Class Project

Scene Classification

Please present a report with all your answers, explanations, and sample images or plots. Submit also a soft copy of the source code and binaries used to generate these results. Please note that copying of any results or source code will result in ZERO credit for the whole project.

In this project you will implement a scene classification system. The goal is to identify the type of scene in the input image from fifteen different scene types. The algorithm you will be implementing is based on the Bag of Words (BoW) model (Szeliski Ch. 14.4.1) and an extension to encode some spatial information in the histograms.

Description

The details of the algorithm is in this paper: "*Beyond Bag of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories*" by Lazebnik et al. The main idea of the algorithm is encoding some coarse spatial information in the histogram of visual words.

Bag of Words

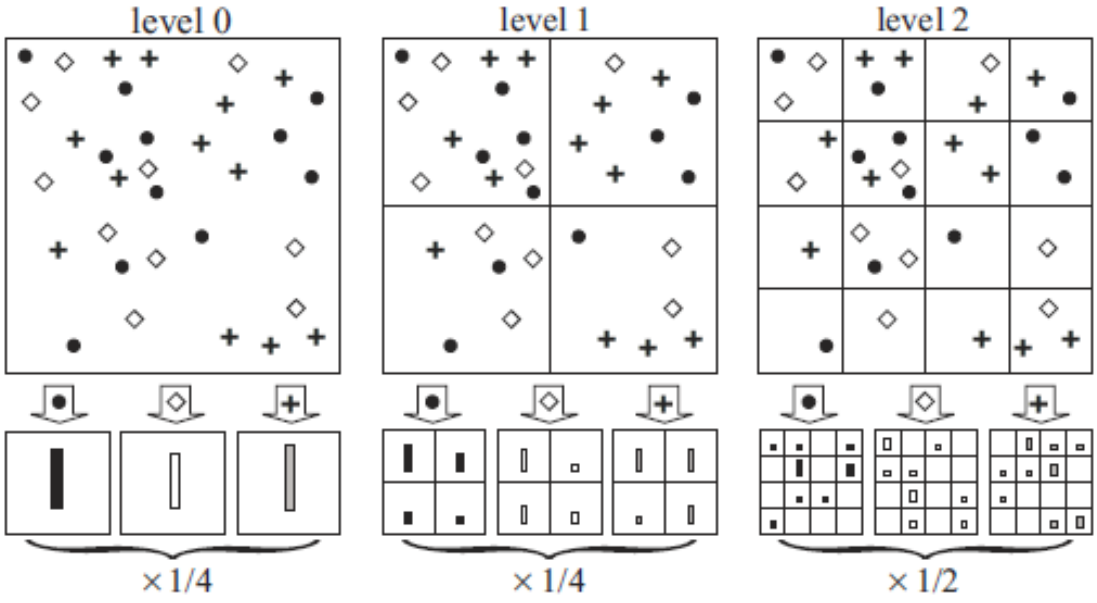
Visual words is an anlogy to text processing. In images we don't have words, so we devise "visual words". They are computed by collecting a large number of features (e.g. SIFT features) from the set of training images, then clustering them into a number of clusters (usually using K-Means). Then for every image, we extract features from the image and find the nearest cluster center (visual word) for each feature. The image is then represented as a histogram of these visual words (i.e. how many occurrences of each visual word).

The problem with BoW is that all the spatial information (location and scale) of these features is lost once the histogram is built. Spatial Pyramid Matching tries to solve this problem.

Spatial Pyramid Matching

It tries to encode some spatial information in the BoW approach. The basic idea is to divide the image into a number of spatial bins, and build a histogram for each block. This is repeated for different resolutions to build a pyramid, and the weights of these histograms is different for each pyramid level.

For example in this figure there are three different visual words (cross, diamond, circle) and three different pyramid levels. At level 0, there is just one block, and one histogram. At level 1, there are three spatial blocks and one histogram for each, … etc.



Training

For training the algorithm, you will need to extract features, build visual words, and then train classifiers. The steps are:

1. Divide the set into training and test set.
2. Extract features (e.g. SIFT) from the images in the training set.
3. Cluster the features (or a subset thereof) in the training set using K-Means to get the visual

words.

4. Compute the spatial pyramid histogram for each training image:
    1. For each feature in the image, find its closest visual word
    2. Divide the image into the pyramid blocks
    3. Build a histogram for each block in each level
    4. Compute the final histogram by appropriate weighting of these histograms
5. Build a different classifier for each scene type. The classifier type to be used is the Support Vector Machine (SVM). Every type of scene will have its own SVM, where the positive examples come from this scene, and the negative examples come from all the other scene types. This technique is called One-Vs-All (OVA). Use the basic Linear SVM. Some C++ implementations include LibSVM and SVMLight, and you can also use whatever other implementation.

Classification

To classify an image:

1. Extract its spatial pyramid histogram as above
2. Run it through each SVM. The scene whose SVM produces the maximum score wins, and the image is classified as of that scene type.

Performance Measure

To report performance, you need to classify every test image and find out if it's correctly classified or not. You need to build a "*Confusion Matrix*", which is a CxC matrix (in this case 15x15 matrix) such that the entry at the i^{th} row and j^{th} column is the number of test images belonging to class i that were classified as class j n_{ij} divided by the number of test images of class i N_i i.e. $c_{ij}=n_{ij}/N_i=n_{ij}/\sum_j n_{ij}$.

The diagonal entries of the matrix are the precision of each class i.e. the percentage of images in class i that are correctly classified. The **performance measure** is the average of the per-class classification rate i.e. the average of the diagonal entries of the confusion matrix.

For training, choose **100** images at random **per scene**, and the rest for testing. Since you have this randomness, you need to repeat the training/classification process several times (only **five** times) and report the average performance and standard deviation over these five runs.

Note: The process of computing the visual words is done every time from images in the training set, so as not to bias the performance measures.

Requirements

1. All implementations should be in C++ with OpenCV under Linux.

2. An implementation of the Bag of Words algorithm:
    1. Visual word generation using K-Means.
    2. Histogram generation using the visual words.
3. An implementation of the Spatial Pyramid Histogram.
4. You can use any type of features (e.g. SIFT) you like.
5. To report results, repeat training/testing **five** different times, and compute the average performance of these five runs. For each run, choose randomly 100 training images per scene type.
6. The goal of the project is to obtain the highest classification accuracy. To this end, you should explore the different components of the algorithm and see what works better and what does not. Namely, you need to explore each of the following:
    1. The type of feature to use (SIFT, HOG, ….). Using one type of feature or a combination of features.
    2. How to sample features from the image i.e. using interesting locations (interest point operators) or using a dense grid over the image (extracting features say every 4 pixels across the whole image).
    3. The number of visual words to use. Typical values to try are: 100, 200, 400, 1000.
    4. The height of the spatial pyramid. Typical values are 0 (basic bag of words), 1, and 2.

Deliverables

1. Running implementation of the algorithm.
2. A project report that describes all the details of your implementation, the different approaches you explored, and results you found. The report should include the following:
    1. Introduction
    2. Project Description
    3. The different parameters/directions explored
    4. Results and Discussion
    5. Conclusion
3. A short **10** minutes presentation. Make sure you don't exceed 10 minutes, it will sharply timed.

Resources

- Svetlana Lazebnik, Cordelia Schmid, and Jean Ponce. Beyond Bags of Features: Spatial

Pyramid Matching for Recognizing Natural Scene Categories. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2006.

- Fifteen Scene Categories Dataset.

Acknowledgments

This homework is adapted from James Hays.