Assignment 1 Report
Computer Vision EEL803
Scene Recognition Challenge
Group No 13

Piyush Kaul
Entry No. 2015EEY7544
EE Dept. IITD

Shiva Azimi
Entry No. 2015EEZ8458
EE Dept. IITD

Nilay Pandey
Entry No 2015BSZ8315
EE Dept IITD

***Abstract*—This report covers the ELL803 Computer Vision Assignment 1, on Scene Recognition Challenge.**

I. Problem Statement

Q. Implement Scene Recognition with the following steps.

- Feature extraction: You can choose your favorite set of features for this task (SIFT, HOG, GIST etc.). Choose your features wisely as this plays an important role in the performance. Extra credit: if features are chosen at multiple scales (Hint: you can choose some sampling strategies for choosing features).
- Encoding: Any feature encoding strategy could be used (Bag of visual Words, Bag of Visual Words+ Spatial Pyramid, Fisher Vector, Super Vector, Locality constrained linear encoding (LLC), Vector Quantization, Kernel Codebook Encoding). This link gives an analysis of various encoding strategies: Vary the size of K for the construction of dictionary: 200, 400, 800, 1000.
- Pooling: Max, Sum pooling etc.
- Classifier: SVM (1-vs-all strategy for classification) with different kernel choices. Use cross validation to measure performance. Randomly select the training and test set data for every category rather than fixating the data. Note of mark: Never choose the train data ¿ test data. Best strategy is to equally divide it for each category.

II. Solution

The following steps were implemented.

- Multi-Scale Dense SIFT
- Fisher Encoding
- Sum Pooling
- SVM (1-vs-all) using Linear and Gaussian Kernel.

Entire implementation is done in Matlab [1] and also utilize VL_feat[2] and libsvm[3] libraries.

A. Multi-Scale Dense SIFT

Multi-Scale Dense SIFT is implemented using the following parameters

- Step Size of 2 pixels.
- Scales Used: Gaussian Kernel with Standard Deviation 4/6, 6/6, 8/6, 10/6.
- The SIFT descriptors are 4×4 grid with 8 orientation bins.
  The Dense SIFT descriptors are extracted utilizing the VL_FEAT library

B. Fisher Encoding and VLAD Encoding

Fisher Encoding is used to capture first order and second order differences between image descriptors and centers of GMM. The GMM is learned first with subset of vectors (5%) The GMM $p(x|\theta)$ is a probability density function on R^D given by.

$$p(x|\theta) = \sum_{k=1}^{K} p(X|\mu_k, \Sigma_k)\pi_k$$

$$p(x|\mu_k, \Sigma_k) = \frac{1}{\sqrt{(2\pi)^D det\Sigma_k}} e^{\frac{1}{2}(x-\mu_k)^T \Sigma_K^{-1}(x-\mu_k)}$$

The GMM defines the soft data-to-cluster assignment

$$q_k i = \frac{p(X_i|\mu_k, \Sigma_k)\pi_k}{\sum_{j=1}^{K} p(x_i|\mu_j, \Sigma_j)\pi_j}$$

Given set of descriptors x_1,... x_N sample from image, let q_{ki}, k=1,...,K,i=1,...N be the soft assignment of N descriptors to K Gaussian components.

For each k=1,...K define the vectors.

$$u_k = 1/Nsqrt(\pi_k) \sum_{i=1} N q_{ki} \Sigma_K^{-1/2}(x_j - \mu_k)$$

$$v_k = 1/Nsqrt(2\pi_k) \sum_{i=1} N q_{ki}[(X_i - \mu_k)\Sigma_k^{-1}(X_i - \mu_k) - 1]$$

The covariance matrices Σ_k are diagonal. The Fisher encoding of the set of local descriptors is then given by concatenation of u_k and v_k for all K components giving a encoding of size 2DK

$$f_{Fisher} = [u_1^T, v_1^T, ..., u_k^T, v_k^T];$$

Addiotionally VLAD Encoding (Vector of Locally Aggregated Descriptors) was also tried out. VLAD encodes feature x by considering the residuals

$$v_k = \sum_{i=1}^{N} q_{ik}(x_i - \mu_k).$$

The residuals are then stacked together

$$f_{VLAD} = \begin{bmatrix} \vdots \\ v_k \\ \vdots \end{bmatrix}$$

Different Values of K are tried for Fisher Encoding and VLAD Encoding

1) 200
2) 400
3) 800
4) 1000

C. Pooling

Spatial Pooling is done by dividing the pictures into 2 × 2 sections and calculating the encodings over each of the sections as well as over the original whole image. The encodings are then combined using max or sum function.

D. Learning - Support Vector Machine

libsvm was used for Supervised Learning classification based on the encoded feature vectors.

50% of all images (2259) are used for training and passed on the svmtrain function for SVM modelling. Rest 50% are then used for testing the generated model. The following SVM kernels are tried.

1) Linear Kernel
2) RBF Kernel

III. Results

A. Confusion Matrix

B. Performance Results with different configurations

SN	Kernel	Feature	Encoding	Dimension	Pooling	Accuracy
1	LINEAR	DSIFT	FISHER	1000	yes	71.35%
2	LINEAR	PHOW	FISHER	1000	yes	64.40%
3	LINEAR	DSIFT	FISHER	200	yes	71.35%
4	LINEAR	DSIFT	VLAD	200	yes	70.60%
5	LINEAR	PHOW	FISHER	200	yes	64.40%
6	LINEAR	PHOW	VLAD	200	yes	63.61%
7	LINEAR	DSIFT	FISHER	400	yes	71.35%
8	LINEAR	PHOW	FISHER	400	yes	64.41%
9	LINEAR	DSIFT	FISHER	800	yes	71.36%
10	LINEAR	PHOW	FISHER	800	yes	64.41%
11	RBF	DSIFT	FISHER	200	yes	62.50%
12	RBF	DSIFT	FISHER	400	yes	62.50%

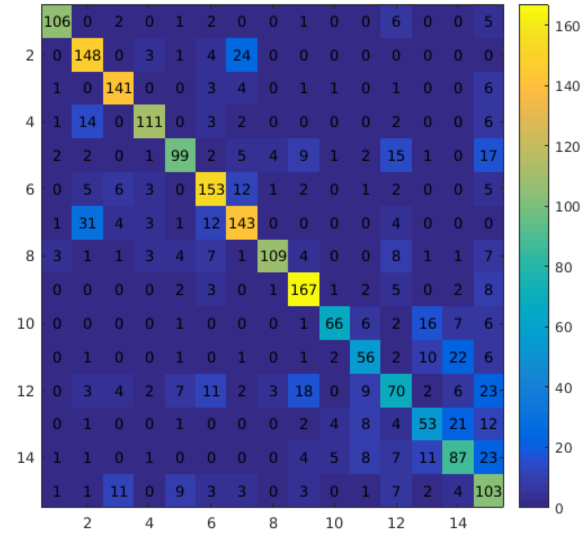

Fig. 1. Confusion Matrix.

IV. Conclusion

The Scene Classification Algorithm using Dense multi-scale SIFT and Fisher encoding along with Linear SVM seems to perform the best. It gives a performance above 71.36% Accuracy and 800 Features. Max and Sum pooling on the other hand don't substantially increase the detection accuracy. Lastly RBF kernel for SVM requires extensive fine-tuning of parameters without which it fails to perform well.

References

[1] Matlab http://www.mathworks.com/

[2] VLFeat Computer Vision Library http://www.vlfeat.org/

[3] LIBSVM – A Library for Support Vector Machines https://www.csie.ntu.edu.tw/ cjlin/libsvm/

[4] S. Lazebnik, C. Schmid and J. Ponce, "Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories," 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)

[5] K. Chatfield, V. Lempitsky, A Vedaldi, and A. Zisserman, "The devil is in the details: an evaluation of recent feature encoding methods"