Object Detection in 20 Questions

Xi (Stephen) Chen He He Larry S. Davis
Department of Computer Science
University of Maryland
College Park, MD 20742
(chenxi,lsd)@umiacs.umd.edu, hhe@cs.umd.edu

Abstract

We propose a novel general strategy for object detection. Instead of passively evaluating all object detectors at all possible locations in an image, we develop a divide-and-conquer approach by actively and sequentially evaluating contextual cues related to the query based on the scene and previous evaluations—like playing a "20 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 reinforcement learning. To demonstrate the efficacy of our generic algorithm, we apply the 20 questions approach in the recent framework of simultaneous object detection and segmentation. Experimental results on the Pascal VOC dataset show that our algorithm reduces about 45.3% of the object proposals and 36% of average evaluation time while achieving better average precision compared to exhaustive search.

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 [8, 31, 3]. Such methods need to evaluate a large number of object hypotheses indiscriminately, and can easily introduce false positives if exclusively considering local appearance.

Instead of checking all hypotheses exhaustively, humans only look for a set of related objects in a given context [4]. Context information is an effective cue for humans to detect low-resolution or small objects in cluttered scenes [24]. Many contextual models have been proposed to capture relationships between objects at the semantic level to reduce ambiguities from unreliable independent detection results. 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 unwanted 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 [23]. 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 [21]. 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 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 cannot 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 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?*

To this end, we propose a generic strategy for object proposal-based object detection to explore the search space dynamically based on learned contextual relations, which achieves favorable speed-accuracy tradeoffs. We formulate the object detection problem as a Markov Decision Process (MDP), and use reinforcement 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 refines the search area for the query class.

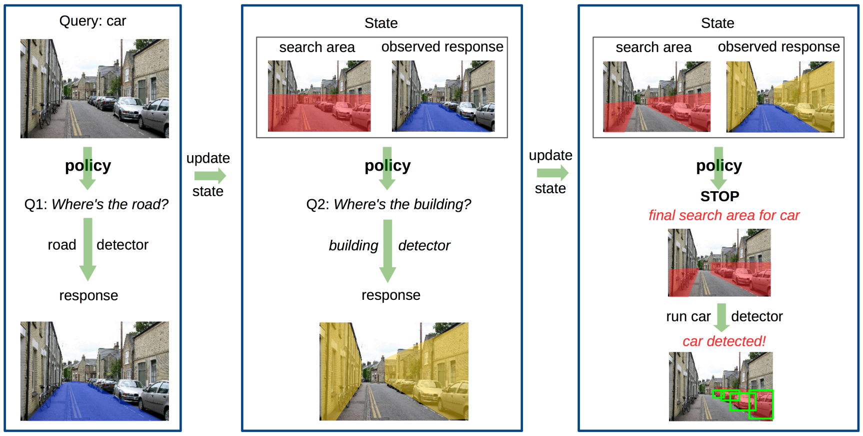

Figure 1: Illustration of our sequential search for query objects in 20 context-driven questions.

We show our framework in Figure 2. Specifically, like playing a 20 Questions game, at each step the policy asks for information about a context class such as road or building based on the query (which is one of the object classes in the dataset, e.g. car) 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 spatial context 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 computation if it determines that there is little chance of the query object being in the scene. The early rejection decision can be taken even before running any object detector; therefore we can eliminate a large amount of unnecessary computation.

To demonstrate the efficacy of our idea, we implement our algorithm based on the Simultaneous Detection and Segmentation (SDS) [14] framework. 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 eliminating 45.3% of object proposals and 36% of total evaluation time compared to an exhaustive detection approach.

2. Related Work

Object Detection. Some common approaches to object detection are based on applying gradient based features over densely sampled sliding windows [10], which 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 [8, 31, 3] have been developed 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 [13, 14, 12] train and apply deep neural network models 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. There are a few recent works to speed up the detection pipeline by using trivial region generation scheme and then regression for accurate bounding box recovery [18, 27], but sliding windows or object proposals based approaches are more commonly used in practice.

Sequential Testing Approaches. The “20 question” approach to pattern recognition dates back to [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. There are several models [11] of objects classification that operate by running classifiers sequentially in an active order [6, 15]. 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. There have been recent attempts to model the computational processes of visual attention [26, 19] for object recognition, which focus on low level salience and are tested in simple scenarios such as MNIST dataset. [7] trains a class-specific attention model for object detection by sequentially transforming the bounding box for the target, but it fails to capture inter-object semantic context and has significant loss of performance compared to exhaustive baseline RCNN.

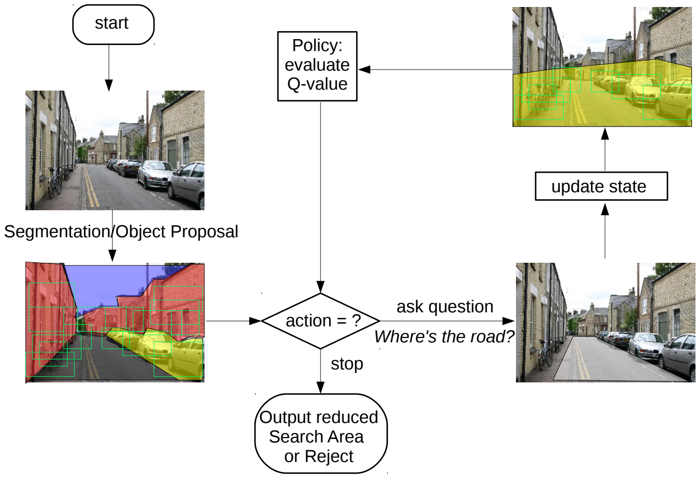

Figure 2: **Flowchart of our context driven object searching algorithm**.We first generate region hypotheses using object proposal algorithms, then the policy evaluates the current state and iteratively selects the action maximizing the Q-value function. Afterwards, the possible search locations are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

Object Recognition using Context. Context has been shown to improve object recognition and detection. In [29, 17], CRF models are used to combine unary potentials based on visual features extracted from superpixels with neighborhood constraints and low level context. [22] shows that using contextual information can improve object detection using conditional random field (CRF) models. However these approaches 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 evaluates the most related context in an active sequence before classification of query class objects is conducted, 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, object classes $\mathcal{C} = \{c_1, c_2, ..., c_n\}$, context classes $\mathcal{X} = \{\chi_1, \chi_2, ..., \chi_m\}$, and a query class $c_q \in \mathcal{C}$ $(q \in 1, .., n)$, we detect instances of the query class by sequentially choosing the next context class to detect, and reduce the search area for the query class based on the responses of the selected context class detectors. 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 $(\mathcal{S}, \mathcal{A}, R(.), \gamma)$:

- The **state** $s_t = (X^t, O^t)$ $(s_t \in \mathcal{S})$, 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 $\mathcal{A} = \{a_1, \ldots, a_m, Stop, Reject\}$, where a_i corresponds to running the detector of context class χ_i, *Reject* corresponds to deciding that the query class does not occur in the image and terminate the process, and *Stop* terminates querying context classes and applies the detector of the query class to the current search area;
- The **reward** function $R : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ evaluates the utility of taking 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 $\sum_{t=0}^{T} \gamma^t R(s_t, a_t)$.

The query agent follows a **policy** $\pi : \mathcal{S} \to \mathcal{A}$ that determines which action to take in a given state. We define our policy by first defining the *immediate reward* R as the immediate gain in an intersection/union model of the search space:

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

where X^t is the updated search area after executing action a_t in state s_t, determined by the context models described in the Approach section. X_q is the groundtruth mask of the query object instances in the image (known only, of course, during training).

4. Approach

We show our framework in Figure 2. Given a query and an image, we first generate object hypotheses as well as a small number of regions corresponding to contextual classes; then the policy sequentially either a) rejects the occurrence of the query, b) poses a question about a context class, or c) stops and runs the query class detector. After an action is taken, the search location for the query class is updated based on the responses. In this section, we first present the reinforcement learning algorithm for learning a policy in a 20-questions approach; we then describe how to refine the search area of the query given responses of contextual classifiers evaluated in response to previous questions.

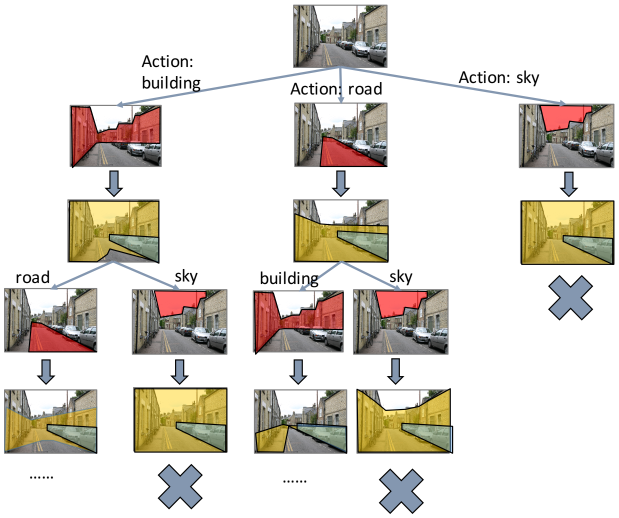

Figure 3: **Collecting training samples**. Our approach collects training samples using depth-first search in the state space. The approach searches by performing each action in a trial-and-error manner, where after taking each action, the search area of the query is predicted and rewards are computed w.r.t. the groundtruth object locations. We prune branches where there are no positive immediate rewards.

4.1. Learning the Policy by Reinforcement Learning

Our goal is to learn a policy for the object detection MDP that guides the search process. We use the value function estimation method based on the sampling sequences from depth-first search. To select an action in a state, the state-action value (Q-value) is defined recursively as

$$Q(s_t, a_t) = R(s_t, a_t) + \gamma \mathbb{E}_{s_{t+1}} \left[\max_{a_{t+1}} Q(s_{t+1}, a_{t+1}) \right]. \quad (2)$$

At test time, the policy simply selects the action with the maximum Q-value.

$$\pi(s) = \arg\max_{a \in A} Q(s, a). \quad (3)$$

Since it is not possible to learn Q-values for all possible s and a, we reduce the problem to learning a linear function approximator for generalization:

$$Q(s, a) = \theta_{\pi_a}^T \phi(s), \quad (4)$$

where $\phi(s) = \phi((X^t, O^t))$ is a feature representation of the state consisting of the search area X^t and observations O^t. Equation 4 approximates the Q-value at state s for executing action a.

To obtain the Q-values for different states and actions, we perform depth-first search with pruning, as illustrated in Figure 3. We then collect examples $\{(s_t, a_t, Q_t)\}$ from the search trajectory, where the Q-values are the sum of discounted R_t along a trajectory. Since there are an exponential number of states, we prune branches with no positive immediate reward. We also train the Q-value predictor for the rejection action with samples in which the query objects do not occur in the image. Due to the large number of such training examples, we sample them by early pruning when the immediate reward is negative, which imitates the action of early rejection. After example collection, we train the policy (predict the optimal Q-values) by regression using the CNN features as in [14]. We train the predictor using ridge regression, but our approach is generic and any other standard regression algorithm can be used, such as Deep Q-Network [20].

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 and quickly 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.

We capture the spatial relationships in a non-parametric manner. Figure 4 illustrates our model. During training, the bounding box of a region s^i indexed by i is represented by $b^i = (x^i, y^i, \sigma_x^i, \sigma_y^i)$ with x, y as its center location and σ_x, σ_y as the scale of aspect ratio w.r.t. the image. For each pair of co-occurring regions belonging to class χ_k and

c_q respectively, we index this pair as j and store the corresponding displacement vector $T_j = T(b_k^j, b_l^j)$ which includes translation $(\Delta x, \Delta y)$ and change of aspect ratio 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 a test image. Given an action a_k to detect context class χ_k at time t, to model the context between class χ_k and query class c_q, we define a probabilistic vote map $p(c_q|\chi_k, s^i)$ as follows.

Let (s_k^j, s_l^j) be the j-th training pair of co-occurring regions of class χ_k and c_q, and b_k^j and b_q^j be their corresponding bounding boxes. Let $s_k^i \subset X_c$ be the context region i detected as class χ_k in the test image. We retrieve those training pairs (s_k^j, s_l^j) between class χ_k and c_q and compute the RBF kernel $W(.)$ measuring the similarity of the features of train/test segments of class χ_k as $W(s_k^i, s_k^j; \theta^W)$, where θ^W is the kernel parameter. We then model $p(c_q|\chi_k, X_c)$ as a weighted vote from the co-occurring region pairs of classes χ_k and c_q in training scenes.

$$p(c_q|\chi_k, s^i) = \frac{1}{Z_c}\sum_i\sum_j W(s_k^i, s_k^j; \theta^W).T(b_k^j, b_q^j) \quad (5)$$

where Z_c is the normalization function. Figure 5 shows a few examples of vote maps. We can see that with the exemplar based and semantic spatial voting, the resulting vote maps give more accurate search areas for the query objects.

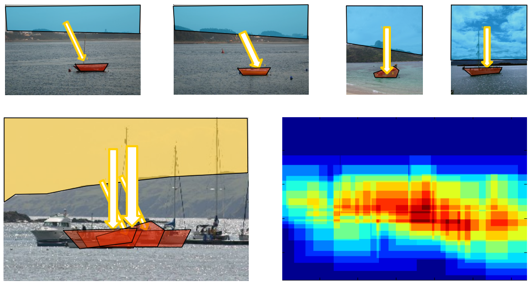

Figure 4: **Our context voting model**. The first row shows example training pairs of sky and boat. The second row shows a test image and the weighted voting map. The arrows denote applying the weighted displacement vectors $T(b_k^j, b_q^j)$ from the training pairs to the test pairs of sky and boat (highlighted in yellow and blue respectively).

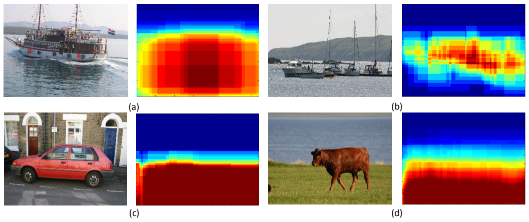

Figure 5: Examples of context vote maps. Each pair of images corresponds to the original image and the predicted probability map of object location given observed context. From (a) - (d) are the vote maps from water to boat, sky to boat, road to car and grass to cow, respectively. Best viewed in color.

5. Implementation Details

5.1. Object Proposals

We use MCG object proposals from [3] as object candidates. Since the object proposals mainly cover the objects, we also generate a small number (20∼30 per image) of segments using the stable segmentation algorithm from [25] to cover regions corresponding to contextual classes. To reduce 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 semantic spatial information.

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 [22] 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 [22] and train our policy on the Pascal Context training set, and test our algorithm and baselines on the validation set using the 20 object classes in the Pascal VOC dataset as the query classes. We also test our policy on the MSRC dataset [29] 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 SDS-C deep neural network model in [14] 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 [30] and learn one-vs-all SVM models for classification.

6. Experiments

6.1. Baselines

We compare with two recent popular exhaustive detection baselines, RCNN [13] and SDS [14]. RCNN adapts the CNN pretrained for image classification [16] to the task of object detection by fine-tuning the network on warped object bounding boxes, then applies the network to extract CNN features on each object proposal for detection. SDS further extends RCNN to the task of segmentation by training and testing on region-based proposals. Both approaches need to extract features and run class-specific detectors exhaustively on all object proposals. We implement our algorithm based on the SDS framework. We also compare with random search which randomly samples the same number of object proposals for detection, window selection driven by context from [2], detection using object proposals in selective search [31] and objectness [1]. For average precision we also compare with a recently proposed contextual model in [22] which considers global and local context in a Markov Random Field framework based on a deformable part-based (DPM) 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.

6.2. Speed-accuracy tradeoff

Figure 6 shows for the Pascal VOC 2010 dataset the mean average precision (mAP) performance vs the (amortized) number of detectors/classifiers evaluated on the object proposals. The amortized number of proposals consists of not only the resulting proposals for the query, but also the average overhead evaluation including context classifiers and the Q-value evaluations on the state masks, so it reflects the total computational cost. Our algorithm has significantly reduced both the number of object proposals for the query and the total computation time. Compared to SDS, the reduction of proposals for the object is 45.3%, and the overall reduction of time is 36%. Empirically it takes SDS about 13.3s to evaluate features for 2000 proposals for a class. With our algorithm, the average number of object proposal drops 45.3% resulting in computation of around 7.1s, plus about 0.8s for evaluating Q-values and 0.6s for context detectors. This is a 36% reduction in amortized run time. 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.

In comparison to [2], which is closely related to our approach, context class lookup in [2] is between 2.55 and 5.7s+0.26s to update the vote map, while our method only takes 0.6s, achieving 7x∼10x speedup. Although we use MCG object proposals that are already highly precise in object location, we still achieve over 45% reduction on average.

In comparison to the recent related work [7], they sequentially transform the bounding boxes for queries using a Deep Q-Network. However, after processing 200 windows their mAP on Pascal VOC 2007 is 46.1% which is a significant loss in mAP compared to RCNN's 54.2%. Our approach, in contrast, achieves similar mAP to the RCNN baseline on the Pascal VOC 2010 dataset after processing 200 windows(43.4% vs. 44.0%), which shows a better speed-accuracy trade-off. Note that these numbers are not directly comparable because the datasets are different (VOC 2010 contains a subset of the VOC2007 images).

6.3. Detection precision

Table 1 shows the classwise mAP of our 20 questions approach with other context based methods and their corresponding baselines. We compare our model with SDS and RCNN as well as [22] denoted as "Pascal 20/30 Context" in the table, and deformable part-based model with context denoted as "DPM(+context)". Both the SDS and the 20 question methods start with 2000 object proposals per image. Our 20 question detection approach outperforms exhaustive search baselines SDS and RCNN as well as DPM based context approaches while reducing 45.3% of proposals.We can see that classes that empirically enjoy strong contextual relations with other objects in the scenes have significant gain in precision over exhaustive search, such as boat, car, chair, cow, sofa etc..

6.4. Search space accuracy

To measure the quality of our predicted search areas, we evaluate the mean intersect vs. union (IU) of the search areas produced by our 20 questions approach with the groundtruth objects. We compare with the search area of the original detector, produced by the union of the object proposals with high scores. We also compare with the search area generated by the "Oracle", which is defined as the search area predicted by the optimal sequence among all paths explored during depth-first search. The mean IU of the original detectors, our 20 questions approach and the oracle are 64.12%, 73.9% and 78.2% respectively. We can see that our approach significantly improves the accuracy of overlap 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 reinforcement learning has

Table 1: Avg. detection precision of ours and other algorithms on Pascal VOC10 dataset.

	Aero	Bike	Bird	Boat	Bottle	Bus	Car	Cat	Chair	Cow	Table	Dog	Horse	Motor	Person	Plant	Sheep	Sofa	Train	TV	Avg.
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 Ctx.	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
Pascal 20 Ctx.	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
Pascal 33 Ctx.	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	**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	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
Ours (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**

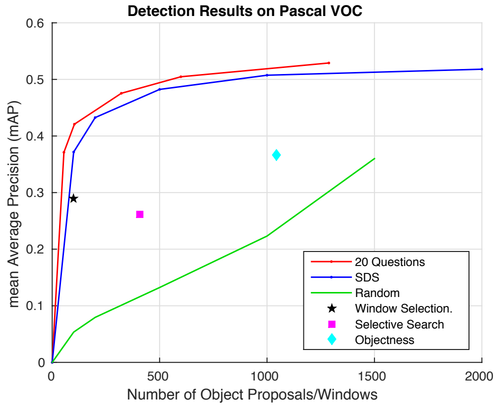

Figure 6: **Speed-accuracy tradeoff** mAP vs. number of amortized evaluated object proposals on Pascal VOC dataset. Best viewed in color.

learned a good policy that closely mimics the oracle's behavior.

6.5. 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 [14] in the simultaneous detection and segmentation task using the AP^r metric proposed in [14]. Table 2 and Table 3 show the performance on Pascal VOC10 and the MSRC datasets respectively. We outperform 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.

Figure 7 shows some qualitative results of for detection and segmentation of the MSRC object classes. We can see that using our sequential search approach the localization of the objects is more accurate because of the refined search areas, while the the probabilities are higher on the true object locations given observed context.

7. Conclusion

We propose a generic approach for object detection and segmentation as a sequential and dynamic process, in which a policy actively selects a context-related question adapting to a query and responses from previous context-related questions, then more accurately refines the search area or rejects the object query early without running many detectors. We frame the object detection problem as a Markov Decision Process to learn a policy by reinforcement learning. We use non-parametric spatial models to represent semantic context between objects. To demonstrate the efficacy of our approach, we apply this active detection scheme to a recent object detection and segmentation framework, and achieved higher average precision compared to the baselines with significant computational savings.

Acknowledgement: This work was partially supported by the US Government through ONR MURI Grant N000141010934.

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. C. Caicedo, F. U. K. Lorenz, and S. Lazebnik. Active object localization with deep reinforcement learning.

[8] 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.

[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.

Table 2: AP^r performance of simultaneous detection and segmentation on Pascal VOC10 dataset.

	Aero	Bike	Bird	Boat	Bottle	Bus	Car	Cat	Chair	Cow	Table	Dog	Horse	Motor	Person	Plant	Sheep	Sofa	Train	TV	Avg.
SDS	**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**

Table 3: AP^r performance of simultaneous detection and segmentation on MSRC dataset.

	cow	sheep	bird	chair	cat	dog	boat	body	car	bike	plane	mean
SDS	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**

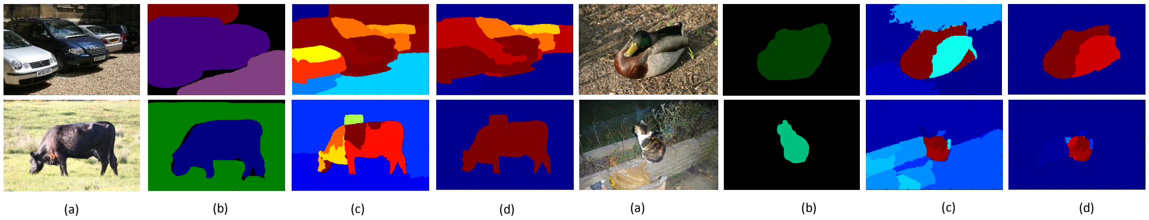

Figure 7: Qualitative results for detection and segmentation of the MSRC object classes. Columns (a) to (d) correspond to the original image, groundtruth label, probability map of the query object given by exhaustive search and by our sequential search respectively. The probability map from red to blue corresponds to the probability from high to low. Best viewed in color.

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

[12] R. Girshick. Fast r-cnn. *arXiv preprint arXiv:1504.08083*, 2015.

[13] 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.

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

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

[16] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 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] K. Lenc and A. Vedaldi. R-cnn minus r. In M. W. J. Xianghua Xie and G. K. L. Tam, editors, *Proceedings of the British Machine Vision Conference (BMVC)*, pages 5.1–5.12. BMVA Press, September 2015.

[19] V. Mnih, N. Heess, A. Graves, et al. Recurrent models of visual attention. In *Advances in Neural Information Processing Systems*, pages 2204–2212, 2014.

[20] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

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

[22] 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.

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

[24] 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.

[25] A. Rabinovich, A. Vedaldi, C. Galleguillos, E. Wiewiora, and S. Belongie. Objects in context. In *Computer vision, 2007. ICCV 2007. IEEE 11th international conference on*, pages 1–8. IEEE, 2007.

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

[27] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. *arXiv preprint arXiv:1506.02640*, 2015.

[28] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *arXiv preprint arXiv:1506.01497*, 2015.

[29] 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.

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

[31] 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.