Exercise for MA-INF 2213 Computer Vision SS23
10.05.2023
Submission deadline: 28.05.2023 Exercise: 14.06.2023

Important: Please use **Python 3.8** for your solutions. You are **not** allowed to use any additional python modules beyond the ones imported in the template. Otherwise you won't get any points.

Grading: Submissions that generate runtime errors or produce obviously rubbish results (e.g. nans, inf or meaningless visual output in future exercises) will receive at most 50% of the the points.

Plagiarism: Plagiarism in any form is prohibited. If your solution contains code copied from any source (e.g. other students, or solutions from the web. This includes ChatGPT!), you will receive **0 points** for the entire exersice sheet.

Submission: You can complete the exercise in groups of two, but only one submission per group is allowed. Include a *README.txt* file with your group members into each solution. Points for solutions without readme file will only be given to the uploader.

1 Support Vector Machines (SVMs)

1.1 Extracting HOG Features:

In this task, you will extract the HOG features and train the SVM yourself. Specifically, in order to train your own SVM, you have to extract HOG features from a set of positive and negative training images, i.e. images that contain the target object (person) and images that do not.

1. Extract HOG features from all the positive training images using HOGDescriptor::compute(). You need to crop each image to a size of 64 x 128. The training images are located in Sheet03/task_data. In total, you will extract one HOG descriptor for every positive image.

2. Now extract HOG features from the negative training images. Since these images are bigger (as described in [1] under "Methodology"), extract 10 patches (64 x 128) at random locations from each image, compute their HOG features and use them as your negative training samples.

3. All training features together with their corresponding label information should be stored and then passed to the SVM object for training.

(4 points)

1.2 Train the SVM:

In case of a linear SVM, the training algorithm finds the hyperplane which best separates the positive training samples from the negative ones. In many cases,

the data contains outliers or is not linearly separable, and then you allow the classifier to misclassify some samples for a cost such that you can increase the margin. The cost of this error is determined by how far the sample is from the margin. The value of the parameter C which you have to choose during training determines the trade-off of reducing the margin versus the number of misclassifcations. Let us now examine the influence of the value C on the results.

1. Choose C = 0.01, C = 1 and C = 100, i.e. train three different SMVs (using the HOG features extracted in task 2).
2. Use all 3 SVMs to classify the given test images. For the positive images, there is only one HOG desciptor per image. For the negative images, there are multiple HOG descriptors, depending on the size of the input image. Make sure your patches have dimension 3780.
3. Create a file containing the confidence scores (distance to the margin) for each training sample.

(8 points)

1.3 Plotting the results:

In general, there is a trade-off between precision and recall, since increasing one usually happens at the cost of the other. In this task, you are going to examine the influence of the choice of C by presenting the results as precision-recall plots. In this plot, you vary the hit-threshold which gives a set of (recall, precision) value pairs representing the (x, y) values on the graph. The formulas for these two metrics are given by:

$$precision = \frac{TP}{FP + TP} \tag{1}$$

$$recall = \frac{TP}{FN + TP} \tag{2}$$

where TP = true positives, FP = false positives, FN = false negatives. You will get different classification results from the SVM by changing the threshold.

(4 points)

1.4 Different kernel:

Use other SVM kernel (like Histogram intersection kernel) and repeat the above process.

(2 points)

References

[1] Dalal N. and Triggs B., Histograms of Oriented Gradients for Human Detection. CVPR 2005