anirudh choudhary

Project 4: Scene recognition with bag of words

In this project, we utilize the bag of words formulation for classifying scenes from Caltech 101 database. We create a vocabulary of words from training images by clustering SIFT features using K-Means/GMM and consequently, develop feature vectors for each image using histogram/ Kernel/ Fisher encoding. To start with, we utilize k-NN classifier predict the scene for test images. this gives us an accuracy within 55-60%. To improve the accuracy, we train SVM classifiers on the feature set using linear and RBF Kernels consequently achieving the highest accuracy of

Key implementations

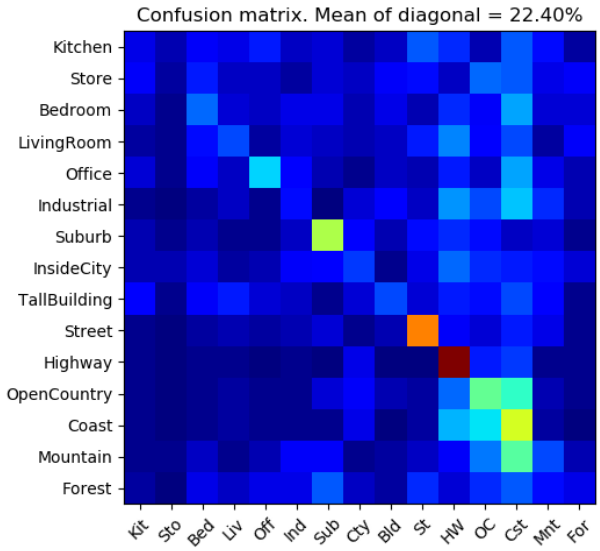
1. Tiny images representation and nearest neighbor classifier (highest accuracy : 22.40%)
2. Bag of SIFT representation and nearest neighbor classifier (highest accuracy : 59.13%)
3. Bag of SIFT representation and linear SVM classifier (highest accuracy : 77.47%)
4. Cross-validating linear and RBF Kernel based SVM for accuracy comparison

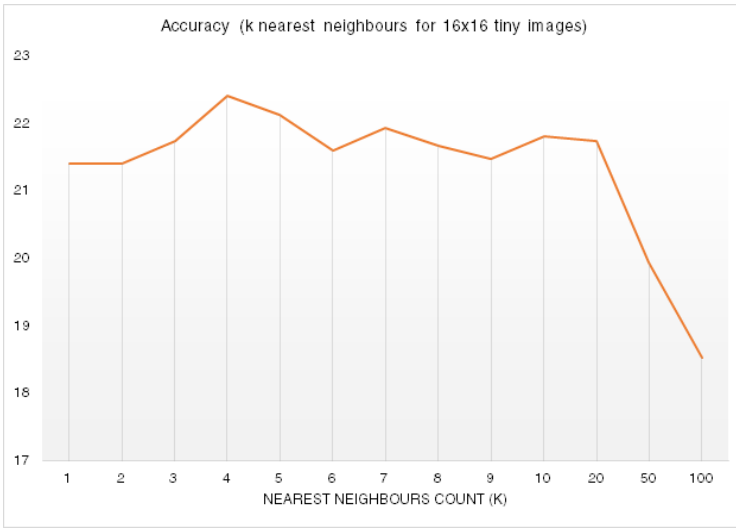
Extras implementations

1. Kernel Encoding (Soft histogram assignment)
2. Fisher Encoding
3. SVM with RBF Kernel

Tiny images evaluation

In this case, we resize the images to 16x16 pixels and create a feature vector of length 256 for each image using pixel values. The images are clas using k-nearest neighbour classifier. Various k values are compared and its found that k=4 gives highest accuracy of 22.4%. Street, Highway anc Coast have highest accuracies.

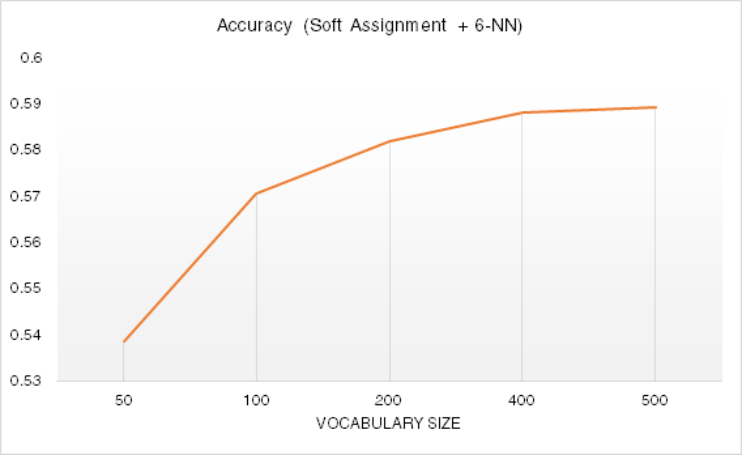




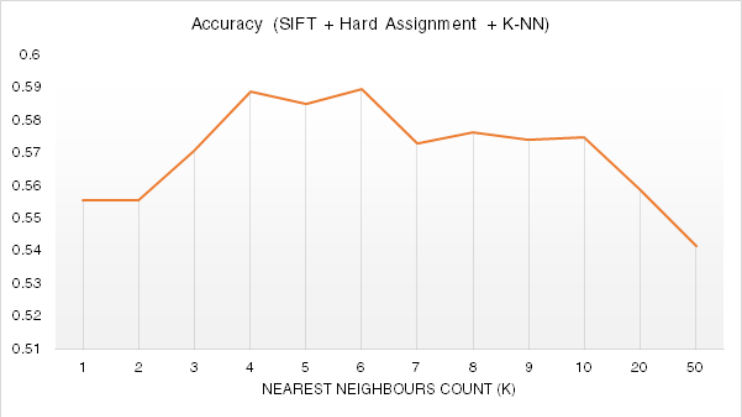
Bag of SIFT + k - Nearest Neighbour

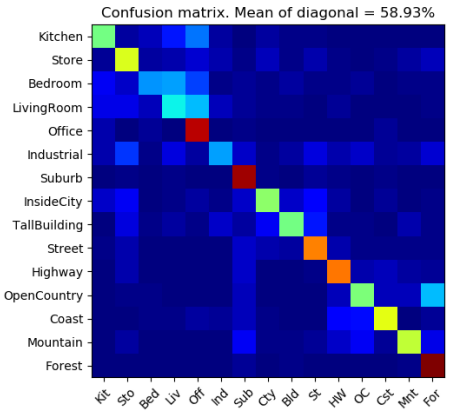
In this case, we compute dense SIFT features for training images using step size of 10 (optimum). These SIFT features are clustered using K-M Clustering and a vocabulary of 400 words is created (the size is selected by testing various vocabularies (50,100,200,400,500) on classification accuracy). We use the 'Fast' version of function DSIFT from VLFEAT to increase the computation speed. For feature creation, SIFT features o dimension 128 are computed for all images using step size of 5. Size 5 is found to be optimum and is also supported by Dense SIFT implementa in [1].

Using 400 words, feature encoding is done for training and test images using both hard assignment (histogram counts) and soft assignment (Ke encoding). For soft assignment, weight assignment to k=3 nearest classes is found to be most optimum(proof in next section). Soft assignment implemented basis 'Codeword uncertainty' metric proposed in [2] since their evaluation finds it to be most accurate.
K-NN classifier is evaluated for various K values and the accuracy corresponding to K = 4, K = 6 and K = 8 is found to be the highest.

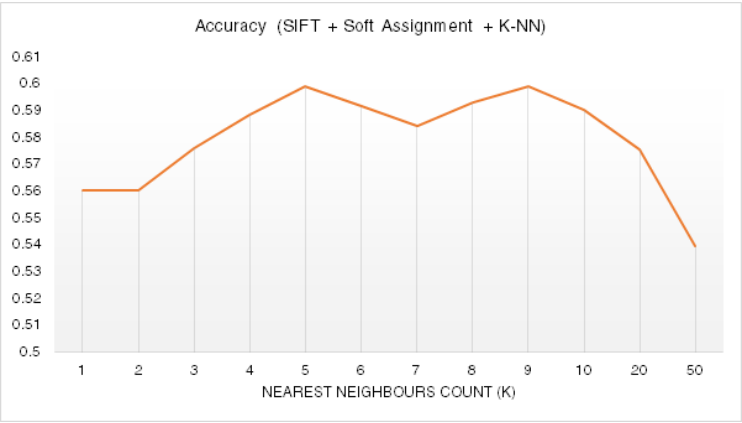


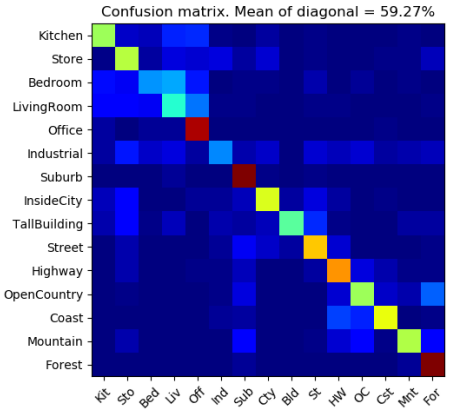
Accuracy of 6-NN classifier with Soft Assignment





Highest Accuracy for K = 6 and Hard Assignment

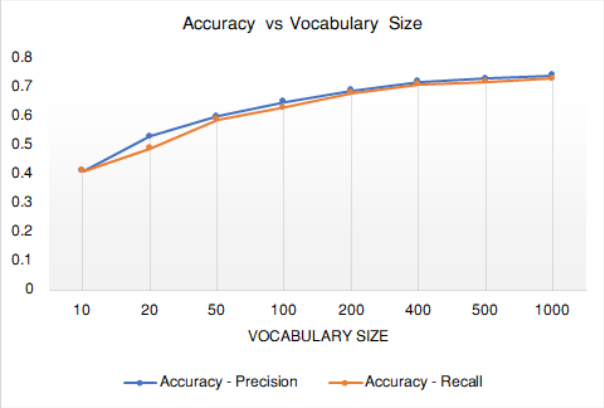




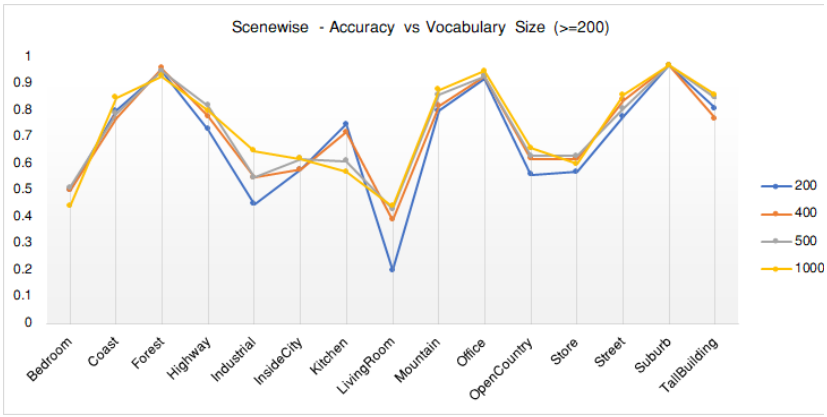
Highest Accuracy for K = 8 and Soft Assignment

Bag of SIFT + SVM

In this case, we compute dense SIFT features for training images using step size of 10 (optimum). 2 clustering algorithms are evaluated for crea vocabulary. For Histogram/ Kernel Encoding, K-Means clustering is used to compute word centres. Vocabulary of 400 words is found to be th most optimal in terms of accuracy and speed. We compare various vocabularies (10,20,50,100,200,400,500,1000) on classification accuracy usin 'one vs all' linear SVM with C = 5 (decided basis cross-validation). Beyond 400 words, the increase in accuracy is minor while the computation increases a lot, hence we use word vocabulary of 400.



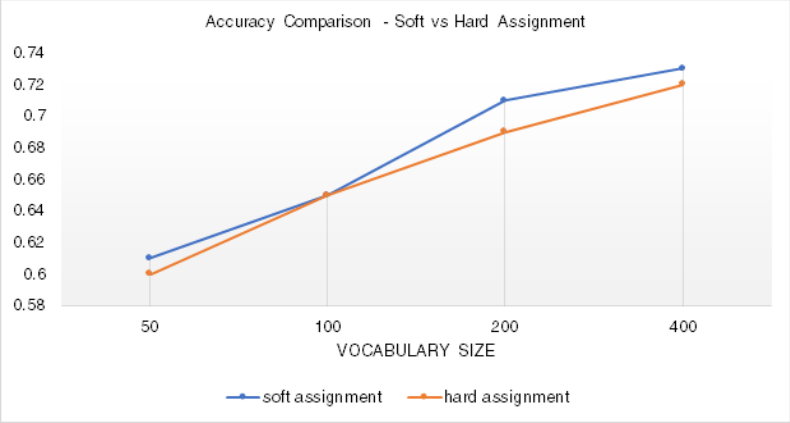
Precision and Recall for various vocabularies



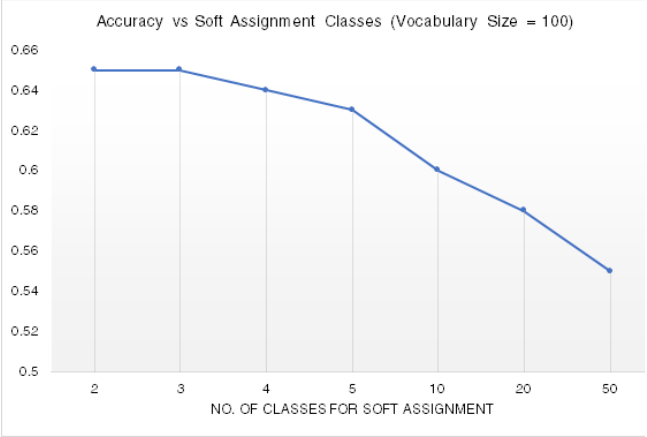
Increase in accuracy from vocabulary size 200 to 400 is the highest

For feature creation, SIFT features of dimension 128 are computed for all images using step size of 5. Using 400 words, feature encoding is don training and test images using histogram/ Kernel encoding for K-Means based vocabulary and Fisher encoding for GMM based vocabulary.

Kernel Encoding : Soft assignment using Kernel encoding is found to increase classification accuracy compared to histogram encoding based ha assignment.For soft assignment, weight assignment to k=3 nearest classes is found to be most optimum.



Accuracy comparison of Hard vs Soft Assignment (SIFT + Encoding + linear SVM)



For vocabulary size = 100, number of classses = 3 is the best for soft assignme

SVM classifiers are evaluated using 5 fold cross-validation on the training dataset.

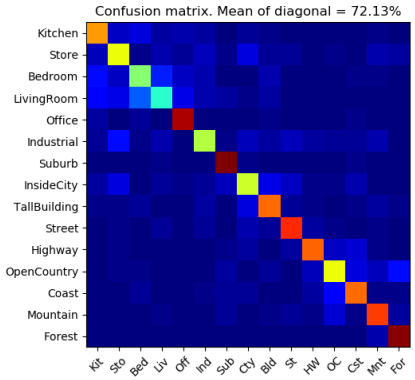
1. For linear 'one vs all' SVM, regularization parameter C = 5 is found to be the most optimum.
2. For RBF Kernel based SVM, regularization parameter C = 5 and gamma=1 is found to be the most optimum. The highest accuracy is for gamma=0.5 but the classifier has higher standard error with minor change in accuracy while classifier with C=5 and gamma=1 has lower standard error

10 fold Cross-validation results - Linear/RBF SVMs, Vocabulary size = 400, Soft Assignment Classes = 3

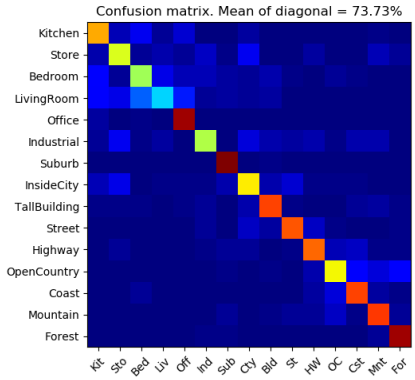
Mean Accuracy (+/- Standard Deviation)

```
Using SVM classifier to predict test set categories
Performing cross-validation with linear and RBF
The best parameters are {'kernel': 'rbf', 'gamma': 0.5, 'C': 5} with a score of 0.70
0.627 (+/-0.066) for {'kernel': 'rbf', 'gamma': 5, 'C': 0.5}
0.645 (+/-0.042) for {'kernel': 'rbf', 'gamma': 4, 'C': 0.5}
0.655 (+/-0.057) for {'kernel': 'rbf', 'gamma': 3, 'C': 0.5}
0.661 (+/-0.051) for {'kernel': 'rbf', 'gamma': 2, 'C': 0.5}
0.663 (+/-0.057) for {'kernel': 'rbf', 'gamma': 1, 'C': 0.5}
0.639 (+/-0.069) for {'kernel': 'rbf', 'gamma': 0.5, 'C': 0.5}
0.548 (+/-0.077) for {'kernel': 'rbf', 'gamma': 0.1, 'C': 0.5}
0.656 (+/-0.037) for {'kernel': 'rbf', 'gamma': 5, 'C': 1}
0.665 (+/-0.035) for {'kernel': 'rbf', 'gamma': 4, 'C': 1}
0.675 (+/-0.050) for {'kernel': 'rbf', 'gamma': 3, 'C': 1}
0.682 (+/-0.048) for {'kernel': 'rbf', 'gamma': 2, 'C': 1}
0.674 (+/-0.055) for {'kernel': 'rbf', 'gamma': 1, 'C': 1}
0.669 (+/-0.049) for {'kernel': 'rbf', 'gamma': 0.5, 'C': 1}
0.590 (+/-0.075) for {'kernel': 'rbf', 'gamma': 0.1, 'C': 1}
0.661 (+/-0.041) for {'kernel': 'rbf', 'gamma': 5, 'C': 5}
0.673 (+/-0.044) for {'kernel': 'rbf', 'gamma': 4, 'C': 5}
0.683 (+/-0.053) for {'kernel': 'rbf', 'gamma': 3, 'C': 5}
0.684 (+/-0.043) for {'kernel': 'rbf', 'gamma': 2, 'C': 5}
0.694 (+/-0.043) for {'kernel': 'rbf', 'gamma': 1, 'C': 5}
0.699 (+/-0.060) for {'kernel': 'rbf', 'gamma': 0.5, 'C': 5}
0.675 (+/-0.052) for {'kernel': 'rbf', 'gamma': 0.1, 'C': 5}
0.661 (+/-0.041) for {'kernel': 'rbf', 'gamma': 5, 'C': 10}
0.671 (+/-0.045) for {'kernel': 'rbf', 'gamma': 4, 'C': 10}
0.682 (+/-0.053) for {'kernel': 'rbf', 'gamma': 3, 'C': 10}
0.683 (+/-0.044) for {'kernel': 'rbf', 'gamma': 2, 'C': 10}
0.688 (+/-0.053) for {'kernel': 'rbf', 'gamma': 1, 'C': 10}
0.691 (+/-0.057) for {'kernel': 'rbf', 'gamma': 0.5, 'C': 10}
0.687 (+/-0.048) for {'kernel': 'rbf', 'gamma': 0.1, 'C': 10}
0.661 (+/-0.041) for {'kernel': 'rbf', 'gamma': 5, 'C': 15}
0.671 (+/-0.045) for {'kernel': 'rbf', 'gamma': 4, 'C': 15}
0.682 (+/-0.053) for {'kernel': 'rbf', 'gamma': 3, 'C': 15}
0.683 (+/-0.044) for {'kernel': 'rbf', 'gamma': 2, 'C': 15}
0.689 (+/-0.056) for {'kernel': 'rbf', 'gamma': 1, 'C': 15}
0.689 (+/-0.051) for {'kernel': 'rbf', 'gamma': 0.5, 'C': 15}
0.695 (+/-0.052) for {'kernel': 'rbf', 'gamma': 0.1, 'C': 15}
0.661 (+/-0.041) for {'kernel': 'rbf', 'gamma': 5, 'C': 20}
0.671 (+/-0.045) for {'kernel': 'rbf', 'gamma': 4, 'C': 20}
0.682 (+/-0.053) for {'kernel': 'rbf', 'gamma': 3, 'C': 20}
0.683 (+/-0.044) for {'kernel': 'rbf', 'gamma': 2, 'C': 20}
0.689 (+/-0.056) for {'kernel': 'rbf', 'gamma': 1, 'C': 20}
0.683 (+/-0.050) for {'kernel': 'rbf', 'gamma': 0.5, 'C': 20}
0.695 (+/-0.062) for {'kernel': 'rbf', 'gamma': 0.1, 'C': 20}
0.649 (+/-0.074) for {'kernel': 'linear', 'C': 0.5}
0.678 (+/-0.048) for {'kernel': 'linear', 'C': 1}
0.694 (+/-0.072) for {'kernel': 'linear', 'C': 5}
0.687 (+/-0.075) for {'kernel': 'linear', 'C': 10}
0.685 (+/-0.075) for {'kernel': 'linear', 'C': 15}
0.688 (+/-0.060) for {'kernel': 'linear', 'C': 20}
```

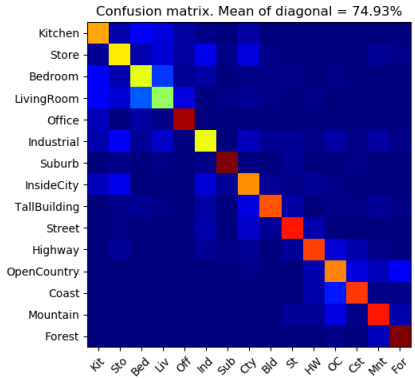
Results of SIFT + Soft/Hard Assignment + Linear/RBF SVM



Hard Assignment with Linear (one vs all) SVM



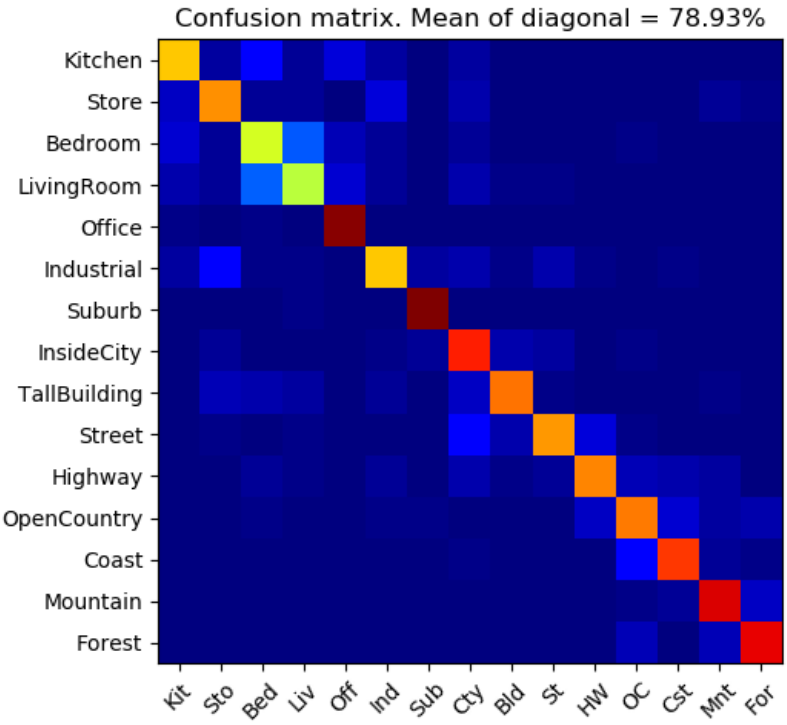
Soft Assignment with Linear (one vs all) SVM



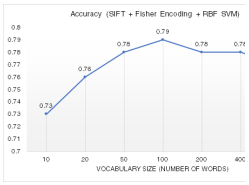
Soft Assignment with RBF SVM

Improving accuracy by using RBF Kernel based SVM with Fisher Encoding

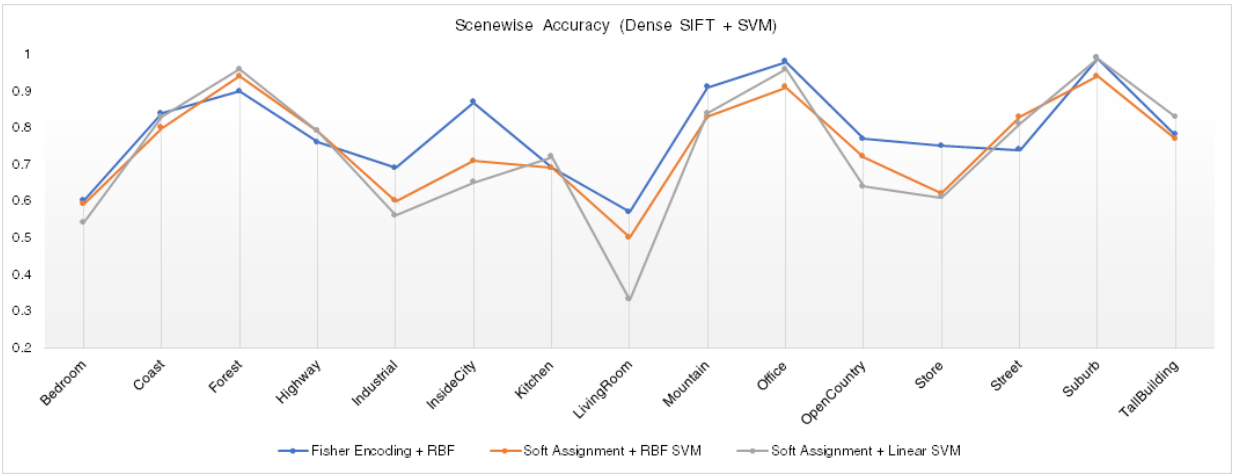
We use Fisher encoding using VLFEAT's fisher function (with Improved option set to Truw which takes care of normalization) and compute t accuracy across all fisher vocabularies. The highest accuracy is achieved for 100 word vocabulary and the accuracy decreases beyond 200 word vocabulary slightly which might be due to the higher dimensionality of features (50k+). RBF SVM classifier is used with C = 5 and gamma = 1.



Highest Accuracy (gamma = 1 and C = 5)



Highest accuracy achieved fo
200 word vocabulary



Fisher Encoding improves in
InsideCity and Industrial Cat
while RBF Kernel shows h
accuracy in classifying Living
and OpenCountry catego

References:

1. SUN Database: Large-scale Scene Recognition from Abbey to Zoo; Jianxiong Xiao James Hays† Krista A. Ehinger Aude Oliva Antonio Torralba
2. Kernel Codebooks for Scene Categorization; Jan C. van Gemert, Jan-Mark Geusebroek, Cor J. Veenman, and Arnold W.M. Smeulders
3. Cross-Validation for Digit Recognizer