Scene Recognition using Bag-of-Words

Sarthak Ahuja
B.Tech Computer Science
Indraprastha Institute of Information Technology
Okhla, Delhi 110020
Email: sarthak12088@iiitd.ac.in

Anchita Goel
B.Tech Computer Science
Indraprastha Institute of Information Technology
Okhla, Delhi 110020
Email: anchita12019@iiitd.ac.in

***Abstract*—Scene recognition is an important upcoming real world problem which finds application in the fields of robotics(autonomous systems), surveillance(wearable camera footage, evidence photographs, etc), and personal assistance devices(Google Glass). There has been a steady progress in the field and this report explains the developments made in this field leading to the current state-of-the-art approach, documenting the results obatined from our implementation with standard benchmarks and classifiers.**

***Keywords*—scene recognition. sift, object recognition, k-means, bag-of-words, visual words, object filters**

I. Motivation

Our motivation towards taking up this projects is it's multiple applications in a varied domains. Firstly, it is of great use for an **autonomous agent** to get as much information as possible about it's environment tomake smarter decisions irrespective of what it's ultimate goal is. Secondly, with the growing trend of wearable cameras by **security personnel**, obtaining added information about their current situation is an added advantage. Finally, information regarding the current scene will in many ways add to the level of service and options **wearable devices like Google Glass** can provide.

II. Problem

The basic approach to the problem is to find a set of image descriptors - something that is distinctive for a particular scene and can distinguish it from the other scenes. These image descriptors are then learnt by a classifier. So the right set of questions to ask are -

1) **Which set of image descriptors work the best?**
2) **How can these descriptors be best represented?**
3) **Which classification algorithm performs the best?**

III. Literature Survey

The most basic and earliest approach[1] is of resizing the images in the training set into tiny photos of unit length each and then running a unsupervised clustering algorithm like nearest neighbour to compare the L2 pixel-wise distance between the testing sample and the training set images. Next with the introduction of SIFT feature points, the idea for a bag-of-words[2] (bag-of-visual words) was introduced to capture information about particular scene using the unique scale invariant feature points. In the last couple of years there have been subtle advances which include higher order vector representation[3] and spatial pyramid matching[2]. Some other state-of-the-art approaches that have come up recently are based on the above approaches - one of them uses distinctly trained object detection SVMs as a filter bank and uses the response from these filters to create the histogram. Another approach goes even further by detecting these objects for the filter bank directly from the training set. **Note: In this report we do not evaluate the latter two as they do not lead to significant improvements in accuracy and are beyond the scope of this minor project.**.

IV. Data

A. Image Acquisition

The dataset used for evaluation in this report combines three popular databases - fifteen scene categories[4] , Caltech-101[5] , and Graz[6] to give a final 15 Scene database. Details regarding distribution are given in table 2 and are depicted in figure 2.

B. Preprocessing

All images used in our case are of 200x200 pixel dimension and in gray scale. For the image descriptors generally PCA is used to reduce the dimensionality of feature vector **F**. This is a crucial step as the low-level local descriptors are strongly correlated, which results in many challenges at later stages such as K-means and GMM for dictionary creation. For example - The feature vector of length [G]x128 is reduced to a lower dimensions depending on the value of G.

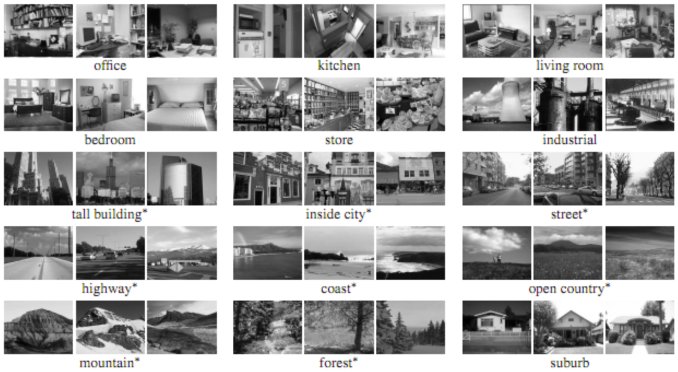

Fig. 1. The 15 scenes used for classification [7]

TABLE I. DATA DESCRIPTION

Class	Training Samples	Testing Samples
Office	100	115
Kitchen	100	110
Living Room	100	189
Bedroom	100	116
Store	100	215
Industrial	100	211
Tall Building	100	256
Inside City	100	208
Street	100	192
Highway	100	160
Coast	100	260
Open Country	100	315
Mountain	100	274
Forest	100	228
Suburb	100	141

V. PRELIMINARIES

There are some terms and concepts which are important to get accustomed to as they are used in rest of the report. Through out this report

1) **K** refers to Cluster Size/Vocabulary Size
2) **D** refers to size of a SIFT keypoint descriptor which is 128. (4x4x8)
3) **F** refers to final representation of an image post encoding. (**GxD**)
4) **G** refers to the number of interest points in an image, in our case - Grid Size.

A. *SIFT features*

SIFT stands for Scale Invariant Feature transform. For an image we require two things to represent it- interest points (given by a detector) and a way to describe the point (given by a descriptor). As pointed out above, this gives us **F=GxD**. SIFT points in an image are calculated as follows:

1) Interest points are selected by calculating the maximas and minimas in a difference of gaussian images set after discarding points along the edges.
2) The SIFT descriptor at these interest points is obtained from image gradients sampled over a 1616 array of locations in the neighbourhood of the specific interest point.
3) For each 4x4 region sample, the above obtained gradients are accumulated into a gradient orientation histogram with 8 quantized orientations. The figure 2 depicts the steps involved.

These features are universally used for such tasks of object detection and scene recognition because of their scale and affine invariance. As can be guessed the value **G** is dependent on the image size. To make our job easier we use a variant DSIFT on same sized images. (Using this allows us to keep **G** as a constant and not worry about comparing different sized feature vectors. This variant skips the detector steps and computes the descriptor in a regularly spaced grid. We use the VLfeat library [8] to compute SIFT features.

B. *GIST features*

GIST features [9] represent a variety of features of a scene such as naturalness, openness, roughness, expansion, ruggedness using filter banks. Given an input image, a GIST descriptor is computed by:

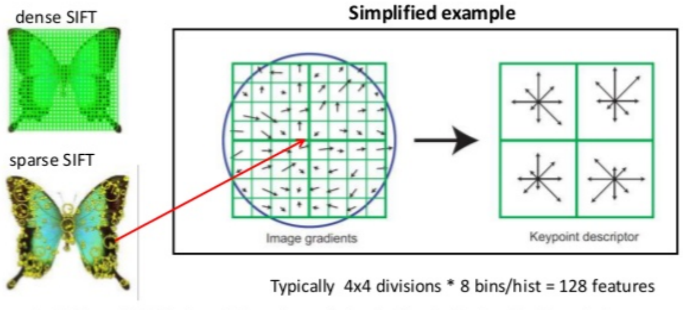

Fig. 2. Procedure for SIFT feature extraction [7]

1) Convolving the image with 32 Gabor filters at 4 scales, 8 orientations, producing 32 feature maps of the same size for an input image.
2) Divide each feature map into 16 regions (by a 4x4 grid), and then average the feature values within each region.
3) Concatenate the 16 averaged values of all 32 feature maps, resulting in a 16x32=512 GIST descriptor.

C. *Bag-of-Words*

For each image (during training or testing) we substitute every SIFT descriptor with the nearest centroid. We count how many of these descriptors for each image you get of each of he centroids. For every test image we compute its bag of keypoints, feed it to the classifier and get the classification.

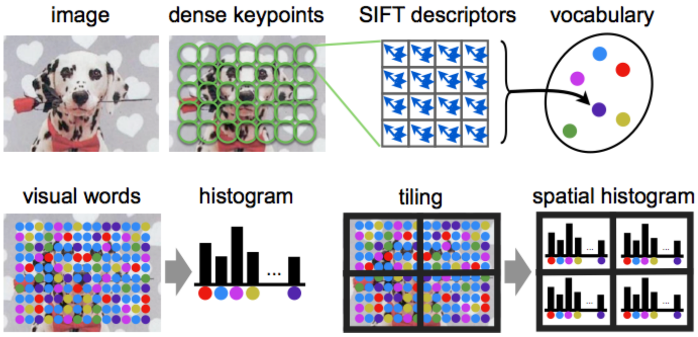

Fig. 3. Bag of Words Approach [2]

D. *Data Encoding*

There is an issue with the conventional BOW encoding method - it does not account for uncertainity and distribution across the centroid. It gives hard boundaries on choosing the centroid. Different encoding approaches try to tackle this issue in a different way and in turn often end up increasing the dimensionality. Some popular approaches are Histogram encoding (K), Kernel codebook encoding (KCB), Fisher encoding (FK) and Supervector encoding (SV). Based on our literature survey, the Fisher Encoding provides the best results with an added advantage of using GMMs as representations to store statistics of the difference between dictionary elements and pooled local features. We haven't gone into the details of the algorithm but can summarise it as follows -

1) GMM is used to model the visual vocabulary
2) The gradient of the log likelihood with respect to parameters of the model are calculated.
3) The fisher vector of a concatenation of these derivatives gives the direction in which the parameters must move to give the best fit.

The figure 2 depicts the steps involved. We use the VLfeat library [8] to encode Fischer vectors.

E. Spatial Pyramid Matching

The idea behind Spatial Pyramid matching is to define a kernel which preserves the spatial information of the features which is often lost by the simple unordered bag-of-words model. The feature vector is supposed to get mapped to a multiresolution histogram to preserve the distinctiveness of each feature at the finest level. These histogram pyramids are compared using a weighted histogram intersection. In the end the spatial pyramid matching simply turns out to be a bag-of-keypoints performed on separate spatial regions and concatenated together after a specific weighting. The figure 3 depicts the spatial pyramid being computed on three scales. As can be observed level 0 represents the least quantized level in which a single histogram is computed from the entire image. In the subsequent levels the degree of the spatial histogram increases by a factor of 2 and are concatenated with each other. The weights are multiplied at each level calculated as function of the level itself.
The spatial pyramid proved to be very effective and most of the subsequent papers that came after it's introduction started to leverage on it.

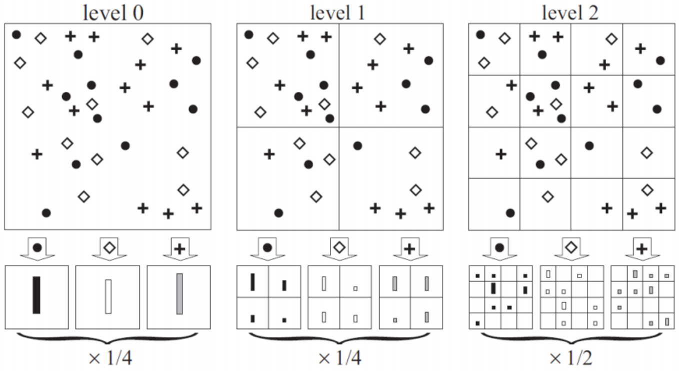

Fig. 4. Spatial Pyramid Representation [2]

VI. Approach

After referring to [10] we got to know about some good practices to keep in mind while performing the evaluation. This mainly includes carrying out preprocessing ,using higher order representation applying spatial pyramids and using dense features (all have majorly been described above). We use libsvm to carry out SVM [**?**] based classifications and the deep learning toolkit for matlab for Neural Network.

A. Direct Nearest Neighbour classification based on L2 distance

1) Method: In this we simply take the L2 pixel wise distance between the training and test images to classify the test images. This leads to extremely poor results - an accuracy of **22.34%** to be specific.

B. Nearest Neighbour classification based on dictionary of visual words

1) Method: This is an extension to the previous method where instead of pixel wise comparisons we first compute DSIFT on all the images and use the L2 distance between the descriptors as the distance function for the nearest neighbour classification system. Doing this improves the performance by almost double to **50%**.

2) Depending Factors: This method depends on the features we calculate for the images. For example we used SIFT in this case. We can further experiment by using different features like GIST or FAST, but considering SIFT is the most popular image descriptor used in state-of-the-art algorithms we don't expect improvements on using the other features. Instead the nearest neighbour approach seems to be the one that requires change.

C. Supervised classification on dictionary of visual words

1) Method: We use various methods to obtain features(in our case we call them visual words) from an image for classification. We use various tools like SIFT, GIST, Fischer Vector encoding and Spatial Pyramid Matching to extract visual words and features from an image. We use SVM classifier as from our literature survey, we inferred that this classifier gives the best results.

2) Depending Factors: The results from using the above features and classifier depend on the vocabulary size of the visual words obtained. The vocabulary size in terms of bags-of-words model refers to the number of clusters formed. We tried out a range of values for vocabulary size and report our results ahead.
The type of kernel used in SVM also affects the results of the classification. We use mainly two types of kernel - linear SVM and RBF kernel.

D. Improvements

1) Fischer Vector Representation and GIST feature: We calculate SIFT descriptors of all the training images. Using GMM, we obtain the parameters needed to obtain Fischer encoding for the SIFT descriptors. We also include GIST descriptors of an image as its features. We later use a linear SVM to classify this data. We obtained an accuracy as high as 66%.

2) Spatial Pyramid Representation: We also attempted to find out features from an image using Spatial Pyramid Matching. Using the vector obtained from the matching for each image as its feature representation, we used a RBF kernel and Linear kernel for SVM. We achieved an accuracy of 69.49% using this feature representation.

VII. Results

A. Classification Accuracy

TABLE II. Classification Accuracy

Approach	Accuracy(per)	Precision(Mean)
Direct Nearest Neighbour Classification	22.34%	-
Nearest-Neighbour Classification (dSIFT)	50%	-
SVM Classification (dSIFT), Vocabulary = 200	58.19%	56.82%
SVM Classification (dSIFT), Vocabulary = 500	60.46%	58.42%
SVM Classification (Fischer), Vocabulary = 50	65.19%	63.38%
SVM Classification (Fischer), Vocabulary = 200	63.28%	63.30%
SVM Classification (dSIFT+GIST), Vocabulary = 200	58.29%	56.92%
SVM Classification (dSIFT+GIST), Vocabulary = 500%	60.5%	58.54%
SVM Classification Fisher(dSIFT)+GIST), Vocabulary = 50	66%	64.48%
SVM Classification Fisher(dSIFT)+GIST), Vocabulary = 200	64.48%	64.15%
SVM Classification SIFT+Spatial, Vocabulary = 200	67.48%	65.21%

1) Confusion Matrix: Confusion matrix obtained for supervised classification with fisher vector encoding of features using a combination of SIFT and GIST features incorporating spatial pyramid matching is depicted in 5.

VIII. Conclusion

We have not been able to obtain appreciable results in this problem. However, using Spatial Pyramid Matching obtains the best results since it is able to encode the spatial information of an image the best. But Fischer encoding also doesn't do that worse than Spatial Pyramid. Linear SVMs give better results than RBF kernels in this case.

References

[1] F. Torralba and Freeman.

[2] R. F. L. Fei-Fei and P. Perona., "Learning generative visual models from few training examples: an incremental bayesian approach tested on 101 object categories."

[3] K. Chatfield, V. Lempitsky, A. Vedaldi, and A. Zisserman, "The devil is in the details: an evaluation of recent feature encoding methods," in *British Machine Vision Conference*, 2011.

[4] L. Fei-Fei and P. Perona., "A bayesian hierarchical model for learning natural scene categories.."

[5] R. F. L. Fei-Fei and P. Perona., "Learning generative visual models from few training examples: an incremental bayesian approach tested on 101 object categories."

[6] A. P. A. Opelt, M. Fussenegger and P. Auer., "hypotheses and boosting for generic object detection and recognition."

[7] C. Grana and G. Serra, "Recent advancements on the bag of visual words model for image classification and concept detection."

[8] A. Vedaldi and B. Fulkerson, "VLFeat: An open and portable library of computer vision algorithms," http://www.vlfeat.org/, 2008.

[9] A. T. Aude Oliva, "Modeling the shape of the scene: a holistic representation of the spatial envelope."

[10] X. W. Y. Q. Xiaojiang Peng, Limin Wang, "Bag of visual words and fusion methods for action recognition: Comprehensive study and good practice."

TABLE III. CONFUSION MATRIX

	Kitchen	Store	Bedroom	L.Room	Office	Industrial	Suburb	I.City	TallBuild.	Street	Highway	O.Country	Coast	Mountain	Forest
Kitchen	54	5	15	18	11	5	0	1	1	0	0	0	0	0	0
Store	10	136	2	14	4	12	1	14	0	4	1	1	0	10	6
Bedroom	11	2	52	31	5	7	0	1	0	1	0	3	0	2	1
L.Room	24	18	38	72	8	12	3	6	0	2	3	0	0	3	0
Office	22	2	6	5	79	0	0	0	0	0	0	0	1	0	0
Industrial	4	30	9	19	1	82	9	12	7	11	12	4	2	8	1
Suburb	0	5	0	3	1	2	124	2	0	1	1	0	0	0	2
I.City	12	15	1	8	1	10	5	130	7	16	1	1	0	1	0
TallBuild.	6	12	1	1	3	20	1	22	165	16	1	1	1	4	2
Street	0	14	0	1	0	21	2	11	1	135	4	1	0	1	1
Highway	0	4	2	1	0	7	2	1	2	3	121	12	4	1	0
O.Country	0	2	1	2	0	2	4	0	0	0	14	212	32	17	24
Coast	0	0	1	4	2	2	1	0	0	0	17	41	188	3	1
Mountain	0	1	3	0	1	1	0	0	1	3	9	21	4	212	18
Forest	0	2	1	0	0	0	0	0	0	2	0	4	0	4	215