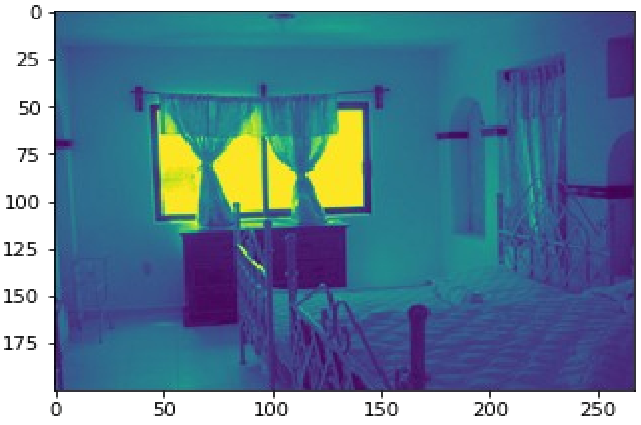
Computer Vision
Assignment 4: Bag of Words

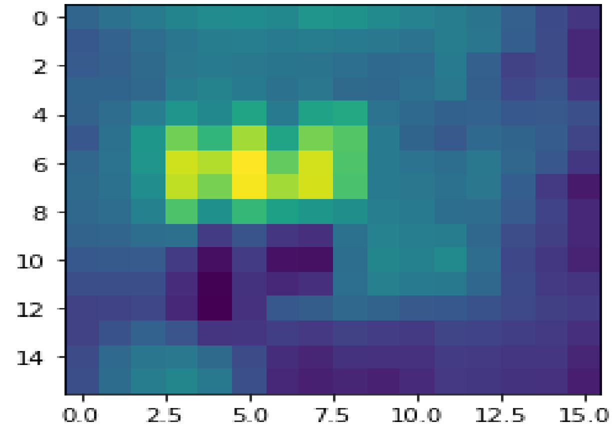
Junaid Maqbool
MSEE18005

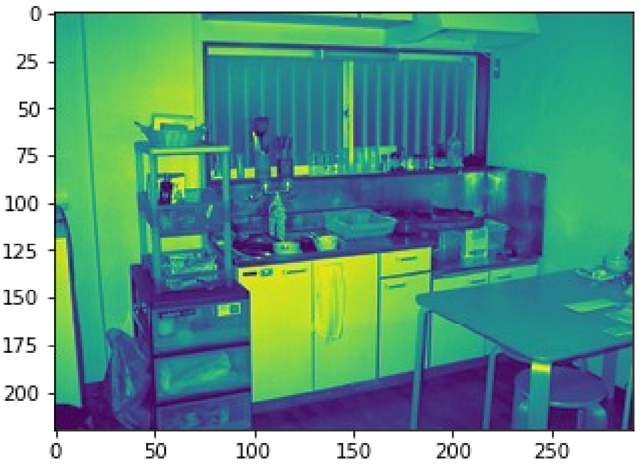
1. Tiny images features and nearest neighbor:

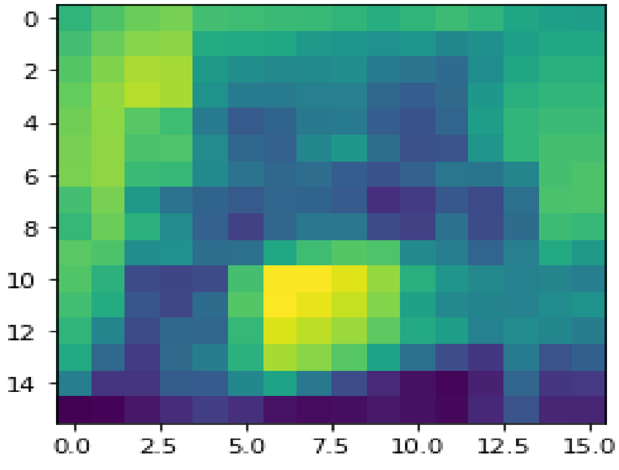
In tiny images implementation we used a naive approach to classify images by using the vectorized down-scaled image as features of that image. Motivation is that even at low resolution enough information is present in the image to classify the image.

True Image: **Tiny Image as feature:**





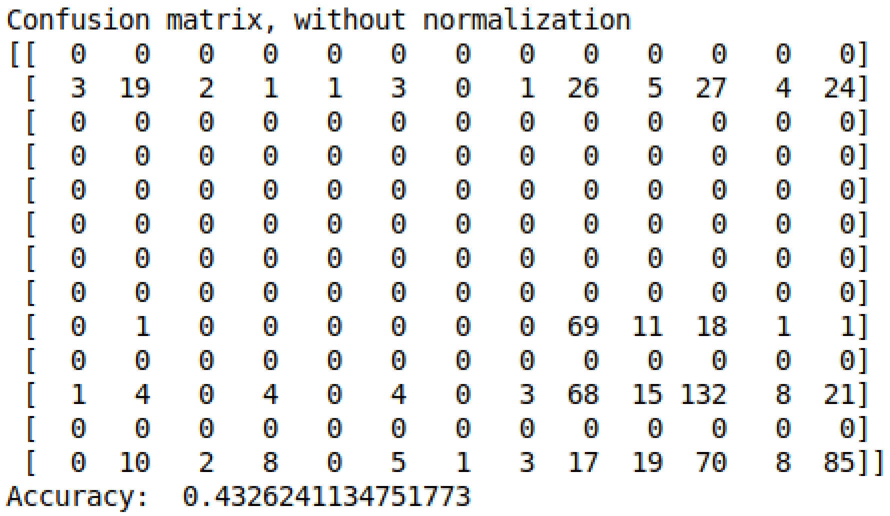




It can be observed in the images that low resolution image have the information to classify.
Tiny image features was easy to implement we normalized features (vectorized tiny image) and notice improvement in the accuracy.

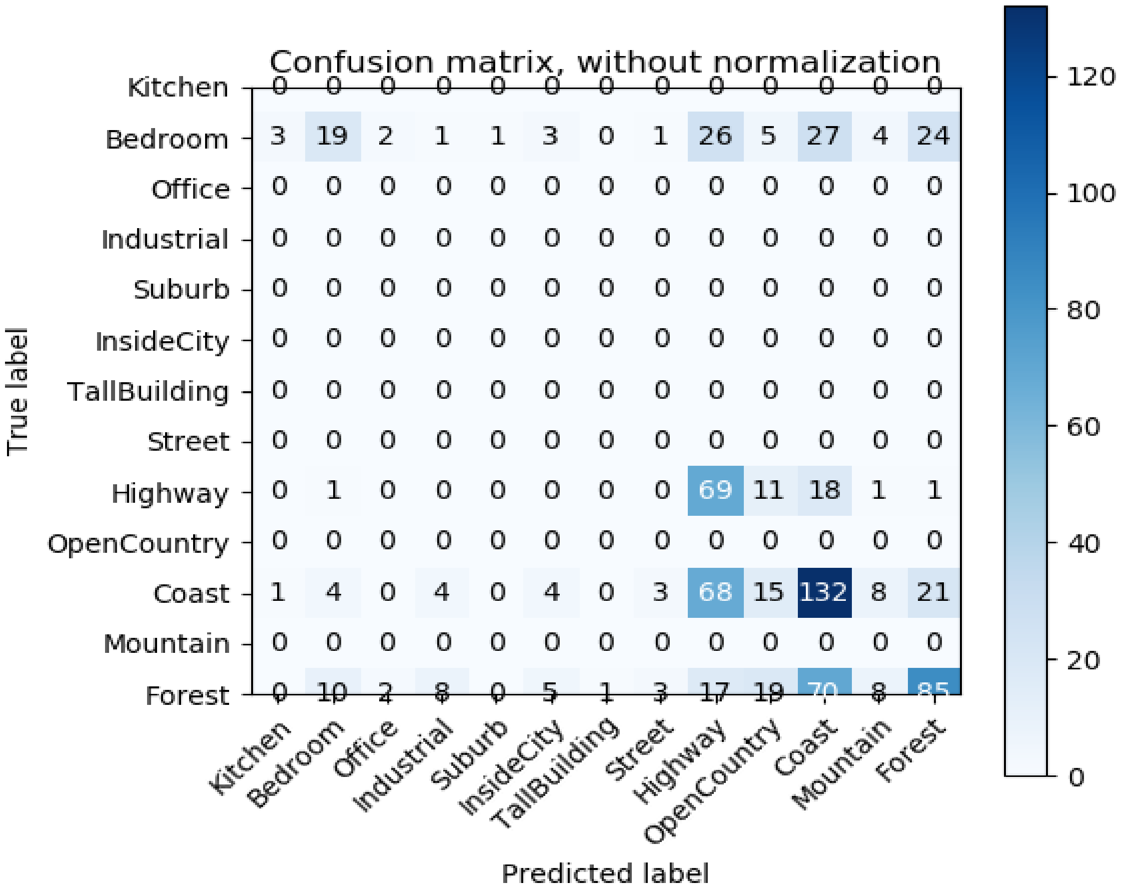
Final Accuracy: 43.26%

Confusion Matrix:



```
Confusion matrix, without normalization
[[  0   0   0   0   0   0   0   0   0   0   0   0   0]
 [  3  19   2   1   1   3   0   1  26   5  27   4  24]
 [  0   0   0   0   0   0   0   0   0   0   0   0   0]
 [  0   0   0   0   0   0   0   0   0   0   0   0   0]
 [  0   0   0   0   0   0   0   0   0   0   0   0   0]
 [  0   0   0   0   0   0   0   0   0   0   0   0   0]
 [  0   0   0   0   0   0   0   0   0   0   0   0   0]
 [  0   0   0   0   0   0   0   0   0   0   0   0   0]
 [  0   1   0   0   0   0   0   0  69  11  18   1   1]
 [  0   0   0   0   0   0   0   0   0   0   0   0   0]
 [  1   4   0   4   0   4   0   3  68  15 132   8  21]
 [  0   0   0   0   0   0   0   0   0   0   0   0   0]
 [  0  10   2   8   0   5   1   3  17  19  70   8  85]]
Accuracy:  0.4326241134751773
```

Table of Classifiers:



2. Bag of SIFT features and nearest neighbor:

In this implementation we used SIFT descriptors as image features and classify the image based on these features. As we know that SIFT features are scale and rotation invariant, highly descriptive and gives local features so it is good idea to used SIFT descriptors as image features.
There are many free parameters in this implementation effect of changing these parameters is observed by experimentation and explained here.

Vocabulary size:
By increasing vocabulary size we got greater precision to classify the images and now we can classify images with minor details that is why accuracy of the classification increases.

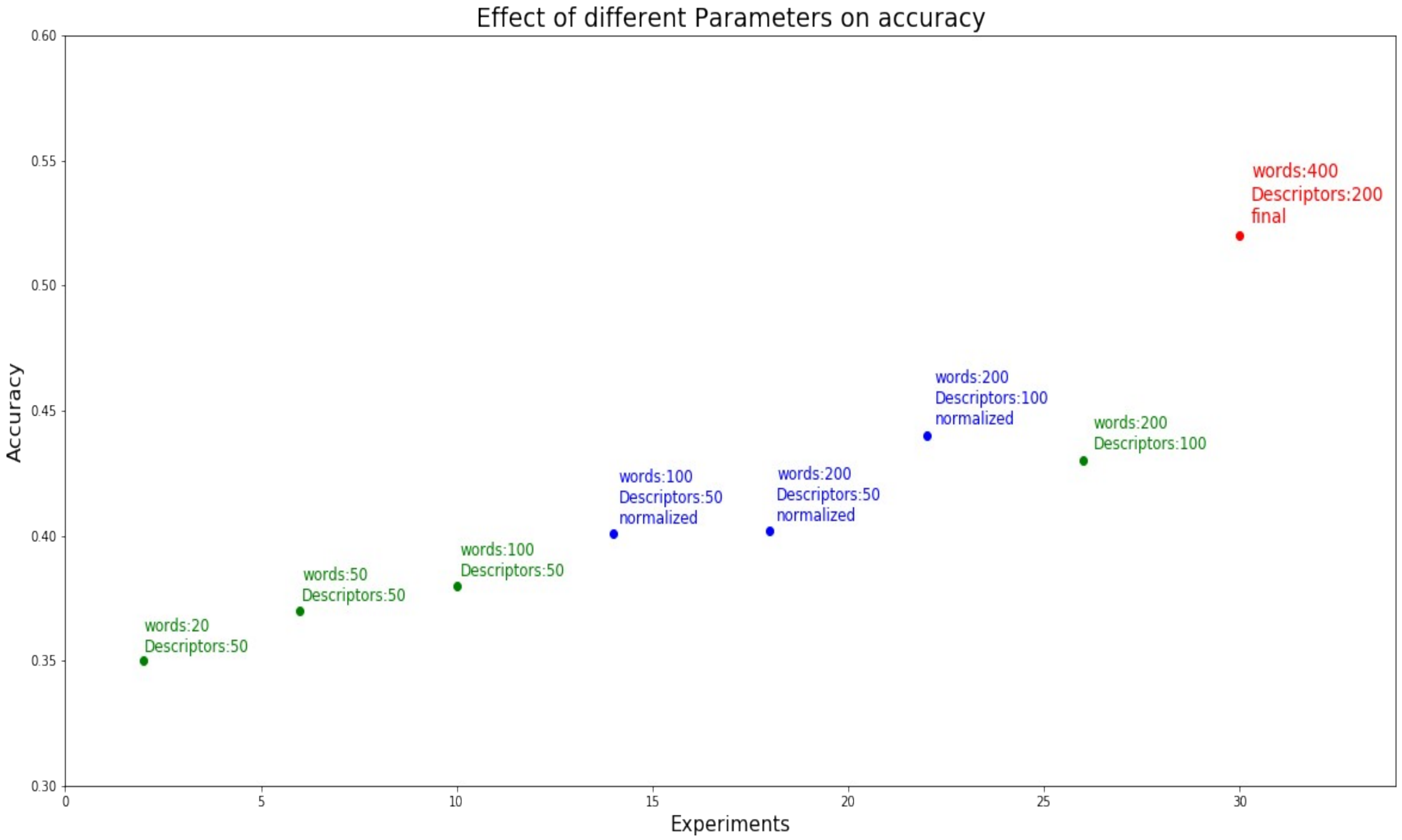
Descriptor Size:
Number of maximum descriptors returned by SIFT can be controlled by choosing greater number of descriptors accuracy increases but by increasing to a limit accuracy decreases or remains same because descriptors having less information are also invluded as feature of image.

Normalizing Descriptors:
Effect on Accuracy of the classifier by normalizing the descriptors is investigated. It is observed that accuracy of the model increases by making norm of the descriptor unity.

Number of Neighbors K:
Parameter K of KNN classifier is also tuned and it is giving best results at K=3.

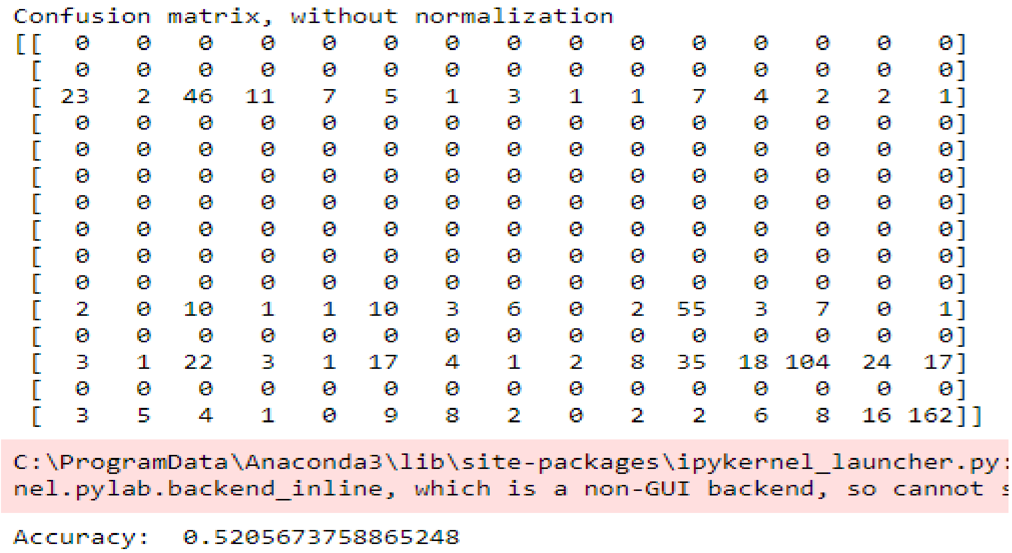
Plot shows the accuracy of the KNN at different parameters:

Following scatter plot shows the accuracy of the KNN at different combination of parameters.



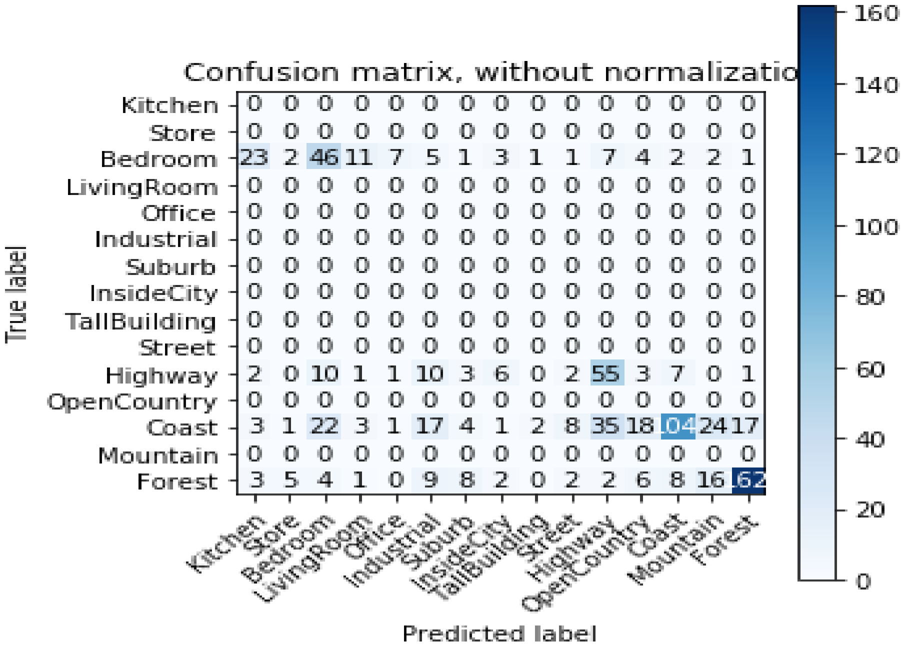
Final Accuracy: 52.05%

Confusion Matrix:



```
Confusion matrix, without normalization
[[  0   0   0   0   0   0   0   0   0   0   0   0   0   0   0]
 [  0   0   0   0   0   0   0   0   0   0   0   0   0   0   0]
 [ 23   2  46  11   7   5   1   3   1   1   7   4   2   2   1]
 [  0   0   0   0   0   0   0   0   0   0   0   0   0   0   0]
 [  0   0   0   0   0   0   0   0   0   0   0   0   0   0   0]
 [  0   0   0   0   0   0   0   0   0   0   0   0   0   0   0]
 [  0   0   0   0   0   0   0   0   0   0   0   0   0   0   0]
 [  0   0   0   0   0   0   0   0   0   0   0   0   0   0   0]
 [  0   0   0   0   0   0   0   0   0   0   0   0   0   0   0]
 [  0   0   0   0   0   0   0   0   0   0   0   0   0   0   0]
 [  2   0  10   1   1  10   3   6   0   2  55   3   7   0   1]
 [  0   0   0   0   0   0   0   0   0   0   0   0   0   0   0]
 [  3   1  22   3   1  17   4   1   2   8  35  18 104  24  17]
 [  0   0   0   0   0   0   0   0   0   0   0   0   0   0   0]
 [  3   5   4   1   0   9   8   2   0   2   2   6   8  16 162]]
C:\ProgramData\Anaconda3\lib\site-packages\ipykernel_launcher.py:
nel.pylab.backend_inline, which is a non-GUI backend, so cannot s
Accuracy:  0.5205673758865248
```

Table of Classifiers:

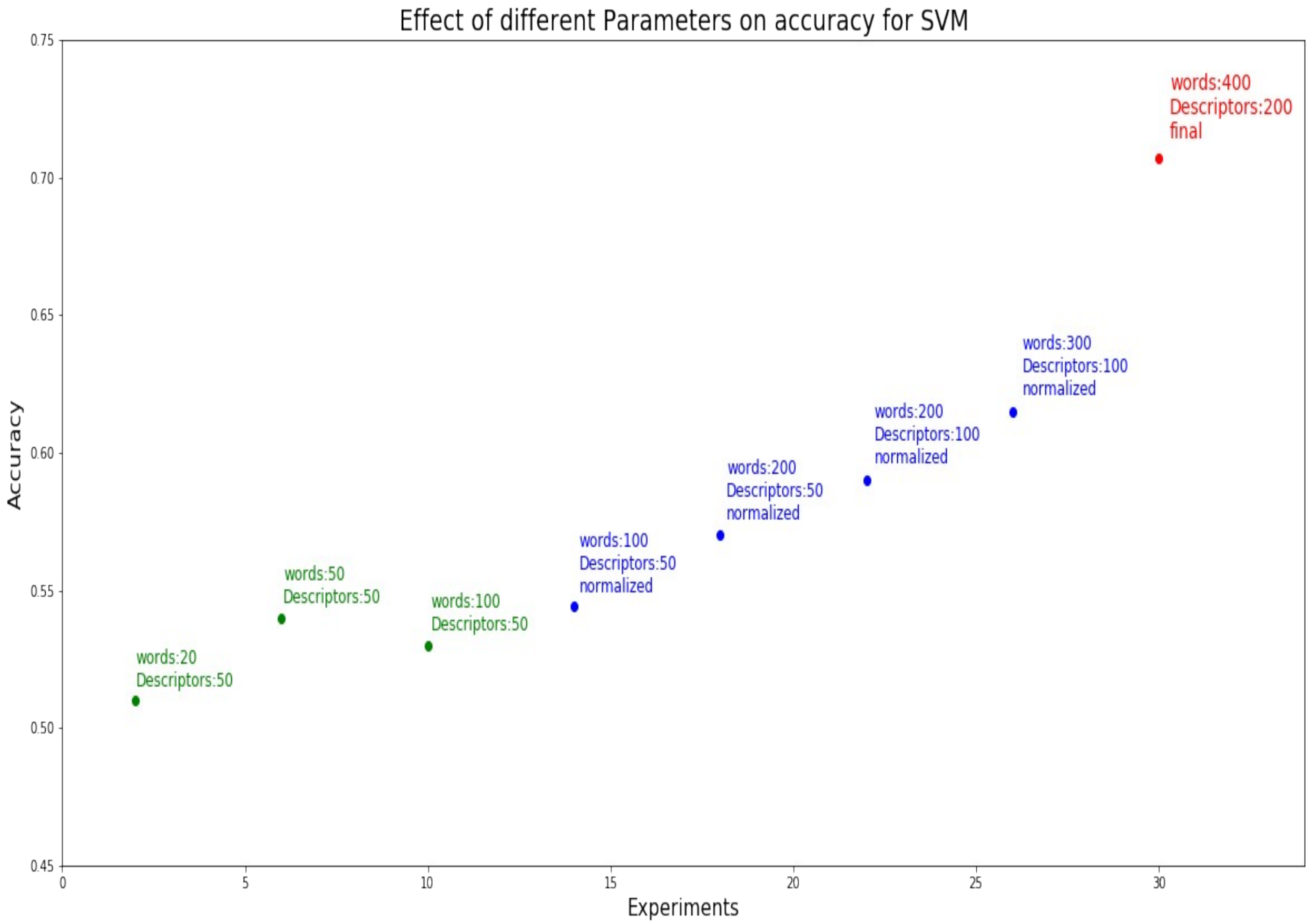


	Kitchen	Store	Bedroom	LivingRoom	Office	Industrial	Suburb	InsideCity	TallBuilding	Street	Highway	OpenCountry	Coast	Mountain	Forest
Kitchen	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Store	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bedroom	23	2	46	11	7	5	1	3	1	1	7	4	2	2	1
LivingRoom	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Office	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Industrial	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Suburb	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
InsideCity	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
TallBuilding	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Street	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Highway	2	0	10	1	1	10	3	6	0	2	55	3	7	0	1
OpenCountry	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Coast	3	1	22	3	1	17	4	1	2	8	35	18	104	24	17
Mountain	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Forest	3	5	4	1	0	9	8	2	0	2	2	6	8	16	162

3. Bag of SIFT features and linear SVM:

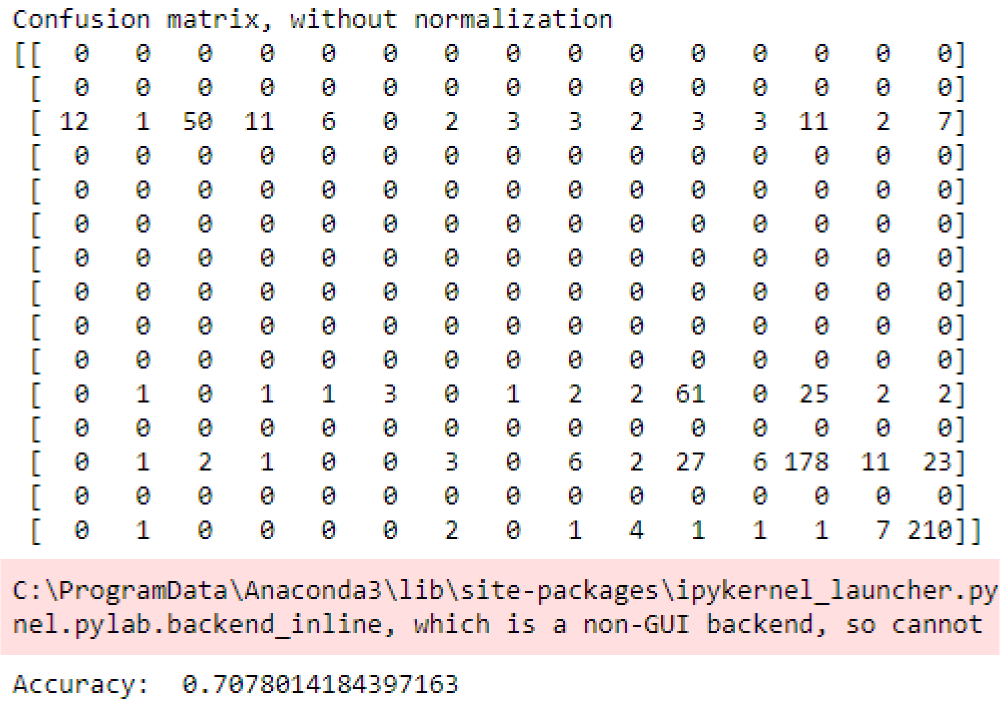
In this part we used Linear Support Vector Machine (SVM) as classifier the effect of different parameters is same in case of SVM with the difference that SVM give better results than KNN on all combination of parameters .

Following plot shows the accuracy of the SVM at different parameters:
Following scatter plot shows the accuracy of the SVM at different combination of parameters.



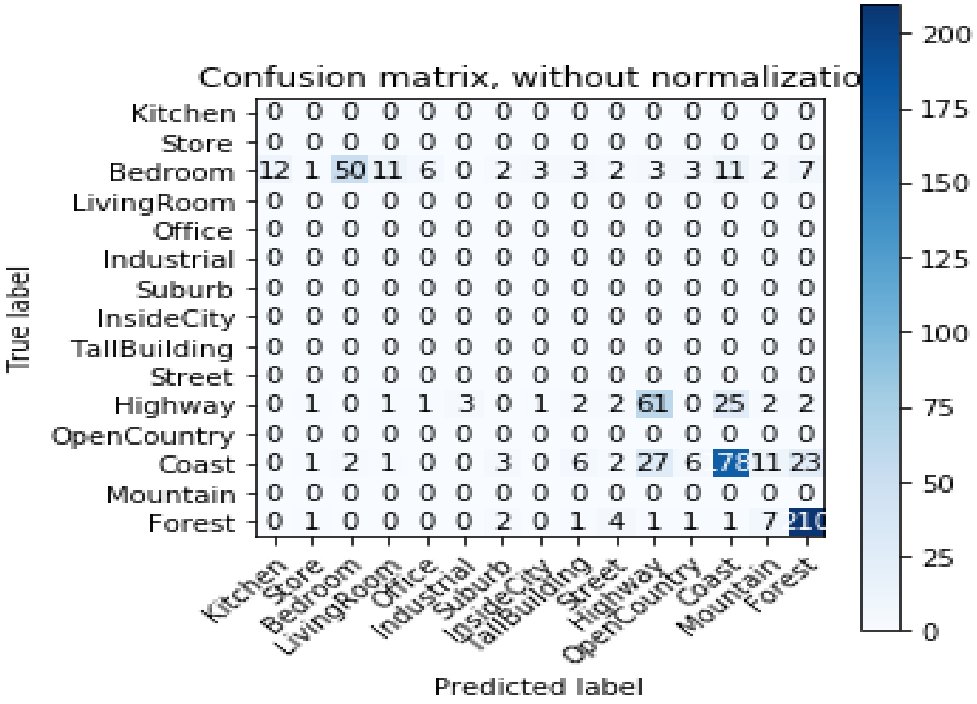
Final Accuracy: 70.7%

Confusion Matrix:



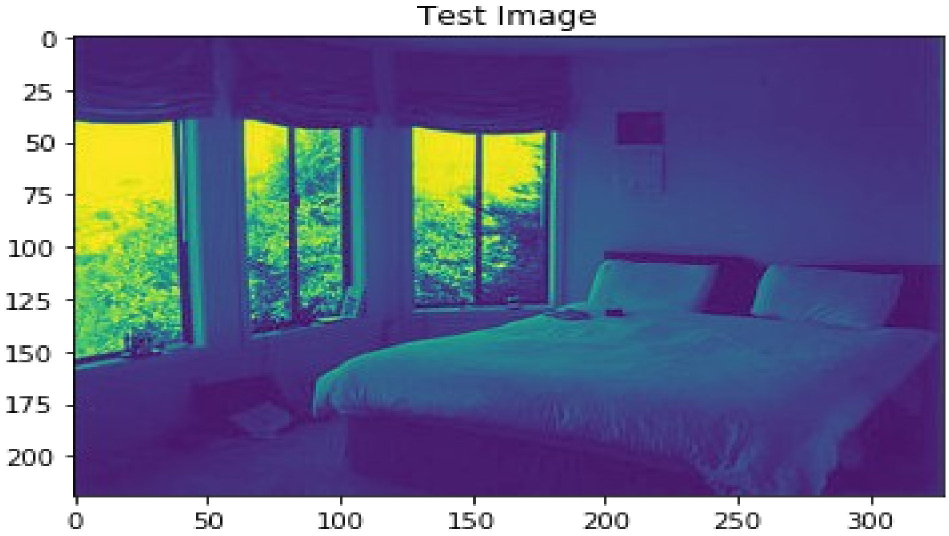
```
Confusion matrix, without normalization
[[  0   0   0   0   0   0   0   0   0   0   0   0   0   0   0]
 [  0   0   0   0   0   0   0   0   0   0   0   0   0   0   0]
 [ 12   1  50  11   6   0   2   3   3   2   3   3  11   2   7]
 [  0   0   0   0   0   0   0   0   0   0   0   0   0   0   0]
 [  0   0   0   0   0   0   0   0   0   0   0   0   0   0   0]
 [  0   0   0   0   0   0   0   0   0   0   0   0   0   0   0]
 [  0   0   0   0   0   0   0   0   0   0   0   0   0   0   0]
 [  0   0   0   0   0   0   0   0   0   0   0   0   0   0   0]
 [  0   0   0   0   0   0   0   0   0   0   0   0   0   0   0]
 [  0   0   0   0   0   0   0   0   0   0   0   0   0   0   0]
 [  0   1   0   1   1   3   0   1   2   2  61   0  25   2   2]
 [  0   0   0   0   0   0   0   0   0   0   0   0   0   0   0]
 [  0   1   2   1   0   0   3   0   6   2  27   6 178  11  23]
 [  0   0   0   0   0   0   0   0   0   0   0   0   0   0   0]
 [  0   1   0   0   0   0   2   0   1   4   1   1   1   7 210]]
C:\ProgramData\Anaconda3\lib\site-packages\ipykernel_launcher.py
nel.pylab.backend_inline, which is a non-GUI backend, so cannot
Accuracy:  0.7078014184397163
```

Table of Classifiers:

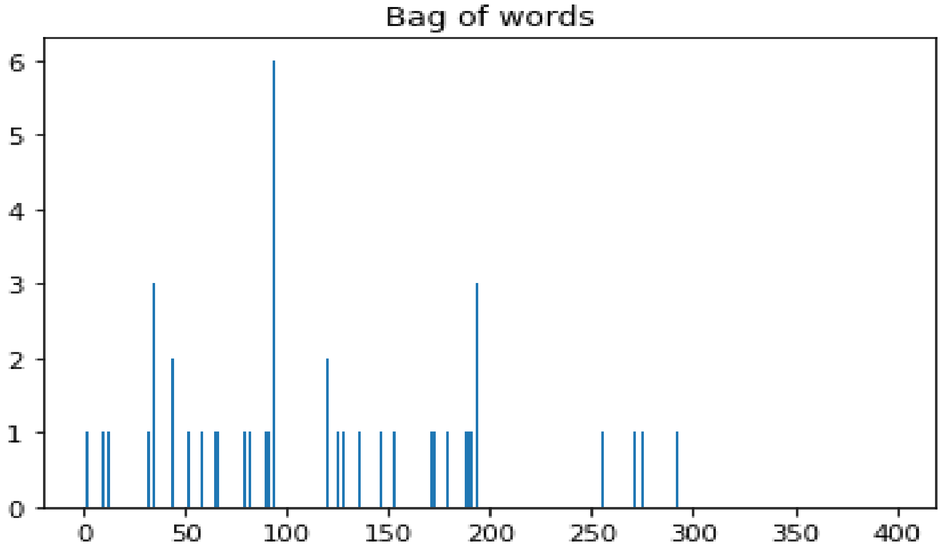


Visualization of SIFT descriptors and histogram of a test image:

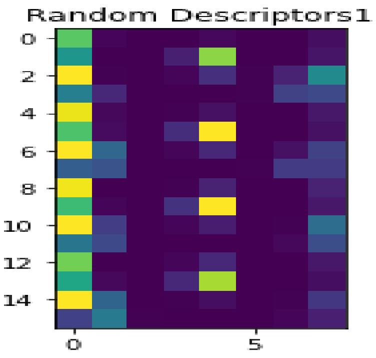
Test Image:

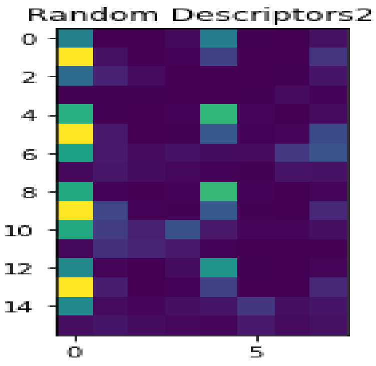


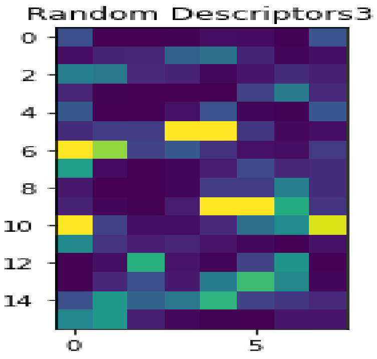
Bag of words histogram:

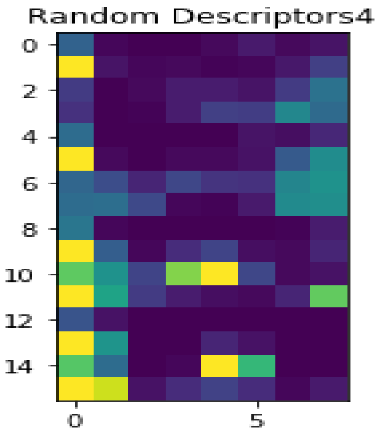


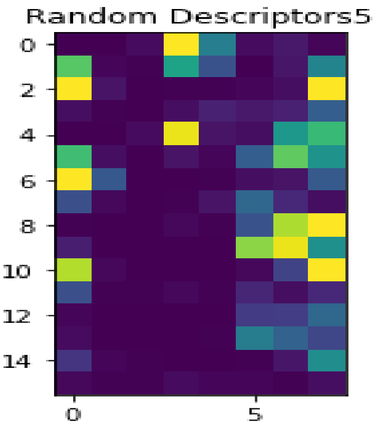
Some random Descriptors(reshape to 16x8):

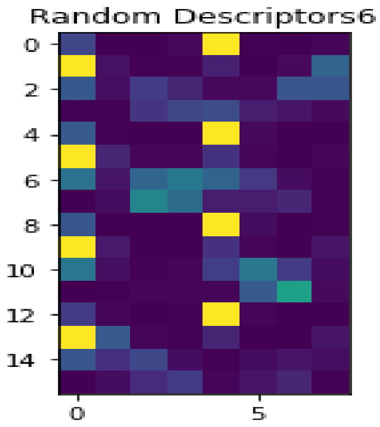







Random Descriptors4
0
2
4
6
8
10
12
14
0
5


Random Descriptors5
0
2
4
6
8
10
12
14
0
5


Random Descriptors6
0
2
4
6
8
10
12
14
0
5

Reading Part:

Read Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories

1. How the proposed approach "Spatial Pyramid Matching" is different from the simple Bag of Visual Words approach and what are the benefits.

To develop representation of an image capable of structured objects segmented from background, Spatial Pyramid Matching uses the approach of pyramid matching of histograms at difference spatial resolutions.

It is different from simple bag of words mainly due to
1. Global representation rather than local order-less features
2. Captures spatial information as well while bag of words discards it
3. Opposite to the approach of using histograms at multiple scales using Gaussians of different variances it divides the image into spatial segments and then computes histograms
4. In pyramid histogram matching matches at higher resolution are given more weightage.

This clever technique of capturing structured information improved dramatically results and has many advantages like

1. Capable of capturing structured information of objects segmented from its background
2. Categorizing features into strong and weak features and uses weighted matching
3. Captures spatial information in images
4. Power to represent highly variable datasets and global scene statistics
5. simple and computationally efficient

2. Interpret the “Pyramid Match Kernel” described in section 3.1 of the paper.

In pyramid matching, we first calculate features and then place grids on the spatial feature space. Cells in the grids at different resolution scales varies in sizes and hence and number of cells as shown in figure. For each cell *i* at each resolution *l* (for each grid) we calculate the number of points/features (*Hl(i)*) (represented by dots, plus and boxes in image) which comes from this cell.

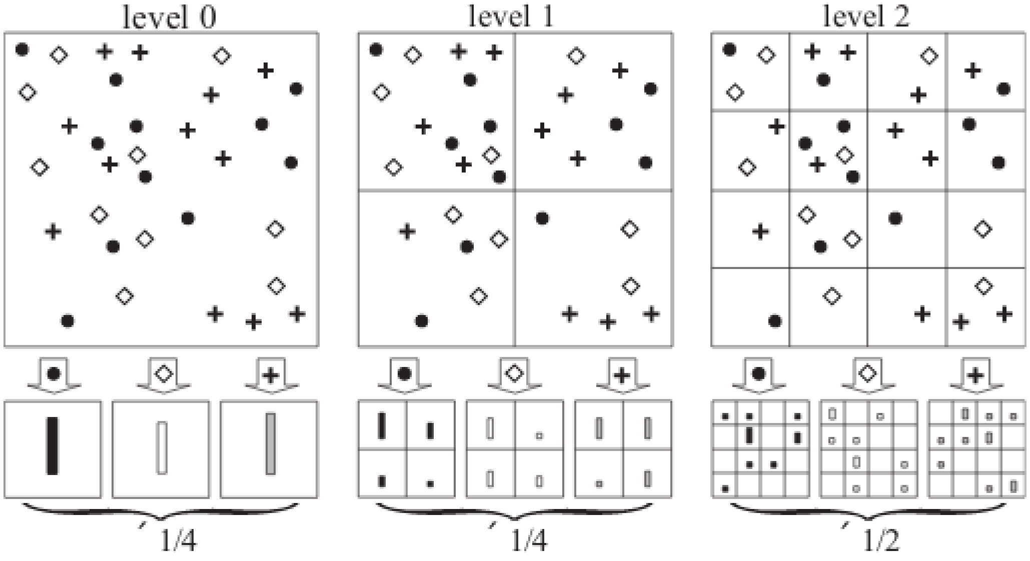


Illustration 1: Grids at three different resolutions

After calculating representations(features) now to find matchings between two images we take histograms at each grid *l* and each cell *i and perform a weighted matching of features or histogram at this level. This task is done by histogram intersection function. This function returns the minimum of the number of features/points in same cells of two images.*

$$\mathcal{I}(H_X^\ell, H_Y^\ell) = \sum_{i=1}^{D} \min\left(H_X^\ell(i), H_Y^\ell(i)\right)$$

Matches at coarser scale (smaller cells) are given more weightage because we want to penalize matches found in larger cells because they involve increasingly dissimilar features.

3. What kind of feature extraction was performed in the paper?

Two kind of features are used in this paper as described earlier in this report
1. Weak features
2. Strong features

***Weak features** (oriented edge points) are points whose gradient magnitude in a given direction exceeds a minimum threshold. These points are extracted at two scales and eight orientations.*

***Strong features** are the dense-SIFT descriptors computed over a grid at the spacing of 8 pixels. Dense-SIFT is used because it gives kind of spatial information and works better for scene classification.*