Scene Recognition – SIFT and Bag Of Words Model

CS 5335 – Project Report

The project aims to evaluate and implement the Bag-Of-Words model to areas image scene recognition and build up on its limitations for a more robust detection algorithm with optimized parameters. The project uses Scale Invariant Feature Transform (SIFT) to recognize and describe image features and then uses the Nearest Neighbor Classifier to classify a set of 1500 images into 15 scenes (as introduced in **Lazebnik et al. 2006**). The project further explores the zero level spatial pyramid (by Lazebnik) and SVM classifier to increase the accuracy of the scene detection.

Project Link:
https://github.ccs.neu.edu/biswarajkar/robotics/tree/master/project/html/index.html

BISWARAJ KAR

Spring 2016

College Of Computer & Information Science

Northeastern University

Problem description:

The goal of this project is to classify images based on the objects contained in the image or the global scene depicted by the image. To accomplish scene detection on a large variety of images (1500 grayscale images, ~300x300), the project evaluates the task of scene recognition with a relatively new technique, *the bags of quantized local features and linear classifiers learned by nearest neighbor classifier and support vector machines*.

Bag of words models is a popular technique applied to image classification, and is inspired by models used in natural language processing and information retrieval. In this model, a text or document is represented as a bag of its words, disregarding grammar or the order of the words. In the realms of Computer Vision, the same model is applied by treating image features as words. The important feature here is that the model ignores word arrangement (spatial information in the image) and classifies based on a histogram of the frequency of visual words. The visual word "vocabulary" is first generated by extracting a large number of local features and then by clustering the features together into visual words. The concept is introduced and described in the book **"Computer Vision: Algorithms and Applications" by Richard Szeliski**.

In this project, we will classify scenes into one of 15 categories by training and testing on the 15 scene database (introduced in **Lazebnik et al. 2006**. The project implements the baseline method the paper discusses (equivalent to the zero level pyramid) and not the more sophisticated spatial pyramid.

Example scenes from of each category in the 15 scene dataset. Figure from Lazebnik et al. 2006.

Algorithms:

The principal approach of the problem of scene recognition using visual words can be divided into the following steps and for each step, an algorithm is chosen based on the features and optimality as illustrated later in this page:

Step 1: Feature Extraction using training images

- Extract local features, build descriptors and enlist/learn "visual vocabulary".
- Possible algorithms for feature detection:
    a. **Scale Invariant Feature Transform (SIFT)[1]**
    b. Harris Detector (using image gradient and Eigen values).
    c. Dense Detector (using every nth pixel as interest point).
- Possible algorithms for choice of descriptor for image features:
    a. SIFT[1,8]
    b. Histograms of Oriented Gradients for Human Detection (HOG)[11]
    c. GIST Descriptor[7]

Step 2: Quantize local features using visual vocabulary

- Cluster features extracted in previous step and represent images by frequencies of "visual words" or cluster centroids.
- Possible algorithms to evaluate:
    a. K-Means Clustering[10] (Exclusive Clustering)
    b. Hierarchical clustering
    c. Mixture of Gaussians (Probabilistic Clustering)

Step 3: Classification of Test images

- Learn a decision rule (classifier) assigning bag-of-features representations of images to different classes.
- Possible algorithms to evaluate:
    a. Nearest Neighbor
    b. Support Vector Machines

The choice of SIFT[1,8,9] is by consideration of the following factors:

- Locality: features are local, so they are robust to occlusion and clutter.
- Distinctiveness: individual features can be matched to a large database of objects.
- Quantity: many features can be generated for even small objects.
- Scale Invariant: features are invariant to scale and rotation as orientation is relative.
- Descriptor length: fixed length vector (128) irrespective of number of detections.
- Very successful in classifying images according to the objects they contain.

Code & Results:

Bag of SIFT representation and nearest neighbor classifier

This section basically consisted of two parts where first we build the vocabulary (best with size =400). I observed that just using step was not good enough as some times important feature is captured not just points in step but continuous points. I thus apply a bigger step size first and then use some randomized permutation which captures continuous points in a more intuitive way. Once features are extracted, then we apply K-means to get representative cluster centers.

```
%Number of key-points to consider for each image
No_of_sample_descriptors = 9;

%Step size for SIFT feature detection
step_size=10;

%Total Number of images to process
No_of_images = size(image_paths, 1);

%Store all descriptors in a matrix of size 128 x [1500 x No_of_sample_descriptors]
descriptors = zeros(128, No_of_images * No_of_sample_descriptors);

%% Get SIFT features/descriptors for each image
for counter=1:No_of_images
    % Load each image from the training set
    img = im2single(imread(image_paths{counter}));
    % Get features for each image
    [~, features] = vl_dsift(img, 'Fast', 'Step', step_size);
    tot_no_of_features = size(features,2);
    % We will randomly sample the descriptors from each image instead of taking
    % all descriptors to save memory and speed up the clustering.
    random_features=randi([1 tot_no_of_features],1,No_of_sample_descriptors);
    % So, we take the any of the 'No_of_sample_descriptors' descriptors per
    % image at random using the randi function to generate
    %'No_of_sample_descriptors' random integers from the uniform distribution
    %between 1 and Total number of features.
    descriptors(:, No_of_sample_descriptors * (counter-1) + 1 :
    No_of_sample_descriptors * counter) ...
                        = features(:,random_features);
end
```

```
%% K-Means Clustering/Quantization
%{
Now that we have tens of thousands of SIFT features from many training
images, we will need to cluster them with kmeans/LLOYD algorithm. The
resulting centroids are now our visual word vocabulary.
Matlab has a build in kmeans function, but it is slower, so I use vl_kmeans
[centers, assignments] = vl_kmeans(X, K)
%}
[centroids, ~] =
vl_kmeans(single(descriptors),vocab_size,'Initialization','RANDSEL');
vocabulary = centroids';
```

Classifier Codes (Nearest Neighbor):

Nearest Neighbor classifier, when tasked with classifying a test feature into a particular category, one simply finds the "nearest" training example and assigns the test case the label of that nearest training example.

```
k = 9;
%Compute feature distances between training and test images
distances = vl_alldist2(training_image_feats', test_image_features');
[~, indices] = sort(distances, 1);
%Get Label/Category Information
all_tr_labels = unique(training_labels);
no_of_tr_labels = size(all_tr_labels, 1);
count_of_labels = zeros(no_of_tr_labels, no_of_test_images);

for nth_test_image = 1:no_of_test_images
    %Find the count of nearest k matching categories for each test image
    for nth_label = 1:no_of_tr_labels
        %Take the nearest k labels of each test image
        tr_idx=indices(1:k, nth_test_image);
        top_k_labels = training_labels(tr_idx);
        count_of_labels(nth_label,nth_test_image) =
sum(strcmp(all_tr_labels(nth_label), top_k_labels));
    end
end

%We compute the most matched category for each image
[~ , label_indices] = max(count_of_labels,[],1);
predicted_categories = all_tr_labels(label_indices);
```

Classifier Codes (SVM) [12]:

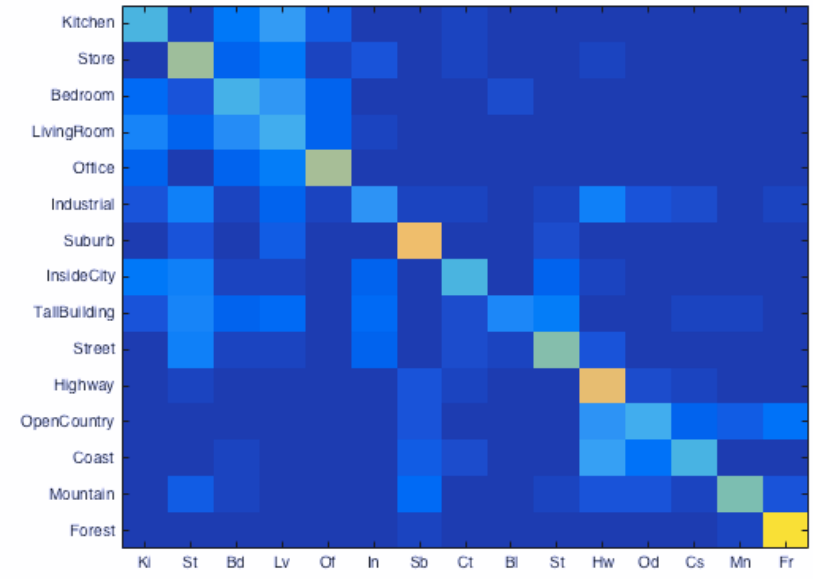
In SVM, a set of training examples, each marked for belonging to one of two categories, an SVM training algorithm builds a model that assigns new examples into one category or the other, making it a non-probabilistic binary linear classifier. An SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall on.

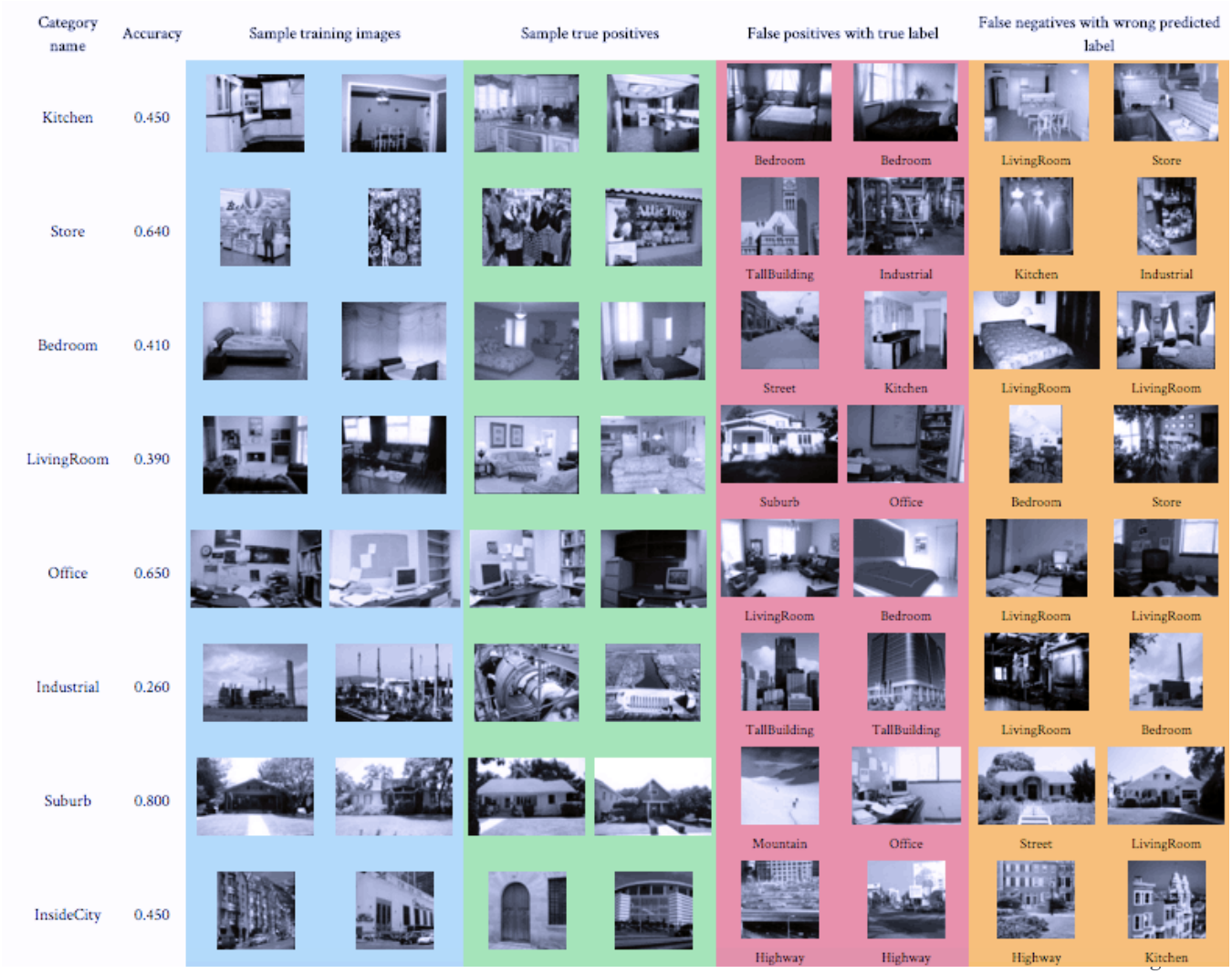
```
LAMBDA = 0.00001;
labels = unique(training_labels);
num_of_categories = length(labels);

for catNum = 1:num_of_categories
    %This function which indices in train_labels match a particular category.
    matching_indices = strcmp(labels(catNum), training_labels);
    matching_indices = +matching_indices;
    matching_indices(matching_indices==0) = -1;
    This function trains linear svms based on training examples, binary labels
    [W B] = vl_svmtrain(training_image_features', matching_indices, LAMBDA);
    %Confidence, or distance from the margin, is W*X + B
    training_score = [training_score; W'*test_image_features' + B];
end
[~,label_indices] = max(training_score);
```

Confidence Results for Bag of SIFT + Nearest Neighbor Classifier

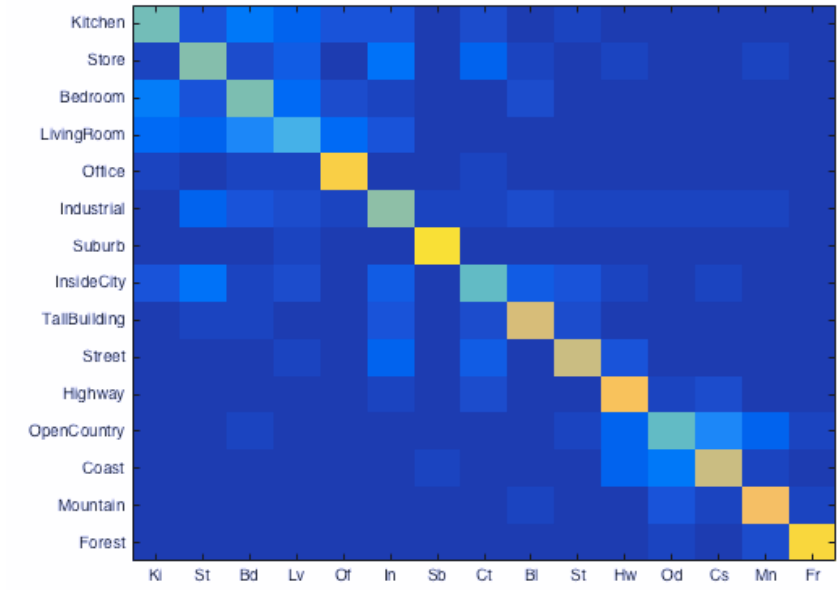
The accuracy is in the range of **0.51 to 0.55**, the confusion matrix is as follows and the same can be located at **html/knn/index.html** in the project directory or at this link[a] in CCIS GitHub.





Confidence Results for Bag of SIFT + Support Vector Machines Classifier

The accuracy is in the range of **0.66 to 0.71**, the confusion matrix is as follows and the same can be located at **html/svm/index.html** in the project directory or at this link[b] in CCIS GitHub.



References:

- [1] David G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints"
- [2] Richard Szeliski, "Computer Vision: Algorithms and Application".
- [3[Svetlana Lazebnik; Cordelia Schmid; Jean Ponce, "Beyond Bag of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories".
- [4] Mubarak Shah, "Fundamentals of Computer Vision".
- [5] Center For Research in Computer Vision (University Of Central Florida), http://crcv.ucf.edu/
- [6] Graphics, Visualization and Interaction Group, Brown University
- [7] Aude Oliva; Antonio Torralba, "Modeling the shape of the scene: a holistic representation of the spatial envelope", *International Journal of Computer Vision, Vol. 42(3): 145-175, 2001*
- [8] Lowe, David G., "Object recognition from local scale-invariant features". *Proceedings of the International Conference on Computer Vision. pp. 1150–1157.*
- [9] Andrew Witkin, "Scale-space filtering: A new approach to multi-scale description", *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP '84., Vol 9, pp. 150-153.*
- [10] J. B. MacQueen, "Some Methods for classification and Analysis of Multivariate Observations", *Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability.*
- [11] Navneet Dalal, Bill Triggs, "Histograms of Oriented Gradients for Human Detection", *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) (Volume:1), pp. 886 – 893.*
- http://www.computervisiononline.com/
- [12] https://en.wikipedia.org/wiki/Support_vector_machine

Project Links:

- [a] https://github.ccs.neu.edu/biswarajkar/robotics/blob/master/project/html/knn/index.html
- [b] https://github.ccs.neu.edu/biswarajkar/robotics/blob/master/project/html/svm/index.html
- [c] https://github.ccs.neu.edu/biswarajkar/robotics/blob/master/project/html/index.html