# CSCI B657 – Computer Vision

# Final Project Report

# Classification of images into specific scene categories

Pradeep Kumar Ravilla
Manikandan Murugesan
Srikanth Srinivas Holavanahalli

# Introduction

Object recognition has been a broad area of research in computer vision. Scene classification is a problem under the domain of object recognition. Scene classification is a very interesting problem of automatically labeling an image among a set of semantic categories.

Scene classification has varied range of applications for example; it can be used to see if the pictures uploaded in Yelp/Foursquare/TripAdvisor sites were uploaded in the correct category. The image can be classified if it is a Scenic Lookout site/Hotel/Restaurant/Sports Facility. This classification can be used to automatically update the features of those places instead of manual update by the users.

# Background and Related Work

We had referred a paper titled "Scene classification using a multi-resolution bag-of features model"[1] which discusses the procedure we decided to implement. The paper discusses in detail the steps to do multi resolution bag of features model. To implement k-means clustering, we referred another paper titled "An efficient kmeans clustering algorithm: analysis and implementation" [2]. We got our dataset from the resources mentioned in the paper "Scene classification using a multiresolution bag-of-features model".

Our scene classifier works using modified bag-of-words model that utilizes the spatial correlation among small sub-samples of the images. We use multiresolution images for generating a codebook which would be used to generate features required for SVM to classify the images. A part of this work has been implemented in A3 in course work.

# Methods

Our approach has 5 steps for building the Scene classification using a multi-resolution bag of features model.

Step1: Creating Multiresolution images

Original images are converted to grayscale and subsampled with scales differing by a factor of 2. We obtained 3 images in this process, original image, subsampled images scaled down by a factor of 2 and 4.

Step2: Spatial binning of multiresolution images

The original image is resized to 256*256 and small horizontal and vertical sections of size 1*4 and 4*1 are taken. For each of these bins, we compute denseSift of each of these sections instead of normal Sift. Similarly for image of size one-half (128*128 ) we compute denseSift for sections of size 1*2 and 2*1 and for image of size one-fourth (64*64) we compute denseSift for sections of size 1*1. We ensure we obtain 32 bins for each resolution image.

Step3: Codebook generation

We use K-means clustering algorithm of OpenCV [3] to generate the visual codebook for each resolution image. We concatenate features generated from each resolution image to generate 1 visual codebook.

Step 4: Training SVM

With the obtained features, we train the SVM to learn the features and build a model for the train images.

Step 5: Visual code book generation for test images

We repeat the same binning and visual code book generation for each test image to be classified.

Step 6: Run SVM to classify the test images

We use the model built from train images and the features generated from test images and classify the image. This step is repeated for all the test images.

# Results and implementation details

<u>Implementation specific details</u>

- We implemented the scene classification using the regular BOW model. We have used some of the code from A3 of the assignment.
- To implement BOW with spatial information, we used certain inbuilt functions of OpenCV like kmeans, vconvat, etc.
- All the functions that we were trying to use were not available in the opencv installation that was there in burrow. So instead we set up OpenCV locally on our machine with contrib module.
- To classify the images, we have used SVMlight provided by Cornell.
- We experimented with various non-linear kernels like RBF and Polynomial kernels, but we found that the linear kernel worked better for us.
- Linear SVM takes hardly 1 min to train on a feature set for 150*15 images and it takes less time to classify a set of 50*15 images.
- Since Burrow doesn't have the installation of opencv that we were looking for, it was not able to run our program. So instead, we have uploaded pre trained datasets.
- To test the code for the respective implementation(Spatial and without Spatial), move into the particular folder and type in:
    - python3 score.py <predicitons_file> <model_file>
    - python3 score.py bow_predictions bow_test_features.svm

*k=50*

bow_train_features_50.svm bow_test_features_50.svm bow_predictions_50 bow_model_50

*k=100*

bow_train_features_100.svm bow_test_features_100.svm bow_predictions_100 bow_model_100

*k=200*

bow_train_features_200.svm bow_test_features_200.svm bow_predictions_200 bow_model_200

# RESULTS

The **main goal** of this project was to improve the accuracy obtained from plain Sift and bag-of-words model using spatial correlation.

We have obtained vast improvement in our results when spatial correlation was used. We got **46%** accuracy with error rate set to 1% as against random guess of 6%.

We analyzed the results that we got, and we found that **outdoor scenes** gave us very promising results.

<u>Results without spatial correlation</u>

For k = 200

```
[sriksrin@silo Without Spatial Correlation]$ python3 score.py bow-predictions bow-test-features
         suburb  coast   forest  highway insidec mountai opencou street  tallbui office  bedroom industr kitchen livingr store
suburb    |0|     0       1       13      1       2       7       9       4       2       2       3       0       4       2
coast     0       |15|    0       14      0       3       16      0       0       0       0       0       0       2       0
forest    0       0       |10|    1       2       5       2       5       8       0       1       3       2       3       8
highway   0       3       0       |34|    1       1       10      0       0       0       0       1       0       0       0
insidec   0       0       2       13      |7|     3       6       7       1       1       1       2       3       1       3
mountai   0       3       0       12      2       |5|     14      2       1       1       1       4       1       1       3
opencou   0       4       1       27      0       2       |15|    0       1       0       0       0       0       0       0
street    0       0       0       9       0       6       1       |28|    1       1       1       2       1       0       0
tallbui   0       0       1       10      3       4       9       3       |7|     4       2       2       1       3       1
office    0       0       2       4       1       4       4       9       1       |2|     7       3       3       5       5
bedroom   0       1       0       6       4       2       5       7       1       1       |4|     2       2       13      2
industr   0       0       1       12      3       3       6       5       2       3       0       |7|     2       5       1
kitchen   0       1       0       1       8       1       1       5       3       4       6       2       |5|     6       7
livingr   0       1       3       2       5       2       2       5       2       3       6       3       2       |10|    4
store     0       1       1       2       3       3       6       5       6       1       1       1       0       8       |12|
Accuracy = 21.466666666666665
[sriksrin@silo Without Spatial Correlation]$ cd
```

<u>Results with spatial correlation</u>

For k = 50

```
[sriksrin@silo Spatial Correlation]$ python3 score.py bow_predictions_50 bow_test_features_50.svm
         suburb  coast   forest  highway insidec mountai opencou street  tallbui office  bedroom industr kitchen livingr store
suburb    |0|     0       12      2       4       1       3       24      0       0       0       0       0       3       1
coast     0       |9|     1       24      0       3       13      0       0       0       0       0       0       0       0
forest    0       0       |47|    0       0       1       0       0       0       0       0       0       0       0       2
highway   0       0       0       |49|    0       0       0       1       0       0       0       0       0       0       0
insidec   0       1       6       4       |15|    1       2       13      3       2       0       0       0       0       3
mountai   0       0       21      0       0       |17|    8       2       0       0       0       0       0       0       2
opencou   0       2       17      10      0       1       |18|    2       0       0       0       0       0       0       0
street    0       0       10      0       0       0       0       |38|    1       0       0       0       0       1       0
tallbui   0       0       3       0       1       0       0       0       |46|    0       0       0       0       0       0
office    0       0       1       2       10      1       1       2       13      |8|     0       0       0       10      2
bedroom   0       0       9       3       1       0       0       7       10      1       |7|     1       4       6       1
industr   0       0       23      1       2       0       2       3       8       0       0       |2|     2       0       7
kitchen   0       0       4       1       2       1       1       2       13      0       1       0       |12|    7       6
livingr   0       0       10      0       2       0       1       5       10      3       0       0       2       |12|    5
store     0       0       29      0       2       0       2       0       9       0       0       0       0       0       |8|
Accuracy = 38.4
```

For k = 100

```
[sriksrin@silo Spatial Correlation]$ python3 score.py bow_predictions_100 bow_test_features_100.svm
         suburb  coast   forest  highway insidec mountai opencou street  tallbui office  bedroom industr kitchen livingr store
suburb    |0|     0       13      3       5       1       2       23      1       0       0       0       0       2       0
coast     0       |18|    1       18      0       2       11      0       0       0       0       0       0       0       0
forest    0       0       |48|    0       0       1       0       0       0       0       0       0       0       0       1
highway   0       0       0       |49|    0       0       0       1       0       0       0       0       0       0       0
insidec   0       1       8       1       |17|    1       2       13      2       1       1       0       1       0       2
mountai   0       0       19      3       0       |17|    7       2       0       0       0       0       0       0       2
opencou   0       5       13      7       0       0       |24|    1       0       0       0       0       0       0       0
street    0       0       4       0       0       0       0       |45|    1       0       0       0       0       0       0
tallbui   0       0       2       0       1       0       0       0       |47|    0       0       0       0       0       0
office    0       0       2       1       7       2       1       2       10      |14|    0       0       1       9       1
bedroom   0       0       8       4       1       1       0       4       9       5       |9|     1       2       5       1
industr   0       0       23      1       1       1       1       5       7       1       0       |3|     1       0       6
kitchen   0       0       3       2       1       2       1       1       15      0       2       0       |13|    9       1
livingr   0       0       8       0       1       0       1       2       8       5       0       0       3       |18|    4
store     0       0       32      0       3       0       2       0       5       0       0       0       1       0       |7|
Accuracy = 43.86666666666667
```

For k = 200

```
[sriksrin@silo Spatial Correlation]$ python3 score.py bow_predictions_200 bow_test_features_200.svm
         suburb  coast   forest  highway insidec mountai opencou street  tallbui office  bedroom industr kitchen livingr store
suburb    |0|     0       12      4       5       1       1       24      0       0       0       0       0       3       0
coast     0       |23|    1       15      0       1       10      0       0       0       0       0       0       0       0
forest    0       0       |49|    0       0       1       0       0       0       0       0       0       0       0       0
highway   0       0       0       |49|    0       0       1       0       0       0       0       0       0       0       0
insidec   0       1       5       0       |19|    0       2       17      3       1       1       0       0       0       1
mountai   0       0       20      4       0       |17|    6       3       0       0       0       0       0       0       0
opencou   0       5       14      6       0       0       |24|    1       0       0       0       0       0       0       0
street    0       0       3       0       0       1       0       |45|    1       0       0       0       0       0       0
tallbui   0       0       2       0       1       0       0       0       |47|    0       0       0       0       0       0
office    0       0       1       2       6       2       0       1       11      |17|    0       0       1       8       1
bedroom   0       0       8       3       1       0       1       2       10      5       |11|    0       2       6       1
industr   0       0       19      1       4       2       2       4       7       1       0       |5|     1       1       3
kitchen   0       0       2       1       3       1       1       2       10      0       2       0       |15|    10      3
livingr   0       0       8       1       1       0       0       3       8       4       0       0       4       |19|    2
store     0       0       30      0       2       0       2       0       7       0       0       0       0       0       |9|
Accuracy = 46.53333333333333
```

As seen from the above results for k = 200, results obtained with spatial correlation was 46% against 28% accuracy obtained when spatial correlation was not used.

# Conclusions and Future work

We wanted to investigate the comparison of scene classification accuracies obtained from with and without spatial correlation .We can conclude that the spatial correlation used scene classification provides better accuracies. As a future work, we would want to use other classifiers such deep learning. We would want to experiment with different image section sizes and compare the results.

# References

1 - “Scene classification using a multi-resolution bag-of-features model” by Li Zhou,

Zongtan Zhou, Dewen Hu

http://www.sciencedirect.com/science/article/pii/S0031320312003330

2 - “An efficient k-means clustering algorithm: analysis and implementation” by

D. M. Mount ; N. S. Netanyahu ; C. D. Piatko ; R. Silverman ; A. Y. Wu

http://ieeexplore.ieee.org/xpl/abstractAuthors.jsp?arnumber=1017616

3 - “Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural

Scene Categories, in: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2006” by S. Lazebnik, C. Schmid, J. Ponce

http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=1641019

4 - “Scene categorization via contextual visual words” by Jianzhao Qin, Nelson H.C. Yung

http://www.sciencedirect.com/science/article/pii/S0031320309004245