
The Imitation Game: Object Detection in 20 Questions

Anonymous Author(s)

Affiliation

Address

email

Abstract

We propose a new strategy for simultaneous object detection and segmentation. Instead of evaluating all object detectors at all possible locations in the image, we develop a divide-and-conquer approach by sequentially posing questions about the query and its related context—like playing a “Twenty Questions” game—to decide where to search for the object. We formulate the problem as a Markov Decision Process and learn a search policy by imitation learning. At each step, the policy dynamically selects a question based on the query, the scene and observed responses given by object detectors evaluated so far. Experimental results show that our algorithm reduces over 30% of the object proposal evaluation while improving performance in average precision compared to exhaustive search. Our learned policy also achieves better speed-accuracy tradeoff than the random search strategy.

1 Introduction

Object detection and segmentation in complex scenes is a central and challenging problem in computer vision and robotics. This problem is usually tackled by running multiple object detectors exhaustively on densely sampled sliding windows [9] or category-independent object proposals [7, 24, 2]. Such methods are time-consuming since they need to evaluate a large number of object hypotheses, and can easily introduce false positives if exclusively considering local appearance.

Instead of checking all hypotheses indiscriminately and exhaustively, humans only look for a set of related objects in a given context [4, 15]. Context information is an effective cue for humans to detect low-resolution or small objects in cluttered scenes [22]. Many contextual models have been proposed to capture relationships between objects at the semantic level to reduce ambiguities from unreliable independent detection results [13, 10, 17]. However, such methods still need to evaluate the high order co-occurrence statistics and spatial relations of the query object with *all* other object classes in the scene, some of which may not be informative and even introduce unnecessary confusion.

By contrast, humans do not process the whole scene at once: it is an active process that sequentially samples the optic array in an intelligent, task-specific way [21]. Research in neuro-science has revealed that when humans search for a target, those objects that are associated to the query will reinforce attention with the query and weaken recognition of unrelated distractions [19]. For instance, in Figure 1, when we search for cars, knowing the top of the scene is sky does not help distinguish whether there is a car or a boat since both can be under the sky; on the other hand, observing a road instead of water in the lower part gives a strong indication of the existence of cars. Additionally, we know from experience that roads, cars and buildings are likely to co-occur and cars are often found beside buildings. Therefore to find cars, humans tend to first look for roads and then search around buildings, instead of looking up for the sky. This motivates us to raise the question: *can*

object detection algorithms decide where to look for objects of a query class more efficiently and accurately by exploring a few related context cues dynamically, similar to humans?

We formulate the object detection problem as a Markov Decision Process (MDP). We use imitation learning to learn a context-driven policy that sequentially and dynamically selects the most informative context class to explore based on past observations, and gradually refine the search area for the query class. We show our framework in Figure 2. Specifically, like playing a Twenty Questions game, at each step the policy asks for information about a context class based on the query and responses from previous contextual classifiers. We then run the detector/classifier of the inquired context class. Based on the responses, we further refine the search area for the query class using spatial-aware contextual models. This process of context querying and search area refining is repeated until the policy thinks enough contextual information has been gathered and decides to stop. Finally, we run the query object detector in the refined search area and output the result. Besides asking for contextual information, our policy can reject a query early to avoid unnecessary detection if it determines that there is a small chance of having the query object in the scene. The decision of early rejection can be made even before running any object detector, therefore we can eliminate a large amount of unnecessary computation.

Object detection experimental results on images of complex scenes (the PASCAL VOC 2010 dataset) show that our algorithm produces a search area that has better overlap with the target object by leveraging its context, thus significantly reduces the number of object proposals to consider and detectors to evaluate[hhe: Add the number here]. Although with less computation, our method achieves mean average precision (mAP) comparable to the exhaustive search method. To the best of our knowledge, this is one of the first few approaches that solve the challenging task of simultaneous object detection and segmentation in complex real scenes by applying imitation learning to learn a policy fully driven by semantic context.

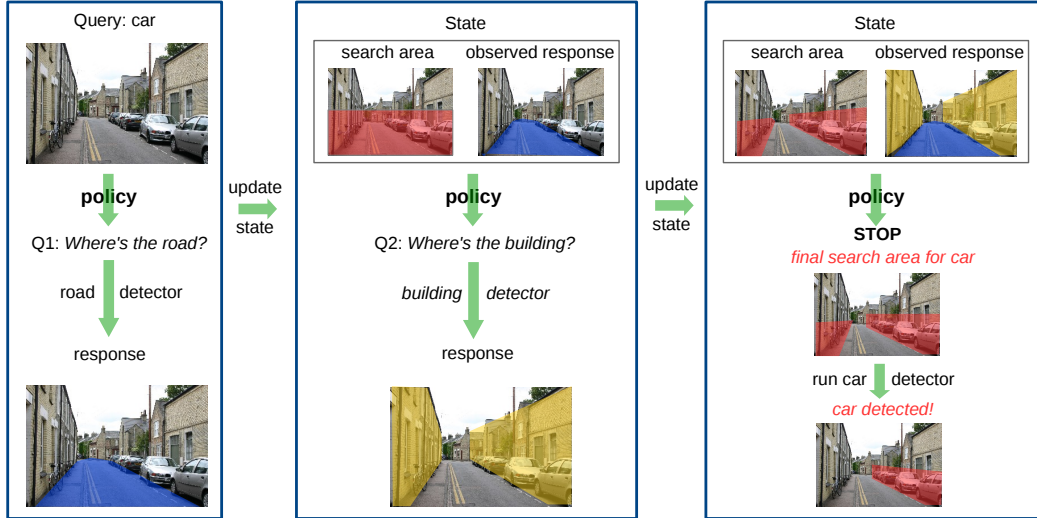


Figure 1: Illustration of our sequential search for objects in 20 context driven questions.

2 Related Work

Sequential Testing. The “20 question” approach to pattern recognition dates back to Blanchard and Geman [5], motivated by the scene interpretation problem with a large number of possible explanations. Their work provides a theoretical foundation for the design of sequential algorithms. “20 questions” approaches recently have been used to generating questions for users in applications such as image binary segmentation [?] and “visual Turing test” [?]. But such methods involve

humans in the loop during test time, which is expensive and hard to scale up. There have been recent attempts to model the computational processes of visual attention [?, ?] for object recognition. Such methods focus on low level salience and are tested in simple scenario such as MNIST dataset.

There are several models [11] of objects classification that operate by running classifiers sequentially in an active order. [6] proposed an information gain based approach to iteratively pose questions for users and incorporate human responses and computer vision detector results for fine-grained classification. [16] formulated object classification as a Markov decision process to select classifiers under certain time constraints. However, these approaches only focus on classifying objects. They have not addressed the challenging problem of simultaneous segmentation and localization of objects in a multi-class scene as we do in this paper, and did not exploit inter-object spatial context.

Object Detection. A common approach to object detection is based on applying gradient based features over densely sampled sliding windows [9]. Such methods achieve good results on classes like human and vehicles, but they are very inefficient since they evaluate up to hundreds of thousands of windows in an image, and false positive detections arise. To reduce the number of windows evaluated, category independent object proposals [7, 24, 2] have been proposed which generate a small number of high quality regions or windows that are likely to be objects. These approaches dramatically reduce the number of candidates and reduce false positive detections. Using these object proposals [12, 14] train and apply deep neural network models on large datasets to learn the feature extractor and classifiers, and achieve state-of-the-art performance on the Pascal VOC detection challenge. However, such category independent proposals do not adapt to different query classes and still lead to a significant amount of unnecessary detector computation.

Object Recognition using Context. Context has been shown to improve object recognition and detection. In [13, 23, 17], CRF models are used to combine unary potentials based on visual features extracted from superpixels with neighborhood constraints and low level context. Inter-object context in the scene has also been shown to improve recognition [10, 8]. [?] shows that using contextual information can improve object detection using CRF models. However these approaches need to evaluate the high order co-occurrence statistics with *all* other object classes appearing in the scene altogether, some of which may not be informative. Our framework, in contrast, only needs to evaluate the most related context in an active sequence before classifications of all objects are made, and goes beyond simple co-occurrence statistics. [1] applied a sequential decision making framework to window selection by voting for the next window. However, the voting process needs to look up nearest neighbors in hundreds of thousands of exemplar window pairs in the training set because their context is purely based on appearance similarity at the instance level, which is highly inefficient. By contrast, our model is based on context between semantic classes so we do not compute nearest neighbors over hundreds of thousands of windows in a high dimensional descriptor space to retrieve the voters, which greatly reduces computational complexity.

3 Problem Formulation

Given an image X and a query object of class c_q ($q \in 1, \dots, C$), we detect the query object by sequentially choosing one context class to detect, and narrow down the search area for the query object based on observed detector responses. The sequential decision-making problem can be formulated as a Markov Decision Process (MDP).

Definition 1. The **Object Detection MDP** is defined by the tuple $(\mathcal{S}, \mathcal{A}, T(\cdot), R(\cdot), \gamma)$:

- The **state** $s_t = (X^t, O^t)$, where X^t is the search area for the query at time t (initially X^0 is the entire image X), $O^t = \{o_1, o_2, \dots, o_t\}$ is a sequence of observed responses from applied contextual classifiers;
- The **action** set $\mathcal{A} = \{a_1, \dots, a_C, \text{Stop}, \text{Reject}\}$, where a_i corresponds to running the detector of class c_i , *Reject* means to reject the query class as occurring in the image and terminate the process, and *Stop* outputs the search area and run the detector of the query class;
- The **state transition** function $T(s'|s, a)$ defines a next-state distribution after action a is taken in state s , which is deterministic in our case;

- The **reward** function $R(s, a) \rightarrow \mathbb{R}$ evaluates how good it is to take action a in state s ;
- The **discount** factor γ is a constant controlling the tradeoff between greedily maximizing the immediate reward and the long term expected reward.

We define the *reward* R as the immediate gain in an intersection/union model of the search space:

$$R(s_t, a_t) = \frac{X^{t+1} \cap X_q}{X^{t+1} \cup X_q} - \frac{X^t \cap X_q}{X^t \cup X_q} \quad (1)$$

where X^{t+1} is the updated search area after executing action a_t in state s_t , determined by the context models described in Section 4.2. X_q is the groundtruth mask of the query object instances in the image.

The query agent follows a *policy* $\pi : S \rightarrow A$ that determines which action to take in a given state. Given an optimal policy π^* which yields a state-action sequence that maximizes the discounted cumulative reward, the optimal Q -value is recursively defined as $Q^*(s_t, a_t) = R(s_t, a_t) + \gamma \max_{a_{t+1}} Q^*(s_{t+1}, a_{t+1})$, where a_t is chosen by π^* and γ is the discount factor. Our goal is to learn the optimal policy for the object detection MDP.

4 Approach

We show our framework in Figure 2. Given a query, we first generate region hypotheses using object proposal algorithms, then the policy sequentially selects an action that rejects the occurrence of the query, poses a question about a context class, or stop and run the query detector. After an action is taken, the search locations are updated based on the responses and new posterior probabilities of each category are computed to update the state. In this section, we first present the imitation learning algorithm for learning a policy that plays the twenty questions game; we then describe how to refine the search area of the query given responses of contextual classifiers evaluated by previous questions.

4.1 Learning the Policy by Imitation

Typically, MDPs are solved by reinforcement learning (e.g., Q-learning [?], REINFORCE [?]). However, given our exponential search space, such trial-and-error approaches can take long to converge. We therefore take the imitation learning approach, where we assume direct supervising signals from the oracle are available and learn to mimic the oracle’s behavior.

Assuming we know the optimal Q -values, the optimal policy is straightforward

$$\pi^*(s) = \arg \max_{a \in A} Q^*(s, a). \quad (2)$$

To learn Q^* , we assume these values are given by an *oracle* at training time; thus the problem can be reduced to learning a linear approximation:

$$Q^*(s, a) = \theta_\pi^T \phi(s, a), \quad (3)$$

where $\phi(s, a) = \phi((X^t, O^t), a)$ is the state feature representing the search area X^t and observations O^t after observing detector responses of a_1, \dots, a_t . This can be solved by standard supervised learning approaches.

We compute the oracle’s action sequence by breadth-first search with pruning. The action sequence that maximizes the discounted cumulative reward in the terminal state is selected as the oracle’s action sequence. We then collect examples $\{(s_t, a_t, r_t)\}$ from the oracle’s trajectory for policy training. However, collecting examples from the oracle’s trajectory only may result in mismatch in distributions of training and test data, since the learned policy may go to state the oracle never visited. To solve the mismatch problem, we encourage exploration by search multiple times with random starting state. Due to the large number of negative states for rejection, we sample the negative states by early pruning at negative immediate reward which imitates the early rejection action. After example collection, we train the policy (predict the optimal Q -values) by ridge regression.

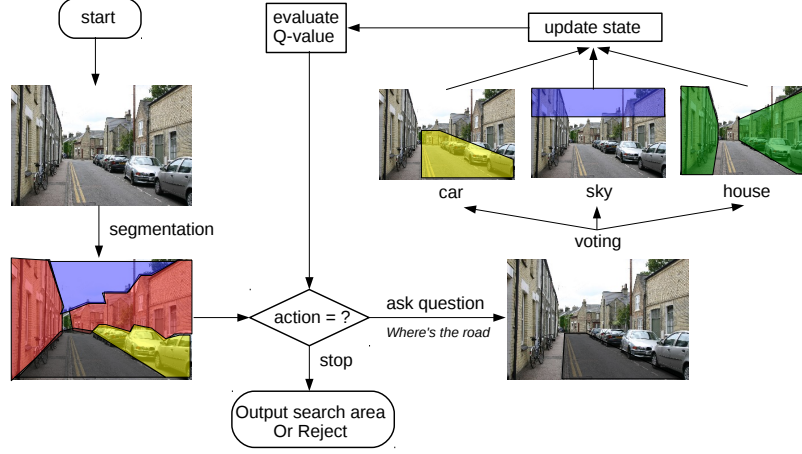


Figure 2: Flowchart of our context driven object searching.

4.2 Context Modeling

Since our task is not only to detect the object but also refine the search space of the query in the image as accurately as possible, conventional modeling of context as simple co-occurrence statistics is inadequate. Instead we present a data-driven location aware approach to represent the spatial correlation between the objects and the scene.

Given an action a_t to detect context class c_t at time t , $X_c \subset X$ is the exploration area for context, here we formulate the context $p(c|c_t, X)$ as a posterior of the probabilistic vote map $p(c_t|c, X_c)$ defined on each pixel $(x_i, x_j) \in X$ over the image, and the responses of class c_t after action a_t :

$$p(c_t|c, X) = \sum_{X_c \subset X} p(c_t|c, X_c) p(c_t|X_c) \quad (4)$$

Given a refined search space $X_c \in X$ of a context class c_t at time t , we formalize $p(c_t|c, X)$ as a weighted vote from the cooccurring region pairs of class c_t and c in training scenes. Let $(s_{c_t}^i, s_c^i)$ be the i -th pair of co-occurring regions of class c_t and c , and $b_{c_t}^i$ and b_c^i be their corresponding bounding boxes. We can now define the probabilistic vote map $p(c_t|c, X)$ as:

$$p(c_t|c, X_c) = \frac{1}{Z_c} \sum_i W(s_{c_t}^i, s; \theta^W) \cdot T(b_{c_t}^i, b_c^i) \quad (5)$$

where $s \in X^t$ is a region within the search space of the context class c_t . Z_c is the normalization function. $W(\cdot)$ is a kernel measuring similarity of region s with a training region s_i . $T(b_{c_t}^i, b_c^i)$ models the transformation from $b_{c_t}^i$ to b_c^i , including translation and scaling. Figure 4 shows a few examples of the vote maps. We can see that with the exemplar based and semantically aware voting, the resulting vote maps give more accurate search areas for the query objects.

The final context probabilistic vote map is given by

$$p(c_t|c, X) = \sum_{s \in X_c} p(c_t|X_c) \sum_i W(s_{c_t}^i, s; \theta^W) \cdot T(b_{c_t}^i, b_c^i) \quad (6)$$

where $p(c_t|X_c)$ is the probabilities of s as class c_t after taking the action a_t to run classification at time t .

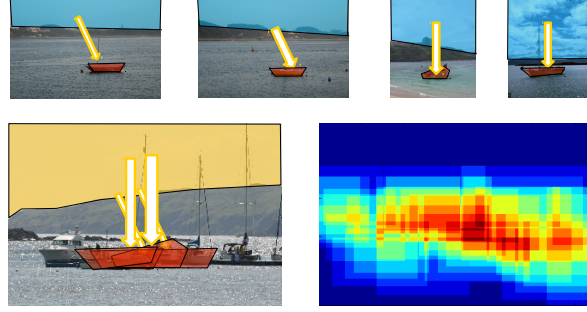


Figure 3: Examples of our weighted vote map for the context from sky to boat. The first rows are the training sample pairs of sky and boat and the second row is the test image and the resulted weighted voting map. The widths of the arrows denote the weighted similarity $W(s_{c_t}^i, s; \theta^W)$ between the test segment of sky (highlighted in yellow) and a training instance of sky segment (in light blue)

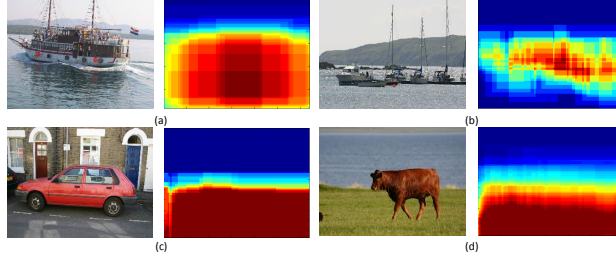


Figure 4: Examples of our context vote maps. Each pair of images corresponds to the original image and the vote-based probability map of object location from observed context. From (a) - (d) are the vote maps from water to boat, sky to boat, road to car and grass to cow, respectively. Best viewed in color.

4.3 Update Responses and Search Area

After taking action a_t and receiving response $o_t = p(c_t|c, X)$ from context class c_t , we integrate the response into observations from previous sequence of actions. Assuming the detectors and context classifiers are trained independently per category, the aggregated responses can be modeled as:

$$p(O^t|c, X) = \prod_t p(c_t|c, X) \quad (7)$$

We then update the search area for the query class c_q in a probabilistic framework:

$$p(c_q|X, O^t) = \frac{p(O^t|c_q, X)p(c_q|X)}{Z} \quad (8)$$

where $Z = \sum_c^{C^t} p(O^t|c, X)p(c|X)$ is the partition function, $p(c|X)$ is obtained by taking actions and running context classifiers over the context segments, and $C^t = \{c_1, c_2, \dots, c_t, c_q\}$ are the set of observed contextual classes.

5 Implementation Details

5.1 Object Proposals

We use MCG object proposals in [2] as object candidates. Since the object proposals mainly covers the objects, we also generate a small number (20~30 per image) of segments using the stable

segmentation algorithm from [8] to cover the whole scene including contextual classes. To reduce the computational overhead, our context voting step uses only the stable segments. The stable segmentation gives a coarse level of object/context division and reduces the computational complexity of context voting compared to the large number of finer object proposals, while still maintaining a semantically informative contextual inference.

5.2 Datasets

We conduct our training and experiments on the Pascal VOC dataset. [?] which is a *de facto* benchmark for object detection. Since the original dataset does not provide annotation of segmentation and contextual classes, we train our policy using the Pascal Context dataset [?] which fully annotates every pixel of the Pascal VOC 2010 train and validation sets, with additional contextual classes such as sky, grass, ground, building etc., which is adequate for our purposes. We use the 33 context classes from [?] and train our policy on the Pascal Context training set, and test our algorithm and baselines on the validation set. We also test our policy on the MSRC dataset [23] to show our algorithm can generalize to different data.

5.3 Feature Representation

To classify object proposals, we extract region features and classify them using the deep neural network model in [14] fine-tuned on Pascal VOC 2012. For the policy action classifiers, we also use the same model to extract features for states represented by the masks of search area X^t and observed area O^t in state s_t , then concatenate the features as inputs to the policy. For context classifiers we use a subset of the appearance features for superpixels from [?] and learn one-vs-all SVM models for classification.

6 Experiments

We conduct our experiments on the Pascal VOC 2010 dataset, in consistence with the data to train our context model. We evaluate the object detection performance by mean average precision (mAP). Since we also evaluate the performance of simultaneous detection and segmentation, we use the validation set as our test set which provides object segmentation annotations.

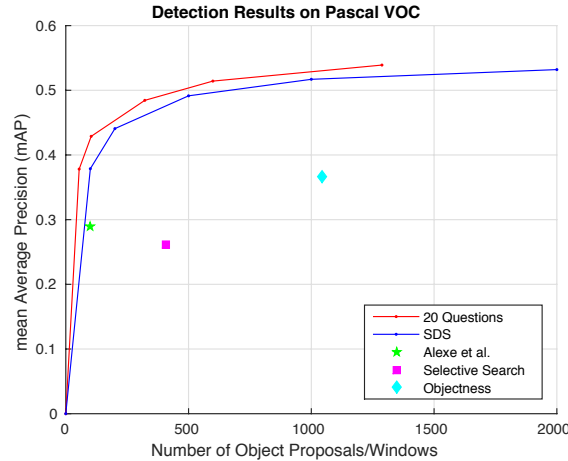


Figure 5: Performance on Pascal VOC dataset. Best viewed in color.

Reduction of Number of Object Proposals Figure 5 shows the average precision performance on the Pascal VOC 2010 dataset with respect to the number of object proposals/windows used. We compared our 20 question approach to the baseline of running detectors in [14] exhaustively over all object proposals, window selection driven by context in [1], detection using object proposals in selective search [24] and objectness [?]. We can see that that our 20 questions detection algorithm

Table 1: Avg. Precision for detection of our algorithm against previous contextual approaches on PASCAL VOC10 dataset.

	Aeroplane	Bicycle	Bird	Boat	Bottle	Bus	Car	Cat	Chair	Cow	Dining Table	Dog	Horse	Motor Bike	Person	Potted Plant	Sheep	Sofa	Train	TV/Monitor	Average
DPM	44.3	51.3	7.1	8.0	21.8	56.0	41.2	18.4	13.8	11.7	10.4	13.5	38.3	42.7	44.6	3.7	27.0	24.3	38.0	32.2	27.4
DPM + 33 Context	46.4	50.8	7.5	8.2	21.2	55.3	41.6	20.0	14.7	11.8	11.6	13.9	37.9	40.2	45.1	4.2	24.1	27.6	40.8	33.9	27.8
[?]+20 Context	46.9	50.1	9.2	9.5	30.1	57.2	44.1	30.7	12.7	15.1	12.9	14.2	35.6	44.8	44.0	4.9	30.6	20.1	42.2	34.8	29.5
[?]+33 Context	49.8	48.8	12.0	10.8	29.1	55.2	45.6	32.0	14.2	12.6	13.7	16.6	39.8	44.2	45.1	8.2	35.3	26.0	42.3	34.3	30.8
SDS [14]	68.2	52.1	51.6	30.7	34.2	66.7	52.4	70.9	21.0	50.6	35.7	64.8	58.2	65.0	55.5	25.0	56.4	38.3	58.3	58.4	50.7
SDS+20Q	71.2	58.8	50.9	33.3	30.2	68.7	56.5	68.2	25.8	55.1	35.8	62.3	56.1	70.4	61.9	28.0	61.2	39.4	63.8	60.6	52.9

can effectively reduce a large amount of object proposals (30% \sim 40%) with little loss compared to exhaustive detection on all object proposals. Thus our approach can achieve comparable results with much less object proposals and corresponding detector evaluation. Our approach has also outperformed random search approach which randomly samples the same number of object proposals for detection. With the increase of the number of object proposals, our algorithm can even gain improvement due to the reduction of false positives.

Comparison with context based methods We compare our model with the SDS exhaustive search baseline and a recent proposed contextual model in [?], which considers global and local context in a Markov Random Field framework based on deformable part-based model. This model has high computational cost since it needs to evaluate hundreds of thousands of windows as well as the context deformation term between all context boxes in the graph. Table 1 shows the classwise mAP of our 20 questions-based object detection with other context based methods and their corresponding baselines. Our 20 question detection approach has outperformed the SDS exhaustive search baseline as well as DPM based context approaches. The initial object proposal number is 2000 for both the SDS and the 20 question method; after the 20 questions refinement of the search space the average number of detector evaluations (including the overhead of context classifiers evaluated) has decreased 36%. We can see that although there are performance drops in some classes due to possible rejection of true object candidates, for some classes that are closely related in some types of scenes such as car, boat,

6.1 Improvement on the search space accuracy

References

- [1] B. Alexe, N. Heess, Y. W. Teh, and V. Ferrari. Searching for objects driven by context. *NIPS*, 2012.
- [2] P. Arbeláez, J. Pont-Tuset, J. T. Barron, F. Marques, and J. Malik. Multiscale combinatorial grouping. *CVPR*, 2014.
- [3] A. G. Barto. *Reinforcement learning: An introduction*. MIT press, 1998.
- [4] I. Biederman, R. J. Mezzanotte, and J. C. Rabinowitz. Scene perception: Detecting and judging objects undergoing relational violations. *Cognitive psychology*, 14(2):143–177, 1982.
- [5] G. Blanchard and D. Geman. Hierarchical testing designs for pattern recognition. *Annals of Statistics*, pages 1155–1202, 2005.
- [6] S. Branson, C. Wah, F. Schroff, B. Babenko, P. Welinder, P. Perona, and S. Belongie. Visual recognition with humans in the loop. In *Computer Vision—ECCV 2010*, pages 438–451. Springer, 2010.
- [7] J. Carreira and C. Sminchisescu. Cpmc: Automatic object segmentation using constrained parametric min-cuts. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(7):1312–1328, 2012.
- [8] X. Chen, A. Jain, A. Gupta, and L. S. Davis. Piecing together the segmentation jigsaw using context. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 2001–2008. IEEE, 2011.
- [9] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(9):1627–1645, 2010.
- [10] C. Galleguillos and S. Belongie. Context based object categorization: A critical survey. *Computer Vision and Image Understanding*, 114(6):712–722, 2010.
- [11] T. Gao and D. Koller. Active classification based on value of classifier. *NIPS*, 2011.
- [12] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Computer Vision and Pattern Recognition*, 2014.

- [13] S. Gould, R. Fulton, and D. Koller. Decomposing a scene into geometric and semantically consistent regions. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 1–8. IEEE, 2009.
- [14] B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik. Simultaneous detection and segmentation. In *European Conference on Computer Vision (ECCV)*, 2014.
- [15] H. S. Hock, G. P. Gordon, and R. Whitehurst. Contextual relations: the influence of familiarity, physical plausibility, and belongingness. *Perception & Psychophysics*, 16(1):4–8, 1974.
- [16] S. Karayev, T. Baumgartner, M. Fritz, and T. Darrell. Timely object recognition. *NIPS*, 2012.
- [17] L. Ladicky, C. Russell, P. Kohli, and P. H. Torr. Graph cut based inference with co-occurrence statistics. In *Computer Vision–ECCV 2010*, pages 239–253. Springer, 2010.
- [18] C. H. Lampert, M. B. Blaschko, and T. Hofmann. Efficient subwindow search: A branch and bound framework for object localization. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 31(12):2129–2142, 2009.
- [19] E. Moores, L. Laiti, and L. Chelazzi. Associative knowledge controls deployment of visual selective attention. *Nature neuroscience*, 6(2):182–189, 2003.
- [20] R. Mottaghi, X. Chen, X. Liu, N.-G. Cho, S.-W. Lee, S. Fidler, R. Urtasun, and A. Yuille. The role of context for object detection and semantic segmentation in the wild.
- [21] J. Najemnik and W. S. Geisler. Optimal eye movement strategies in visual search. *Nature*, 434(7031):387–391, 2005.
- [22] D. Parikh, C. L. Zitnick, and T. Chen. Exploring tiny images: the roles of appearance and contextual information for machine and human object recognition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(10):1978–1991, 2012.
- [23] J. Shotton, J. Winn, C. Rother, and A. Criminisi. Textonboost: Joint appearance, shape and context modeling for multi-class object recognition and segmentation. In *Computer Vision–ECCV 2006*, pages 1–15. Springer, 2006.
- [24] K. E. Van de Sande, J. R. Uijlings, T. Gevers, and A. W. Smeulders. Segmentation as selective search for object recognition. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 1879–1886. IEEE, 2011.