What If Localization Worked?

Anonymous CVPR submission

Paper ID ****

Abstract

Real world computer vision systems typically have some intrinsic value in their underlying business use. Serving the the right image in a search result ad might be worth $0.001 and counting nuclear particles in material images might be worth $10,000. In general, we want to build systems which produce sufficiently accurate results within a given budget. Although an interaction with human workers can improve accuracy in many algorithms, it also increases the cost. Most computer vision research focuses on purely automated algorithms, arguing that human labor is much too expensive to be included as subroutines in operational algorithms. In this work, we put this argument into perspective by investigating joint algorithms using computers and human labor. In particular, we focus on how different degrees of human involvement effect the algorithm's accuracies. We focus on the representative computer vision task of localization, however, our general methodology can similarly be applied to other tasks, e.g., Object Detection or Image Matching. We introduce several general strategies for combining existing computer vision algorithms with human labor, e.g. (To be filled). We evaluate our results on three reference data sets, i.e., UIUC Cars, Caltech Pedestrian and Street View House numbers, and show how the accuracy of the algorithms scale with varying degrees of human involvement. Finally, we provide an outlook on what computer vision applications become possible, if localization works.

1. Introduction

TODO

2. Background

TODO A place to describe

1. definition of what we understand "localization" is
   (a) Where localization fits into a typical classification pipeline, which helps us motivate why we chose to focus on this problem in particular
   (b) The difference between localization, semantic segmentation, object detection, classification, ...
   (c) State of the art
   (d) data sets
   (e) applications
2. Crowd work:
   (a) What kinds of tasks humans are good at and are not good at?
   (b) previous work on this area (the bubble game, Fei-Fei's bounding boxes, etc!)

TODO We're going to have to be extremely consistent with how we use our terminology. "Detection"? "Localization"? What's the difference? After we decide this (and this section is probably a good place to choose), we should do a Ctrl+F on all instances of both words and change them to match.

3. Data Sets

1. Describe in detail the picked data sets (refer to previous section why they are important.)
2. Describe the measure of accuracy on those data sets

3.1. UIUC Cars

For our first example, we chose the classic UIUC cars dataset, introduced in [1, 5]. This dataset makes a particularly good baseline because it is well-established and because there is no recognition task after the initial detection/localization step. This dataset allows us to measure the performance of a simple single-class open-set detection task. The 550 positive samples in the test set are side views of cars, some of which are partially occluded, and the 500 negative training samples contain natural scenes, various other vehicles, etc.

On this dataset, performance is evaluated using standard precision/recall measures. F-measure and absolute number of false positives is also typically reported. In the single-scale case, the output bounding box is 100×40px, and it counts as a correct detection if the center lies within a certain ellipse of the groundtruth's bounding box center. Since there may be many cars per image; statistics are aggregated over each individual car in the test set, not per image. The original parts-based representation in [1] achieves about 77.58% F-measure.

TODO What is the state of the art?

3.2. Caltech Pedestrian Detection Benchmark

The Caltech Pedestrian Detection Benchmark, introduced by [2], is a challenging pedestrian detection dataset. This set is several orders of magnitude larger than the UIUC Cars set, containing 350,000 groundtruth pedestrian bounding boxes. The research that uses this dataset is summarized by [3].

To evaluate performance, this dataset asks authors to calculate a standard ROC curve of their detection results. However, in the open set "self-driving car looking for pedestrians" scenario, the amount of negative data is potentially infinite, which means plotting TP/TR is not very insightful. Instead, authors report miss rate versus number of false positives per image ("*FPPI*"). This ROC curve is summarized as the "*log-average miss rate,*" computed by averaging the miss rate at nine FPPI values along logarithmically-spaced intervals [3].

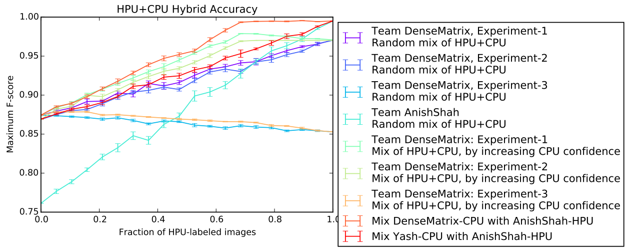
A detection counts as correct if its intersection-over-union score is greater than 0.5, meaning $\frac{\text{AREA}(A \cap B)}{\text{AREA}(A \cup B)} \geq 0.5$, where A and B are the detected and groundtruth bounding boxes respectively.

3.3. Street View House numbers

[4]

Problem: The first paper that uses this dataset [4] only reports classification accuracy, which is separate from the precision/recall tasks. It's unclear to me where they're assuming they have groundtruth segmentation and where they do not start from the crop set. We might be able to spin the story as "Well what if it wasn't already solved? How would the existing algorithms perform if they didn't already have good crops? And how can we minimize this performance drop? This is a hard problem and nobody's thinking about it". Or we could just pick a different dataset. Not sure.

Question for you guys: did anyone find previous work that measures localization accuracy on this dataset?



4. Computer-Human Localization

A place to briefly describe our different HCOMP algorithms

4.1. Removing false positives

TODO Describing HPU algorithms that work by removing bounding boxes that were incorrectly detected

4.2. Removing true positives

TODO Describing HPU algorithms that work by asking the human to add boxes that were never found

4.3. Using crowds to adjust detection thresholds (@Chinmaya and teammates)

TODO

4.4. CCA clustering (Team LiteVision)

TODO

4.5. How much stimulus is necessary? (@boom and teammates)

TODO

4.6. Real-time HPU implementations (Team BlackHatProcrastinators)

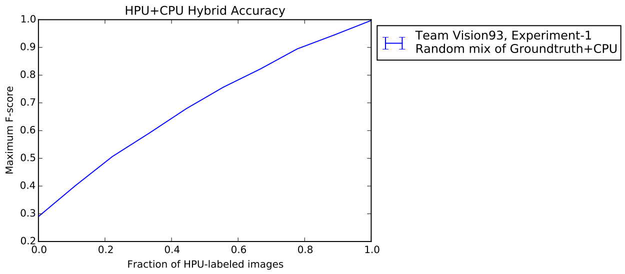
TODO (see commented-out progress)

4.7. (Others ...)

TODO

5. Experiments

A place for experiments and results



5.1. UIUC Cars

TODO Team DenseMatrix: What is the difference between HPU Experiments 1, 2, 3? Please describe exactly what the task was, whether you started from CPU or HPU bounding boxes, etc.

TODO Team AnishShah, Yash's team: Please include confidence values for each bounding box in your CPU results. That way, we can ask the human worker to label the least-confident images first, which will help quickly improve performance

TODO Team DenseMatrix: Thanks for including confidence values, but they are taken from the set $\{1, 2\}$. Is there any way to get confidence more fine-grained?

5.2. Caltech Pedestrians

TODO Team Vision93: Thanks for including bounding boxes, but we note that the groundtruth has 4,024 images and your CPU results list bounding boxes for 1,400 images. Is this intentional? This would mean that the CPU algorithm starts at $\approx$ 30% recall.

TODO All teams: Does anyone have bounding boxes for Caltech?

5.3. Street view house numbers

TODO Everyone: It looks like we may have to come up with our own strategy to evaluate SVHN.

6. Conclusion

References

[1] S. Agarwal and D. Roth. Learning a sparse representation for object detection. In *Computer VisionECCV 2002*, pages 113–127. Springer, 2002. 1, 2

[2] P. Dollár, C. Wojek, B. Schiele, and P. Perona. Pedestrian detection: A benchmark. In *CVPR*, June 2009. 2

[3] P. Dollár, C. Wojek, B. Schiele, and P. Perona. Pedestrian detection: An evaluation of the state of the art. *PAMI*, 34, 2012. 2

[4] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y. Ng. Reading digits in natural images with unsupervised feature learning. In *NIPS workshop on deep learning and unsupervised feature learning*, volume 2011, page 5. Granada, Spain, 2011. 2

[5] A. A. Shivani Agarwal and D. Roth. Learning to detect objects in images via a sparse, part-based representation. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 26, pages 1475–1490. 2004. 1