CS 484, Fall 2017
Term Project: Object Localization

The goal of this project is to develop an object localization method that can be used for generating object proposals for recognition systems. Commonly used data sets for training object recognition systems include images that are tightly cropped around objects so that the learning algorithms can focus on the characteristics of the objects of interest. A straightforward way of applying these methods on test images that contain multiple objects on varying backgrounds is to slide windows at different scales and run the recognition algorithms for each window. An alternative to this computationally inefficient procedure is to run object localization algorithms and obtain object proposals so that the more complex recognition algorithms are run only on these image regions.

In this project, you are asked to developed a superpixel grouping based object localization algorithm. You are expected to work in groups of two or three. Specifications for the components of the object localization procedure are given below.

1 Data

A set of 200 images from the PASCAL Visual Object Classes (VOC) Challenge (`http://host.robots.ox.ac.uk/pascal/VOC/`) will be used in this project. Example images are shown in Figure 1. Each image contains one or more bounding boxes for objects belonging to the following categories: cat, cow, dog, horse, sheep, boat, bus, car, train, and tvmonitor. The bounding boxes are stored in text files with the same names as the image files. Each bounding box is represented by four numbers, x_1 and y_1 for the upper-left corner, x_2 and y_2 for the lower-right corner, and an additional integer that indicates the object class. The class ids are listed in a file named `classes.txt`. The bounding boxes will be used as reference data for performance evaluation.

2 Background

As mentioned above, the object localization algorithm is based on grouping superpixels. Read the following papers to learn more about the problem of interest and the methodology:

- R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, S. Susstrunk, "SLIC Superpixels Compared to State-of-the-art Superpixel Methods," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 34, no. 11, pp. 2274–2282, May 2012.
- J. R. Uijlings, K. E. van de Sande, T. Gevers, A. W. Smeulders, "Selective search for object recognition," International Journal of Computer Vision, vol. 104, no. 2, pp. 154–171, September 2013.
- C. L. Zitnick, P. Dollar, "Edge boxes: Locating object proposals from edges," in European Conference on Computer Vision (ECCV), pp. 391–405, 2014.
- Y. Xiao, C. Lu, E. Tsougenis, Y. Lu and C.-K. Tang, "Complexity-adaptive distance metric for object proposals generation," in IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 778–786, 2015.

In particular, we will use the algorithm described in the first paper to obtain the superpixels and the algorithm described in the last paper to group them.

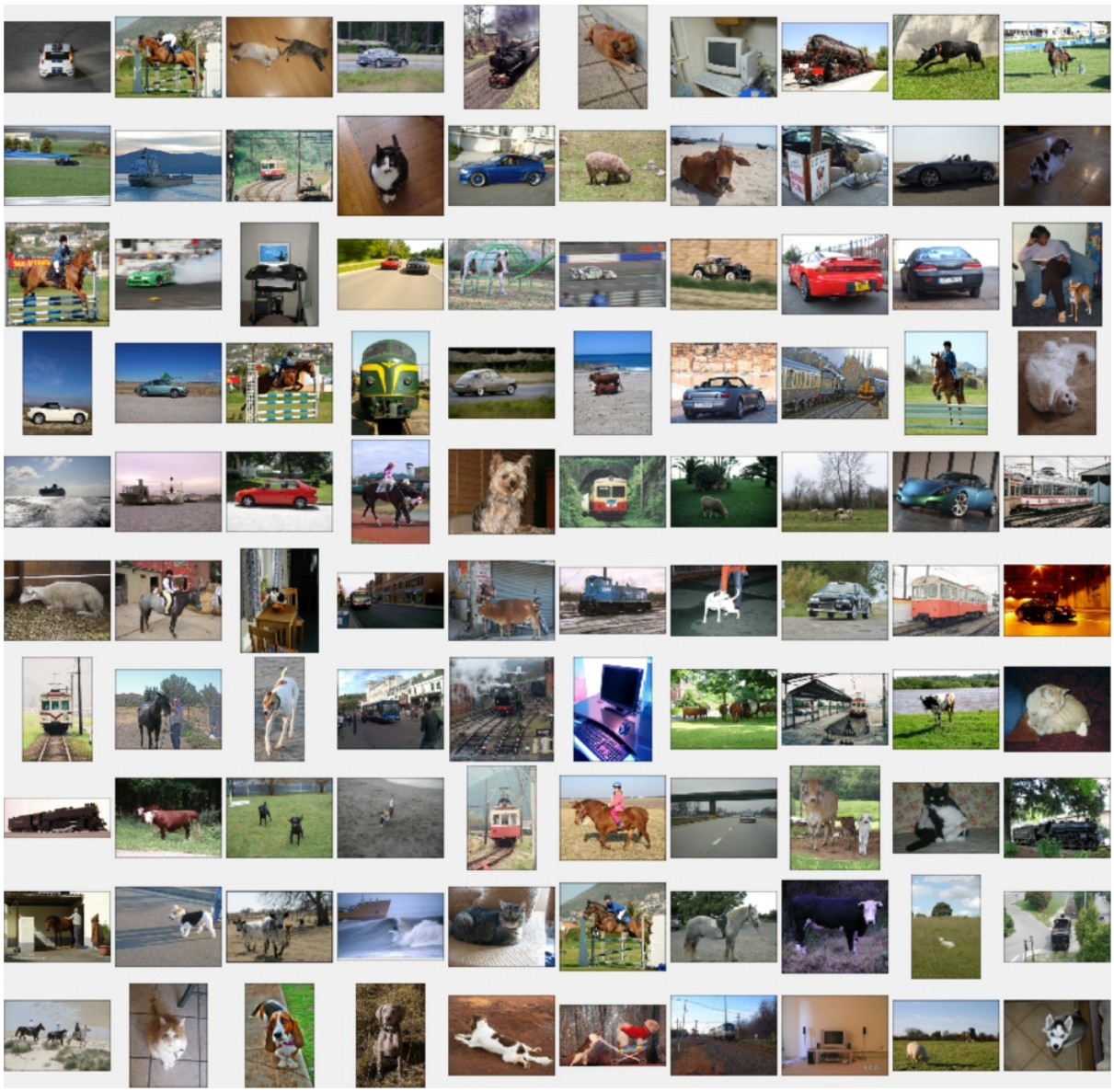

Figure 1: Example images from the PASCAL VOC data set.

3 Pre-processing

Superpixels are locally homogeneous groups of pixels that preserve the object boundaries. Oversegmentation using superpixels can be used as a preprocessing stage that provides an abstraction and simplifies the computation at later stages.

Use the code at `http://ivrl.epfl.ch/research/superpixels` to obtain superpixels for each image in our data set. There are two versions (SLIC and SLIC0). You can use either version. Read the instructions for the version you choose carefully. The output that will be necessary in the rest of the project is a matrix that contains an integer label for each pixel where each label indicates the corresponding pixel's superpixel id. You may need to experiment with the parameters in the code to obtain a meaningful oversegmentation.

After trying the code on several images, fix the parameters and use the same settings for segmenting all images. The output of this part for each image is an integer label image where each pixel stores the id of the region it belongs to.

4 Grouping

The main procedure that you are asked to implement for object localization is based on grouping superpixels. The details of the algorithm are provided in the fourth reference above (Xiao et al., CVPR 2015). Your solution to the grouping problem must be your own implementation of the algorithm proposed in that paper.

The grouping process is based on a complexity adaptive region growing approach. The algorithm groups superpixel sets by considering both low-complexity and high-complexity grouping scenarios with the help of a distance metric where, in each iteration, two superpixel sets with the smallest distance are merged. The complexity of a superpixel set is modeled according to the diversity of the superpixels in the set. A low-complexity scenario corresponds to a case where the mean color can be used to adequately capture the distance between the two superpixel sets. In a high-complexity scenario, while the mean color of the two superpixel sets are significantly different, it is still reasonable to group them into a single set (see Figure 1 in the paper).

Distance computation has three basic components (see the paper for details):

1. Color and texture feature distance: For each superpixel, the color content is represented using color histograms and texture is modeled using gradient orientation histograms. The color and texture distance between a pair of superpixels is computed using the L_1 distance between their corresponding feature vectors.

2. Edge cost: Edge cost measures the edge responses along the common border of the segments. Even though the paper uses the structured edge algorithm for edge detection, you can use other available methods like Canny, Prewitt, etc., instead of the structured edge algorithm.

3. Graph distance: It is aimed to regularize the grouping process by favoring spatially close superpixels. Note that, three-person groups must implement the full graph distance as described in the paper. Two-person groups can use a simplified version.

The definitions of the low-complexity distance and the high-complexity distance that use these basic components as well as the final complexity-adaptive distance that combines the low-complexity and high-complexity distances are given in the paper. You can run the grouping procedure for a fixed number of iterations for all images. You can set the number of iterations experimentally. You can also fix the hyperparameters of the method to the suggested values in the paper: $\eta = 2$, $\sigma = 0.1$, $\lambda = 6$, $\kappa = 1$.

The final step is to rank the object proposals formed by groups of merged superpixels. After computing the score that is based on edge evidence as an indicator for the existence of an object in the region enclosed by each group of superpixels, the groups can be ranked based on their scores in descending order. Each group can also be represented by its bounding box that is computed from the top-most, left-most, bottom-most, and right-most pixels. That is, each group of superpixels is represented by four numbers, x_1 and y_1 for the upper-left corner, and x_2 and y_2 for the lower-right corner like the bounding boxes in the reference data. The final output of the procedure is a list of bounding boxes ranked according to the detection score.

5 Evaluation

The grouping algorithm produces candidate windows that are likely to contain objects. Some of these windows are correct proposals but some may be poorly aligned with the target objects or do not correspond to an object at all. Quantitative evaluation of the performance of object localization methods is done in terms of the overlap between the detected windows and the reference windows in the literature. The "Intersection over Union (IoU)" measure is defined as the number of pixels in the intersection area of the proposed box and the ground truth box divided by the number of pixels in the area of their union. If there are more than one proposal bounding boxes corresponding to a ground truth box, we take the proposal box having the largest IoU. A box is considered a successful detection if its IoU is greater than 0.5.

Given the list of ranked object proposals (bounding boxes), you can choose a threshold and produce a list of detected objects (the boxes that have a score above this threshold). Overall performance can be evaluated using precision and recall that are computed as

$$\text{precision} = \frac{\text{\# of correctly detected objects}}{\text{\# of all detected objects}}, \tag{1}$$

$$\text{recall} = \frac{\text{\# of correctly detected objects}}{\text{\# of all objects in the ground truth}}. \tag{2}$$

Recall can be interpreted as the number of true positives detected by the algorithm, while precision evaluates the tendency of the algorithm for false positives. You can change the threshold and get a new precision-recall pair for each threshold.

You are asked to provide a summary of the detection results in terms of a "precision versus recall" curve computed for a set of thresholds. This curve must be computed by combining the reference and detected objects from all images. You are also asked to provide one plot that shows "recall versus IoU threshold" for each class of objects only by considering the reference objects belonging to that class. The code that performs the evaluation must be your own implementation.

Submit:

You must submit the final report and the developed code through the online form by the deadline given on the course web page.

1. Report
    - Must be readable and well-organized.
    - Provide proper explanation of the details of the approach, the implementation strategies, the results obtained, and the observations made.
    - Provide examples for superpixel segmentation.

- Provide examples for superpixel grouping (as in Figure 2 in the paper) (you should show example groupings when different criteria are used, e.g., only D_{min}, only D_{max}, different combinations of D_{min}, D_{max}, D_{edge}, and D_{graph} like in D_{L} and D_{H}).
- Provide examples for detection results (as in Figures 7 and 8 in the paper). You should show examples for both successful and unsuccessful detections.
- Provide the performance evaluation curves.
- An important part of your report is the discussion where you comment on the performance of your system according to different implementation decisions. You should also analyze the results for individual object types, i.e., which object types were easy and which ones were more difficult, and why you think they were easy/difficult. You can also experiment with the parameters (initially fixed as described above) and discuss how the performance changes.
- All figures must have proper captions and must be properly referenced in the text.
- The report must follow the IEEE Computer Society two-column format as shown on the course web page.
- Each team member should also provide a written description of her/his own specific contributions to the project, and should include this information as an appendix of the report.

2. Code

- Provide well-documented code.
- You are free to use the superpixel detection code given in the SLIC implementation but the codes for grouping and evaluation must be your own implementation (with a few exceptions for low-level operations like histogram or edge detection). Each team member should be prepared to answer questions about all parts.