
Hierarchical Object Detection with Deep Reinforcement Learning

Anonymous Author(s)

Affiliation

Address

email

Abstract

1 We present a method for performing hierarchical object detection in images guided
2 by a deep reinforcement learning agent. The key idea is to focus on those parts
3 of the image that contain richer information and zoom on them. We train an
4 intelligent agent that, given an image window, is capable of deciding where to
5 focus the attention among five different predefined region candidates (smaller
6 windows). This procedure is iterated providing a hierarchical image analysis.

7 We compare two different candidate proposal strategies to guide the object search:
8 with and without overlap. Moreover, our work compares two different strategies to
9 extract features from a convolutional neural network for each region proposal: a
10 first one that computes new feature maps for each region proposal, and a second
11 one that computes the feature maps for the whole image to later generate crops for
12 each region proposal.

13 Experiments indicate better results for the overlapping candidate proposal strategy
14 and a loss of performance for the cropped image features due to the loss of spatial
15 resolution. We argue that, while this loss seems unavoidable when working with
16 large amounts of object candidates, the much more reduced amount of region
17 proposals generated by our reinforcement learning agent allows considering to
18 extract features for each location without sharing convolutional computation among
19 regions.

20 1 Introduction

21 When we humans look at an image, we always perform a sequential extraction of information in
22 order to understand its content. First, we fix our gaze to the most salient part of the image and, from
23 the extracted information, we guide our look towards another image point, until we have analyzed all
24 its relevant information. This is our natural and instinctive behavior to gather information from our
25 surroundings.

26 Traditionally in computer vision, images have been analyzed at the local scale following a sliding
27 window scanning, often at different scales. This approach analyses the different image parts inde-
28 pendently, without relating them. Just by introducing a hierarchical representation of the image, we
29 can more easily exploit the relationship between regions. We propose to use a top-down scanning
30 which firstly takes a global view of the image to sequentially focus on the local parts that contain the
31 relevant information (e.g. objects or faces).

32 Our algorithm is based on an intelligent agent trained by reinforcement learning that is capable of
33 making decisions to detect an object in a still image, similarly to [2]. The agent first analyzes the
34 whole image, and decides in which region of the image to focus among a set of predefined ones.
35 Inspired by [9], our agent can top-down explore a set of five different predefined region candidates:

36 four regions representing the four quadrants plus a central region. Two different strategies have been
37 studied: proposing overlapping or non-overlapping candidates. The agent stops its search when
38 it finds an object. Reinforcement learning is useful for our task because there is no single way of
39 completing it. The agent can explore the hierarchical representation in different ways and still achieve
40 its goal. Then, instead of programming every step that the agent should do, we train it so that it
41 makes decisions under uncertainty to reach its target. Notice that the final goal of object detection is
42 to define a bounding box around the object and that, in our work, these bounding boxes are limited to
43 the predefined regions in the hierarchy.

44 Most state of the art solutions for object detection analyze large amounts of region proposals. These
45 algorithms need to leverage the bottleneck of describing all these proposals by reusing convolutional
46 feature maps of the whole image. In our work though, as the reinforcement learning agent and the
47 hierarchy allow us to analyze a very reduced number of regions, we can feed each region visited by
48 the agent through a convolutional network to extract its features, allowing us to work with region
49 representations of higher spatial resolution, which are also more informative than those cropped from
50 a feature map of the whole image. To study this trade-off, we have trained and compared two different
51 models based on each of these two principles: the *Image-Zooms* model, which extracts descriptors
52 at each region, and the *Pool45-Crops* model, which reuses feature maps for different regions of the
53 same image.

54 The first contribution of our work is the introduction of a hierarchical representation to top-down
55 (zoom-in) guide our agent through the image. We explore how the design of the hierarchy affects
56 the detection performance and the amount of visited regions. The second contribution is the study
57 between extracting features for each region instead of reusing feature maps for several locations.
58 We show the gain of the region-specific features for our scheme, and argue that the computational
59 overhead is minor thanks to the very reduced amount of regions considered by the agent.

60 2 Related Work

61 Reinforcement learning is a powerful tool that has been used in a wide range of applications. The
62 most impressive results are those from DeepMind [11], who have been able to train an agent that
63 plays Atari 2600 video games by observing only their screen pixels, achieving even superhuman
64 performance. Also they trained a computer that won the Go competition to a professional player for
65 the first time [16]. More specifically to traditional computer vision tasks, reinforcement learning has
66 been applied to learn spatial glimpse policies for image classification [10, 1], for captioning [20] or
67 for activity recognition [21]. It has also been applied for object detection in images [2], casting a
68 Markov Decision Process, as our approach does.

69 The traditional solutions for object detection are based on region proposals, such as Selective Search
70 [19], CPMC [3] or MCG [13] and other methods based on sliding windows such as EdgeBoxes [23].
71 The extraction of such proposals was independent of the classifier that would score and select which
72 regions compose the final detection. These methods are computationally expensive because rely on
73 a large number of object proposals. Then the first trends based on Convolutional Neural Networks
74 appeared, such as Fast R-CNN [6], that already studied how to share convolutional computation
75 among locations, as they identified that the extraction of features for the hypothesized objects was the
76 bottleneck for object detection.

77 More recent proposals such as Faster R-CNN [15] have achieved efficient and fast object detection
78 by obtaining cost-free region proposals sharing full-image convolutional features with the detection
79 network. Directly predicting bounding boxes from an image is a difficult task, and for this reason,
80 approaches such as Faster R-CNN rely on a number of reference boxes called anchors, that facilitate
81 the task of predicting accurate bounding boxes by regressing these initial reference boxes. One key
82 of our approach is the refinement of bounding box predictions through the different actions selected
83 by the reinforcement learning agent. Besides Faster R-CNN or other approaches such as YoLo
84 [14] or MultiBox [4] based on anchors, there are other works that are based on the refinement of
85 predictions. Yoo et al. [22] propose the AttentionNet. They cast an object detection problem as an
86 iterative classification problem. AttentionNet predicts a number of weak directions pointing to the
87 target object so that a final accurate bounding box is obtained. The state-of-the-art in object detection
88 is the Single Shot MultiBox Detector (SSD) [8], which works with a number of default boxes of

89 different aspect ratios and scales per each feature map location, and also adjusts them to a better
90 match to the object shape.

91 Another approach that supports this idea is the Active Object Localization method proposed by
92 Caicedo and Lazebnik [2]. Their method trains an intelligent agent using deep reinforcement learning
93 that is able to deform bounding boxes sequentially until they fit the target bounding box. Each action
94 that the agent does to the bounding box can change its aspect ratio, scale or position. Our main
95 difference to this approach is that we add a fixed hierarchical representation that forces a top-down
96 search, so that each action zooms onto the predicted region of interest.

97 How to benefit from super-resolution has also been studied by other works. In the paper of Lu et al.
98 [9] a model is trained to determine if it is required to further divide the current observed region
99 because there are still small objects on it, and in this case, each subregion is analyzed independently.
100 Their approach could also be seen as hierarchical, but in this case they analyze each subregion when
101 the zoom prediction is positive, whereas we just analyze the subregion selected by the reinforcement
102 learning agent. On contrast to their proposal, we analyze fewer regions and then we can afford
103 extracting high-quality descriptors for each of them, instead of sharing convolutional features.

104 3 Hierarchical Object Detection Model

105 In this work we define the object detection problem as the sequential decision process of a goal-
106 oriented agent interacting with a visual environment that is our image. At each time step the agent
107 should decide in which region of the image to focus its attention so that it can find objects in a few
108 steps. We cast the problem as a Markov Decision Process, that provides a framework to model
109 decision making when outcomes are partly uncertain.

110 3.1 MDP formulation

111 In order to understand the models for the object detection task that we have developed, we first define
112 how the Markov Decision Process is parameterized.

113 **State** The state is composed by the descriptor of the current region and a memory vector. The
114 type of descriptor defines the two models we compare in our work: the *Image-Zooms* model and the
115 *Pool45-Crops* model. These two variations are explained in detail in Section 3.3. The memory vector
116 of the state captures the last 4 actions that the agent has already performed in the search for an object.
117 As the agent is learning a refinement of a bounding box, a memory vector that encodes the state of
118 this refinement procedure is useful to stabilize the search trajectories. We encode the past 4 actions in
119 a one-shot vector. As there are 6 different actions presented in the following section, the memory
120 vector has 24 dimensions. This type of memory vector was also used in [2].

121 **Actions** There are two types of possible actions: *movement actions* that imply a change in the
122 current observed region, and the *terminal* action to indicate that the object is found and that the
123 search has ended. One particularity of our system is that each movement action can only transfer the
124 attention top-down between regions from a predefined hierarchy. A hierarchy is built by defining five
125 subregions over each observed bounding box: four quarters distributed as 2×2 over the box and a
126 central overlapping region. We have explored two variations of this basic 2×2 scheme: a first one
127 with *non-overlapped* quarters (see Figure 1), and a second one with *overlapped* quarters (see Figure
128 2), being the size of a subregion $3/4$ of its ancestor. Then, there are five movement actions, each one
129 associated to one of the yellow regions. If, on the other hand, the terminal action is selected, there is
130 no movement and the final region is the one marked with blue.

131 **Reward** The reward functions used are the ones proposed by Caicedo and Lazebnik [2]. The
132 reward function for the movement actions can be seen in Equation 1 and the reward function for the
133 terminal action in Equation 2. Given a certain state s , a reward is given to those actions that move
134 towards a region b' with a greater Intersection Over Union (IoU) with the ground truth g than the
135 region b considered at the previous step. Otherwise, the actions are penalized. For the trigger action,
136 the reward is positive if the Intersection Over Union of the actual region b with the ground truth is

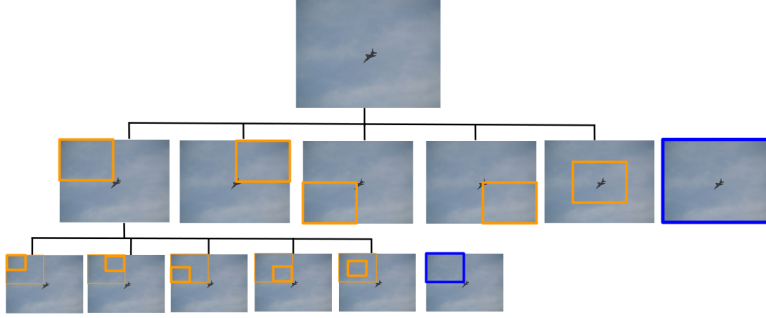


Figure 1: Image hierarchy of three levels with non-overlapped quarters

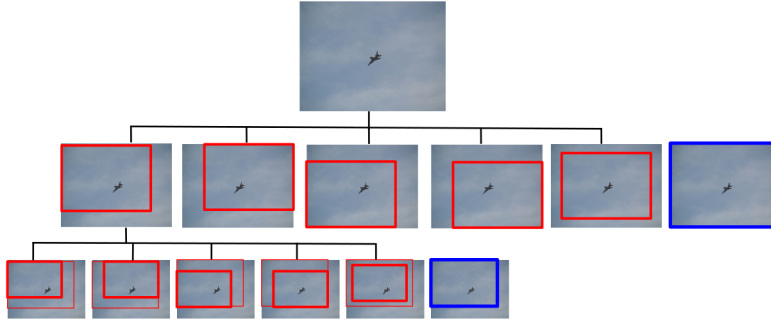


Figure 2: Image hierarchy of three levels with overlapped quarters

137 greater than a certain threshold τ , and negative otherwise. We consider $\tau = 0.5$, because it is the
 138 threshold for which a detection is considered positive, and η is 3, as in [2].

$$R_m(s, s') = \text{sign}(\text{IoU}(b', g) - \text{IoU}(b, g)) \quad (1)$$

$$R_t(s, s') = \begin{cases} +\eta & \text{if } \text{IoU}(b, g) \geq \tau \\ -\eta & \text{otherwise} \end{cases} \quad (2)$$

139 3.2 Q-learning

140 The reward of the agent depending on the chosen action a at state s is governed by a function $Q(s, a)$,
 141 that can be estimated with Q-learning. Based on $Q(s, a)$, the agent will choose the action that is
 142 associated to the highest reward. Q-learning iteratively updates the action-selection policy using the
 143 Bellman equation 3, where s and a are the current state and action correspondingly, r is the immediate
 144 reward and $\max_a Q(s', a')$ represents the future reward. Finally γ represents the discount factor. In
 145 our work, we approximate the Q-function by a Deep Q-network trained with Reinforcement Learning
 146 [11].

$$Q(s, a) = r + \gamma \max_a Q(s', a') \quad (3)$$

147 3.3 Model

148 In our work we study two different approaches for visual feature extraction, which are used to train
 149 a Deep Q-Network. Figure 3 depicts the two variations with the common reinforcement learning
 150 network.

151 We compare two models to extract the visual features that define the state of our agent: the *Image-Zooms*
 152 model and the *Pool45-Crops* model. For the *Image-Zooms* model, each region is resized to
 153 224x224 and its visual descriptors correspond to the feature maps from Pool5 layer of VGG-16 [17].
 154 For the *Pool45-Crops* model, the image at full-resolution is forwarded into VGG-16 [17] through
 155 Pool5 layer. As Girshick [6], we reuse the feature maps extracted from the whole image for all the
 156 regions of interest (ROI) by pooling them (ROI pooling). As in SSD [8], we choose which feature
 157 map to use depending on the scale of the region of interest. In our case, we only work with the Pool4
 158 and Pool5 layers, that are the two last pooling layers from VGG-16. Once we have a certain region of
 159 interest from the hierarchy, we decide which feature map to use by comparing the scale of the ROI
 160 and the scale of the feature map. For large objects, the algorithm will select the deeper feature map,
 161 whereas for smaller objects a shallower feature map is more adequate.

162 The two models for feature extraction result into a feature map of 7x7 which is fed to the common
 163 block of the architecture. The region descriptor and the memory vector are the input of the Deep
 164 Q-network that consists of two fully connected layers of 1024 neurons each. Each fully connected
 165 layer is followed by a ReLU [12] activation function and is trained with dropout [18]. Finally the
 166 output layer corresponds to the possible actions of the agent, six in our case.

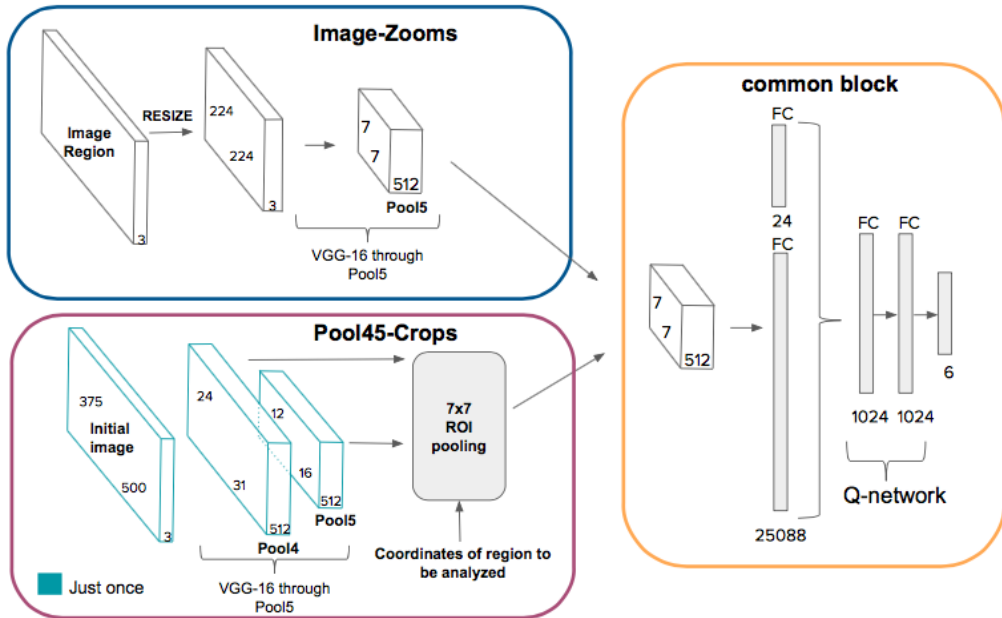


Figure 3: Hierarchical Object Detection Models

167 3.4 Training

168 In this section we will explain the particularities that we chose to train the Q-network.

169 **Exploration-Exploitation** To train the deep Q-network with reinforcement learning we use an
 170 ϵ -greedy policy, that starts with $\epsilon=1$ and decreases until $\epsilon=0.1$ in steps of 0.01. Then, we start with
 171 random actions, and at each epoch the agent takes decisions relying more on the already learnt policy.
 172 Actually, in order to help the agent to learn the terminal action, which in random could be difficult to
 173 learn, we force it each time the current region has a IoU > 0.5. With this approach we can accelerate
 174 the training. Notice that we always do exploration, so we do not get stuck into a local minimum.

175 **Learning trajectories** One fact that we detected while training was that we should not impose
 176 which object of the image to look first. At each time step, the agent will focus on the object in the
 177 current region with the highest overlap with its ground-truth. This way, it is possible then that the
 178 target object changes during the top-down exploration.

179 **Training parameters** The weights for the Deep Q-network were initialized from a normal distribu-
180 tion. For learning, we used Adam optimizer [7] with a learning rate of 1e-6 to avoid that the gradients
181 explode. We trained each model for 50 epochs.

182 **Experience Replay** As we have seen previously, Bellman Equation 3 learns from transitions
183 formed by (s, a, r, s') , which can also be called experiences. Consecutive experiences in our algorithm
184 are very correlated and this could lead to inefficient and unstable learning, a traditional problem in
185 Q-learning. One solution to make the algorithm converge is collecting experiences and storing them
186 in a replay memory. Random minibatches from this replay memory are used to train the network. We
187 used an experience replay of 1,000 experiences and a batch size of 100.

188 **Discount factor** To perform well in the long-run, the future rewards should also be taken into
189 account and not only the most immediate ones. To do this, we use the discounted reward from
190 Bellman Equation 3 with a value of $\gamma = 0.90$. We set the gamma high because we are interested in
191 balancing the immediate and future rewards.

192 4 Experiments

193 Our experiments on object detection have used images and annotations from the PASCAL VOC
194 dataset [5]. We trained our system on the trainval sets of 2007 and 2012, and tested it on the test
195 set of 2007. We performed all the experiments for just one class, the *aeroplane* category, and only
196 considering pictures with the target class category. This experiment allows us to study the behavior
197 of our agent and estimate the amount of regions that must be analyzed to detect an object.

198 4.1 Qualitative results

199 We present some qualitative results in Figure 4 to show how our agent behaves on test images. These
200 results are obtained with the *Image-Zooms* model with overlapped regions, as this is the one that
201 yields best results, as argued in the following sections. We observed that for most images, the model
202 successfully zooms towards the object and completes the task in a few steps. As seen in the second,
203 third and fourth rows, with just two or three steps, the agent selects the bounding box around the
204 object. The agent also performs accurately when there are small instances of objects, as seen in the
205 first and last rows.

206 4.2 Precision-Recall curves

207 We will analyze the precision and recall curves for different trained models, considering that an object
208 is correctly detected when the Intersection over Union (IoU) of its bounding box compared to the
209 ground truth is over 0.5, as defined by the Pascal VOC challenge [5].

210 The Precision-Recall curves are generated by ranking all regions analyzed by the agent. The sorting
211 is based on the reward estimated by the sixth neuron of the Q-network, which corresponds to the
212 action of considering the region as terminal.

213 **Upper bound and random baselines** Our results firstly include baseline and upper-bound ref-
214 erences for a better analysis. As a baseline we have programmed an agent that chooses random
215 actions and detection scores at each time step. As an upper-bound, we have exploited the ground
216 truth annotations to manually guide our agent towards the region with the greatest IoU. The result
217 of these random baselines and upper-bounds for hierarchy type can be seen in Figure 5. It is also
218 important to notice that the best upper-bound option does not even achieve a recall of 0.5. This
219 poor performance is because more than half of the ground truth objects do not fit with the considered
220 region proposals, so they cannot be detected in our framework.

221 **Overlapped and non-overlapped regions** The results obtained with the upper-bound and baseline
222 methods provide enough information to compare the overlapped and non-overlapped schemes. The
223 overlapped regions scheme is the one that provides higher precision and recall values, both for the
224 upper-bound and the random models. This superiority of the overlapped case can be explained by the
225 slower reduction of spatial scale with respect to the non-overlapped model: as bounding box regions



Figure 4: Visualizations of searches for objects

226 are larger due to the overlap, their division in equal-size subregions also generate larger subregions.
 227 This also implies that the agent will require more steps to reach a lower resolution, but this finer
 228 top-down exploration is shown as beneficial in our experiments as the chances of missing an object
 229 during the descent are also lower.

230 **Model comparison** The *Image-Zooms* model and the *Pool45-Crops* model are compared in Figure
 231 6. Results clearly indicate that the *Image-Zooms* model performs better than the *Pool45-Crops* model.
 232 We hypothesize that this loss of performance is due to the loss of resolution resulting from the
 233 ROI-pooling over Pool4 or Pool5 layers. While in the *Image-Zooms* model the 7×7 feature maps
 234 of Pool5 have been computed directly from a zoom over the image, in the *Pool45-Crops* model the
 235 region crops over Pool4 or Pool5 could be smaller than 7×7 . While these cases would be upsampled
 236 to the $7 \times 7 \times 512$ input tensor to the deep Q-Net, the source feature maps of the region would be of
 237 lower resolution than their counterparts in the *Image-Zoom* model.

238 **Models at different epochs** We study the training of our *Image-Zooms* model by plotting the
 239 Precision-Recall curves at different epochs in Figure 7. As expected, we observe how the performance
 240 of the model improves with the epochs.

241 4.3 Number of regions analyzed per object

242 An histogram of the amount of regions analyzed by our agent is shown in 8. We observe that the
 243 major part of objects are already found with a single step, which means that the object occupies the
 244 major part of the image. With less than 3 steps we can almost approximate all objects we can detect.

245 5 Conclusions

246 This paper has presented a deep reinforcement learning solution for object detection. Our solution is
 247 characterized by a top-down exploration of a hierarchy of regions guided by an intelligent agent.

248 Our experiments indicate that objects can be detected with very few proposals from an appropriate
 249 hierarchy, but that working with a predefined set of regions clearly limits the recall. A possible

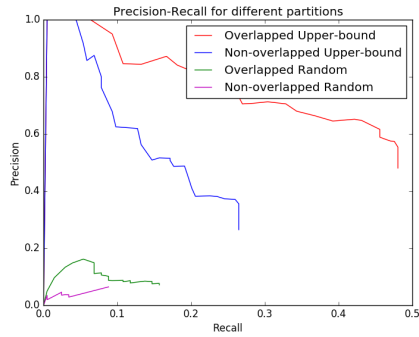


Figure 5: Baselines and upper-bounds

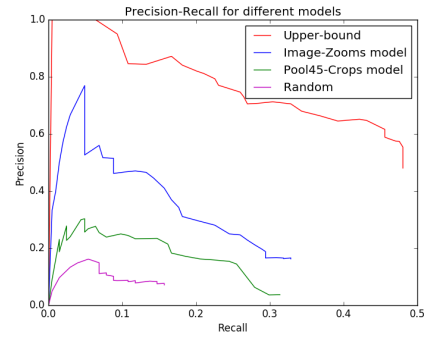


Figure 6: Comparison of the two models

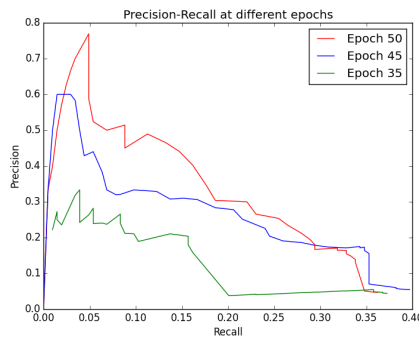


Figure 7: Image-Zooms at different epochs

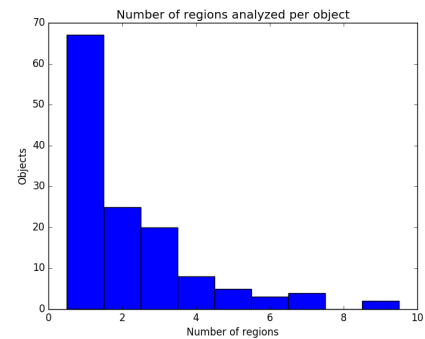


Figure 8: Regions analyzed per object

250 solution to this problem would be refining the approximate detections provided by the agent with a
 251 regressor, as in [15].

252 Finally, our results indicate the limitations of cropping region features from the convolutional layers,
 253 especially when considering small objects. We suggest that, given the much smaller amount of region
 254 proposals considered by our reinforcement learning agent, feeding each region through the network
 255 is a solution that should be also considered.

256 References

- 257 [1] Jimmy Ba, Volodymyr Mnih, and Koray Kavukcuoglu. Multiple object recognition with visual attention.
 258 *arXiv preprint arXiv:1412.7755*, 2014.
- 259 [2] Juan C Caicedo and Svetlana Lazebnik. Active object localization with deep reinforcement learning. In
 260 *Proceedings of the IEEE International Conference on Computer Vision*, pages 2488–2496, 2015.
- 261 [3] Joao Carreira and Cristian Sminchisescu. Cpmc: Automatic object segmentation using constrained
 262 parametric min-cuts. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(7):1312–1328,
 263 2012.
- 264 [4] Dumitru Erhan, Christian Szegedy, Alexander Toshev, and Dragomir Anguelov. Scalable object detection
 265 using deep neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern
 266 Recognition*, pages 2147–2154, 2014.
- 267 [5] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The
 268 pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2):303–338,
 269 2010.
- 270 [6] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision*,
 271 pages 1440–1448, 2015.

- 272 [7] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint*
273 *arXiv:1412.6980*, 2014.
- 274 [8] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, and Scott Reed. Ssd: Single shot
275 multibox detector. *arXiv preprint arXiv:1512.02325*, 2015.
- 276 [9] Yongxi Lu, Tara Javidi, and Svetlana Lazebnik. Adaptive object detection using adjacency and zoom
277 prediction. *arXiv preprint arXiv:1512.07711*, 2015.
- 278 [10] Volodymyr Mnih, Nicolas Heess, Alex Graves, et al. Recurrent models of visual attention. In *Advances in*
279 *Neural Information Processing Systems*, pages 2204–2212, 2014.
- 280 [11] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare,
281 Alex Graves, Martin Riedmiller, Andreas K Fiedjeland, Georg Ostrovski, et al. Human-level control through
282 deep reinforcement learning. *Nature*, 518(7540):529–533, 2015.
- 283 [12] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In
284 *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pages 807–814, 2010.
- 285 [13] Jordi Pont-Tuset, Pablo Arbeláez, Jonathan T. Barron, Ferran Marques, and Jitendra Malik. Multiscale
286 combinatorial grouping for image segmentation and object proposal generation. *IEEE Transactions on*
287 *Pattern Analysis and Machine Intelligence (TPAMI)*, 2016.
- 288 [14] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time
289 object detection. *arXiv preprint arXiv:1506.02640*, 2015.
- 290 [15] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection
291 with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.
- 292 [16] David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian
293 Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of go
294 with deep neural networks and tree search. *Nature*, 529(7587):484–489, 2016.
- 295 [17] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition.
296 *CoRR*, abs/1409.1556, 2014.
- 297 [18] Nitish Srivastava, Geoffrey E Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout:
298 a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1):
299 1929–1958, 2014.
- 300 [19] Koen EA Van de Sande, Jasper RR Uijlings, Theo Gevers, and Arnold WM Smeulders. Segmentation
301 as selective search for object recognition. In *2011 International Conference on Computer Vision*, pages
302 1879–1886. IEEE, 2011.
- 303 [20] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhutdinov, Richard S
304 Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention.
305 *arXiv preprint arXiv:1502.03044*, 2(3):5, 2015.
- 306 [21] Serena Yeung, Olga Russakovsky, Greg Mori, and Li Fei-Fei. End-to-end learning of action detection from
307 frame glimpses in videos. *arXiv preprint arXiv:1511.06984*, 2015.
- 308 [22] Donggeun Yoo, Sunggyun Park, Joon-Young Lee, Anthony S Paek, and In So Kweon. Attentionnet:
309 Aggregating weak directions for accurate object detection. In *Proceedings of the IEEE International*
310 *Conference on Computer Vision*, pages 2659–2667, 2015.
- 311 [23] C Lawrence Zitnick and Piotr Dollár. Edge boxes: Locating object proposals from edges. In *European*
312 *Conference on Computer Vision*, pages 391–405. Springer, 2014.