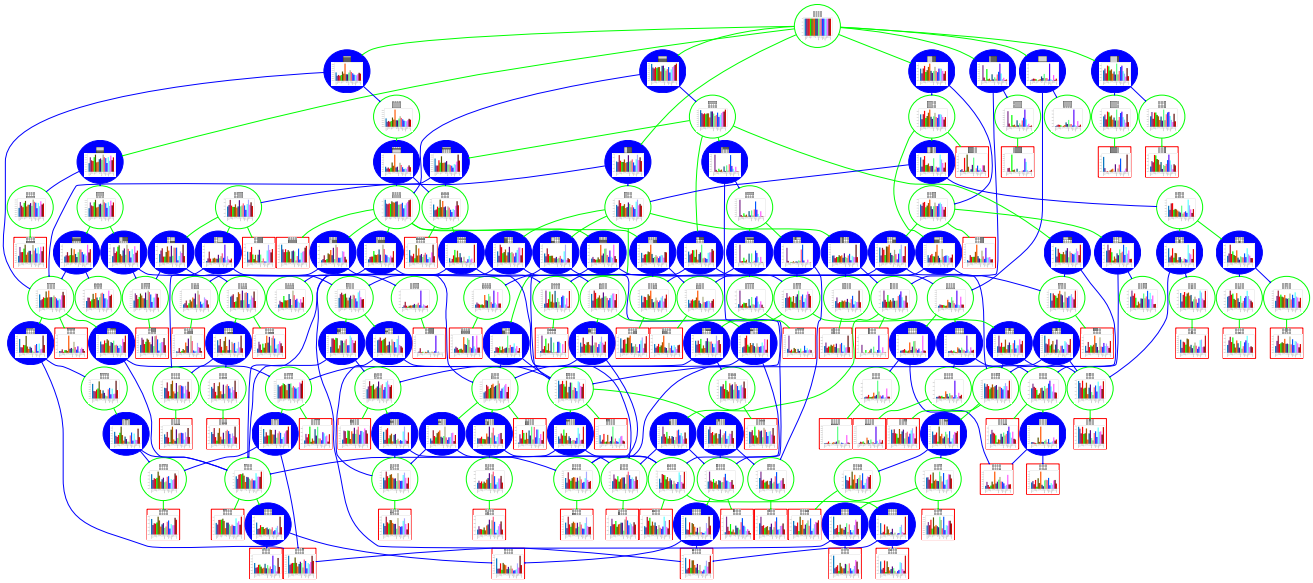

Figure 3. The AOG$_{5,5,1}$ + adv learned in the PASCAL VOC 2007 and 2012 *trainval* datasets. In the each, we plot the category distribution showing the proportion of the true positive per class. (Best viewed in color and magnification)

code platform [5]. We build on Faster R-CNN [50] with ResNet101 [24] and feature pyramid network (FPN) [39] as backbones. We maintain the model complexity comparable by tuning the feature dimension C in computing the Terminal-node sensitive features and the value sub-network. The inference time of our method is mostly comparable to the baseline. We conduct experiments with three different AOGs, $\mathcal{G}_{3,3,1}$, $\mathcal{G}_{5,5,1}$ and $\mathcal{G}_{7,7,1}$ in PASCAL VOC 2007. We only test $\mathcal{G}_{5,5,1}$ in COCO. We following the default hyper-parameter settings (e.g., the total number of epochs, the initial learning rate and its schedule) provided in the MMDetection platform. For the folding-unfolding learning, we usually use half number of epochs for folding and the other half for folding. We note that the proposed method can be tested in other systems implemented in the MMDetection platform in a straightforward way.

The proposed method obtains consistently comparable accuracy performance with the baseline system. Table 1 summarizes the results. We note that the observed fluctuations of performance may be caused by not tuning some of hyper-parameters. We will present and update more results with tuned training parameters in our Github repository. Interestingly, we observe that for the three policies of applying the value sub-network, the vanilla one obtains the best performance, and the adversarial attack one is better than the Top-k sparsity-inducing one. In the current implementation, the value sub-network is simple focusing on Terminal-nodes only without considering the AOG structures. And, the hard way of removing non-selected Terminal-nodes in both the Top-k and its counterpart may need to be relaxed to some soft versions. In the following, we will focus on analyzing the qualitative interpretability of the proposed method in the following.

Examples in Figure 1 and Figure 2 are all models trained with AOG$_{5,5,1}$ + adv. Figure 3 shows the learned AOG$_{5,5,1}$+adv in PASCAL VOC. Although it is not easy to interpret the "meaningfulness" of the learned AOG, it sheds light on developing interpretability-sensitive objective functions in learning interpretable models from scratch. For example, with the AOGs, we will be able to formulate the following two terms into an interpretability-sensitive object functions.

Explainability and Sparsity in the space of latent part configurations of a sample x. The intuitive idea is that an underlying interpretable model should focus much more on the most "meaningful" latent part configuration for a random sample, which covers the most important underlying semantic regions including both intrinsic and contextual ones, not necessarily connected, of an image w.r.t. the label. Furthermore, the focused latent part configuration should be stable and consistent between the original sample and other new augmented samples. If we could unfold the space of latent part configurations, which is usually huge, we can evaluate the interpretability score in the spirit similar to the masking and scaling operators used in [61] for evaluating information contributions of bottom-up/top-down computing processes in a hierarchical model.

Stability of the focused latent part configurations across different images within a category. The intuitive idea is that the number of distinct focused latent part configurations unfolded for different samples within a category should be small, i.e., most of them shared among a subset of samples.

Limitations and Discussions. The proposed method has two main limitations to be addressed in future work. First,

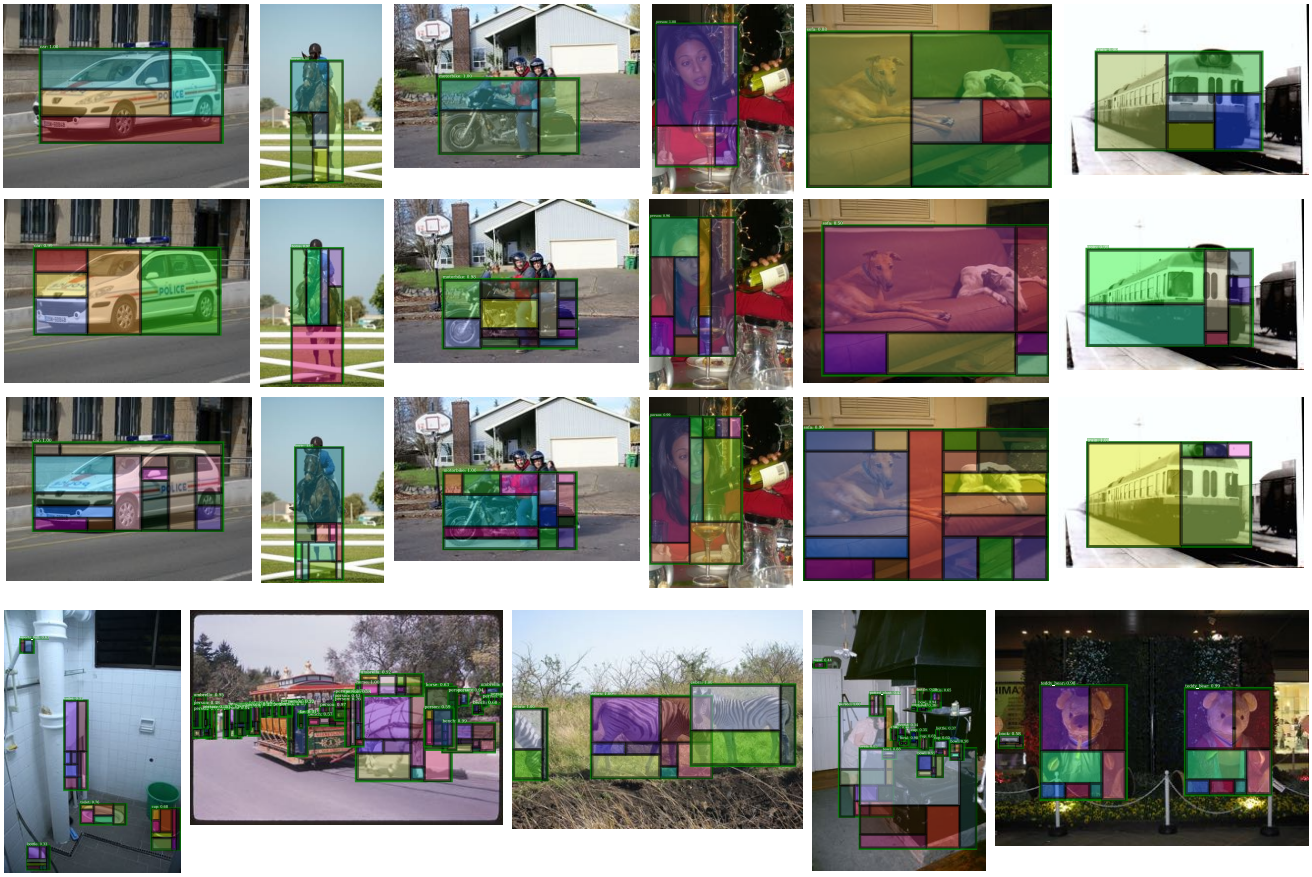

Figure 4. Examples of latent structures unfolded by AOGs. The first three rows show comparisons between results of the three AOGs, $\mathcal{G}_{3,3,1}+adv$, $\mathcal{G}_{5,5,1}+adv$ and $\mathcal{G}_{7,7,1}+adv$ in the PASCAL VOC 2007. For clarity, we show one instance only in each image. The fourth row shows a few detection results in COCO. (Best viewed in color and magnification)

although it can show qualitative extractive rationale in detection in a weakly-supervised way, it is difficult to quantitatively measure the model interpretability. One potential direction for quantitative interpretability is that we will investigate rigorous definitions which can be formalized as a interpretability-sensitive loss term in end-to-end training, as briefly discussed above. Second, current implementation of the proposed method did not improve the accuracy performance although it is not our focus in this paper. We will explore new operators for AND-nodes and OR-nodes in the AOG to improve performance. We hope detection performance will be further improved with the interpretability-sensitive loss terms.

6. Conclusion

This paper presented a method of integrating a generic top-down grammar model (specifically the AND-OR grammar model) with bottom-up ConvNets in an end-to-end way for learning qualitatively interpretable models in object detection using the R-CNN framework. It builds on top the two-stage R-CNN method and proposes an AOGParsing operator that seamlessly integrates with the RoIPooling/RoIAlign operators to unfold the space of latent part configurations. It proposed a folding-unfolding method in learning. In experiments, the proposed method is tested in the PASCAL VOC 2007 and COCO val2017 benchmarks with performance comparable to state-of-the-art baseline R-CNN detection methods. The proposed method computes the optimal parse tree in the AOG as qualitatively extractive rationale in "justifying" detection results. It sheds light on learning quantitatively interpretable models in object detection.

Acknowledgement

The authors would like to thank the anonymous reviewers for their helpful comments. This work was supported in part by ARO grant W911NF1810295, NSF IIS-1909644, Salesforce Inaugural Deep Learning Research Grant (2018) and ARO DURIP grant W911NF1810209. The views presented in this paper are those of the authors and should not be interpreted as representing any funding agencies.

References

[1] Sanjeev Arora, Aditya Bhaskara, Rong Ge, and Tengyu Ma. Provable bounds for learning some deep representations. In *ICML*, pages 584–592, 2014. 2

[2] Anish Athalye and Ilya Sutskever. Synthesizing robust adversarial examples. *CoRR*, abs/1707.07397, 2017. 2

[3] David Bau, Bolei Zhou, Aditya Khosla, Aude Oliva, and Antonio Torralba. Network dissection: Quantifying interpretability of deep visual representations. In *CVPR*, 2017. 3, 4

[4] Ken Chatfield, Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Return of the devil in the details: Delving deep into convolutional nets. In *BMVC*, 2014. 4

[5] Kai Chen, Jiaqi Wang, Jiangmiao Pang, Yuhang Cao, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jiarui Xu, Zheng Zhang, Dazhi Cheng, Chenchen Zhu, Tianheng Cheng, Qijie Zhao, Buyu Li, Xin Lu, Rui Zhu, Yue Wu, Jifeng Dai, Jingdong Wang, Jianping Shi, Wanli Ouyang, Chen Change Loy, and Dahua Lin. Mmdetection: Open mmlab detection toolbox and benchmark. *CoRR*, abs/1906.07155, 2019. 7

[6] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. *CoRR*, abs/1606.03657, 2016. 4

[7] Jifeng Dai, Yi Li, Kaiming He, and Jian Sun. R-FCN: object detection via region-based fully convolutional networks. In *NIPS*, 2016. 1, 2, 3

[8] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. *CoRR*, abs/1703.06211, 2017. 2

[9] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. *CoRR*, abs/1703.06211, 2017. 4

[10] DARPA. Explainable artificial intelligence (xai) program, http://www.darpa.mil/program/explainable-artificial-intelligence, full solicitation at http://www.darpa.mil/attachments/ darpa-baa-16-53.pdf. 2

[11] Emily L. Denton, Soumith Chintala, Arthur Szlam, and Rob Fergus. Deep generative image models using a laplacian pyramid of adversarial networks. In *NIPS*, pages 1486–1494, 2015. 4

[12] Dumitru Erhan, Yoshua Bengio, Aaron Courville, and Pascal Vincent. Visualizing higher-layer features of a deep network. Technical Report 1341, University of Montreal, June 2009. 4

[13] Mark Everingham, S. M. Eslami, Luc Gool, Christopher K. Williams, John Winn, and Andrew Zisserman. The pascal visual object classes challenge: A retrospective. *IJCV*, 111(1):98–136, Jan. 2015. 1, 3, 6

[14] Manaal Faruqui, Jesse Dodge, Sujay Kumar Jauhar, Chris Dyer, Eduard H. Hovy, and Noah A. Smith. Retrofitting word vectors to semantic lexicons. In *NAACL*, 2015. 4

[15] Pedro F. Felzenszwalb. Object detection grammars. In *ICCV-Workshops*, page 691, 2011. 3, 4

[16] Pedro F. Felzenszwalb, Ross B. Girshick, David McAllester, and Deva Ramanan. Object detection with discriminatively trained part-based models. *TPAMI*, 32(9):1627–1645, Sept. 2010. 2, 5

[17] Chuang Gan, Naiyan Wang, Yi Yang, Dit-Yan Yeung, and Alexander G. Hauptmann. Devnet: A deep event network for multimedia event detection and evidence recounting. In *CVPR*, pages 2568–2577, 2015. 4

[18] Stuart Geman, Daniel Potter, and Zhi Yi Chi. Composition systems. *Quarterly of Applied Mathematics*, 60(4):707–736, 2002. 3

[19] Ross Girshick. Fast R-CNN. In *ICCV*, 2015. 1, 3

[20] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, 2014. 4

[21] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C. Courville, and Yoshua Bengio. Generative adversarial nets. In *NIPS*, pages 2672–2680, 2014. 4

[22] Tian Han, Yang Lu, Song-Chun Zhu, and Ying Nian Wu. Learning generative convnet with continuous latent factors by alternating back-propagation. *CoRR*, abs/1606.08571, 2016. 4

[23] Kaiming He, Georgia Gkioxari, Piotr Dollr, and Ross Girshick. Mask R-CNN. In *ICCV*, 2017. 1, 2

[24] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 1, 3, 4, 5, 7

[25] Lisa Anne Hendricks, Zeynep Akata, Marcus Rohrbach, Jeff Donahue, Bernt Schiele, and Trevor Darrell. Generating visual explanations. In *ECCV*, 2016. 4

[26] Michiel Hermans and Benjamin Schrauwen. Training and analysing deep recurrent neural networks. In *NIPS*, pages 190–198, 2013. 4

[27] Max Jaderberg, Karen Simonyan, Andrew Zisserman, and Koray Kavukcuoglu. Spatial transformer networks. In *NIPS*, 2015. 4

[28] Yunho Jeon and Junmo Kim. Active convolution: Learning the shape of convolution for image classification. *CoRR*, abs/1703.09076, 2017. 4

[29] Andrej Karpathy, Justin Johnson, and Fei-Fei Li. Visualizing and understanding recurrent networks. *CoRR*, abs/1506.02078, 2015. 4

[30] Andrej Karpathy and Fei-Fei Li. Deep visual-semantic alignments for generating image descriptions. In *CVPR*, pages 3128–3137, 2015. 4

[31] P. W. Koh and P. Liang. Understanding black-box predictions via influence functions. In *ICML*, 2017. 4

[32] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, pages 1106–1114, 2012. 1, 4

[33] Brenden M. Lake, Ruslan Salakhutdinov, and Joshua B. Tenenbaum. Human-level concept learning through probabilistic program induction. *Science*, 350(6266):1332–1338, 2015. 4

[34] Brenden M. Lake, Tomer D. Ullman, Joshua B. Tenenbaum, and Samuel J. Gershman. Building machines that learn and think like people. *CoRR*, abs/1604.00289, 2016. 2

[35] Svetlana Lazebnik, Cordelia Schmid, and Jean Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *CVPR*, 2006. 3

[36] Quoc V. Le, Marc'Aurelio Ranzato, Rajat Monga, Matthieu Devin, Greg Corrado, Kai Chen, Jeffrey Dean, and Andrew Y. Ng. Building high-level features using large scale unsupervised learning. In *ICML*, 2012. 4

[37] Yann LeCun, Leon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. 1

[38] Jiwei Li, Xinlei Chen, Eduard H. Hovy, and Dan Jurafsky. Visualizing and understanding neural models in NLP. In *NAACL*, pages 681–691, 2016. 4

[39] Tsung-Yi Lin, Piotr Dollár, Ross B. Girshick, Kaiming He, Bharath Hariharan, and Serge J. Belongie. Feature pyramid networks for object detection. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 936–944, 2017. 7

[40] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, Lubomir D. Bourdev, Ross B. Girshick, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: common objects in context. *CoRR*, abs/1405.0312, 2014. 1, 3, 6

[41] Meg Aycinena Lippow, Leslie Pack Kaelbling, and Tomás Lozano-Pérez. Learning grammatical models for object recognition. In *Logic and Probability for Scene Interpretation*, 2008. 4

[42] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C. Berg. SSD: Single shot multibox detector. In *ECCV*, 2016. 2

[43] Jonathan Long, Ning Zhang, and Trevor Darrell. Do convnets learn correspondence? In *NIPS*, 2014. 4

[44] Jiasen Lu, Jianwei Yang, Dhruv Batra, and Devi Parikh. Hierarchical question-image co-attention for visual question answering. In *NIPS*, 2016. 4

[45] Yang Lu, Song-Chun Zhu, and Ying Nian Wu. Learning FRAME models using CNN filters. In *AAAI*, 2016. 4

[46] Junhua Mao, Wei Xu, Yi Yang, Jiang Wang, and Alan L. Yuille. Deep captioning with multimodal recurrent neural networks (m-rnn). *CoRR*, abs/1412.6632, 2014. 4

[47] Anh Mai Nguyen, Jason Yosinski, and Jeff Clune. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *CVPR*, pages 427–436, 2015. 2

[48] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *CoRR*, abs/1511.06434, 2015. 4

[49] Joseph Redmon, Santosh Kumar Divvala, Ross B. Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *CVPR*, 2016. 2

[50] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *NIPS*, 2015. 1, 2, 3, 6, 7

[51] Marco Túlio Ribeiro, Sameer Singh, and Carlos Guestrin. "why should I trust you?": Explaining the predictions of any classifier. *CoRR*, abs/1602.04938, 2016. 4

[52] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *IJCV*, 115(3):211–252, 2015. 3

[53] Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *ICCV*, 2017. 4

[54] David Silver, Aja Huang, Chris J. Maddison, Arthur Guez, Laurent Sifre, George van den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, Sander Dieleman, Dominik Grewe, John Nham, Nal Kalchbrenner, Ilya Sutskever, Timothy Lillicrap, Madeleine Leach, Koray Kavukcuoglu, Thore Graepel, and Demis Hassabis. Mastering the game of Go with deep neural networks and tree search. *Nature*, 529(7587):484–489, 2016. 3

[55] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *CoRR*, abs/1312.6034, 2013. 4

[56] Xi Song, Tianfu Wu, Yunde Jia, and Song-Chun Zhu. Discriminatively trained and-or tree models for object detection. In *CVPR*, pages 3278–3285, 2013. 2, 3, 5

[57] Jiawei Su, Danilo Vasconcellos Vargas, and Sakurai Kouichi. One pixel attack for fooling deep neural networks. *CoRR*, abs/1710.08864, 2017. 2

[58] Christian Szegedy, Sergey Ioffe, and Vincent Vanhoucke. Inception-v4, inception-resnet and the impact of residual connections on learning. *CoRR*, abs/1602.07261, 2016. 1

[59] Tianfu Wu, Bo Li, and Song-Chun Zhu. Learning and-or model to represent context and occlusion for car detection and viewpoint estimation. *TPAMI*, 38(9):1829–1843, 2016. 2

[60] Tianfu Wu, Yang Lu, and Song-Chun Zhu. Online object tracking, learning and parsing with and-or graphs. *TPAMI*, 2016. 2, 3, 5

[61] Tianfu Wu and Song Chun Zhu. A numerical study of the bottom-up and top-down inference processes in and-or graphs. *IJCV*, 93(2):226–252, 2011. 7

[62] Jianwen Xie, Yang Lu, Song-Chun Zhu, and Ying Nian Wu. A theory of generative convnet. In *ICML*, 2016. 4

[63] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron C. Courville, Ruslan Salakhutdinov, Richard S. Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *ICML*, 2015. 4

[64] Qian Yu, Jingen Liu, Hui Cheng, Ajay Divakaran, and Harpreet S. Sawhney. Multimedia event recounting with concept based representation. In *MM*, pages 1073–1076, 2012. 4

[65] Matthew D. Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *ECCV*, pages 818–833, 2014. 4

[66] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. 2016. 2

[67] Jianming Zhang, Zhe L. Lin, Jonathan Brandt, Xiaohui Shen, and Stan Sclaroff. Top-down neural attention by excitation backprop. In *ECCV*, 2016. 4

[68] Bolei Zhou, Aditya Khosla, Àgata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *CVPR*, 2016. 4

[69] Long Zhu, Yuanhao Chen, Yifei Lu, Chenxi Lin, and Alan L. Yuille. Max margin AND/OR graph learning for parsing the human body. In *CVPR*, 2008. 2, 3, 4

[70] Song Chun Zhu and David Mumford. A stochastic grammar of images. *Found. and Trends in Comp. G. and V.*, 2(4):259–362, 2006. 3, 4

[71] Xizhou Zhu, Han Hu, Stephen Lin, and Jifeng Dai. Deformable convnets v2: More deformable, better results. *CoRR*, abs/1811.11168, 2018. 6