Assignment Project #4 “Object Recognition Contest”

Due date: Tuesday, 05/06/2014, 5:00pm

Description:

1. Download the Train/Validation/Test Images from the course website. (We are using Graz02 dataset for your final project.) There are three object classes, i.e., bikes, cars, people, plus background images.
2. You need to use the Train dataset to train the classifiers for recognizing the three object categories; use the validation dataset to tune any parameters; and fix the parameters you tuned and report your final classification accuracy on the test images.
3. Please follow these steps:
    a. **Feature extraction**: you can use the following method
        i. Sample fixed-size image patches on regular grid of the image. You can use a single fixed image patch size, or have several different sizes.
        ii. Sample random-size patches at random locations
        iii. Sample fixed-size patches around the feature locations from, e.g., your own Harris corner detector or any other feature detector you found online. Note if you used code you downloaded online, please cite it clearly in your report.
    b. **Feature Description**: It is recommended that you implement the SIFT descriptor described in the lecture slides for each image patch you generated. But you may also use other feature descriptors, such as the raw pixel values (bias-gain normalized). You don’t need to make your SIFT descriptor to be rotation invariant if you don’t want to do so.
    c. **Dictionary Computation**: Run k-means clustering on all training features to learn the dictionary. Setting k to be 500-1000 should be sufficient. You may experiment with different size.

d. **Compute Image Representation**: Given the features extracted from any image, quantize each feature to its closest code in the dictionary. For each image, you can accumulate a histogram by counting how many features are quantized to each code in the dictionary. This is your image representation.

e. **Classifier Training**: Given the histogram representation of all images, you may use a k-NN classifier or any other classifier you wish to use. If you wanted to push for recognition, you mean consider training an SVM classifier. Please feel free to find any SVM library to train your classifier. You may use the Matlab Toolbox provided in this link (http://theoval.cmp.uea.ac.uk/~gcc/svm/toolbox/).

f. **Recognition Quality:** After you train the classifier, please report recognition accuracy on the test image set. It is the ratio of images you recognized correctly. Please also generate the so-called confusion matrix to enumerate, e.g, how many images you have mis-classified from one category to another category. You will need to report results from both validation and test dataset to see how your algorithm generalize.

g. **Important Notes:** Please follow the train/validation/test protocol and don't heavily tune parameters on the test dataset. You will be asked to report recognition accuracy on train, validation, and test dataset, respectively. Tuning parameters on the test dataset is considered to be cheating.

What to turn in?

There are several things I would expect from your final submission. You need to submit all your source code with a detailed ReadMe.txt file on how to run them. You need to also submit your presentation slides, and your final report.

Your report needs to be as detailed as possible, which should be between 8 to 10 pages. Please try to look into your recognition errors and discuss why you can not make it to be correct and your thoughts on make it better. You should also submit an one-page slide to summarize the main results of your project.

Please name your folder as well as the zip file as

[yourfirstname]_[yourlastname]_FinalProject.zip

If your code cannot run, you may also turn in it with more detailed comments on what you did and tested in your code.

Bonus Points:

No bonus points will be granted. But additional efforts will be rewarded and normalized across class.

Grade: 40%

Your grade will be distributed as the following. Your final presentation accounts for 5 points. Your system will be competing with others in the class, and the champion will get 15 points, with a decrease of 0.5 point for each following rank. Your final report will account for 20 points.

You will be penalized if your code cannot run. You will also be penalized if you did not create the hybrid image in the right way.

Late submission policy applies universally with no exception.

If you have a compelling excuse, you must inform me at least 2 days before the due date. I don't accept excuses such as "I am overloaded by other courses".