Object Detection Analysis Code (v2)

Derek Hoiem
University of Illinois at Urbana-Champaign

July 31, 2014

1 Overview

Object detection is an important part of computer vision. Typically, the task is defined as providing the bounding boxes that correspond to each instance of an object category, such as "person" or "car" in a collection of images. Hoiem et al. [3] introduced a set of tools to interpret the performance of such object detection systems. We have updated those tools. In this document, we provide a brief description of the tools with some added commentary on their use. Consult the text file included in the code package for details on usage. When using the error analysis tool, please cite the original Hoiem et al. ECCV 2012 paper.

2 Updates

Version 2 of the detection analysis tool includes the following updates:

- Code is rewritten to improve readability and make it easier to adapt to new datasets and detectors. See the text file included in the code package for a more detailed description of how to use the code.
- A new visualization that shows both false positive trends and recall as a function of detection rank is included, which provides an easily interpretable summary of detector performance.
- Weak and strong localization criteria can be easily set, though defaults are the same as previous version.

3 Analyzing detector performance

False Negatives: Detectors may miss objects for a variety of reasons, such as unusual appearance, low resolution, occlusion, perspective effects, and so on. Such misses or low-confidence detections are called false negatives. On the PASCAL VOC 2007 dataset, our original tool characterizes performance for objects with various characteristics using the normalized AP measure. This measure normalizes the computation of average precision to balance for the number of positive examples in the subset of objects under consideration. A complete description and justification of these tools is in the original paper [3].

False Positives: Detectors may also incorrectly assign bounding boxes that do not accurately circumscribe an object from the target category. Such errors are called false positives. We characterize false positives as due to localization error (bounding box overlaps with a target object but not well enough to meet the VOC criterion), confusion with similar objects (e.g., confusing a motorcycle for a bicycle), confusion with dissimilar objects (e.g., confusing a potted plant for a bicycle), and confusion with background (e.g., confusing a pair of glasses or another unlabeled object/surface for a bicycle).

The PASCAL VOC criterion for localization is 0.5 intersection/union of the detection bounding box with the ground truth bounding box. We call this criterion "strong" localization and consider "weak" localization to be 0.1 intersection/union. While it is possible for a detection to accidentally achieve 0.1 intersection/union, we find that confident detections that are weakly localized are often reasonable mistakes, such as providing a box around the cat head but missing the body, or grouping a pair of nearby airplanes into one bounding box. In this update, we experimented with changing the weak localization criterion (specifically requiring that the detection box is mostly contained in the ground truth box) but found that doing so incorrectly characterized detections that appeared to be localization errors as background.

See Hoiem et al. [3] for more detail on the false positive analysis tools.

New Visualization: Our first version of the analysis code (described in the paper) focuses on characterizing the types of false positives, such as what percent of confident false positives are due to localization error vs. confusion with similar objects. However, focusing purely on characteristics of errors can sometimes obscure improvements. For example, we might want to know whether a change to a detection algorithm reduced the confusion with similar categories *and* increased the number of true detections at a given rank (rather than swapping one kind of error for another). Also, one detector that makes very few mistakes (whether "reasonable" or not) may be preferred over a detector that makes many reasonable mistakes.

We introduce a new visualization that summarizes detection performance and types of false positives in a single plot. See Figure **??** for examples. The visualization includes stacked area plots that display the cumulative fraction of detections that are correct or due to localization, confusion with similar, confusion with other, or confusion with background. An improvement in a detector algorithm should push the white area upward. The x-axis is normalized by the total number of non-"difficult" objects (VOC notation for objects that are considered too small or occluded to be required for detection) in the category, so a perfect detector would achieve 100% fraction of correct detections up to 1 on the x-axis, and remaining detections would be split among various types of errors. More typically, the top-ranked detections are highly accurate, and the cumulative fraction of errors increases slowly with detection rank. The visualization also includes line plots of recall with strong localization (solid red) and weak localization (dashed red) criteria on the same axes. The y-axis is recall percentage; the x-axis, the number of detections divided by the total number of objects. For a perfect detector, the recall would match the number of detections until the normalized number of detections is 1; then, recall would be saturated at 1. Note, however, that the plot is semi-log so a perfect recall plot is not a straight line. Typically, a detector's recall will steadily improve until the normalized number of detections reaches 1 or 2, and then most remaining detections will be incorrect, with recall plateauing at 60-90%, depending on the category and detector.

4 Using the Detection Analysis Tools

The following process is suggested:

1. Modify detector/dataset parameters in the analysis code. See text file for details.
2. Run `detectionAnalysisScript.m` after checking settings at the top of the file. This produces figures and tex files.
3. Remove white space around the pdf figures (e.g., with a script in Acrobat for batch processing). The figures `plot_impact_strong.pdf` and `plot_fp_dttrendarea_*.pdf` are the most useful for a quick summary. The impact plots are only available for a subset of categories in the VOC 2007 dataset (because they require additional annotation).
4. Compile the latex document `detectionAnalysisAutoReportTemplate.tex` and view.
5. View `false_positive_analysis_*.txt` and `missed_object_characteristis_*.txt` for further details and view additional qualitative results.

References

[1] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 2010.

[2] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, 2014.

[3] D. Hoiem, Y. Chodpathumwan, and Q. Dai. Diagnosing error in object detectors. In *ECCV*, 2012.

[4] A. Vedaldi, V. Gulshan, M. Varma, and A. Zisserman. Multiple kernels for object detection. In *ICCV*, 2009.

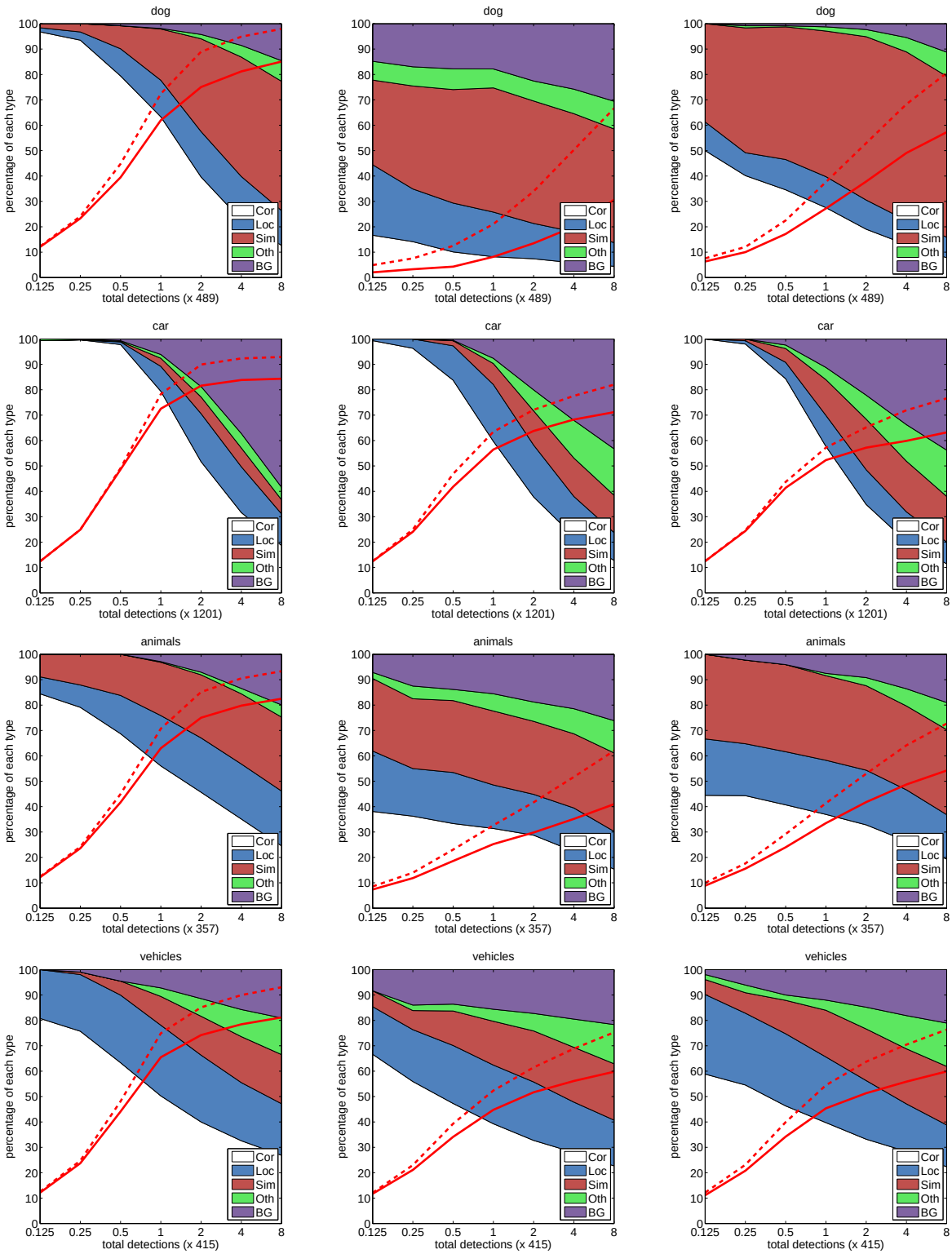

Figure 1: **New visualization of detection performance:** Top two rows: performance comparison on dog and car detection. Bottom two rows: performance summary for animal and vehicle categories. See text for details. Summaries are computed by averaging recall and false positive fractions across categories at points normalized by number of total objects for each category. CNN performs similarly to Vedaldi et al. for dog detection, except that it greatly reduces the confusion with similar categories. CNN's improvement in vehicle performance is less dramatic than for animals at low recall, but it also achieves much higher recall within a limited number of detections.