

CS376: Computer Vision: Assignment 4

Due: Apr. 5th, 11:59 PM

Instruction: You must answer the Short Answer Problems. You may choose either Programming Track(Image Classification) or Theory Track. 100 points in total.

1 Short answer problems [30 pts]

1. What is the relation between image classification and object detection? Please give two concrete examples on how they are related (e.g., one is applied to solve another)?
2. Please compare semantic segmentation, instance segmentation and object detection. Describe the similarities and differences.
3. Describe at least two plausible neural network designs for the task of semantic segmentations.
4. What are the applications of generative models in solving core computer vision tasks.
5. Discuss the differences between convolution neural networks and deconvolution neural networks.

2 Image Classification (70 points)

Our task is to perform image classification using the CIFAR-10 dataset, which consists of 50K training images and 10K testing images. We will experiment with two categories of methods: non-parametric methods and parametric methods. Our goal is to understand the performance of each method on this dataset. Our specific aims are:

- Compare the performance between K-Nearest Neighbor classifier and Adaptive Boosting classifier;

- Analyze the performance of these two methods when varying the size of the training set and the size of the testing set;
- Perform cross-validation to study the hyper-parameters of each method;
- Try different feature representations.

Step 0: Download the CIFAR-10 data set. The dataset can be downloaded from <https://www.cs.toronto.edu/~kriz/cifar.html>.

Step 1 (15 points): K-nearest neighbor classifier. Implement the k-nearest neighbor classifier with $k = 10$. Each image of size $32 \times 32 \times 3$ is represented as a vector with dimension 3072. Please report an overall accuracy as well as a confusion matrix. **Hint:** For saving computation cost, it is recommended that you can use a low-dimensional projection matrix $P \in \mathbb{R}^{m \times 3072}$ to reduce the feature dimension of each image. This projection matrix P may be computed using principal component analysis (via SVD) among the training images.

Step 2 (30 points): Adaptive Boosting classifier. For this part, you are asked to implement an AdaBoosting method to classify the images described above. We show follow the following two major steps:

- Weak feature computation (10 points). We will implement the features used in Viola-Jone's face detector, i.e., the Haar Features. <https://www.cs.cmu.edu/~efros/courses/LBMV07/Papers/viola-IJCV-01.pdf>. See Section 2 of the paper for the detailed description. Basically, you first pre-compute an integral image from each image, you then use this integral image to generate Haar features. For efficiency, it is suggested that you use 1K-2K Haar features rather than all the possibilities (hundreds of thousands). This can be down by choosing a random subset. Note that this random subset has to be consistent across all the training images. Moreover, the image dimension in CIFAR-10 is slightly bigger, i.e., 32×32 versus 24×24 .
- (10 points) Train an Adaboost classifier for each class, i.e., using the one-versus-all mode. Since for each classifier, the number of negative instances is nine times more than the number of positive instances, you have to balance the initial weight. Please refer to slide 10 of Lecture 19 regarding the Adaboosting algorithm.
- (10 points) Combine these one-versus-all classifiers into a multi-class image classifier. Please report the overall accuracy.

Step 3 (5 points): Comparison. Please compare the performance of the K-nearest neighbor classifier and the Adaboosting classifier. Which classes does K-nearest neighbor do better and which classes does Adaboosting do better, and why? How about running time?

Step 4 (10 points): Cross-validation. Perform 5-fold cross-validation for the K-nearest neighbor classifier. Report the optimized hyper-parameter K and the corresponding confusion matrix.

Step 5 (10 points): Cross-validation II. Perform 5-fold cross-validation for the AdaBoosting classifier. Report the optimal value for the number of weak classifiers.

Tips: Please check piazza post for tips.

Submission instructions:

Create a single **zip** file so submit on Canvas that includes

- Your well-commented code, including the files and functions named as specified above.

- A **PDF** writeup of your results with embedded figures where relevant.

Please do not include any saved matrices or images etc. within your zip file.