Problem 1

The goal in this problem will be to explore methods of combining bounding box proposals to classify objects and form predictions about where they are located in a picture. We will be using the Overfeat paper as well as the SUN09 dataset. A set of images from the SUN09 dataset has been provided in your starter code.

(a) First, download a set of object proposals from HContext.html: You will need the detectorOutputsText.tar.gz file. First you will need to gzip and untar the folders. These files are organized into two folders, `train` and `test`, each with subdirectories each specifying an object. Each of these object directories contains many .txt files named after the image to which the object belongs. Each line in the file contains bounding box locations and scores for that image in the format [x_1 y_1 x_2 y_2 `score`].

Your first task is to write a Python script transform this directory into a directory that instead contains a directory for each image in the image set provided, with a set of files for each image containing bounding box information. These files should be labeled {`object name`}.txt. Note: The `os` module is necessary for this task. Another useful module may be the `shutil` module.

Then, for each image, plot the bounding box proposals on top of the image. Use a different color for each bounding box, and symbolize the confidence by varying the thickness of the lines the box is drawn with. (Include a legend specifying which color pertains to which category.)

[Justification for the problem: Sometimes in research and data analysis, datasets are not provided in a nice format and we have to reformat them. Learning how to use useful tools like Python for this sort of task is a useful skill for any researcher who works with data. The visualization portion of the task is important as well for conceptualizing the task individually.]

(b) For each provided image, implement parts $(d) - (g)$ of the greedy merge algorithm for the proposed bounding boxes, as described on pages $10-11$ of the Overfeat paper. You will need to use the bounding box data from both the `train` and `test` folders. In the paper, they discuss the scale s: ignore this aspect of the algorithm, and assume that the bounding boxes of the same class for each image are all the proposals that exist. Be sure to explain and justify your implementation of `match_score`, as only the idea of this algorithm is provided in the paper. Also be sure to assign a confidence score to the resulting merged bounding boxes based on the confidences of the individual bound scores (read pg. 11 closely). Display the merged bounding box for each object class on top of the image as in part (a).

[Justification for the problem: It is important to be able to read a paper and implement algorithms from a simple, sometimes incomplete description and evaluate it for yourself. This question is a simple exercise in implementing a simple algorithm where some information was left out, requiring the reader of the paper to think about and fill in the gaps.]

(c) The task will be to perform an experiment to see how well a common clustering algorithm can distinguish object classes based on bounding boxes. You will apply k-means clustering on each object class of bounding boxes for an image as a replacement to the method defined in the Overfeat paper. Think carefully about how to define a vector for each bounding box: recall that k-means minimizes distortion, or the sum of the squared Euclidean distances between each bounding box vector and the centroid it is closest to. A naïve approach would simply specify each bounding box as a 4-tuple (x_1, y_1, x_2, y_2). What distance are we trying to minimize between bounding boxes? You might use the approach from part (b) as inspiration. Give justification for your bounding box representation. You may find the Python module `scipy.cluster` useful (do not implement your own version of k-means).

[Justification for the problem: Researchers should constantly think about valid approaches to solve problems in papers they read apart from the solutions prevented. This question has the dual benefit of introducing k-means and causing the student to think about the representation problem. Representing something as simple as a bounding box can be done in many ways. It also requires the student to think about the optimization problem at hand, in terms of norms and objectives. The goal here is not for the student to have to implement k-means. Good answers will come up with a sensible vector representation, with coordinates potentially involving the Jaccard similarity (an area-based metric), distance between bounding box centroids, possibly even the confidence scores.]

(d)

(e)

Problem 2

(a)

(b)

(c)

Problem 3

(a)

(b)

(c)