Visual Recognition

Classical Approaches

Scene Classification Project

The goal of this project is to implement a scene recognition approach based on bags of visual words. Approaches based on bags of visual words (BoW) build visual vocabularies where each word describes an image feature, thus, an image can be described by a vector that keeps records of words occurrences.

Different descriptors of image features, have been used in the context of scene recognition:

1.- Histograms of oriented gradients (HoG). [skimage.feature.hog()]

Dalal, N. and Triggs, B. Histograms of Oriented Gradients for Human Detection. IEEE Conference on Computer Vision and Pattern Recognition, 2005.

2.- Local binary patterns (LBP). [skimage.feature.local_binary_pattern()]

T. Ahonen, A. Hadid and M. Pietikainen. Face Description with Local Binary Patterns: Application to Face Recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2006.

3.- Census transform histograms (CENTRIST)

Wu, Jianxin and James M. Rehg. "CENTRIST: A Visual Descriptor for Scene Categorization." IEEE Transactions on Pattern Analysis and Machine Intelligence , 2011.

4.- Scale invariant feature transform (SIFT). [cv2.SIFT.create()]

David G. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. International Journal of Computer Vision, 2004.

5.- Gabor filter features. *[skimage.filters.gabor_kernel()]*

B. Dong, G. Ren. A New Scene Classification Method Based on Local Gabor Features. Mathematical Problems in Engineering, 2015.

5.- GIST. [https://github.com/imoken1122/GIST-feature-extractor]

Aude Oliva, Antonio Torralba. Building the GIST of a scene: The role of global image features in recognition. Progress in Brain Research 155, 23-36. Progress in brain research, 2006.

In the context of scene classification, neither the use of different scales nor rotation invariance have proven to be useful. Furthermore, dense sampling (based on grid points) has been demonstrated to be advantageous over sparse sampling (based on key points). In this project you need to decide the grid-cell size and related to it the size of the window to calculate the descriptors.

1. Image description based on BoW

These approaches firstly need to build a vocabulary of visual words, which will represent the similar local regions across the images in the training dataset. It can be formed by clustering with k-means ([sklearn.cluster.KMeans()]) the many thousands of local feature vectors from the training set. Each cluster centroid will represent a visual word. Once the vocabulary is completed, any image in the target domain could be described by its histogram of visual features ocurrences.

Instead of storing hundreds of densely sampled feature descriptions for each image, BoW approaches count how many feature descriptions fall into each cluster (or vocabulary word).

If, for instance, the vocabulary size is fixed to 100 visual words, then the bag of words representation will be a histogram of 100 dimensions where each bin keeps records of how many times a feature was assigned to that cluster. The histogram should be normalized

(frequency in [0,1]) so that image size does not dramatically change the bag of features magnitude.

Fei-Fei, Li and P. Perona. A Bayesian hierarchical model for learning natural scene categories. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2005.
(Vocabulary building and histogram description for one feature 30 points)

2. Classification

There are numerous methods to learn linear classifiers, among them the **support vector machines**. This classifier is inherently binary, so to decide which of N categories a test case belongs to, you will need to train N binary 1-vs-all SVMs. Fortunately, [sklearn.svm.LinearSVC()] and [sklearn.svm.SVM()] does mutliclass classification in a single function call. These functions automatically infer that they need to do multiclass classification if they are given a training dataset with multiple output labels.

For high-dimensional binary classification tasks, a linear support vector machine is a good choice, but a different method can be used.
(Classifier 30 points)

3. Assessment

You should now measure how well your BoW representation works when paired with a classifier:
1.- Accuracy. [sklearn.metrics.accuracy_score]
2.- Confussion matrix. [sklearn.metrics.confusion_matrix]
(Results and discussion 20 points)

4. Extra credit

There are many design decisions and free parameters (number of clusters, sampling density, sampling scales, feature descriptor parameters, etc.). An analysis of their influence could be done to tweak them and get an enhanced performance.
1.- Add a validation set to your training process to tune learning parameters. This validation set could either be a subset of the training set or some of the otherwise unused test set. (10 points)
2.- Experiment with different vocabulary sizes (number of clusters) and report performance. (10 points)
3.- Consider the combination of 2 features. You could use 2 descriptors based on BoW, or one based on BoW and another one for global structure. (30 points)
4.- The best reported accuracy for a specific combination of features. Results have to be presented in public. (10 points)

5. Dataset

You can use this dataset of indoor scenes: https://web.mit.edu/torralba/www/indoor.html
The database contains 67 Indoor categories, and a total of 15.620 images. The number of images varies across categories, but there are at least 100 images per category.

Choose 10 categories (bathroom, bakery, bookstore, casino, corridor, gym, kitchen,

locker_room, subway, winecellar) and the first 150 images per category. Among them, take the first 80 for training, the next 20 for validation, and the last 50 for testing.

6. Assignment Assessment

Groups of 2 student: grade based on 0-100 scale
Groups of 3 students: grade based on 0-130 scale

7. Hand-in Documentation

- Jupyter (or Google Colab) notebook, or pdf document, including all the code, comments on implementation decisions, results and assessment tables/graphs, and comments on them.