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. Experiments in detection and segmentation show that our algorithm reduces over 30% of the object proposal evaluations while improving performance in average precision as well as producing more accurate search areas 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 [10] or category-independent object proposals [7, 28, 3]. 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, 17]. Context information is an effective cue for humans to detect low-resolution or small objects in cluttered scenes [23]. Many contextual models have been proposed to capture relationships between objects at the semantic level to reduce ambiguities from unreliable independent detection results [15, 19]. 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: human visual perception is an active process that sequentially samples the optic array in an intelligent, task-specific way [22]. 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 [20]. For instance, in Figure 1, when we search for cars, knowing the top of the scene is sky does not help distinguish whether the image contains either a car or a boat since both are equally likely to 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. Therefore, in order to find cars, humans tend to first look for roads instead of sky; additionally, if we can not find cars on the road, we may want to look "beside" the buildings because cars are likely to park next to them. This motivates us to raise

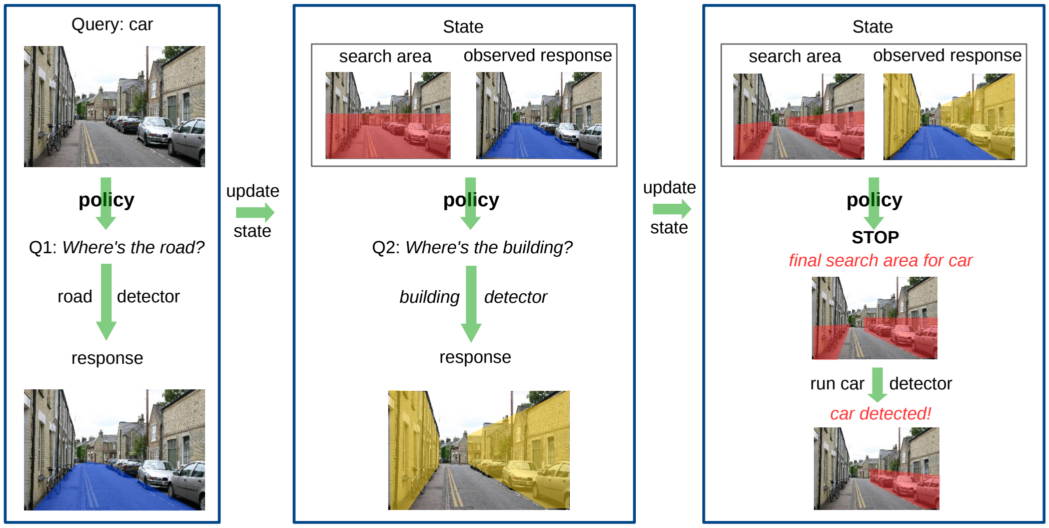


Figure 1: Illustration of our sequential search for query objects in 20 context-driven questions.

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 this 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 selected context class. Based on the responses, we further refine the search area for the query class using spatially-aware contextual models. This process of contextual querying and search area refinement is repeated until the policy determines that sufficient 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 little chance of the query object in the scene. The early rejection decision can be taken even before running the object detector; therefore we can eliminate a large amount of unnecessary computation.

Object detection experiments on the PASCAL VOC dataset show that our algorithm produces a search area that has better overlap with the target object by leveraging its context, thus significantly reducing the number of object proposals to consider and detectors to evaluate by over 30%. Even with less computation, our method achieves mean average precision (mAP) comparable to or even higher than the exhaustive search method. To the best of our knowledge, this is one of the first approaches that solves the challenging task of simultaneous object detection and segmentation in complex scenes in an MDP framework by applying imitation learning to learn a policy fully driven by context.

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 generate questions for users in applications such as image binary segmentation [25] and "visual Turing test" [13]. 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 [24] for object recognition. Such methods focus on low level salience and are tested in simple scenarios such as MNIST dataset.

There are several models [12] 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 incorporates human responses and computer vision detector results for fine-grained classification. [18] formulated object classification as a Markov decision process to select classifiers under given 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 [10].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 positve detections arise. To reduce the number of windows evaluated, category independent object proposals [7, 28, 3] 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 [14, 16] 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 [26, 19], 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 [11, 8]. [21] 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-occurence statistics. [2] 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, which greatly reduces computational complexity.

3 Problem Formulation

Given an image X and a query class c_q ($q \in 1, .., C$), we detect instances of the query class by sequentially choosing one context class to detect, and reduce the search area for the query class based on the observed context class 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 $(\mathcal{S}, \mathcal{A}, T(.), R(.), \gamma)$:

- 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), $O^t = \{o_1, o_2, \ldots, o_t\}$ is a sequence of observed responses from applied contextual classifiers;
- The **action** set $\mathcal{A} = \{a_1, \ldots, a_C, Stop, Reject\}$, where a_i mean run the detector of class c_i, *Reject* means to decide that query class does not occur in the image and terminate the process, and *Stop* runs the detector of the query class on the current search area;
- 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 $R(s, a) \to \mathbb{R}$ 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:

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

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 either a) selects an action that rejects the occurrence of the query, b) poses a question about a context class, or c) stops and runs 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 [29], REINFORCE [30]). However, given our exponential search space, such trial-and-error approaches can take too 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

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

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:

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

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 $a_1, ..., a_t$. 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 states the oracle never visited. To solve the mismatch problem, we encourage exploration by search multiple times with random starting states. 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.

4.2 Context Modeling

Since our task is not only to detect instances of the query object but also to refine the search space of the query in the image as accurately as possible, conventional modeling of context as simple

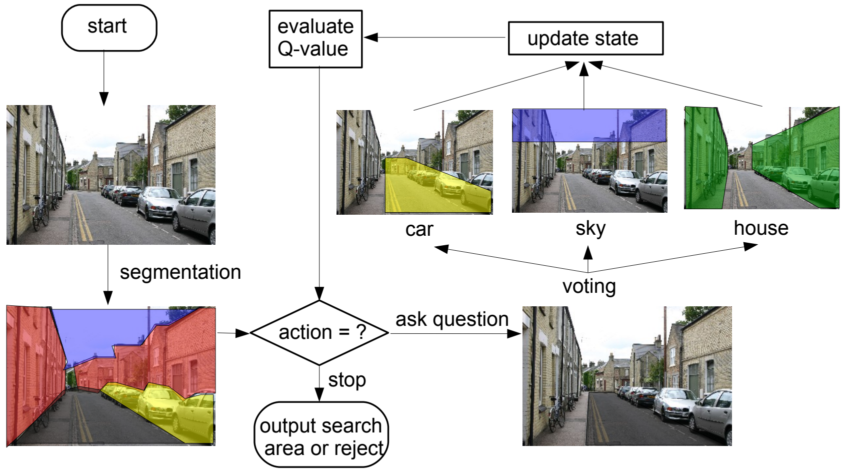


Figure 2: **Flowchart of our context driven object searching algorithm**.We first generate region hypotheses using object proposal algorithms, then the policy evaluates the current state and iteratively selects the action maximizing the Q-value function. Afterwards, the possible search locations are updated and the posterior probabilities of each category are evaluated for the next state.

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.

We capture the spatial relationships in a non-parametric manner. Figure 3a illustrates our model. During training, the bounding box of a region s^i is represented by $b^i = (x^i, y^i, \sigma^i)$ with x, y as its location and σ as the scale. For each pair of co-occurring regions belong to class c_k and c_l respectively, we index this pair as j and store corresponding displacement vector $T_j = T(b^j_k, b^j_l)$ which includes translation $(\Delta x, \Delta y)$ and change of ratio in x, y directions between the two boxes.

During test time, we define $X_c \subset X$ as the *exploration area* for context which excludes the observed regions of other contextual classes in the image. Let $s^i \subset X_c$ be the context region i in the test image. Given an action a_k to detect context class c_k at time t, to model the context between class c_k and another class c_l, here we model the context $p(c_k|c_l, X)$ as the posterior of the probabilistic vote map $p(c_l|c_k, X_c)$ defined for each pixel in the image, and the responses of class c_i after action a_i:

$$p(c_k|c_l, X) = p(c_k|c_l, X_c) = \sum_{s^i \subset X_c} p(c_l|c_k, s^i) p(c_k|s^i) \tag{4}$$

where $p(c_k|s^i)$ is the probabilities of s^i as class c_k after taking the action a_k to run classification at time t.

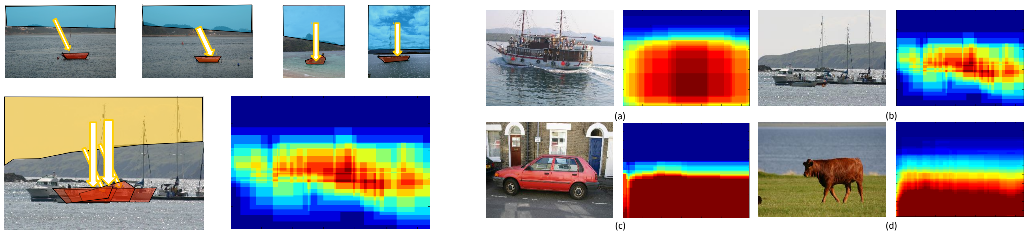
We can now define the probabilistic vote map $p(c_l|c_k, s^i)$ as follows. Let (s^j_k, s^j_l) be the j-th training pair of co-occurring regions of class c_k and c_l, and b^j_k and b^j_l be their corresponding bounding boxes. Let $s^i_k \subset X_c$ be the context region i detected as class c_k in the test image. We retrieve those training pairs (s^j_k, s^j_l) between class c_k and c_l and compute the RBF kernel $W(.)$ measuring the similarity of the features of train/test segments of class c_k as $W(s^i_k, s^j_k; \theta^W)$, where θ^W is the kernel parameter. We then formalize $p(c_l|c_k, X_c)$ as a weighted vote from the co-occurring region pairs of classes c_k and c_l in training scenes.

$$p(c_l|c_k, s^i) = \frac{1}{Z_c} \sum_i \sum_j W(s^i_k, s^j_k; \theta^W).T(b^j_k, b^j_l) \tag{5}$$

where Z_c is the normalization function. Figure 3b 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

$$p(c_k|c_l, X) = \sum_{s^i \subset X_c} p(c_k|s^i) \sum_i \sum_j W(s^i_k, s^j_k; \theta^W).T(b^j_k, b^j_l) \tag{6}$$



3a. Weighted vote map from sky to boat.

3b. Examples of context vote maps.

Figure 3: **a. Our context voting model**. The first row shows example training pairs of the sky and the boat. The second row shows the test image and the weighted voting map. The arrows denote applying the weighted displacement vectors $T(b_k^j, b_l^j)$ from the train pairs to the test pairs of sky and boat (highlighted in yellow and blue respectively). **b. Examples of context vote maps**. Each pair shows the original image and the voted probability map of object locations given observed context. From (a) – (d) are the vote maps from water to boat, sky to boat, road to car and grass to cow. 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 the previous sequence of actions. Assuming the detectors and context classifiers are trained independently per category, the aggregated responses can be modeled as:

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

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

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

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 $C^t = \{c_1, c_2, ..., c_t, c_q\}$ are the set of observed contextual classes.

5 Implementation Details

5.1 Object Proposals

We use MCG object proposals in [3] 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 experiments on the Pascal VOC dataset [9], a standard benchmark for object detection. Since the original dataset does not provide annotation for segmentation of contextual classes, we train our policy using the Pascal Context dataset [21] 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. We use the 33 context classes from [21] 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 [26] to show our algorithm can generalize to different data.

Table 1: Avg. detection precision of ours and other algorithms 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
[21]+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
[21]+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
RCNN [14]	**69.9**	64.2	48.0	30.2	26.9	63.3	56.0	67.6	26.8	44.7	29.6	61.7	55.7	**69.8**	56.4	26.6	56.7	35.6	54.4	**57.7**	50.1
SDS [16]	67.3	63.6	47.1	33.1	**34.3**	67.2	55.8	**74.6**	24.9	44.8	35.7	**62.7**	62.5	64.8	**59.1**	**26.9**	54.2	40.7	61.3	55.7	51.8
SDS+20Q	66.8	**64.3**	**48.9**	**36.1**	32.2	**67.7**	**56.5**	70.4	**28.1**	**58.3**	**37.2**	60.5	**64.7**	65.9	52.1	26.7	**57.8**	**46.6**	**62.9**	53.3	**52.9**

5.3 Feature Representation

To classify object proposals, we extract region features and classify them using the deep neural network model in [16] fine-tuned on Pascal VOC 2012. For the policy action classifiers, we 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 [27] and learn one-vs-all SVM models for classification.

6 Experiments

6.1 Performance of Object Detection

We compare our model with current state-of-the-art exhaustive search baselines SDS [16] and RCNN [14], as well as a recently proposed contextual model in [21], which considers global and local context in a Markov Random Field framework based on a 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 class-wise mAP of our 20 questions approach with other context based methods and their corresponding baselines. Our 20 question detection approach outperforms the SDS and RCNN exhaustive search baselines as well as DPM based context approaches. Both the SDS and the 20 question method start with 2000 object proposals per image; 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 the performance for some classes have dropped due to possible rejection of true object candidates, classes that are more correlated to other objects in the scenes have significant gain over exhaustive search, such as boat, car, chair, cow, sofa etc., thus improving overall average precision.

6.2 Reduction of Object Proposals

Figure 4 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 [16] exhaustively over all object proposals, random search which randomly samples the same number of object proposals for detection, window selection driven by context in [2], detection using object proposals in selective search [28] and objectness [1]. We can see that that our 20 questions detection algorithm can effectively reduce a large amount of object proposals ($30\% \sim 40\%$) with comparable performance to exhaustive detection on all object proposals. With increasing numbers of object proposals, our algorithm can achieve even better results than exhaustive methods due to the reduction of false positives. We also see the random search approach performs poorly, showing the effectiveness of our context driven search approach.

6.3 Improvement on the search space accuracy

To measure the quality of our predicted search areas, we evaluate the mean intersect vs. union (IU) of the search area produced by our 20 questions approach with the groundtruth objects. We also compare with the search area of the original detector, produced by the union of the object proposals with high scores. Table 2 shows that our approach significantly improves the accuracy of overlap

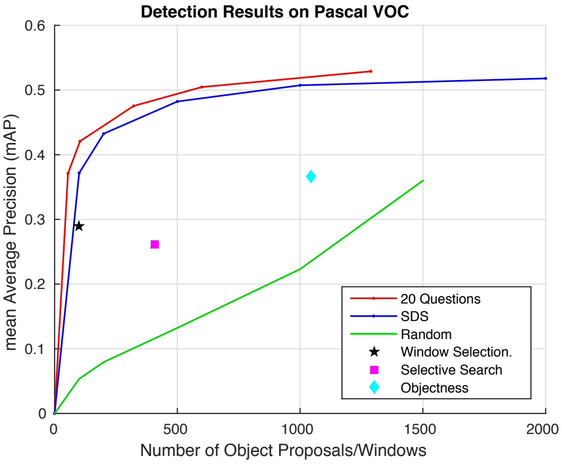


Figure 4: Performance on Pascal VOC dataset. Best viewed in color.

	Original	20Q	Oracle
Mean IU	64.12	73.9	**78.2**

Table 2: Mean intersect VS union (IU) of the predicted search area and the groundtruth objects.

between the predicted search area and the target query object. We also find that the mean IU of the 20 questions search space is close to that predicted by the oracle trajectory, which shows that our imitation learning has learned a good policy that closely mimics the oracle's behavior.

Table 3: AP^r performance 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
SDS [16]	**68.2**	52.1	51.6	30.7	**34.2**	66.7	52.4	**70.9**	21.0	39.6	30.7	**58.8**	**55.2**	**54.0**	**55.5**	25.0	56.4	33.3	61.2	**58.4**	48.8
SDS+20Q	66.7	**55.0**	**52.2**	**33.3**	32.1	**67.2**	**53.7**	66.9	**24.2**	**53.8**	**32.0**	56.9	54.1	53.3	49.6	**25.1**	**58.1**	**35.1**	**61.3**	55.5	**49.3**

	cow	sheep	bird	chair	cat	dog	boat	body	car	bike	plane	mean
SDS [16]	87.4	87.6	**49.6**	**52.2**	75.0	72.3	49.2	62.5	73.0	80.7	93.8	71.9
SDS+20Q	**88.4**	**93.8**	45.8	48.3	**82.6**	**76.7**	**51.7**	**65.5**	**79.0**	**85.2**	**95.7**	**73.2**

Table 4: AP^r performance on MSRC dataset.

6.4 Simultaneous detection and segmentation

Given that we employ segment based object proposals generated by [3], our detection system can also perform segmentation. We compare our algorithm with [16] in the simultaneous detection and segmentation task using the AP^r metric proposed in [16]. Table 3 and Table 4 show the performance on Pascal VOC10 and the MSRC datasets respectively. We outperforms the SDS approach on both datasets, showing our 20 questions algorithm can generalize well from the detection to the segmentation task, as well as generalize to other datasets such as MSRC.

7 Conclusion

We proposed a sequential and dynamic process for object detection and segmentation, in which a policy iteratively selects a context related question adapting to different query and responses from previous context-related questions, then more accurately refines the search area or rejects the object early without running many detectors. Thus it can significantly reduce computational cost while preserving good accuracy for the query class. We formulate the object detection problem as a Markov Decision Process to learn a policy by imitation learning. We present a unified probabilistic framework to model spatial context between objects. We applied this active detection scheme to the problem of object detection and segmentation, and achieved comparable or even higher average precision compared to previous approaches, but with significant computational savings.

References

[1] B. Alexe, T. Deselaers, and V. Ferrari. What is an object? In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 73–80. IEEE, 2010.

[2] B. Alexe, N. Heess, Y. W. Teh, and V. Ferrari. Searching for objects driven by context. *NIPS*, 2012.

[3] P. Arbeláez, J. Pont-Tuset, J. T. Barron, F. Marques, and J. Malik. Multiscale combinatorial grouping. CVPR, 2014.

[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] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88(2):303–338, June 2010.

[10] 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.

[11] C. Galleguillos and S. Belongie. Context based object categorization: A critical survey. *Computer Vision and Image Understanding*, 114(6):712–722, 2010.

[12] T. Gao and D. Koller. Active classification based on value of classifier. *NIPS*, 2011.

[13] D. Geman, S. Geman, N. Hallonquist, and L. Younes. Visual turing test for computer vision systems. *Proceedings of the National Academy of Sciences*, 112(12):3618–3623, 2015.

[14] 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.

[15] 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.

[16] B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik. Simultaneous detection and segmentation. In *European Conference on Computer Vision (ECCV)*, 2014.

[17] 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.

[18] S. Karayev, T. Baumgartner, M. Fritz, and T. Darrell. Timely object recognition. *NIPS*, 2012.

[19] 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.

[20] E. Moores, L. Laiti, and L. Chelazzi. Associative knowledge controls deployment of visual selective attention. *Nature neuroscience*, 6(2):182–189, 2003.

[21] 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. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 891–898. IEEE, 2014.

[22] J. Najemnik and W. S. Geisler. Optimal eye movement strategies in visual search. *Nature*, 434(7031):387–391, 2005.

[23] 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.

[24] M. Ranzato. On learning where to look. *arXiv preprint arXiv:1405.5488*, 2014.

[25] C. Rupprecht, L. Peter, and N. Navab. Image segmentation in twenty questions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3314–3322, 2015.

[26] 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.

[27] J. Tighe and S. Lazebnik. Superparsing: scalable nonparametric image parsing with superpixels. In *Computer Vision–ECCV 2010*, pages 352–365. Springer, 2010.

[28] 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.

[29] C. J. Watkins and P. Dayan. Q-learning. *Machine learning*, 8(3-4):279–292, 1992.

[30] R. J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3-4):229–256, 1992.