MVA 2008 - 2009

Object recognition and computer vision

Final project: Object category detection and localization

Jean Ponce, Cordelia Schmid and Josef Sivic

New! Due date: February 1st 2009

(adapted from R. Fergus, A. Torralba and L. Fei Fei)

The goal of the final project is to experiment with, and improve detection performance of a simple object detector based on boosting. The project will focus on two object categories — side view's of cars and near frontal views of computer screens.

The final project will have about 1.5x the weight of a regular assignment when counting towards you final grade from the class.

Homework materials

For the project you will use the Matlab code and dataset provided by the ICCV'2005 short course on recognizing and learning object categories.

Download the following updated functions plotBoundigBox.m and runDetector.m and use them instead of the corresponding functions provided at the ICCV'2005 short course webpage.

Directions

1. For downloading and running the code follow the instructions at the ICCV'2005 short course webpage.
2. Run the toy classification example with hand-clicked 2D dataset of red-circles and green squares. For details of the GentleBoost algorithm you can have a look at the paper by Friedman, J. H., Hastie, T. and Tibshirani, R., "Additive Logistic Regression: a Statistical View of Boosting." (Aug. 1998) (up to and including section 3).
3. Run the pre-trained object detector for both computer screens and cars. Make sure you understand the code. For details of the algorithm and image features have a look at the short course slides (slides 1-56). Try using different number of weak learnes (e.g. 5, 20). How does the detection performance (measured by the precision-recall curve)change with different number of weak-learners used? You can change the number of weak learners by changing the variable `NweakClassifiers` in function runDetector.m. For computing precision-recall curves use

function LMrecallPrecision.m instead of function LMprecisionRecall.m used by default in the runDetector.m. You will need to modify runDetector.m to output the detected bounding boxes in all test images in the appropriate format specified in LMrecallPrecision.m.

4. Train your own car and computer screen detectors. Again, follow the instructions at the ICCV'2005 short course webpage and make sure you understand the code. For faster feature extraction, you can limit the number of extracted weak learners (features).

5. The next goal is to implement an extension of the detector and improve its performance measured by the precision-recall curve. Below are some ideas for possible extensions. You should implement at least one of the extensions below.However, you are free to implement any other extension you come-up with, such as extraction of different features (e.g. the Histogram of Oriented Gradients—HOG, described here, chapter 4) or using a different type of detector such as a support vector machine (SVM). You are free to use any code available on-line.

   Suggestions for extensions of the object detector:

   a. Implement a multi-scale version of the detector to detect both small and large objects in images. This can be implemented by running the detector on several re-scaled versions of the image as described on the ICCV'2005 short course webpage. You will need to choose a range of scales and scale step, You might also need to implement a multi-scale non-maxima suppression algorithm to merge detections from multiple scales.
   b. Implement enlargement of the training data by perturbing the training examples with small geometric transformations (shift of few-pixels, small scale change) as described in the first paragraph on page 8 in *I. Laptev*, Improving object detection with boosted histograms, Image and Vision Computing, 2009, (in press).
   c. Implement the mean-shift based non-maximum suppression algorithm described in section 5.2 (page 57) of *N. Dalal*, Finding people in images and videos, PhD thesis, Institute National Polytechnique de Grenoble, 2006. You can combine this extension with extension (a.) above or just implement the non-maximum suppression on a single scale.

What to hand in

For the final project, you do not need to hand-in any code. Only hand-in a 2 page report in a format of a submission to the IEEE Computer Vision and Pattern Recognition conferenc (CVPR) . Use the latex or word templates provided at the CVPR Author Guidelines webpage. Note, that your are asked to produce only a 2-page double-column report (in contrast, a standard CVPR submission is up-to 8 pages). The report should include the following:

· Title of the assignment, your name and email address.

· You should spend about 1 page on describing what you did. If you are extending the provided GentleBoost detector briefly summarize (0.5 page = 1 column) in your words the image features and the boosting algorithm. This is to make sure you understand the provided code. Use the reminder of the space to describe the implemented extension(s). You might want to illustrate your extension with a figure showing, e.g. detections over multiple scales for extension (a.), examples of the perturbed training data for extension (b.), or detections before and after the non-maximum suppression for extension (c.).

· You should spend about 1 page on quantitative and qualitative evaluation of your extension of the detector. For quantitative evaluation, you should show and discuss precision-recall curves for both objects (computer screens and cars) for the basic detector and also the extended detector. For qualitative evaluation, show and discuss examples of true positives, false positives and missed detections. Show and discuss and example of an object instance, which was not detected before (e.g. a car missed because it was too small/big) and is fixed with your extension.

.