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 classifiers for all possible locations of objects in the image, we develop a divide-and-conquer approach by sequentially posing questions about the query and 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 (MDP) and learn a policy using imitation learning that can dynamically select questions based on the query, the scene and current observed responses given by object detectors and classifiers. The algorithm reduces over 30% of the object proposals evaluation while maintaining high average precision comparable to exhaustive search. Experiments show promising results compared with baselines of exhaustive search, searching for objects in random sequences and random locations.

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 [6, 28, 2]. Such methods are time-consuming since they need to evaluate a large number of object hypotheses, and 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 [3, 16]. 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 detection results [14, 10, 18]. However, such methods still need to evaluate the high order co-occurrence statistics and spatial relations with *all* other object classes appearing in the scene altogether, some of which may not be informative and even introduce error.

By contrast, human perception does not tend to process the whole scene at once. It 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, knowing the top of the scene is sky is not very helpful to distinguish whether there is a car or a boat since both can be under sky; on the other hand, observing road instead of water in the lower part gives a strong indication of the existence of cars. Additionally, cars are more likely to be found "beside" the buildings after observing the road. So in order to find the car, humans tend to first look for the road, then search around the buildings, instead of looking up to the sky. This motivates us to raise the question: *can object detection algorithms decide where*

to look for the object of a query class more efficiently and accurately by exploring a few related context cues dynamically, similar to humans?

Below we formulate this process as a Markov Decision Process (MDP), and use imitation learning to learn a context-driven policy that sequentially and dynamically selects the most informative context class to explore and refine the search area for the target. We show our framework in Figure 2. Specifically, like playing a Twenty Questions game, at each step the policy makes a decision about which detector of a context class to run given the query and responses from previous contextual classifiers. After taking the action to run context classifiers/detectors and observing responses, it then refines the search area for the query object using spatially-aware contextual models. We also incorporate early rejection as an action to avoid further running detection if the policy determines there is a low chance of having the query object in the scene, and this decision can be made even before running any object detectors so it can eliminate a large amount of unnecessary computation. Finally, we run the query object detector in the predicted search area if the policy thinks enough contextual information has been gathered and decides to stop. Object detection experiments in images of complex scenes like Pascal VOC dataset show that our algorithm can produce a search space that is close to the target object by checking its context, thus significantly reducing the number of object proposals considered and detector evaluations performed while maintaining comparable mean average precision (mAP) to exhaustive search. To the best of our knowledge, this is one of the first approaches to model the challenging task of simultaneous object detection and segmentation in complex real scenes using an imitation learning policy fully driven by semantic context.

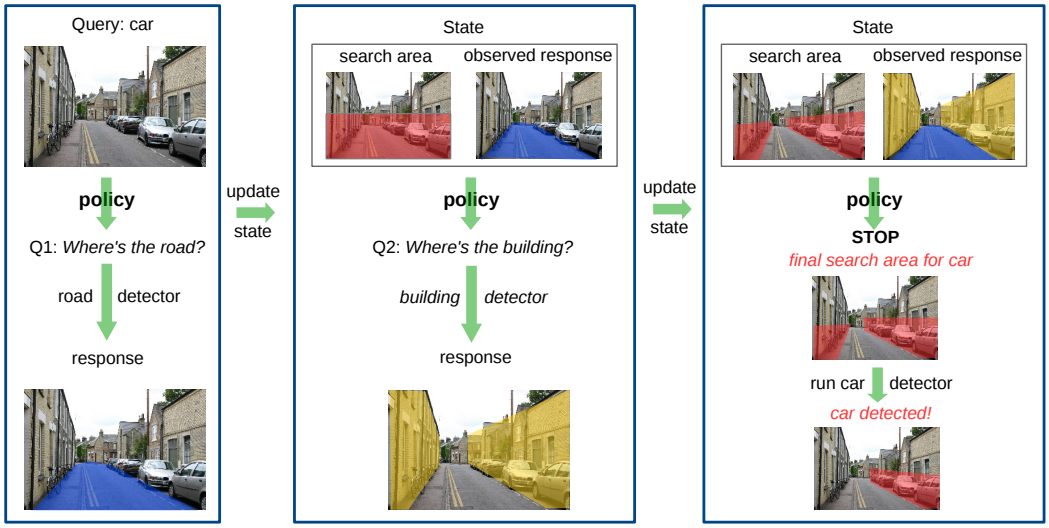

Figure 1: Illustration of our sequential search for objects in 20 context driven questions.

2 Related Work

Sequential Testing. The "20 question" approach to pattern recognition dates back to Blanchard and Geman [4], 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 [25] and "visual Turing test" [12]. 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, 19] 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. [5] 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. [17] 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 positve detections arise. To reduce the number of windows evaluated, category independent object proposals [6, 28, 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 [13, 15] 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 [14, 26, 18], 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, 7]. [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. [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

A sequential decision-making problem can be formulated as a Markov Decision Process (MDP). An MDP is defined by a tuple $(\mathcal{S}, \mathcal{A}, T(.), R(.))$, where S is a set of *states* and A is a set of *actions*. An agent interacts with the environment by following a *policy* $\pi : S \rightarrow A$ that determines which action to take in a given state. After taking action a in state s, the environment takes the agent to the next state s' according to the transition probability $T(s'|s, a)$ and responds with some reward $R(s, a)$. 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.

Given an image X and a query object class c_q ($q \in 1, .., C$), we detect the query class of objects by sequentially making decision of exploring one context/object class, and narrow down the search area for the query based on observed responses. Our framework is shown in Figure 2. We model our problem as an MDP.

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

- **State** s_t is (X^t, O^t), where X^t is the search area for the query at time t, and initially the entire image $X^0 = X$. $O^t = \{o_1, o_2, ..., o_t\}$ is a sequence of observations of responses from applying contextual classifiers
- The **actions** set $\mathcal{A} = \{a_1, ..., a_C, \text{Stop}, \text{Reject}\}$, 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 detector corresponding to the query.

- **State transition** function $T(s'|s, a, X)$ is deterministic in our case.
- The **reward** function $R(s, a, s') \rightarrow \mathbb{R}$ gives the reward for taking action a in state s leading to the next state s'
- The **discount** constant γ captures a tradeoff between taking the action that greedily maximizing the immediate reward or the considering long term expected reward.
- The **policy** $\pi(s) : \mathcal{S} \rightarrow \mathcal{A}$ is a mapping from a state to an action.

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.

4 Approach

4.1 Learning the Policy by Imitation

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

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

To learn Q^*, we assume that these values are given by an oracle *at training time*; thus the problem is 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 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.

The oracle's action sequence maximized the cumulative reward. Since the space of action sequences is exponential, we compute the oracle by breadth-first search with pruning. To avoid collecting examples from the oracle's trajectory only, we encourage exploration by starting from a random state. We collect examples (s, a, r) samples from the oracle's trajectory and use ridge regression to predict the Q-values.

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:

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

Given a refined search space $X_c \in X$ of a context class c_t at time t, we formalize $p(c|c_t, X)$ as a weighted vote from the cooccurring region pairs of class c_t and c in training scenes. Let $(s^i_{c_t}, s^i_c)$ be

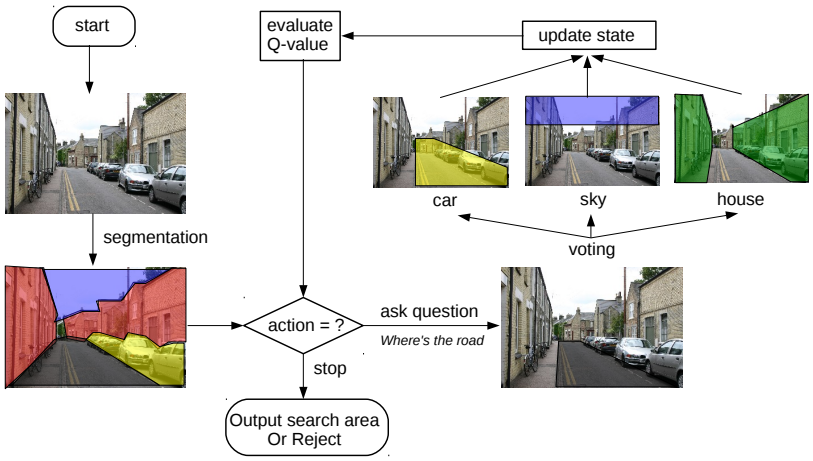

Figure 2: Framework of our context driven object searching. We first generate region hypotheses using object proposal algorithms, then the policy evaluates the current state and iteratively selects the action maximizing the Q-value function. Afterwards, the possible search locations are updated and the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

the i-th pair of co-occurring regions of class c_t and c, and $b^i_{c_t}$ and b^i_c be their corresponding bounding boxes. We can now define the probabilistic vote map $p(c|c_t, X)$ as:

$$p(c|c_t, X_c) = \frac{1}{Z_c} \sum_i W(s^i_{c_t}, s; \theta^W).T(b^i_{c_t}, b^i_c) \tag{5}$$

where $s \in X^t$ is a region within the search space of the context class c_t. Z_c is the normalization function. $W(.)$ is a kernel measuring similarity of region s with a training region s_i. $T(b^i_{c_t}, b^i_c)$ models the transformation from $b^i_{c_t}$ to b^i_c, 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

$$p(c_t|c, X) = \sum_{s \in X_c} p(c_t|X_c) \sum_i W(s^i_{c_t}, s; \theta^W).T(b^i_{c_t}, b^i_c) \tag{6}$$

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.

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:

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

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

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

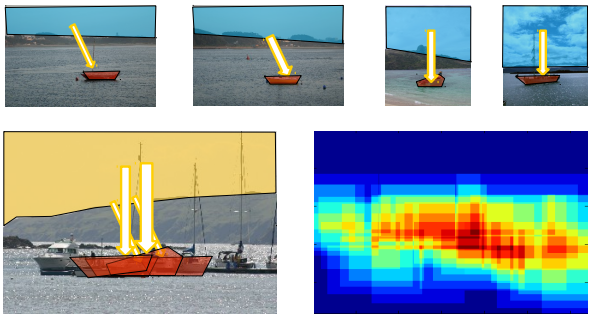

Figure 3: Examples of our weighted vote map for the context from sky to boat. The first rows are the training sample pairs of sky and boat and the second row is the test image and the resulted weighted voting map. The widths of the arrows denote the weighted similarity $W(s^i_{c_t}, s; \theta^W)$ between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

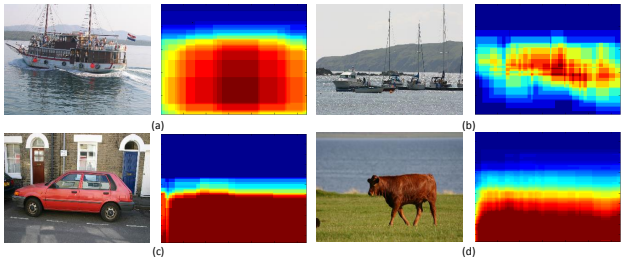

Figure 4: Examples of our context vote maps. Each pair of images corresponds to the original image and the vote-based probability map of object location from observed context. From (a) - (d) are the vote maps from water to boat, sky to boat, road to car and grass to cow, respectively. Best viewed in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

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 [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 [7] 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. [8] 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 [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., which is adequate for our purposes. 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.

5.3 Feature Representation

To classify object proposals, we extract region features and classify them using the deep neural network model in [15] 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 [27] and learn one-vs-all SVM models for classification.

6 Experiments

6.1 Reduction of Number of Object Proposals

Figure **??** shows that our 20 questions detection algorithm can effectively reduce a large amount of object proposals ($30\% \sim 40\%$) while maintaining similar mAP performance compared to exhaustive detection on all object proposals.

6.2 Comparison with other context based methods

6.3 Comparison with random search methods

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] 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.

[4] G. Blanchard and D. Geman. Hierarchical testing designs for pattern recognition. *Annals of Statistics*, pages 1155–1202, 2005.

[5] 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.

[6] 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.

[7] 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.

[8] 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.

[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] 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.

[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] 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.

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

[16] 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.

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

[18] 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.

[19] H. Larochelle and G. E. Hinton. Learning to combine foveal glimpses with a third-order boltzmann machine. In *Advances in neural information processing systems*, pages 1243–1251, 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.