Searching for Objects using Structure in Indoor Scenes

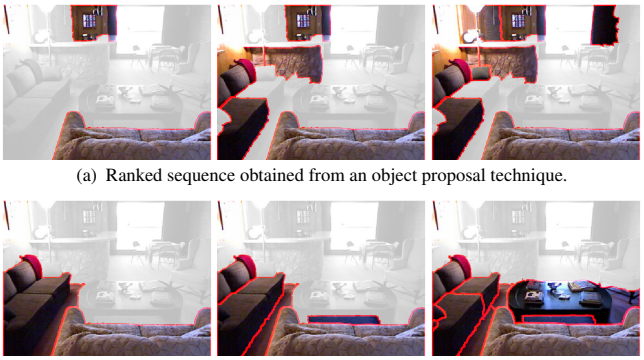
Varun K. Nagaraja
varun@umiacs.umd.edu
Vlad I. Morariu
morariu@umiacs.umd.edu
Larry S. Davis
lsd@umiacs.umd.edu

University of Maryland
College Park, MD. USA.

Consider the situation where a computer vision system needs to identify the presence or location of a particular object in an image. In a passive computer vision system, if we ask a specific question like "Where is the chair in this room?", it would process all the regions in the image to detect a chair instance. Such a vision system does not exploit the structure in the scene to efficiently process the image. In this work, our goal is to locate objects of interest in an image by processing as few image regions as possible using scene structure. Given a set of region proposals, we propose a search technique that sequentially processes the regions such that the regions that are more likely to correspond to the query class object are explored earlier. At each step during the search, we use the labels of the explored regions and spatial context to predict the likelihood that each unexplored region is an instance of the target class. We then select a few regions with highest likelihood, obtain the class label from the region classification module and add them to the explored set. The process is repeated with the updated set of explored regions.

We frame our sequential exploration problem as a Markov Decision Process (MDP) and use a reinforcement learning technique to learn an optimal search policy. Let R be the set of indices of the regions in the image and t correspond to a step index. Let R_e^t be the set of indices of the explored image regions and $R_u^t = R \setminus R_e^t$ be the set of indices of the unexplored image regions at a step t. To state our problem in the reinforcement learning setting, a state s_t is the set of all the image regions (r) explored until that step $s_t = \{r_i | i \in R_e^t\}$. An action corresponds to selecting the next image region to explore, $a_t = r_j$ where $j \in R_u^t$. A policy π is a function that maps states to actions $\pi(s)$. The goal is to find a policy that will maximize a cumulative function of the reward. However, it is challenging to manually specify a reward function for the search policy. The true reward function is unknown for our sequential exploration problem since the underlying distribution from which a spatial arrangement of objects in an image is generated is unknown, analogous to a game generated by a hidden emulator. But we have access to an oracle's actions in the individual images. Learning an optimal policy in such situations is known as imitation learning [3] where an oracle predicts the actions it would take at a state and the search policy learns to imitate the oracle and predict similar actions. The oracle in our image exploration problem selects the next set of regions to explore based on the groundtruth labels.

We use the DAgger (Dataset Aggregation) algorithm of Ross *et al*. [3] that trains a classifier as the search policy which takes in features extracted at a state as input and predicts an action as the output. In typical imitation learning algorithms, the policy is learned by training a classifier on the dataset of state features and actions obtained by sampling sequences produced by an oracle policy. They make an i.i.d assumption about the states encountered during the execution of a learned policy which does not hold for our problem since the policy's prediction affects future states. During the test stage, if the policy encounters a state that was not generated by the oracle policy, it could predict an incorrect action that can lead to compounding of errors. But DAgger does not make the i.i.d assumptions about states. During training, it starts with an initial classifier and runs through the states, predicting labels for each state. Based on its predictions, it is assigned a loss value at each state. At the end of an iteration, all the features, the predicted labels and the loss values for all states are collected. The aggregate of all the collected datasets until the current iteration is used to train a cost sensitive classifier, which becomes the policy for the next iteration. Since the training states are generated by the policy being learned, it is a non i.i.d supervised learning problem. DAgger is available in the Vowpal Wabbit library through a programming abstraction proposed by Daumé III *et al*. [1] where a developer writes a single *predict* function that encodes the algorithm for the testing stage and the training is done by making repeated calls to this predict function.



(a) Ranked sequence obtained from an object proposal technique.

(b) Sequence obtained from a search strategy that uses structure in the scene.

Figure 1: **Searching for a table**. Each step in the above sequence shows exploration of three additional regions in the image. The search strategy learned using our method utilizes the room structure and the presence of other objects in the image to discover the table region much earlier than using the ranked sequence from an object proposal technique.

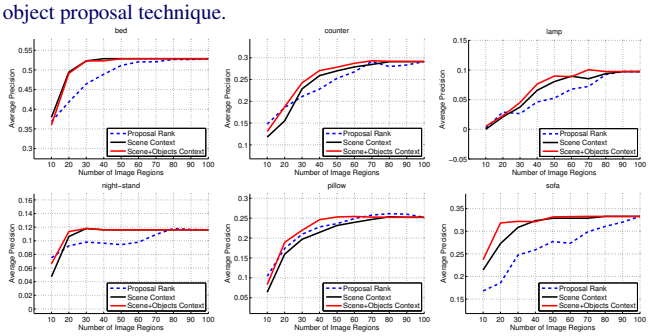


Figure 2: **Average Precision (AP) vs. number of processed regions.** A classifier trained for a query class with unary scene context features alone can achieve a significantly high average precision by processing very few regions. Classes like *bed, nightstand* and *sofa* need only 20-25% of the regions when compared to the proposal ranking sequence. A search strategy trained for a query class using both object-object context and scene-context features further improves the performance for classes like *counter, lamp, pillow* and *sofa*.

Since structure in the scene is essential for search, we work with indoor scene images as they contain both unary scene context information and object-object context in the scene. We demonstrate our approach on the NYU depth v2 dataset [4]. We use the RCNN-Depth module of Gupta *et al*. [2] for the region classification. Their region proposal module is a modified Multiscale Combinatorial Grouping (MCG) technique that incorporates depth features. Their feature extraction module is RCNN which includes CNNs fine-tuned on the depth images. We perform experiments on the NYU-depth v2 dataset and show that the unary scene context features alone can achieve a significantly high average precision while processing only 20-25% of the regions for classes like *bed* and *sofa*. By considering object-object context along with the scene context features, the performance is further improved for classes like *counter*, *lamp*, *pillow* and *sofa*. Our sequential search process adds a negligible overhead when compared to the time spent on extracting CNN features, hence the reduction in number of regions leads directly to a gain in computation speed of the object detection process.

[1] Hal Daumé III, John Langford, and Stephane Ross. Efficient programmable learning to search. *arXiv preprint arXiv:1406.1837*, 2014. URL `http://arxiv.org/abs/1406.1837`.

[2] Saurabh Gupta, Ross Girshick, P Arbeláez, and J Malik. Learning Rich Features from RGB-D Images for Object Detection and Segmentation. In *ECCV*, 2014.

[3] Stéphane Ross, Geoffrey J Gordon, and J Andrew Bagnell. A Reduction of Imitation Learning and Structured Prediction. In *AISTATS*, 2011.

[4] Nathan Silberman, Pushmeet Kohli, Derek Hoiem, and Rob Fergus. Indoor Segmentation and Support Inference from RGBD Images. In *ECCV*, 2012.