INFORMATION TECHNOLOGY UNIVERSITY

Report (Assignment 3)

Submitted To: Dr. Mohsen

Submitted By: Sheeza Shabbir

Subject: Computer Vision

Roll No: MSDS20028

Contents

Bag of SIFT features and nearest neighbor classifier 3

Experiment 1: 3

Experiment 2: 4

Experiment 3: 5

Experiment 4: 6

Experiment 5: 7

Experiment 6: 8

Experiment 7: 9

Experiment 8: 10

Experiment 9: 11

Experiment 10: 12

Experiment 11: 13

Experiment 12: 14

Experiment 13: 15

TP,FP,FN Visulaization Table 16

Predictions 17

KNN Discussion: 18

Analysis of experiments: 19

Bag of SIFT features and linear SVM classifier 20

Experiment 1: 20

Experiment 2: 21

Experiment 3: 22

Experiment 4: 23

Experiment 5: 24

TP,FP,FN Visulaization Table 25

Predictions: 26

Experiment 6: 28

TP,FP,FN Visulaization Table 29

Predictions: 30

Experiment 7: 32

TP,FP,FN Visulaization Table 33

Predictions 34

Experiment 8: 36

TP,FP,FN Visulaization Table 37

Predictions: 38

SVM Discussion: 39

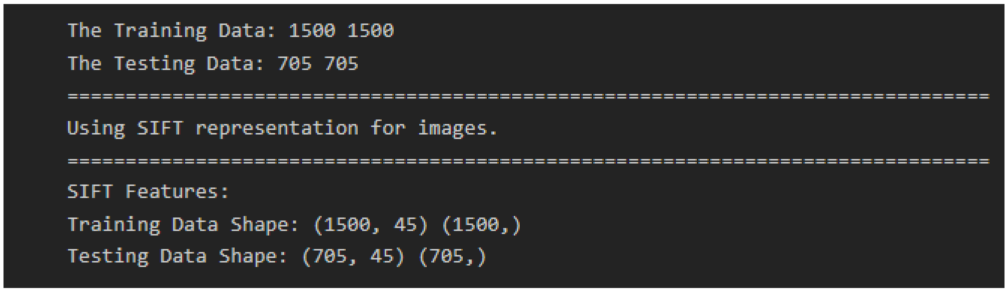
Analysis of experiments: 39

Reading Part 40

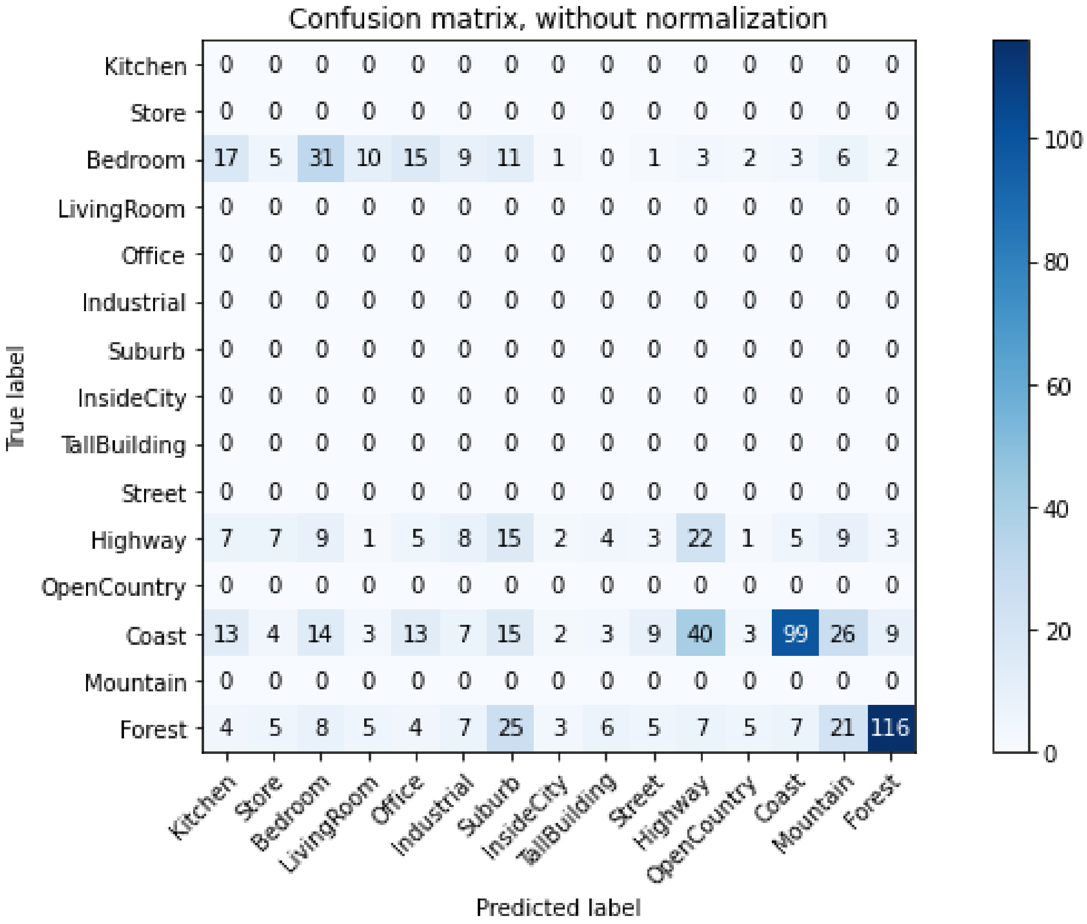
Bag of SIFT features and nearest neighbor classifier

Experiment 1:

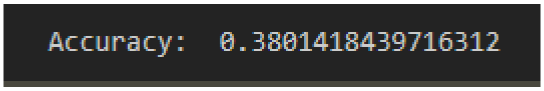
Features	Vocab size	KMeans(# of K)	Knn(# of K)
30	45	45	34



```
The Training Data: 1500 1500
The Testing Data: 705 705
================================================================================
Using SIFT representation for images.
================================================================================
SIFT Features:
Training Data Shape: (1500, 45) (1500,)
Testing Data Shape: (705, 45) (705,)
```



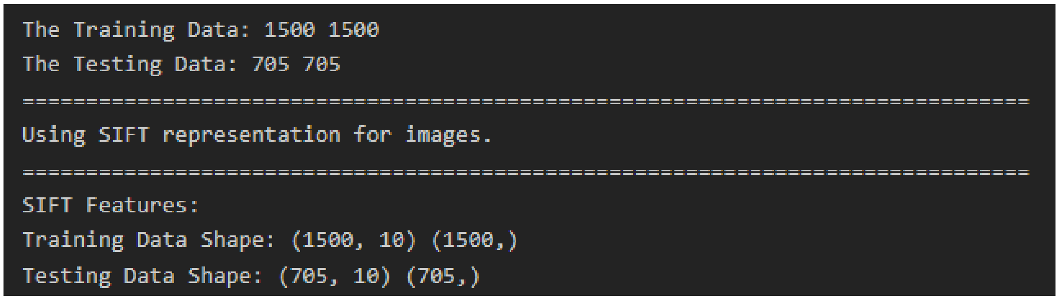
	Kitchen	Store	Bedroom	LivingRoom	Office	Industrial	Suburb	InsideCity	TallBuilding	Street	Highway	OpenCountry	Coast	Mountain	Forest
Kitchen	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Store	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bedroom	17	5	31	10	15	9	11	1	0	1	3	2	3	6	2
LivingRoom	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Office	0	0	0	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	7	7	9	1	5	8	15	2	4	3	22	1	5	9	3
OpenCountry	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Coast	13	4	14	3	13	7	15	2	3	9	40	3	99	26	9
Mountain	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Forest	4	5	8	5	4	7	25	3	6	5	7	5	7	21	116



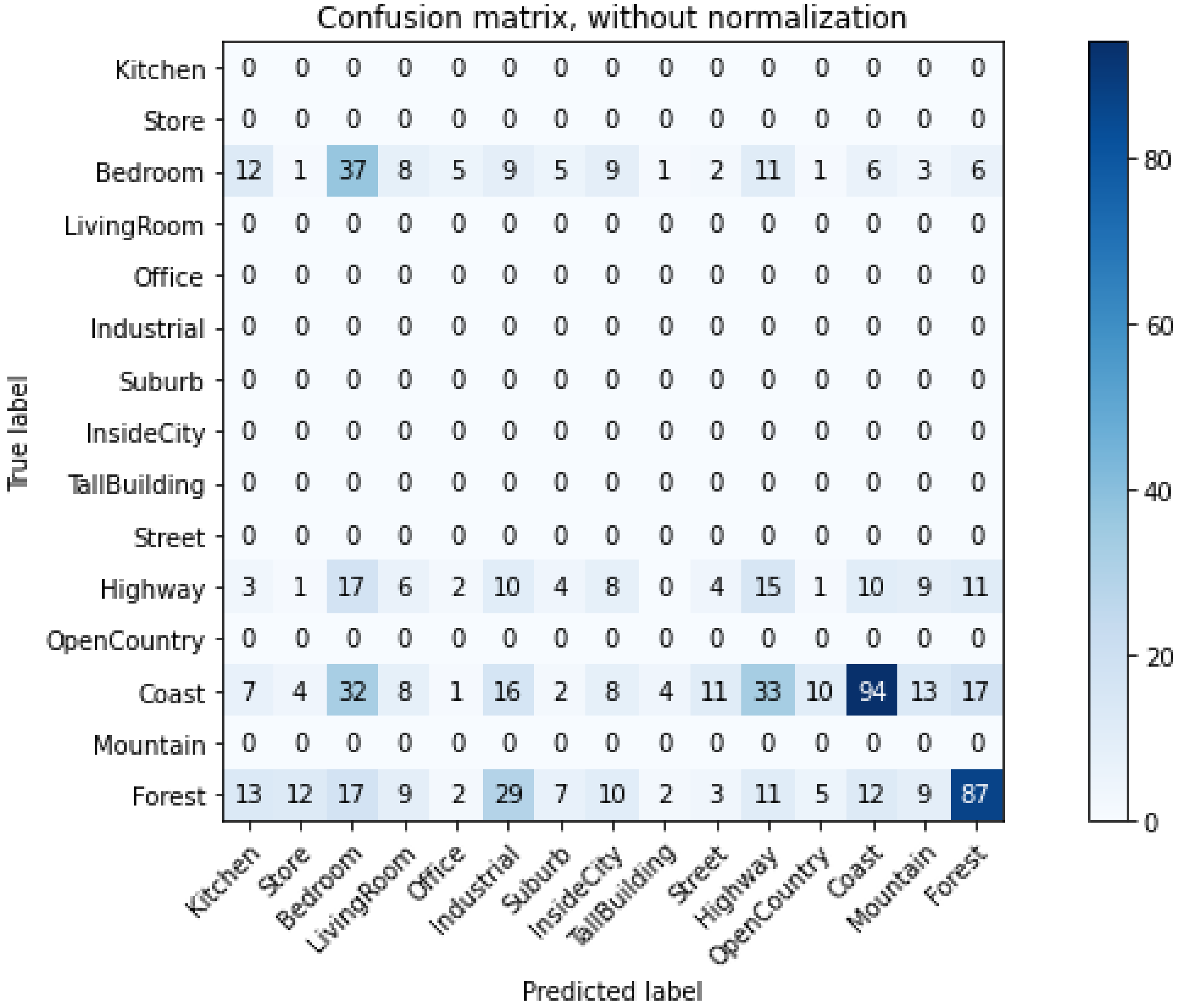
```
Accuracy:  0.3801418439716312
```

Experiment 2:

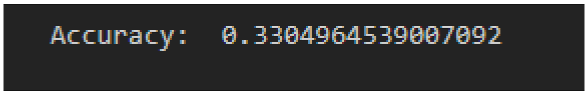
Features	Vocab size	KMeans(# of K)	Knn(# of K)
30	10	10	3



```
The Training Data: 1500 1500
The Testing Data: 705 705
================================================================================
Using SIFT representation for images.
================================================================================
SIFT Features:
Training Data Shape: (1500, 10) (1500,)
Testing Data Shape: (705, 10) (705,)
```



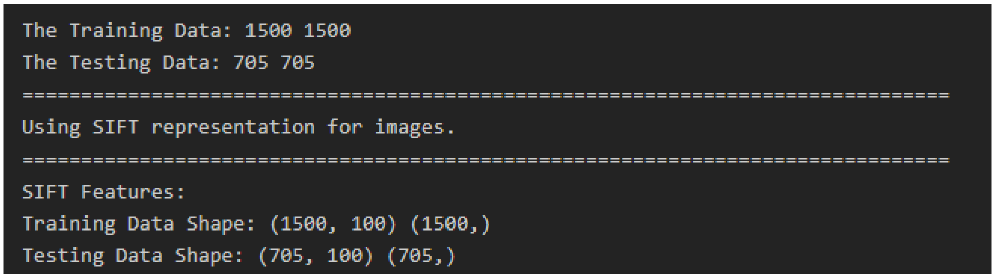
	Kitchen	Store	Bedroom	LivingRoom	Office	Industrial	Suburb	InsideCity	TallBuilding	Street	Highway	OpenCountry	Coast	Mountain	Forest
Kitchen	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Store	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bedroom	12	1	37	8	5	9	5	9	1	2	11	1	6	3	6
LivingRoom	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Office	0	0	0	0	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	3	1	17	6	2	10	4	8	0	4	15	1	10	9	11
OpenCountry	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Coast	7	4	32	8	1	16	2	8	4	11	33	10	94	13	17
Mountain	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Forest	13	12	17	9	2	29	7	10	2	3	11	5	12	9	87



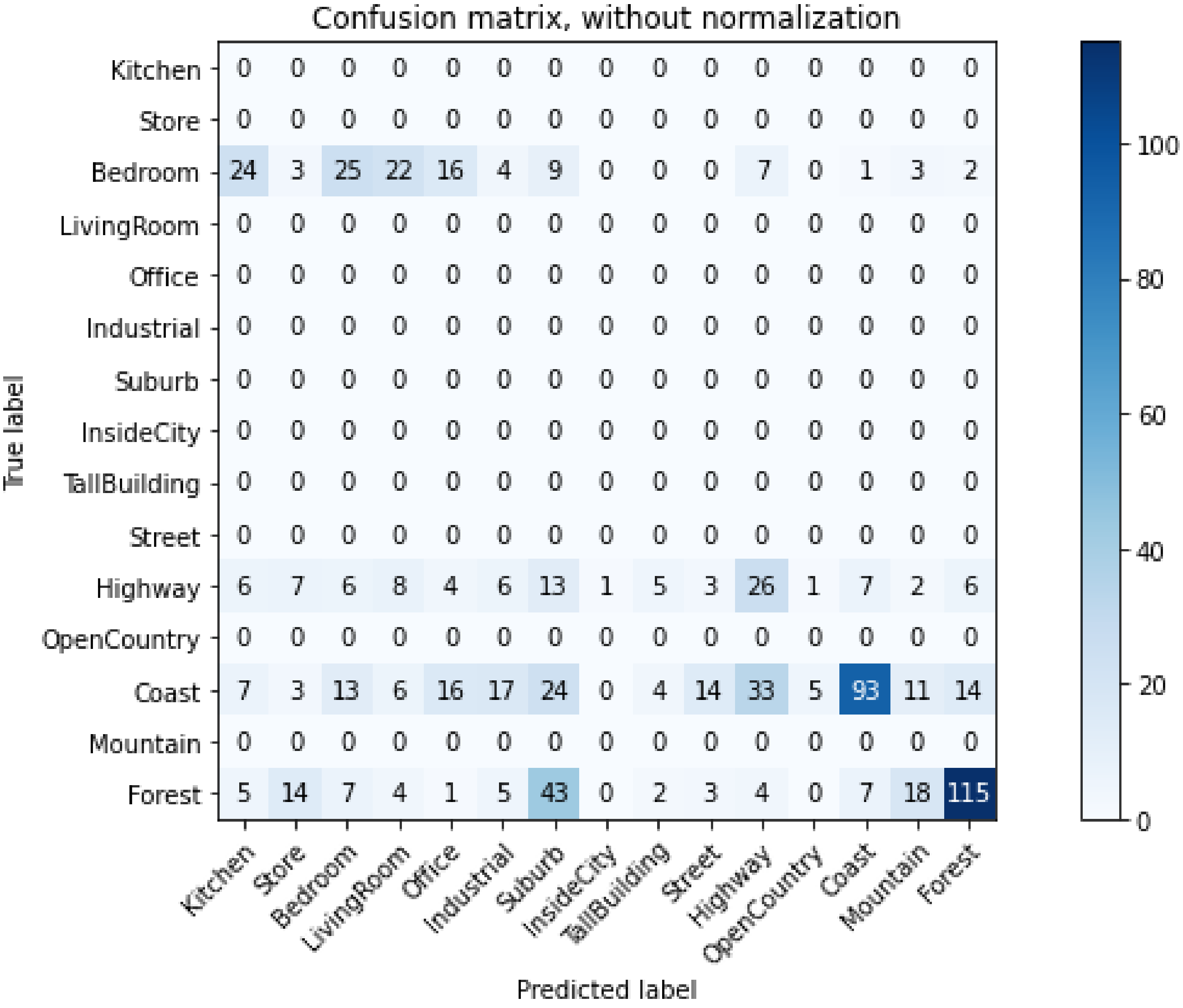
```
Accuracy:  0.3304964539007092
```

Experiment 3:

Features	Vocab size	KMeans(# of K)	Knn(# of K)
30	100	100	48

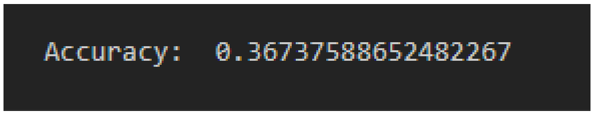


```
The Training Data: 1500 1500
The Testing Data: 705 705
==============================================================================
Using SIFT representation for images.
==============================================================================
SIFT Features:
Training Data Shape: (1500, 100) (1500,)
Testing Data Shape: (705, 100) (705,)
```



Confusion matrix, without normalization

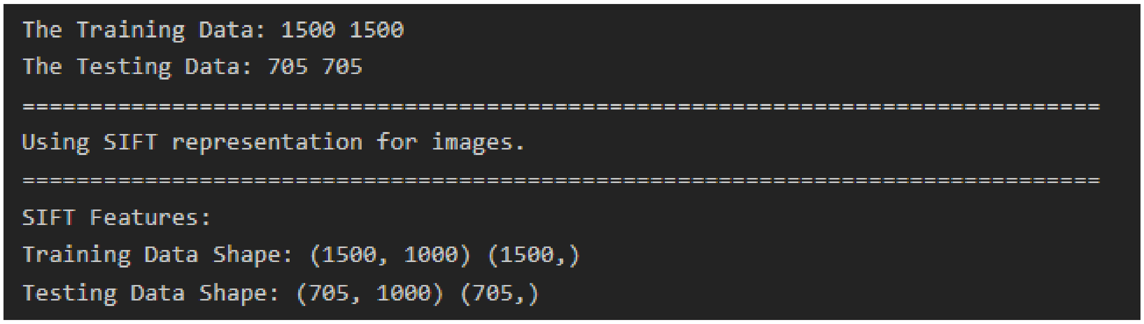
True label \ Predicted label	Kitchen	Store	Bedroom	LivingRoom	Office	Industrial	Suburb	InsideCity	TallBuilding	Street	Highway	OpenCountry	Coast	Mountain	Forest
Kitchen	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Store	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bedroom	24	3	25	22	16	4	9	0	0	0	7	0	1	3	2
LivingRoom	0	0	0	0	0	0	0	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	6	7	6	8	4	6	13	1	5	3	26	1	7	2	6
OpenCountry	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Coast	7	3	13	6	16	17	24	0	4	14	33	5	93	11	14
Mountain	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Forest	5	14	7	4	1	5	43	0	2	3	4	0	7	18	115



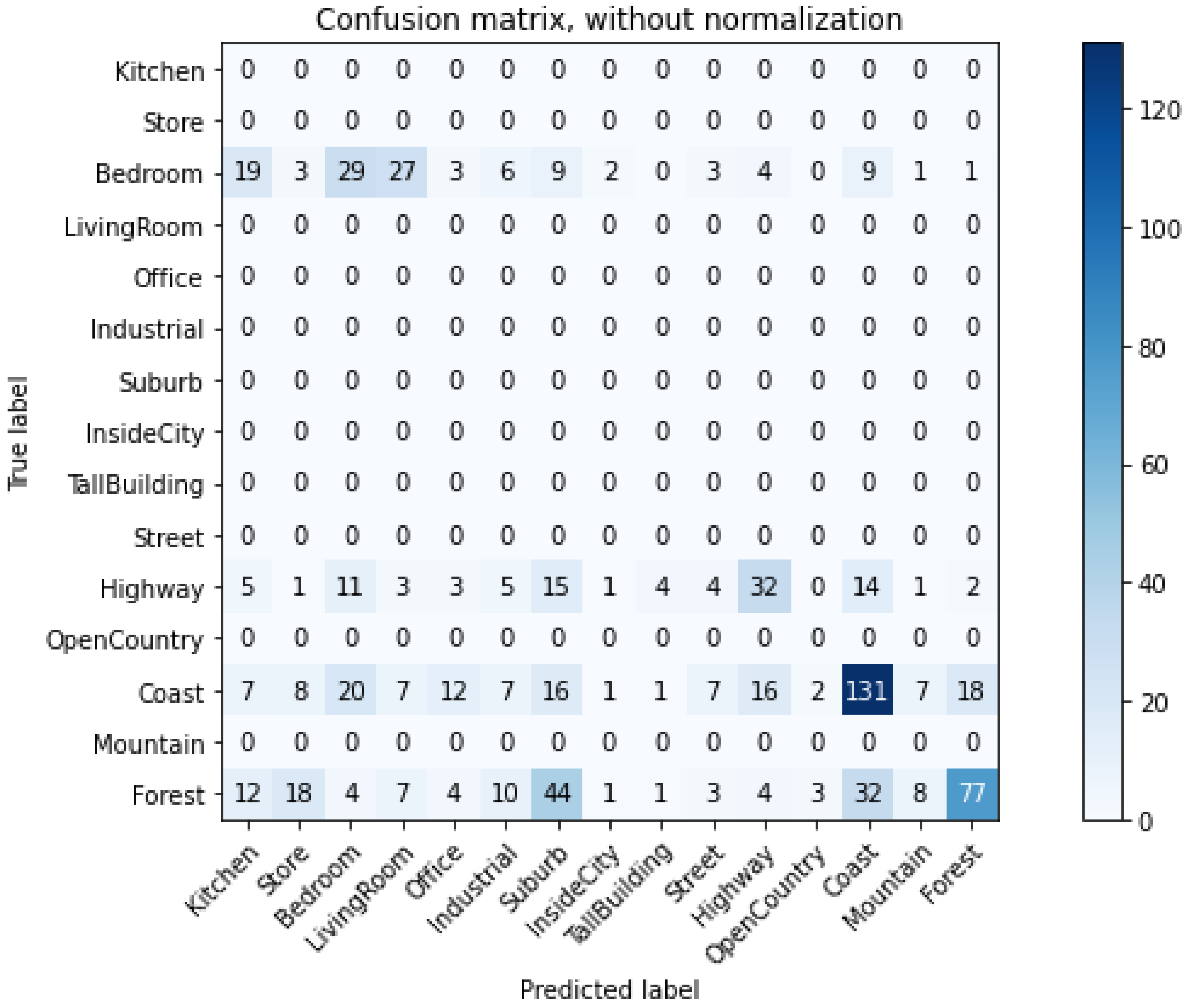
```
Accuracy:  0.36737588652482267
```

Experiment 4:

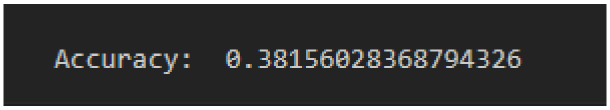
Features	Vocab size	KMeans(# of K)	Knn(# of K)
30	1000	1000	24



```
The Training Data: 1500 1500
The Testing Data: 705 705
================================================================================
Using SIFT representation for images.
================================================================================
SIFT Features:
Training Data Shape: (1500, 1000) (1500,)
Testing Data Shape: (705, 1000) (705,)
```



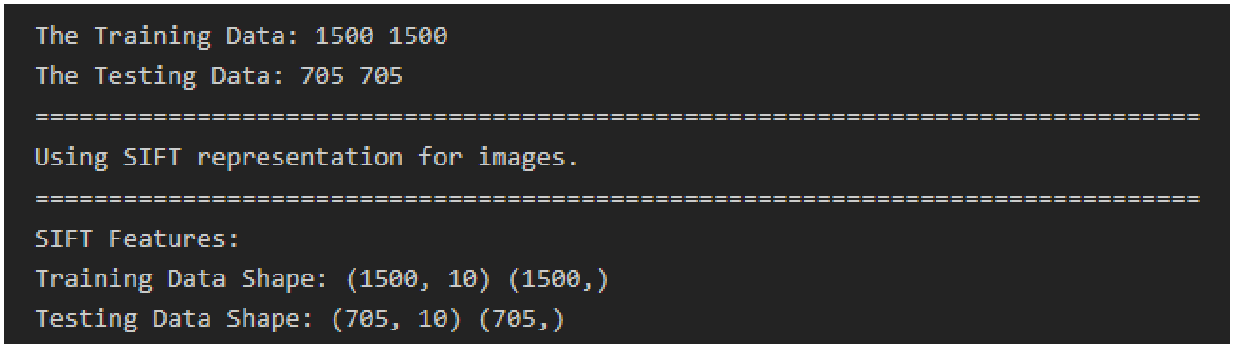
	Kitchen	Store	Bedroom	LivingRoom	Office	Industrial	Suburb	InsideCity	TallBuilding	Street	Highway	OpenCountry	Coast	Mountain	Forest
Kitchen	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Store	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bedroom	19	3	29	27	3	6	9	2	0	3	4	0	9	1	1
LivingRoom	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Office	0	0	0	0	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	5	1	11	3	3	5	15	1	4	4	32	0	14	1	2
OpenCountry	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Coast	7	8	20	7	12	7	16	1	1	7	16	2	131	7	18
Mountain	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Forest	12	18	4	7	4	10	44	1	1	3	4	3	32	8	77



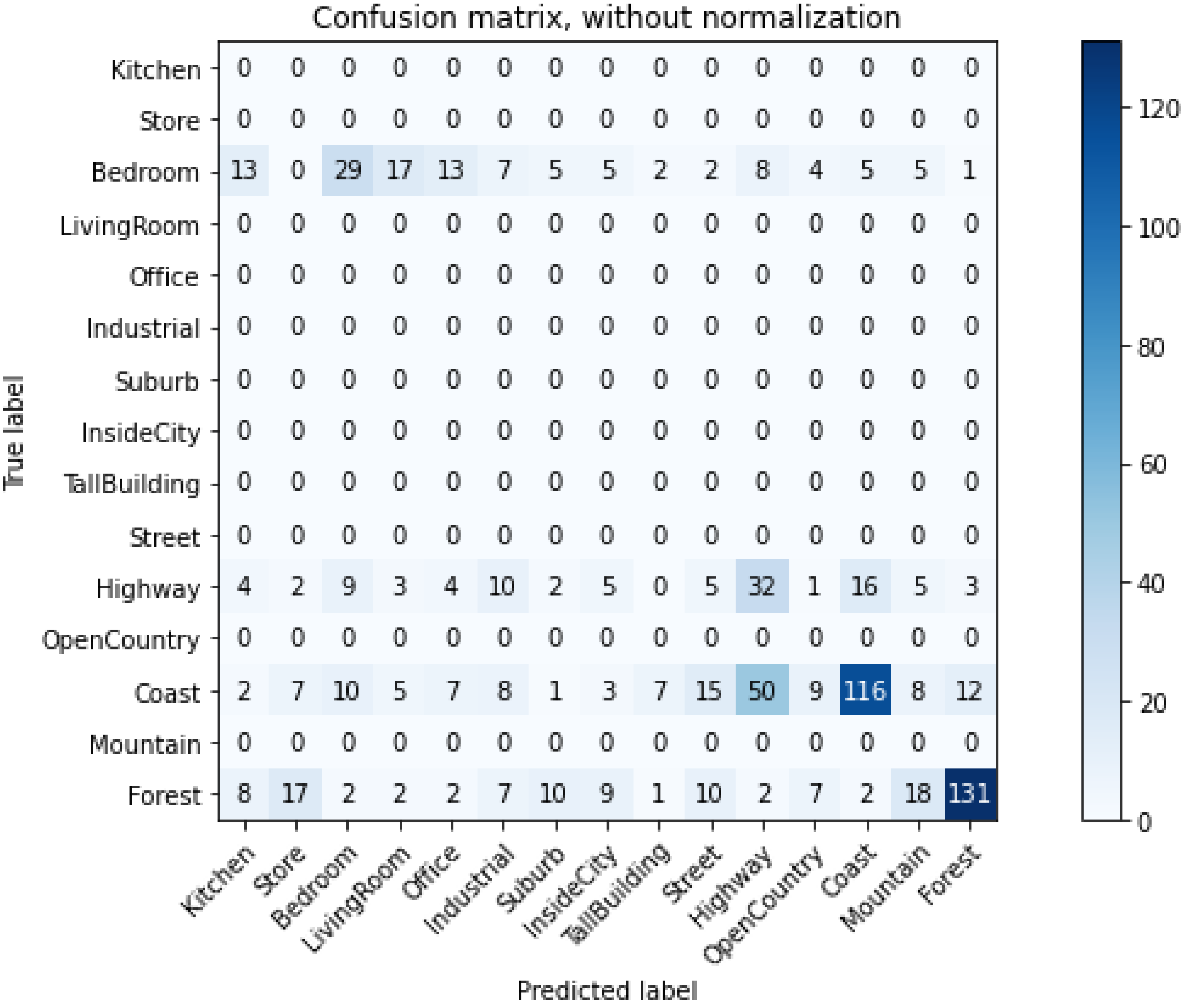
```
Accuracy:  0.38156028368794326
```

Experiment 5:

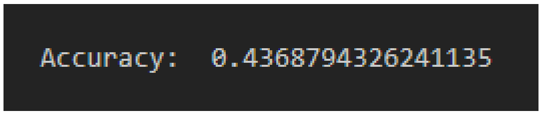
Features	Vocab size	KMeans(# of K)	Knn(# of K)
100	10	10	8



```
The Training Data: 1500 1500
The Testing Data: 705 705
================================================================================
Using SIFT representation for images.
================================================================================
SIFT Features:
Training Data Shape: (1500, 10) (1500,)
Testing Data Shape: (705, 10) (705,)
```



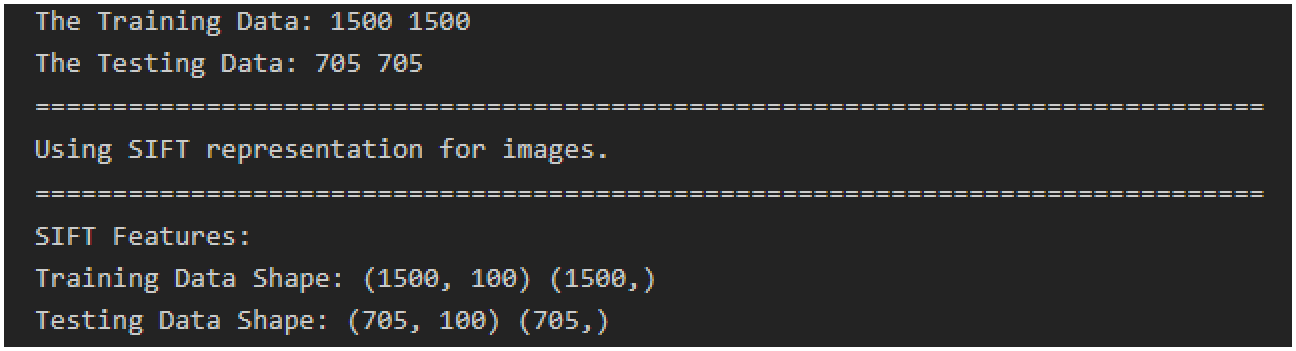
	Kitchen	Store	Bedroom	LivingRoom	Office	Industrial	Suburb	InsideCity	TallBuilding	Street	Highway	OpenCountry	Coast	Mountain	Forest
Kitchen	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Store	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bedroom	13	0	29	17	13	7	5	5	2	2	8	4	5	5	1
LivingRoom	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Office	0	0	0	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	4	2	9	3	4	10	2	5	0	5	32	1	16	5	3
OpenCountry	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Coast	2	7	10	5	7	8	1	3	7	15	50	9	116	8	12
Mountain	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Forest	8	17	2	2	2	7	10	9	1	10	2	7	2	18	131



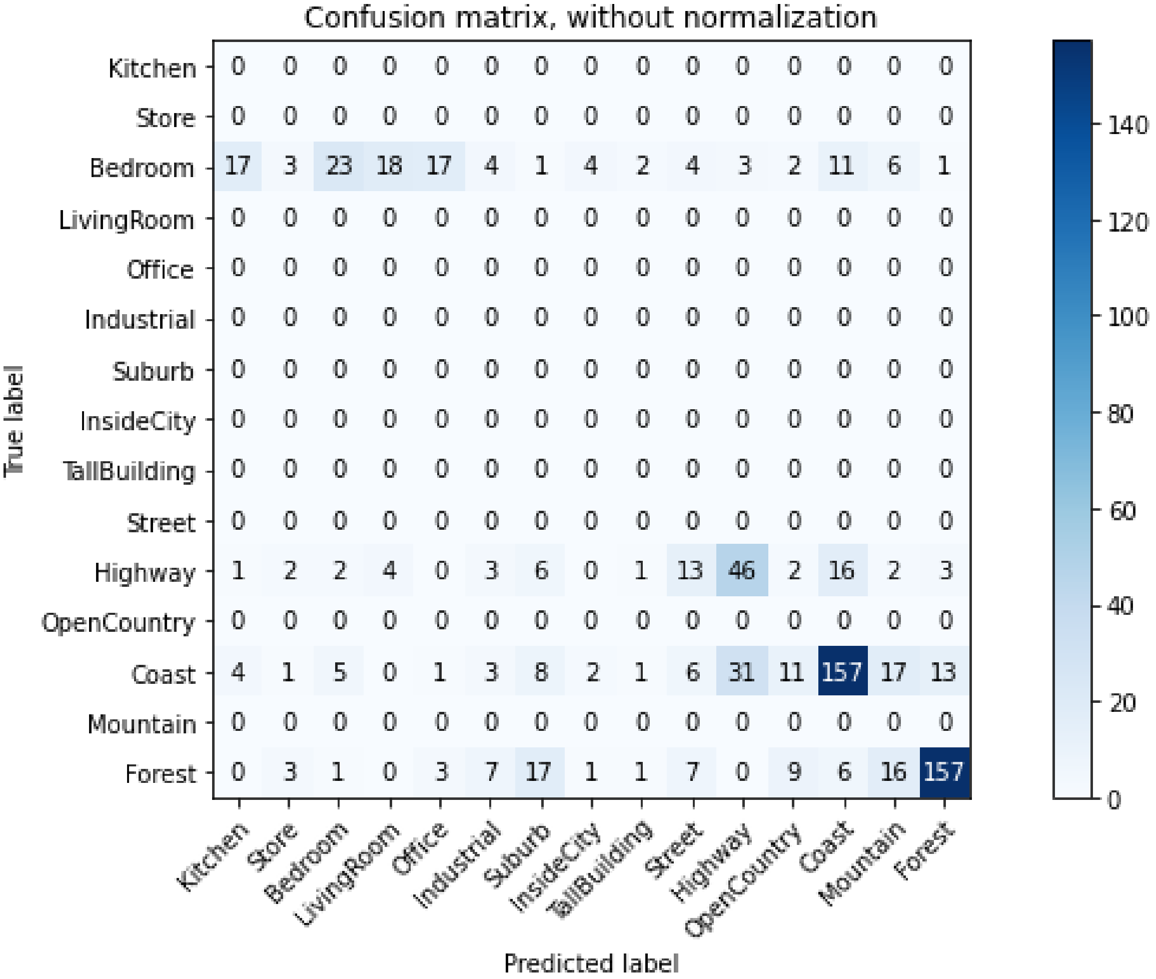
```
Accuracy:  0.4368794326241135
```

Experiment 6:

Features	Vocab size	KMeans(# of K)	Knn(# of K)
100	100	100	20

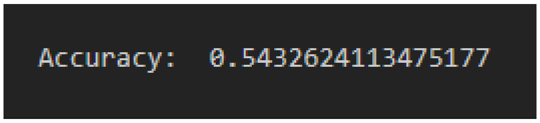


```
The Training Data: 1500 1500
The Testing Data: 705 705
===============================================================================
Using SIFT representation for images.
===============================================================================
SIFT Features:
Training Data Shape: (1500, 100) (1500,)
Testing Data Shape: (705, 100) (705,)
```



Confusion matrix, without normalization

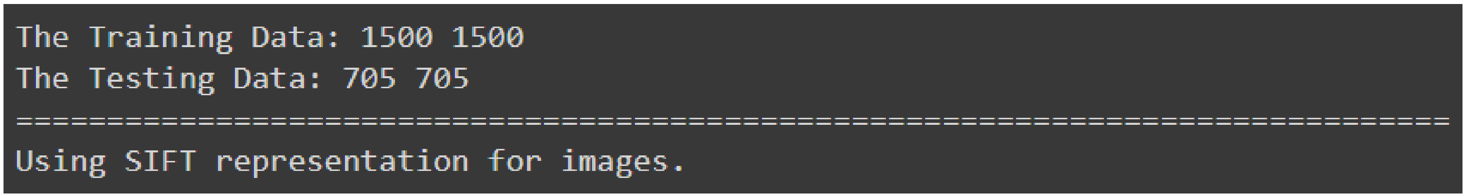
True label \ Predicted label	Kitchen	Store	Bedroom	LivingRoom	Office	Industrial	Suburb	InsideCity	TallBuilding	Street	Highway	OpenCountry	Coast	Mountain	Forest
Kitchen	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Store	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bedroom	17	3	23	18	17	4	1	4	2	4	3	2	11	6	1
LivingRoom	0	0	0	0	0	0	0	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	1	2	2	4	0	3	6	0	1	13	46	2	16	2	3
OpenCountry	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Coast	4	1	5	0	1	3	8	2	1	6	31	11	157	17	13
Mountain	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Forest	0	3	1	0	3	7	17	1	1	7	0	9	6	16	157



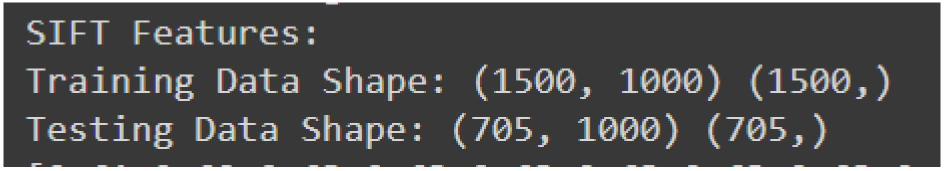
```
Accuracy:  0.5432624113475177
```

Experiment 7:

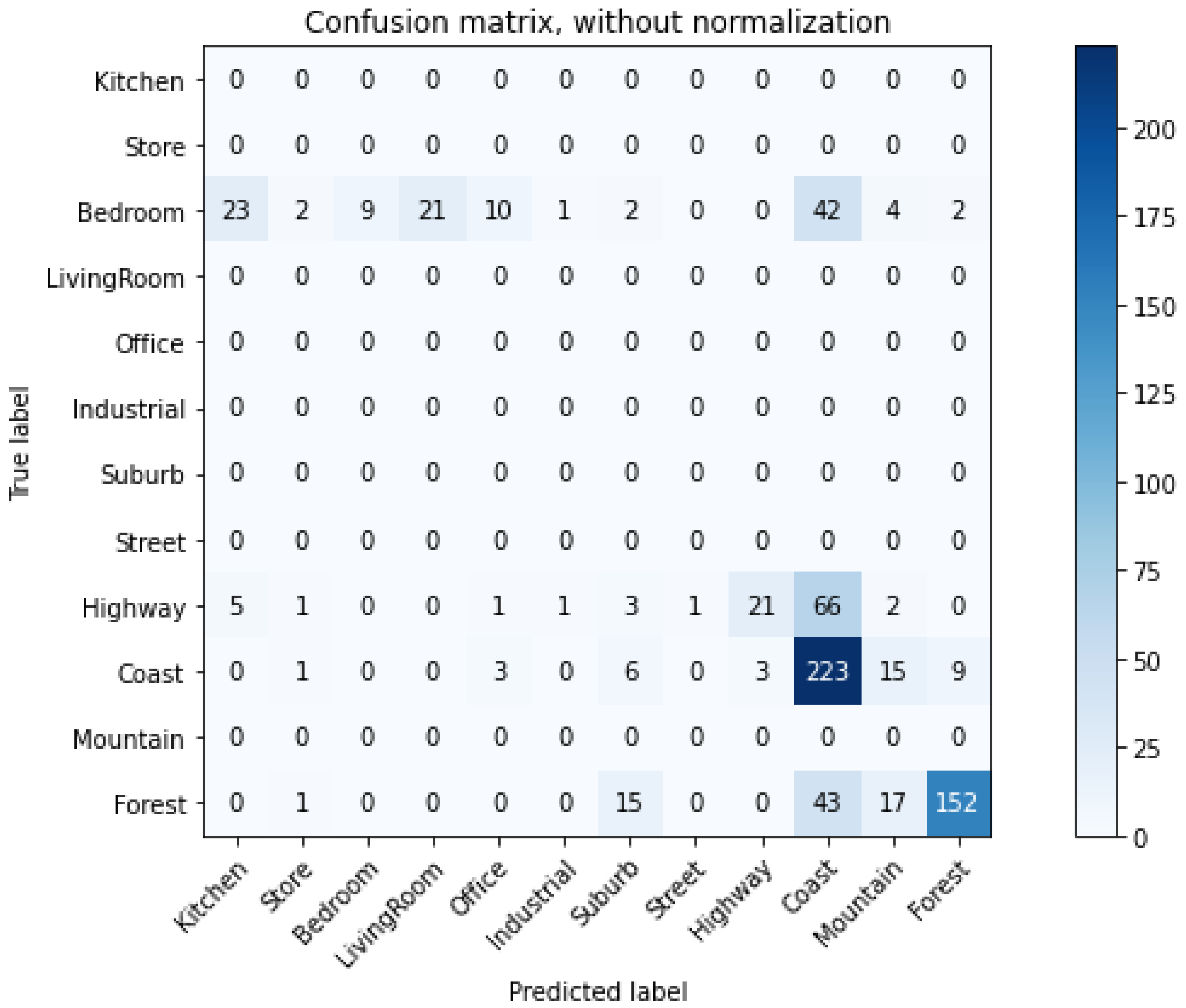
Features	Vocab size	KMeans(# of K)	Knn(# of K)
100	1000	1000	174



```
The Training Data: 1500 1500
The Testing Data: 705 705
==============================================================================
Using SIFT representation for images.
```

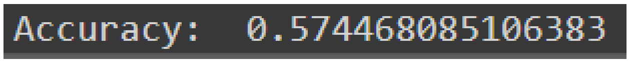


```
SIFT Features:
Training Data Shape: (1500, 1000) (1500,)
Testing Data Shape: (705, 1000) (705,)
```



Confusion matrix, without normalization

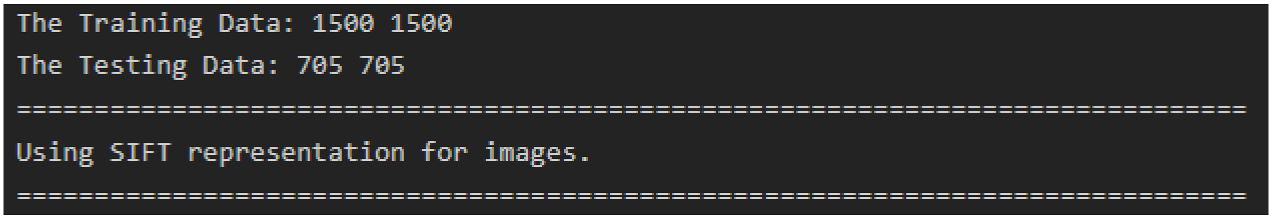
True label \ Predicted label	Kitchen	Store	Bedroom	LivingRoom	Office	Industrial	Suburb	Street	Highway	Coast	Mountain	Forest
Kitchen	0	0	0	0	0	0	0	0	0	0	0	0
Store	0	0	0	0	0	0	0	0	0	0	0	0
Bedroom	23	2	9	21	10	1	2	0	0	42	4	2
LivingRoom	0	0	0	0	0	0	0	0	0	0	0	0
Office	0	0	0	0	0	0	0	0	0	0	0	0
Industrial	0	0	0	0	0	0	0	0	0	0	0	0
Suburb	0	0	0	0	0	0	0	0	0	0	0	0
Street	0	0	0	0	0	0	0	0	0	0	0	0
Highway	5	1	0	0	1	1	3	1	21	66	2	0
Coast	0	1	0	0	3	0	6	0	3	223	15	9
Mountain	0	0	0	0	0	0	0	0	0	0	0	0
Forest	0	1	0	0	0	0	15	0	0	43	17	152



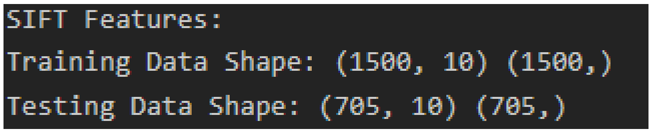
```
Accuracy:  0.574468085106383
```

Experiment 8:

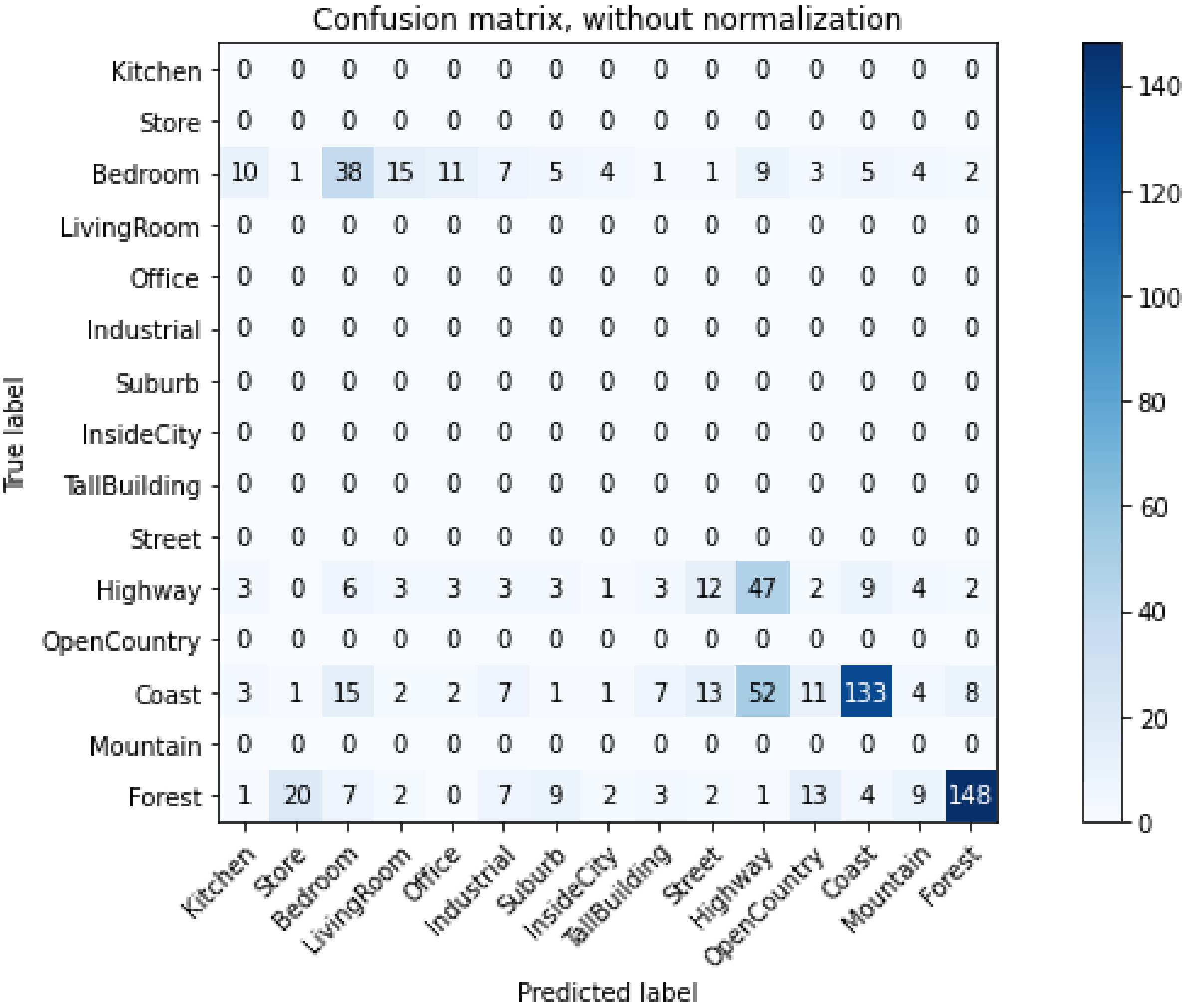
Features	Vocab size	KMeans(# of K)	Knn(# of K)
200	10	10	5



```
The Training Data: 1500 1500
The Testing Data: 705 705
================================================================================
Using SIFT representation for images.
================================================================================
```

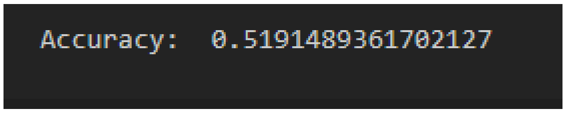


```
SIFT Features:
Training Data Shape: (1500, 10) (1500,)
Testing Data Shape: (705, 10) (705,)
```



Confusion matrix, without normalization

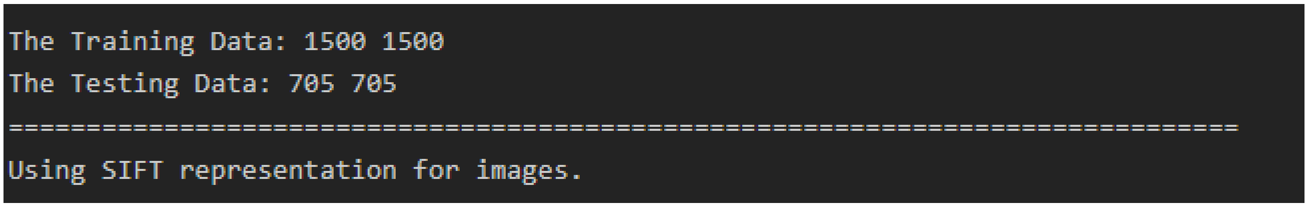
True label \ Predicted label	Kitchen	Store	Bedroom	LivingRoom	Office	Industrial	Suburb	InsideCity	TallBuilding	Street	Highway	OpenCountry	Coast	Mountain	Forest
Kitchen	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Store	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bedroom	10	1	38	15	11	7	5	4	1	1	9	3	5	4	2
LivingRoom	0	0	0	0	0	0	0	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	3	0	6	3	3	3	3	1	3	12	47	2	9	4	2
OpenCountry	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Coast	3	1	15	2	2	7	1	1	7	13	52	11	133	4	8
Mountain	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Forest	1	20	7	2	0	7	9	2	3	2	1	13	4	9	148



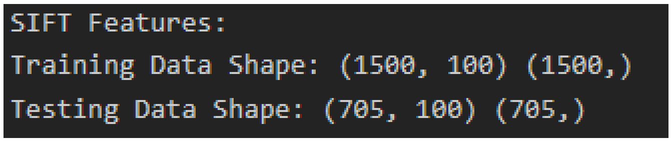
```
Accuracy:  0.5191489361702127
```

Experiment 9:

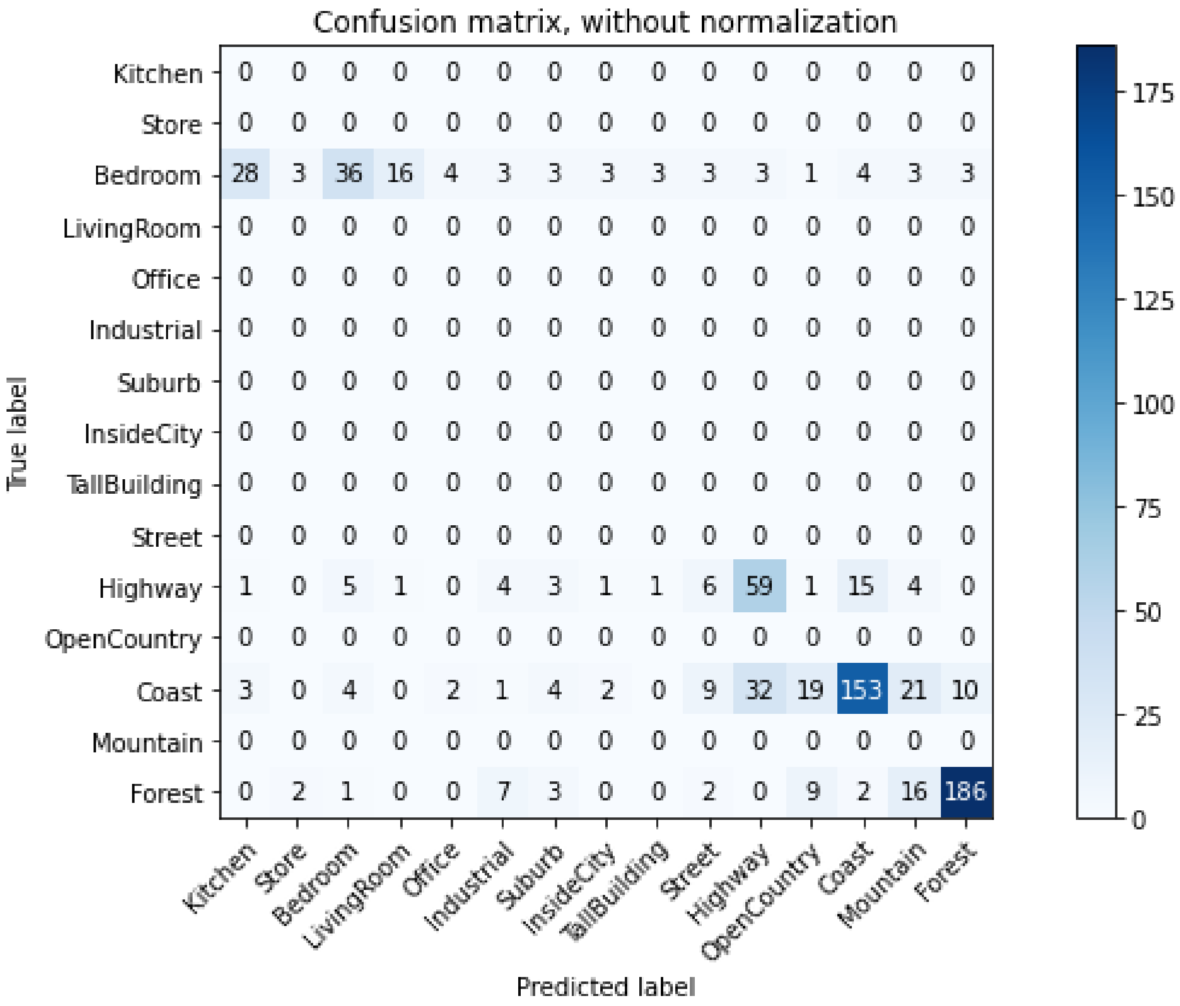
Features	Vocab size	KMeans(# of K)	Knn(# of K)
200	100	100	14



```
The Training Data: 1500 1500
The Testing Data: 705 705
================================================================================
Using SIFT representation for images.
```

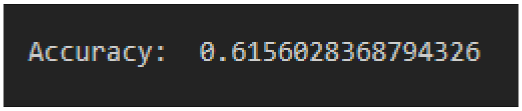


```
SIFT Features:
Training Data Shape: (1500, 100) (1500,)
Testing Data Shape: (705, 100) (705,)
```



Confusion matrix, without normalization

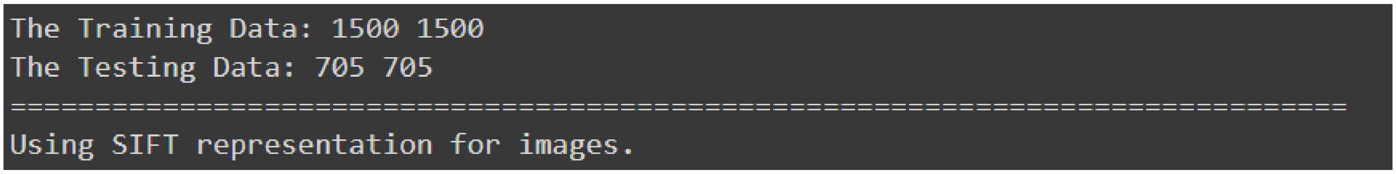
	Kitchen	Store	Bedroom	LivingRoom	Office	Industrial	Suburb	InsideCity	TallBuilding	Street	Highway	OpenCountry	Coast	Mountain	Forest
Kitchen	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Store	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bedroom	28	3	36	16	4	3	3	3	3	3	3	1	4	3	3
LivingRoom	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Office	0	0	0	0	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	1	0	5	1	0	4	3	1	1	6	59	1	15	4	0
OpenCountry	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Coast	3	0	4	0	2	1	4	2	0	9	32	19	153	21	10
Mountain	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Forest	0	2	1	0	0	7	3	0	0	2	0	9	2	16	186



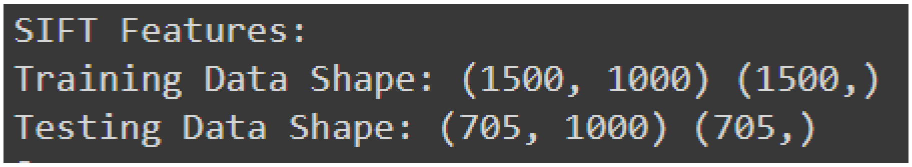
```
Accuracy:  0.6156028368794326
```

Experiment 10:

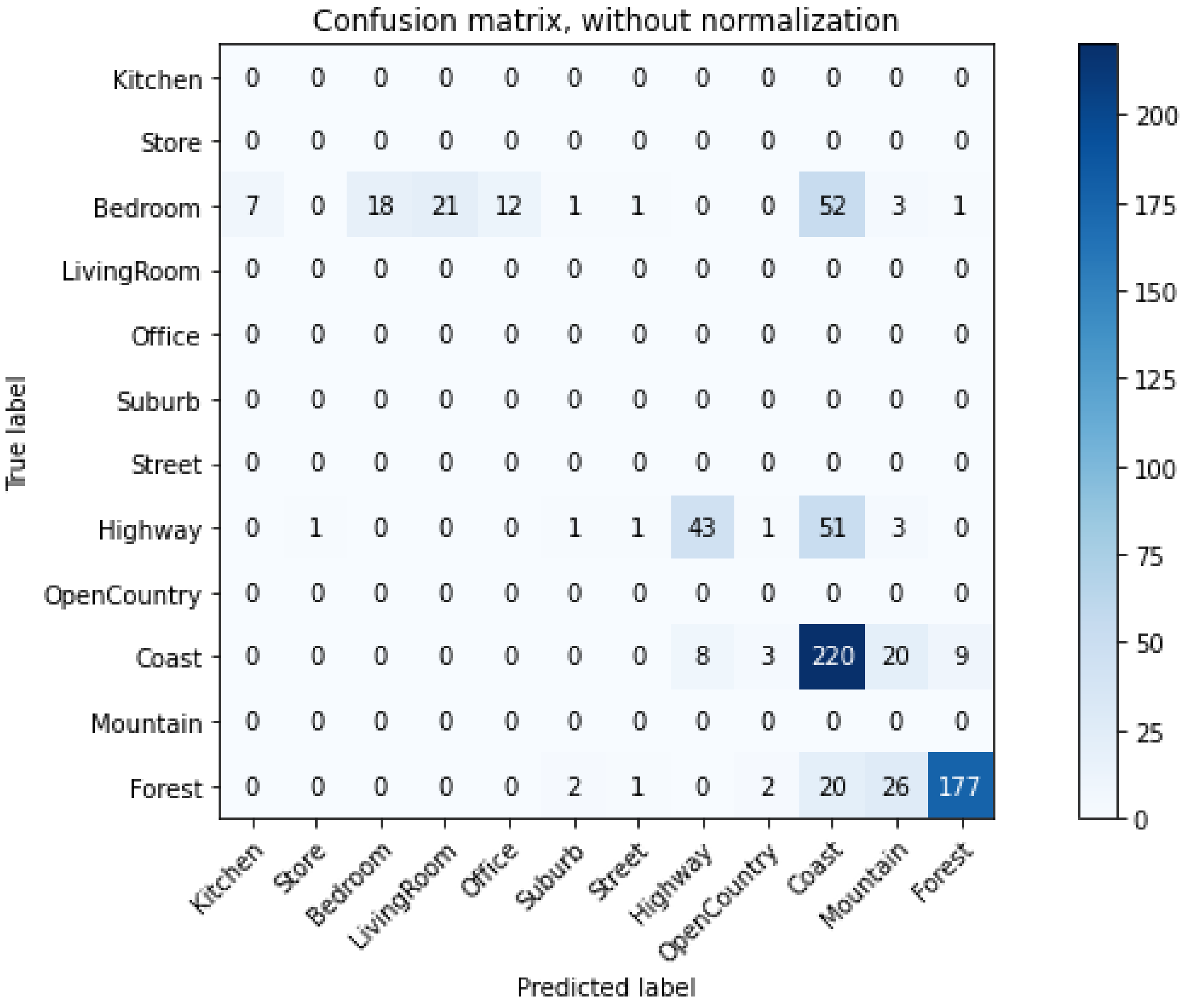
Features	Vocab size	KMeans(# of K)	Knn(# of K)
200	1000	1000	342



```
The Training Data: 1500 1500
The Testing Data: 705 705
================================================================================
Using SIFT representation for images.
```

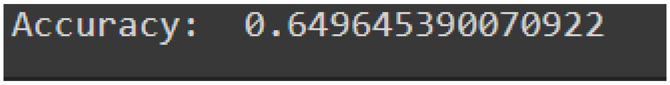


```
SIFT Features:
Training Data Shape: (1500, 1000) (1500,)
Testing Data Shape: (705, 1000) (705,)
```



Confusion matrix, without normalization

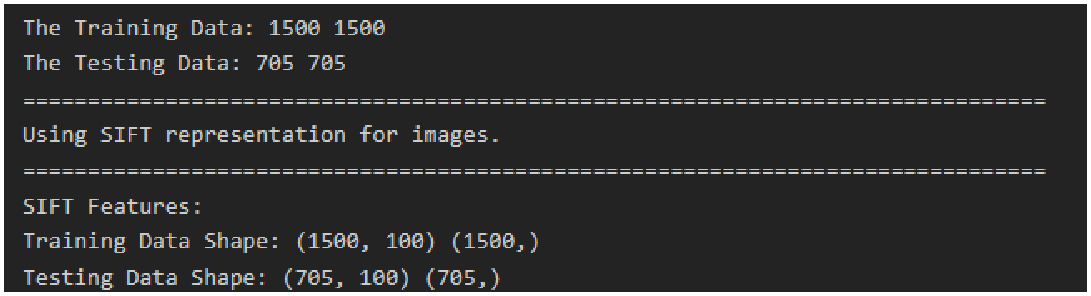
True label \ Predicted label	Kitchen	Store	Bedroom	LivingRoom	Office	Suburb	Street	Highway	OpenCountry	Coast	Mountain	Forest
Kitchen	0	0	0	0	0	0	0	0	0	0	0	0
Store	0	0	0	0	0	0	0	0	0	0	0	0
Bedroom	7	0	18	21	12	1	1	0	0	52	3	1
LivingRoom	0	0	0	0	0	0	0	0	0	0	0	0
Office	0	0	0	0	0	0	0	0	0	0	0	0
Suburb	0	0	0	0	0	0	0	0	0	0	0	0
Street	0	0	0	0	0	0	0	0	0	0	0	0
Highway	0	1	0	0	0	1	1	43	1	51	3	0
OpenCountry	0	0	0	0	0	0	0	0	0	0	0	0
Coast	0	0	0	0	0	0	0	8	3	220	20	9
Mountain	0	0	0	0	0	0	0	0	0	0	0	0
Forest	0	0	0	0	0	2	1	0	2	20	26	177



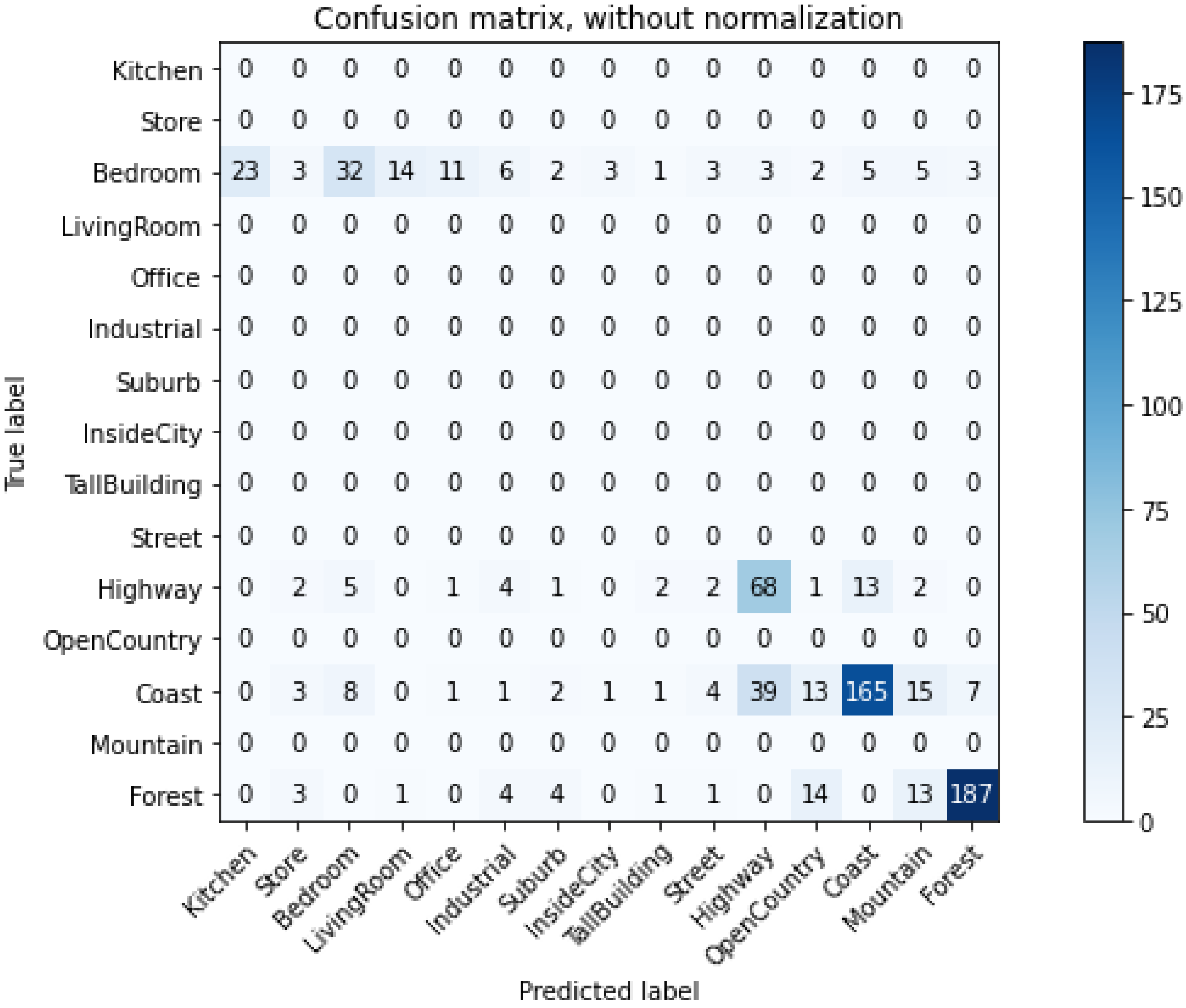
```
Accuracy:  0.649645390070922
```

Experiment 11:

Features	Vocab size	KMeans(# of K)	Knn(# of K)
300	100	100	8

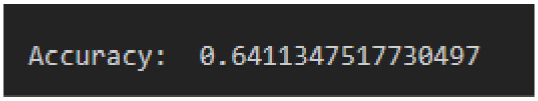


```
The Training Data: 1500 1500
The Testing Data: 705 705
================================================================================
Using SIFT representation for images.
================================================================================
SIFT Features:
Training Data Shape: (1500, 100) (1500,)
Testing Data Shape: (705, 100) (705,)
```



Confusion matrix, without normalization

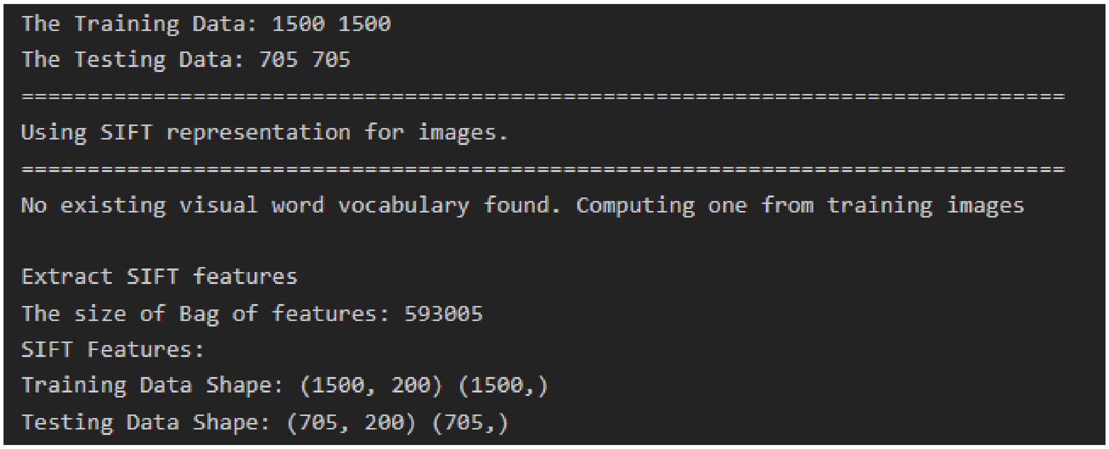
	Kitchen	Store	Bedroom	LivingRoom	Office	Industrial	Suburb	InsideCity	TallBuilding	Street	Highway	OpenCountry	Coast	Mountain	Forest
Kitchen	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Store	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bedroom	23	3	32	14	11	6	2	3	1	3	3	2	5	5	3
LivingRoom	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Office	0	0	0	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	0	2	5	0	1	4	1	0	2	2	68	1	13	2	0
OpenCountry	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Coast	0	3	8	0	1	1	2	1	1	4	39	13	165	15	7
Mountain	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Forest	0	3	0	1	0	4	4	0	1	1	0	14	0	13	187



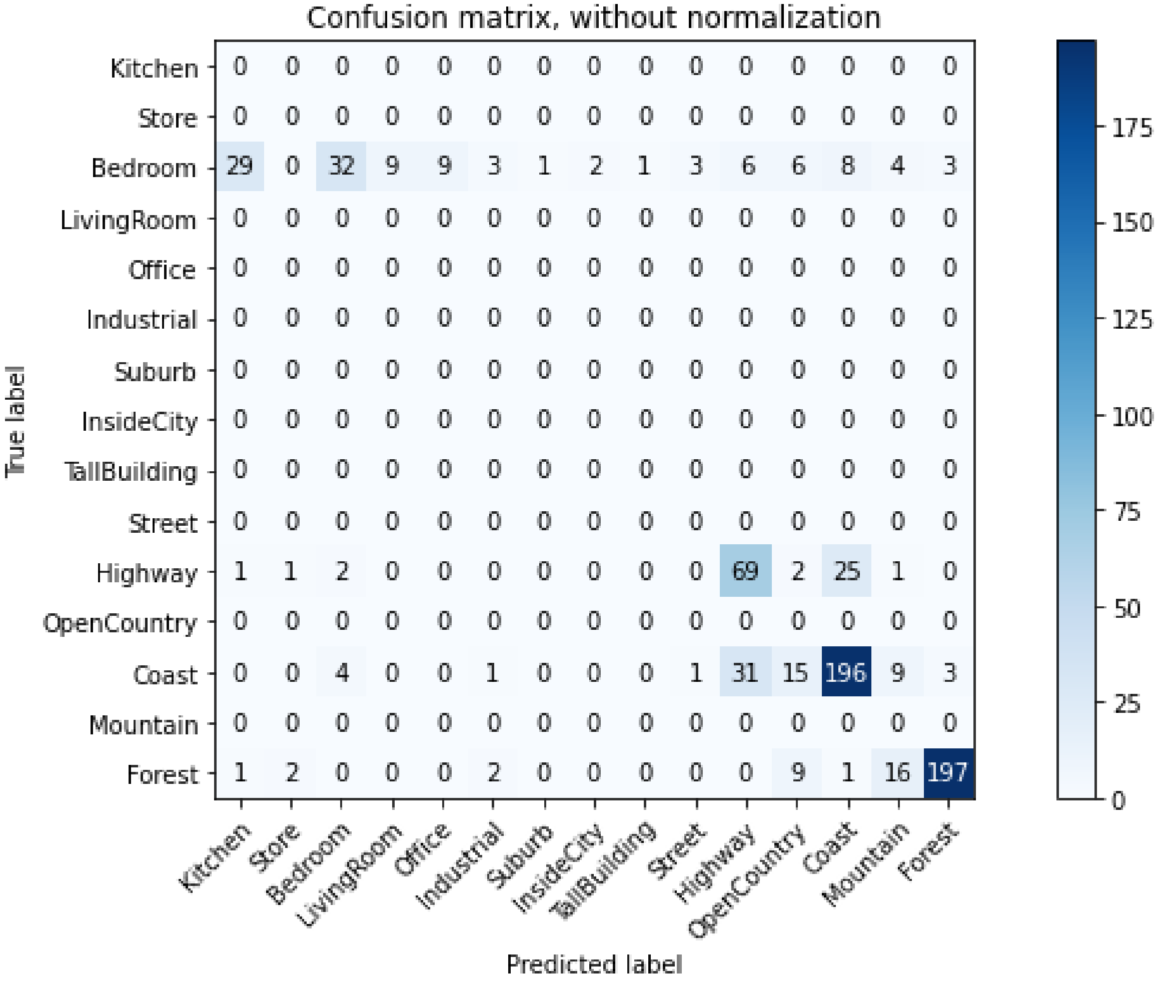
```
Accuracy:  0.6411347517730497
```

Experiment 12:

Features	Vocab size	KMeans(# of K)	Knn(# of K)
500	200	200	9

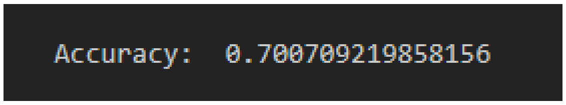


```
The Training Data: 1500 1500
The Testing Data: 705 705
================================================================================
Using SIFT representation for images.
================================================================================
No existing visual word vocabulary found. Computing one from training images

Extract SIFT features
The size of Bag of features: 593005
SIFT Features:
Training Data Shape: (1500, 200) (1500,)
Testing Data Shape: (705, 200) (705,)
```



Confusion matrix, without normalization

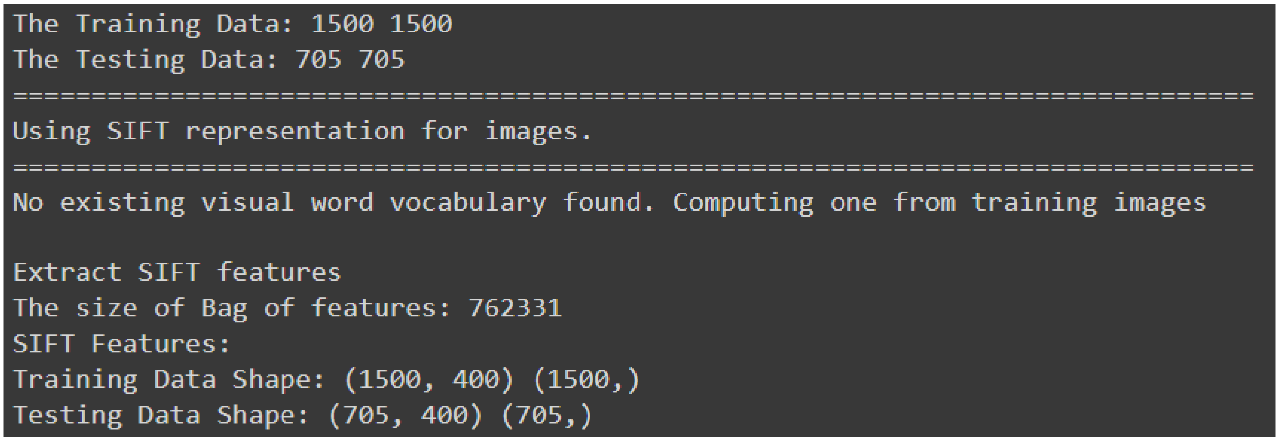
	Kitchen	Store	Bedroom	LivingRoom	Office	Industrial	Suburb	InsideCity	TallBuilding	Street	Highway	OpenCountry	Coast	Mountain	Forest
Kitchen	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Store	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bedroom	29	0	32	9	9	3	1	2	1	3	6	6	8	4	3
LivingRoom	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Office	0	0	0	0	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	1	1	2	0	0	0	0	0	0	0	69	2	25	1	0
OpenCountry	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Coast	0	0	4	0	0	1	0	0	0	1	31	15	196	9	3
Mountain	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Forest	1	2	0	0	0	2	0	0	0	0	0	9	1	16	197



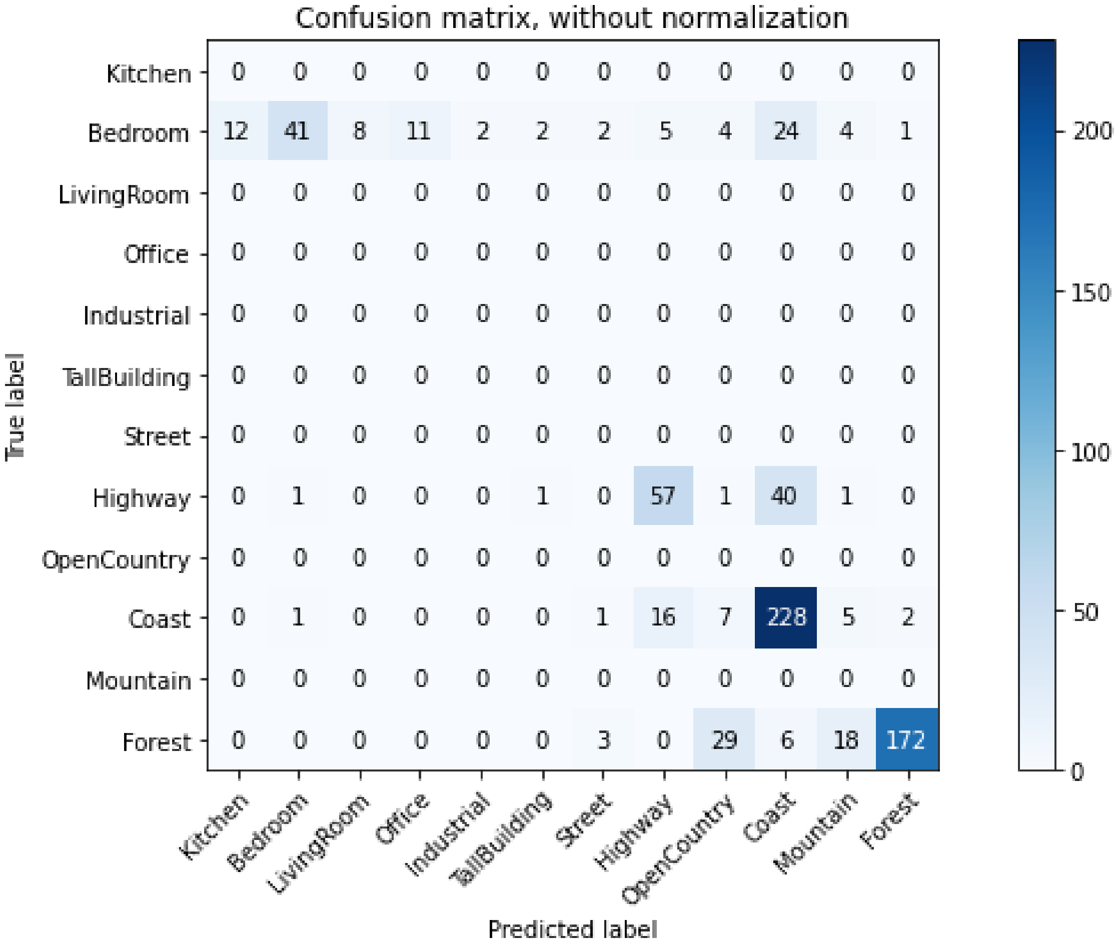
```
Accuracy:  0.700709219858156
```

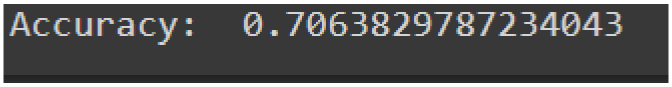
Experiment 13:

Features	Vocab size	KMeans(# of K)	Knn(# of K)
random	400	400	14



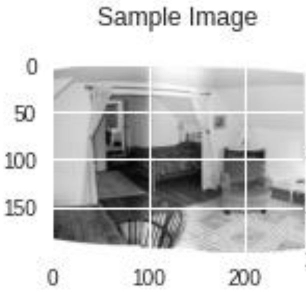
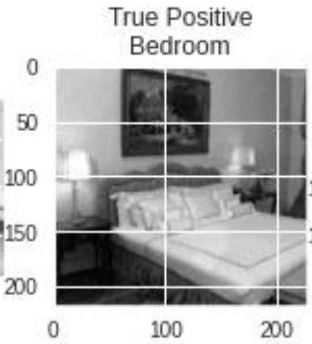
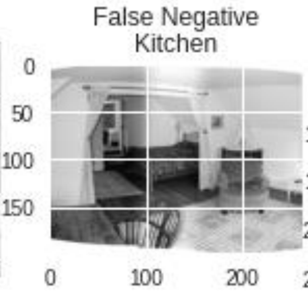
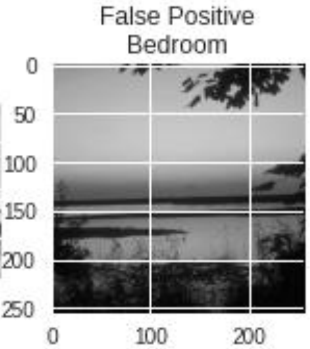
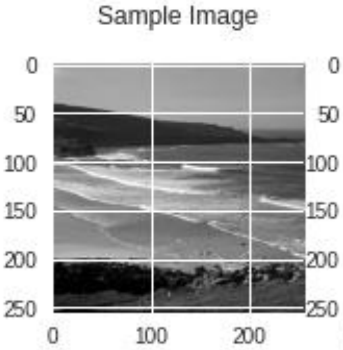
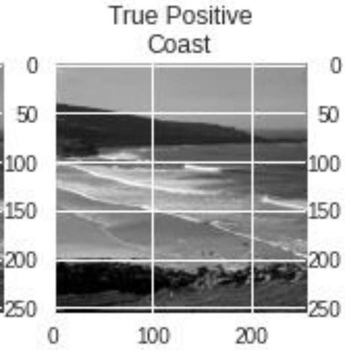
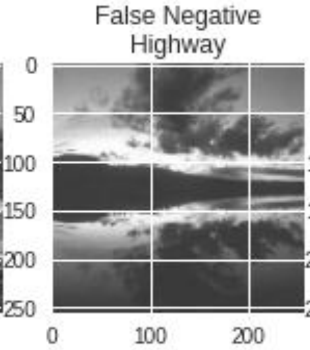
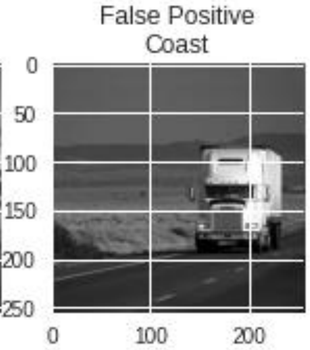
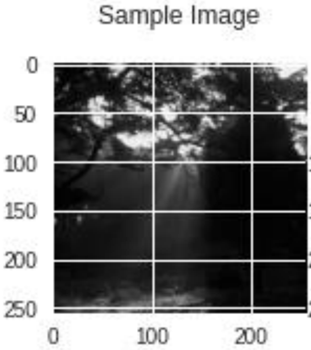
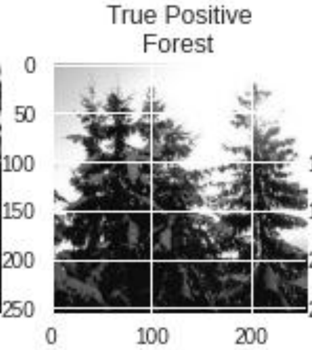
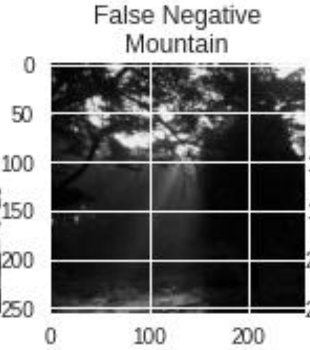
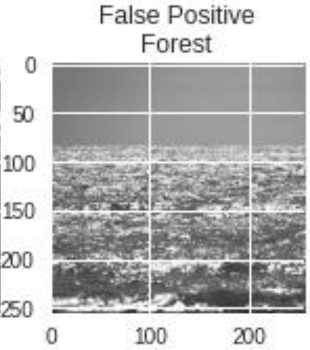
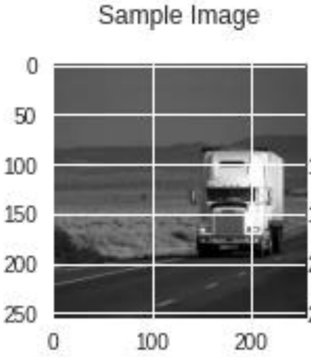
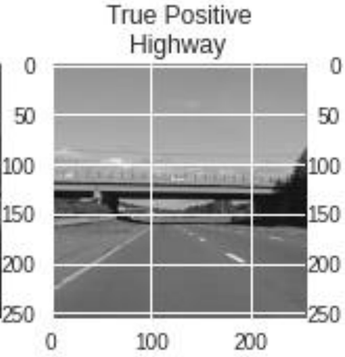
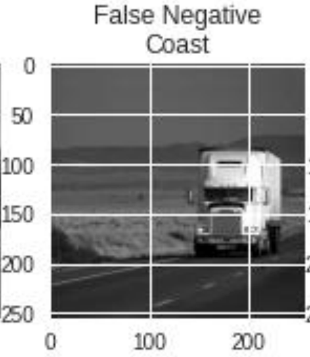
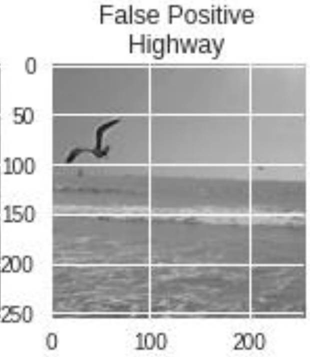
```
The Training Data: 1500 1500
The Testing Data: 705 705
===============================================================================
Using SIFT representation for images.
===============================================================================
No existing visual word vocabulary found. Computing one from training images

Extract SIFT features
The size of Bag of features: 762331
SIFT Features:
Training Data Shape: (1500, 400) (1500,)
Testing Data Shape: (705, 400) (705,)
```



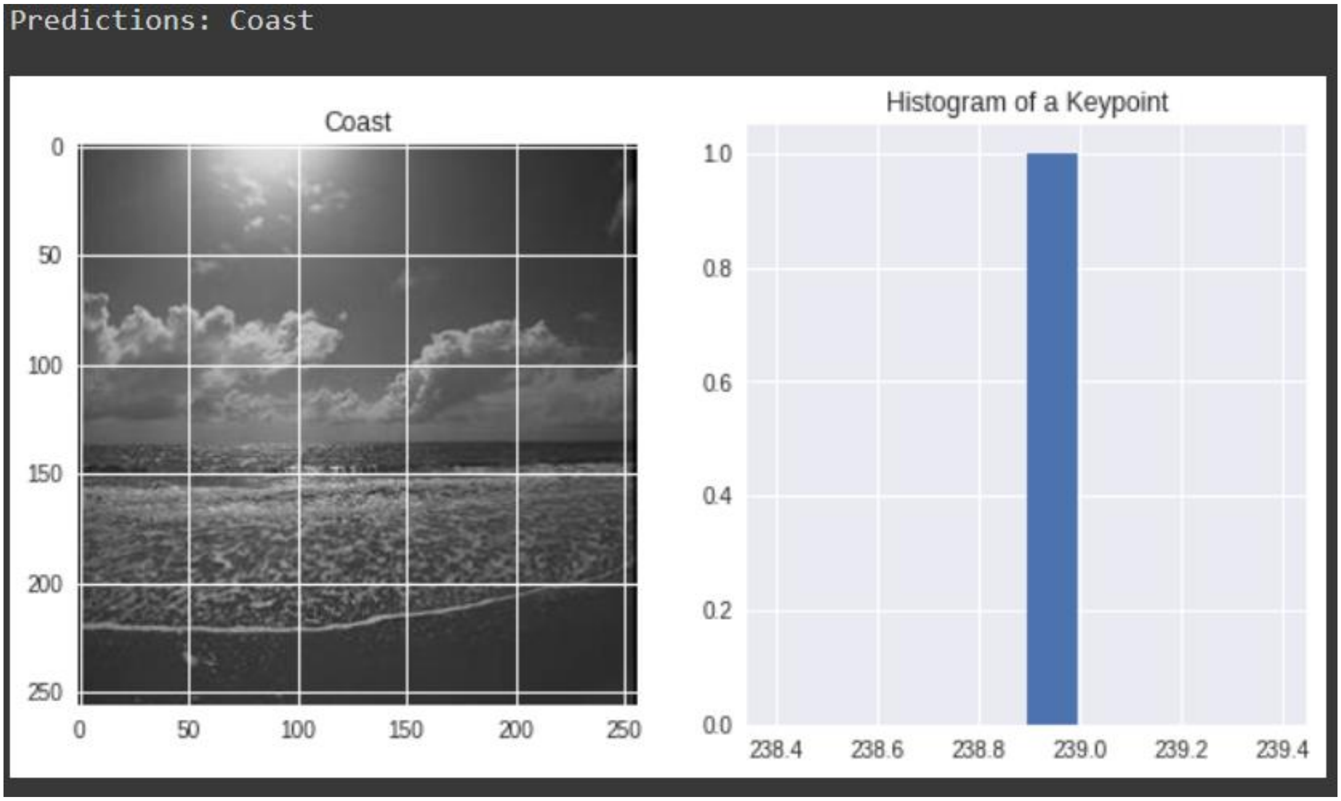


```
Accuracy:  0.7063829787234043
```

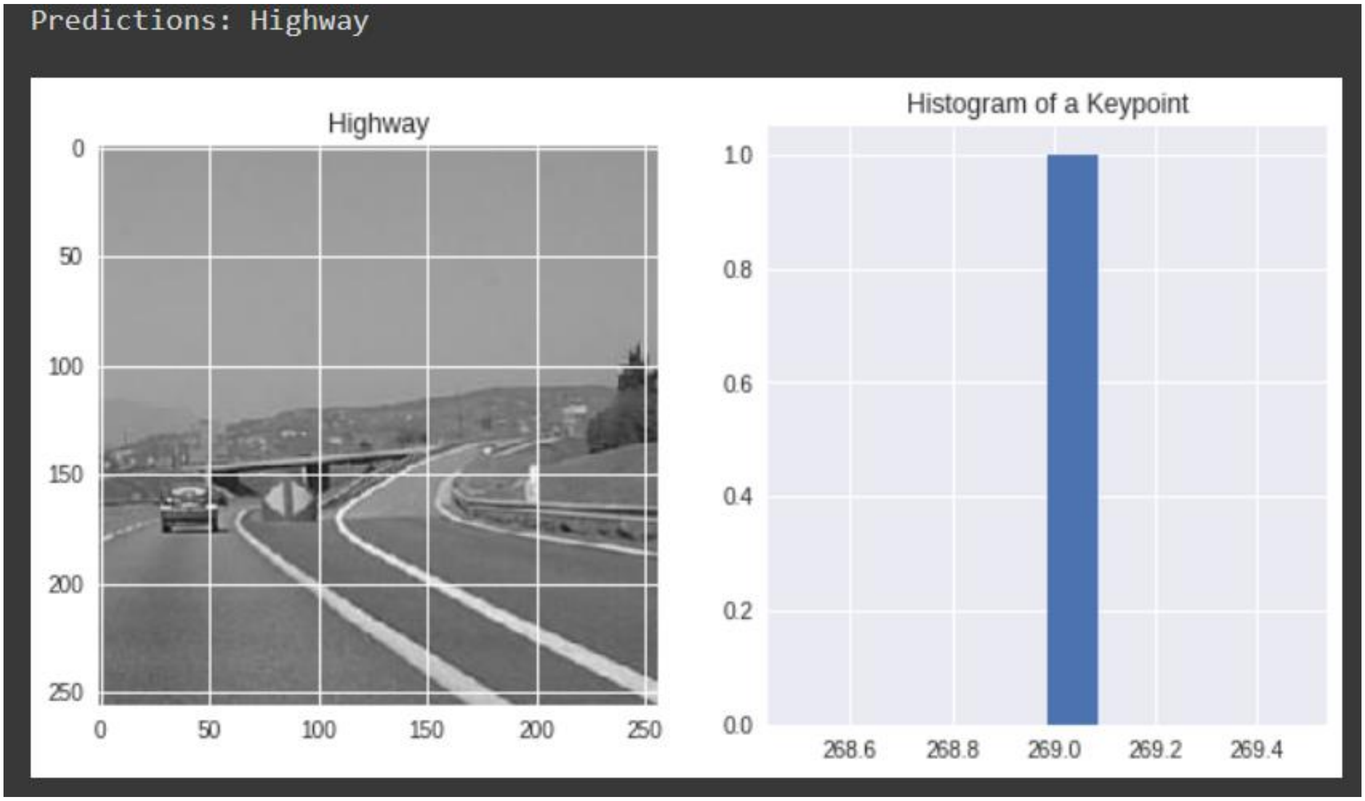
TP,FP,FN Visulaization Table

Category	Sample Image	True Positive	False Negative	False Positive
Bedroom				 False Positive Bedroom 0 50 100 150 200 250 0 100 200
Coast			 False Negative Highway 0 50 100 150 200 250 0 100 200	 False Positive Coast 0 50 100 150 200 250 0 100 200
Forest		 True Positive Forest 0 50 100 150 200 250 0 100 200	 False Negative Mountain 0 50 100 150 200 250 0 100 200	 False Positive Forest 0 50 100 150 200 250 0 100 200
Highway		 True Positive Highway 0 50 100 150 200 250 0 100 200	 False Negative Coast 0 50 100 150 200 250 0 100 200	 False Positive Highway 0 50 100 150 200 250 0 100 200

Predictions



```
Predictions: Coast
```

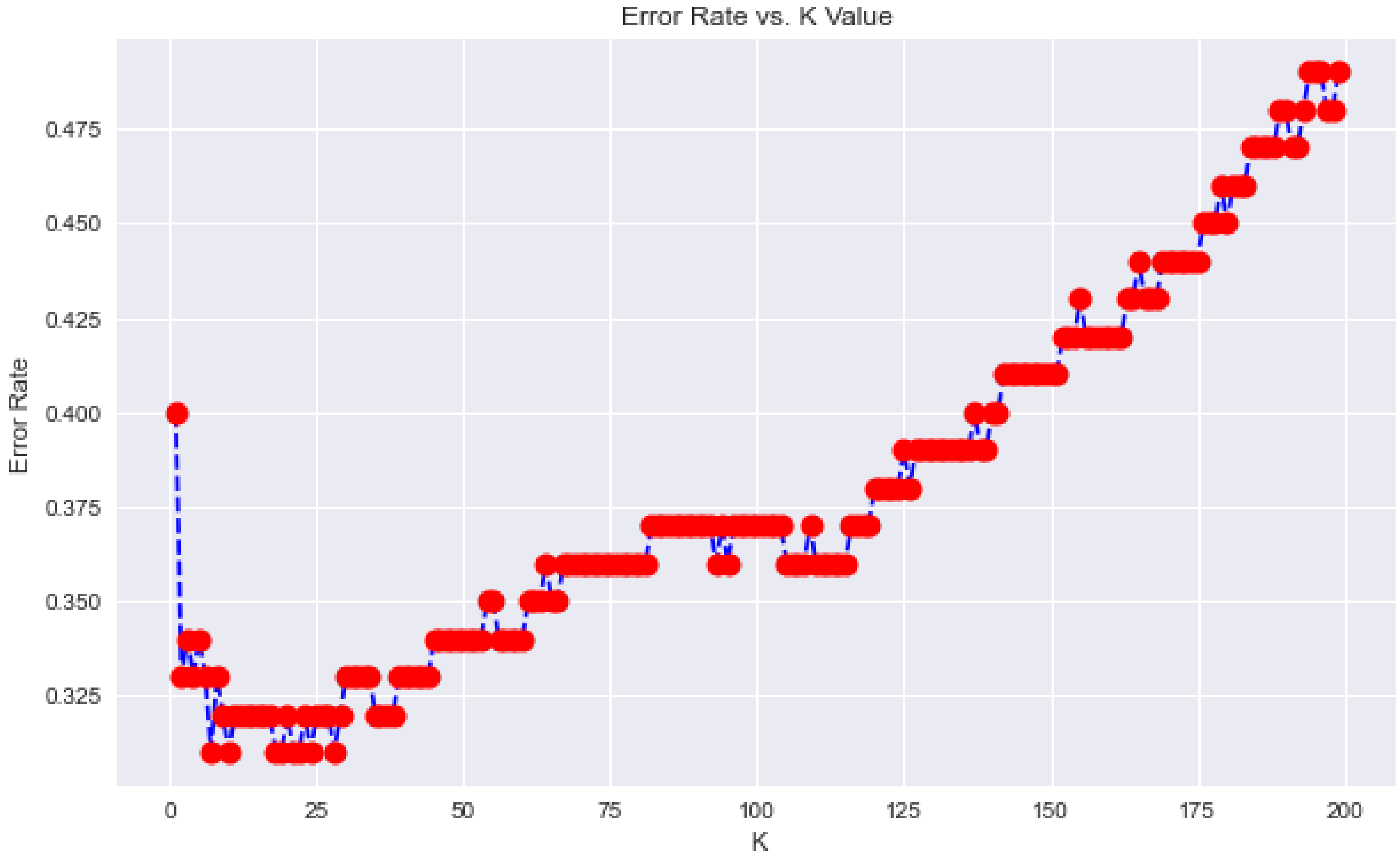


```
Predictions: Highway
```

KNN Discussion:

To discuss the experiments, I am using a table that represents the values of the different parameters used in each experiment. For example number of features, vocab size, the number of clusters, and the number of K in KNN. The next figure shows the distribution of data and shape before feature extraction and after feature extraction. Confusion matrix to display results for each class and then the accuracy of the model with given parameters.

- The implementation of code is the same as it was given in the starter code with the slight change I am using optimalKNN() function to select the optimal value of K for KNN.
- optimalKNN() is being used in all experiments How function works calculate error rate for each given in a range of K and return K with minimum error. That K value I am using for selecting Nearest Neighbors in Classification. For example, the given plot shows results of error rate for 200 k where selected k for minimum errors are [6 9 17 18 20 21 23 27].



Analysis of experiments:

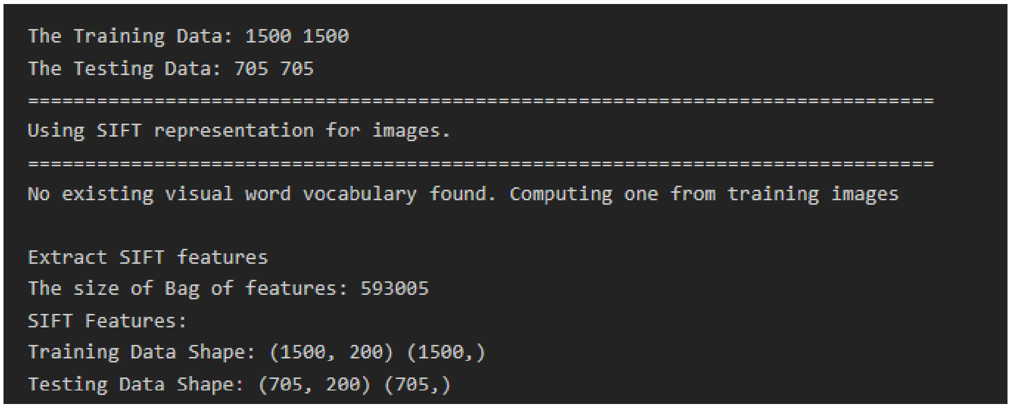
Discussing the Bag of SIFT with KNN, I have done different experiments to capture the relationship between the size of features extracted from the SIFT and the size of vocabulary in terms of model performance. Following are highlights of experiments:

- In the beginning, I started with the very small size of features and vocab like 30 for features and 45 for vocab size as given in my first experiment. The number of clusters is equal to the vocab size in my all experiments as was mentioned in the assignment guidelines. As we can see the accuracy is very low with these sizes.
- To check the effect of the vocab size I kept the size of the feature the same for 2,3 and 4th experiments and increased the vocab size by (10,100,1000) as mentioned in the assignment. The accuracy is between (33-39) not as was expected.
- In experiments 5,6 and 7, I set the features 100 and then again experimented with different vocabulary sizes (10,100,1000). The accuracy increased by (47-57). But there was not much difference in the accuracy of vocab size with 100 and 1000.
- In experiments 8,9 and 10, again I increased the features by 200 and experimented on vocabulary size (10,100,1000). The accuracy is better than the previous experiment and is between (51-64). There is a huge difference in the accuracy of 10 and 100 sizes but a small difference in 100 and 1000. It can be because of Cluster size as increasing the vocab means increasing the clusters.
- Experiment 11 with 300 features and 100 vocab size results in an accuracy of 0.64. One thing I noticed over here increasing the number of features is increasing the accuracy whereas vocab size should be feasible not too much small or large. Therefore in my next experiments, I started increasing features.
- In experiments 12 and 13 the accuracy is 70 with a slight change in point values. These are the best results so far achieved with KNN. Experiment 12 is similar to the above one whereas experiment 13 is a bit different here instead of defining descriptors the SIFT randomly selects the number of descriptors.
- For the final best model I have also shared a table of TP, FP, and FN results for given class and predictions made by the model which are accurate predictions.

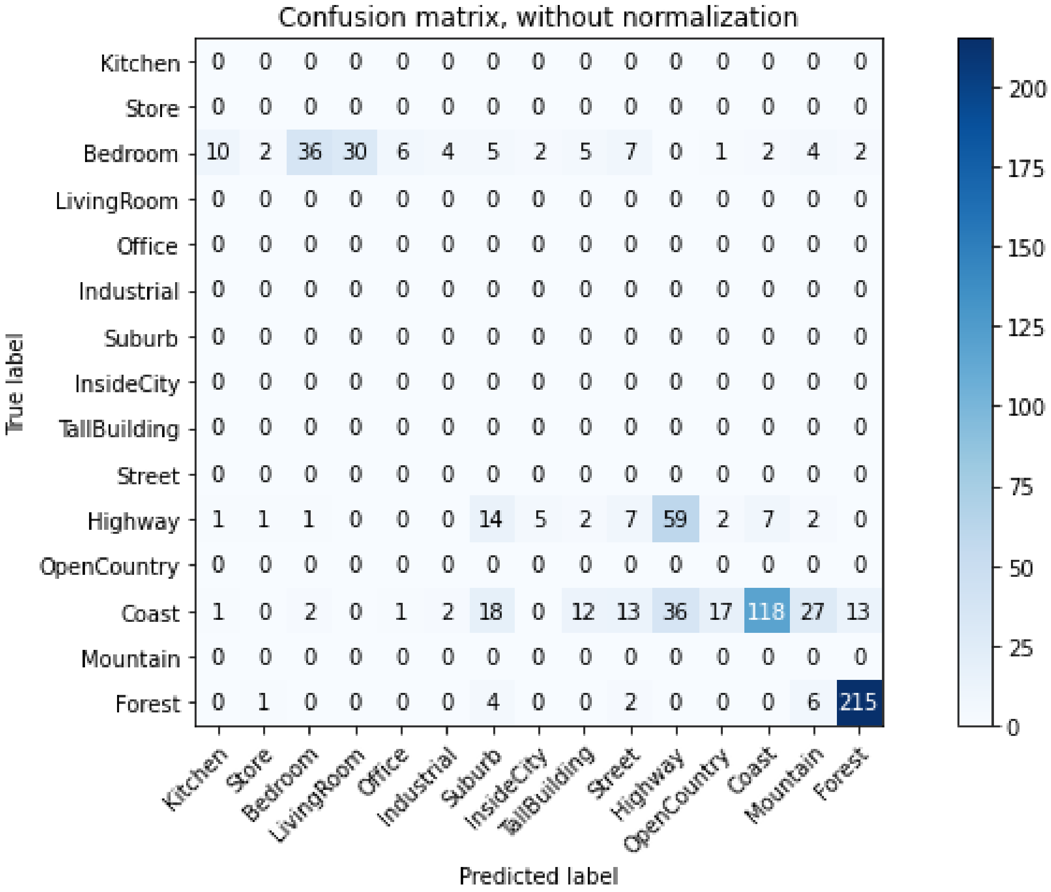
Bag of SIFT features and linear SVM classifier

Experiment 1:

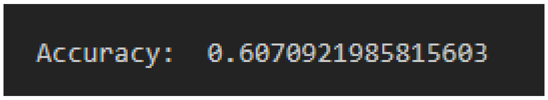
Features	Vocab size	KMeans(# of K)
500	200	200



```
The Training Data: 1500 1500
The Testing Data: 705 705
================================================================================
Using SIFT representation for images.
================================================================================
No existing visual word vocabulary found. Computing one from training images

Extract SIFT features
The size of Bag of features: 593005
SIFT Features:
Training Data Shape: (1500, 200) (1500,)
Testing Data Shape: (705, 200) (705,)
```



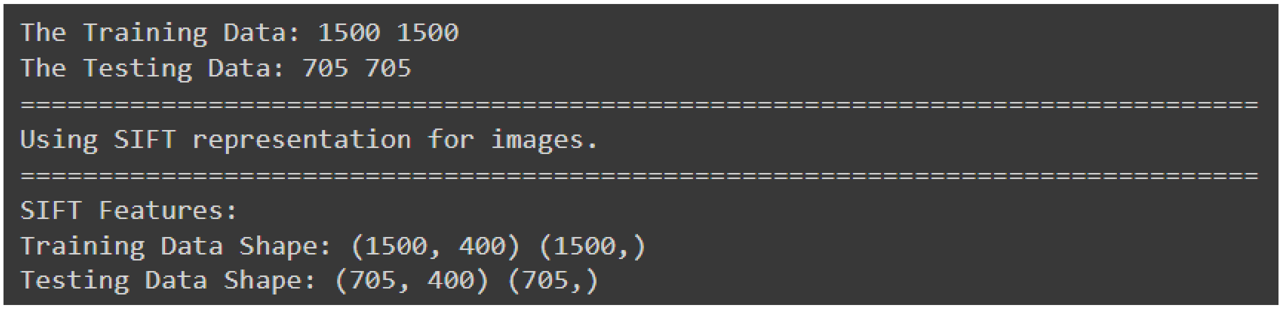
	Kitchen	Store	Bedroom	LivingRoom	Office	Industrial	Suburb	InsideCity	TallBuilding	Street	Highway	OpenCountry	Coast	Mountain	Forest
Kitchen	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Store	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bedroom	10	2	36	30	6	4	5	2	5	7	0	1	2	4	2
LivingRoom	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Office	0	0	0	0	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	1	1	1	0	0	0	14	5	2	7	59	2	7	2	0
OpenCountry	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Coast	1	0	2	0	1	2	18	0	12	13	36	17	118	27	13
Mountain	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Forest	0	1	0	0	0	0	4	0	0	2	0	0	0	6	215



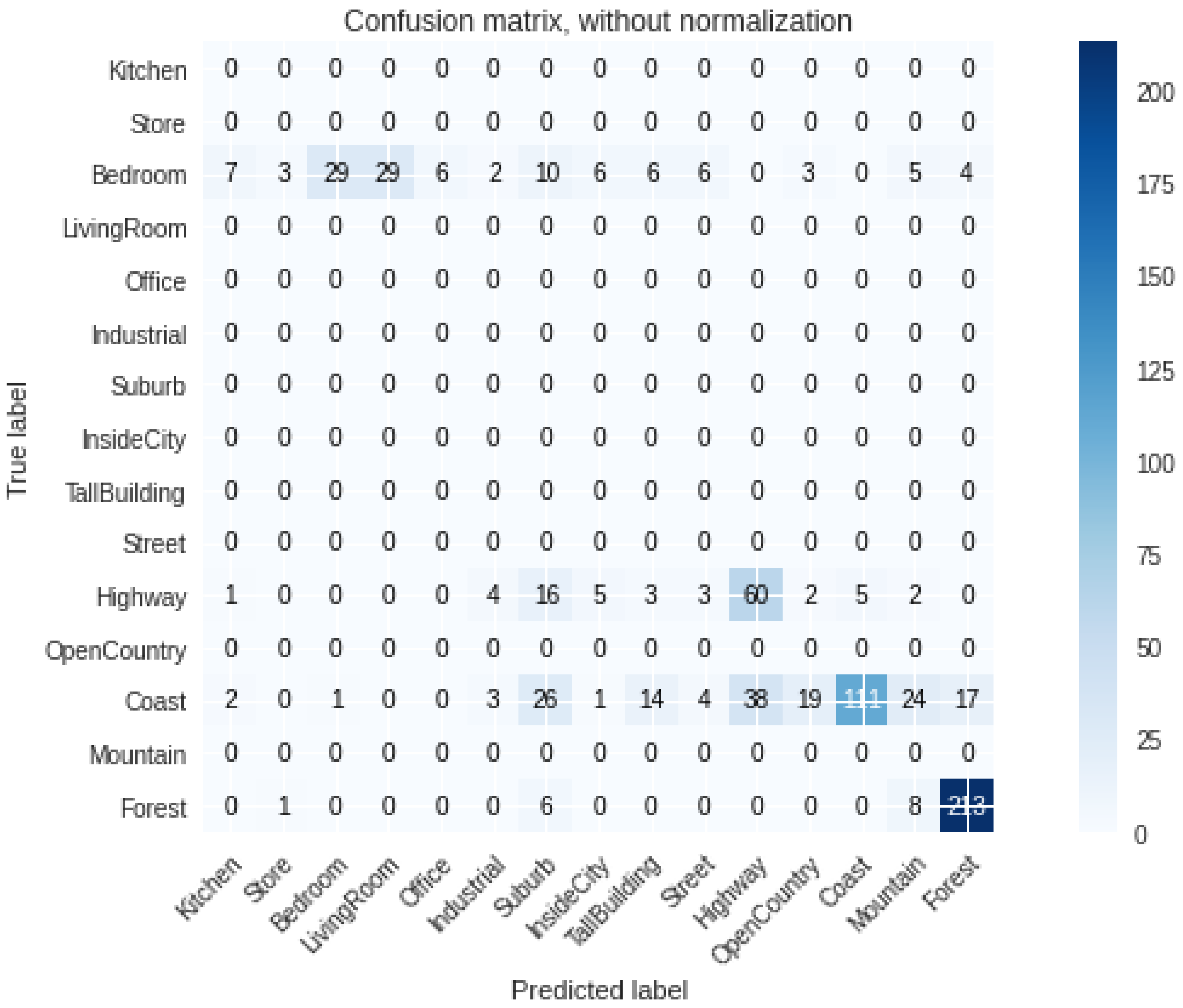
```
Accuracy:  0.6070921985815603
```

Experiment 2:

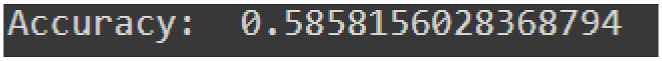
Features	Vocab size	KMeans(# of K)
600	400	400



```
The Training Data: 1500 1500
The Testing Data: 705 705
================================================================================
Using SIFT representation for images.
================================================================================
SIFT Features:
Training Data Shape: (1500, 400) (1500,)
Testing Data Shape: (705, 400) (705,)
```



	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	7	3	29	29	6	2	10	6	6	6	0	3	0	5	4
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	1	0	0	0	0	4	16	5	3	3	60	2	5	2	0
OpenCountry	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Coast	2	0	1	0	0	3	26	1	14	4	38	19	111	24	17
Mountain	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Forest	0	1	0	0	0	0	6	0	0	0	0	0	0	8	213

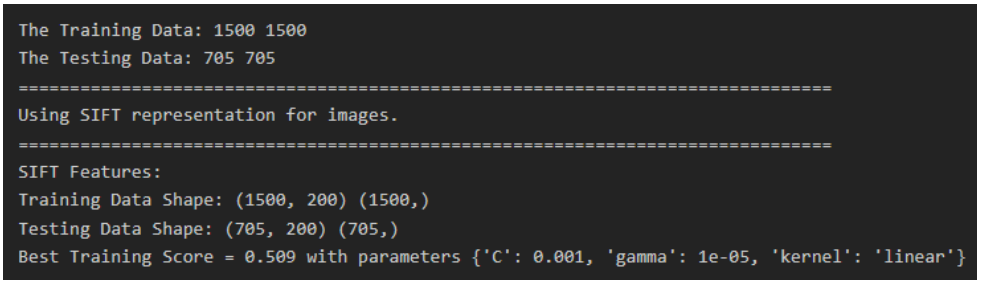


```
Accuracy:  0.5858156028368794
```

Experiment 3:

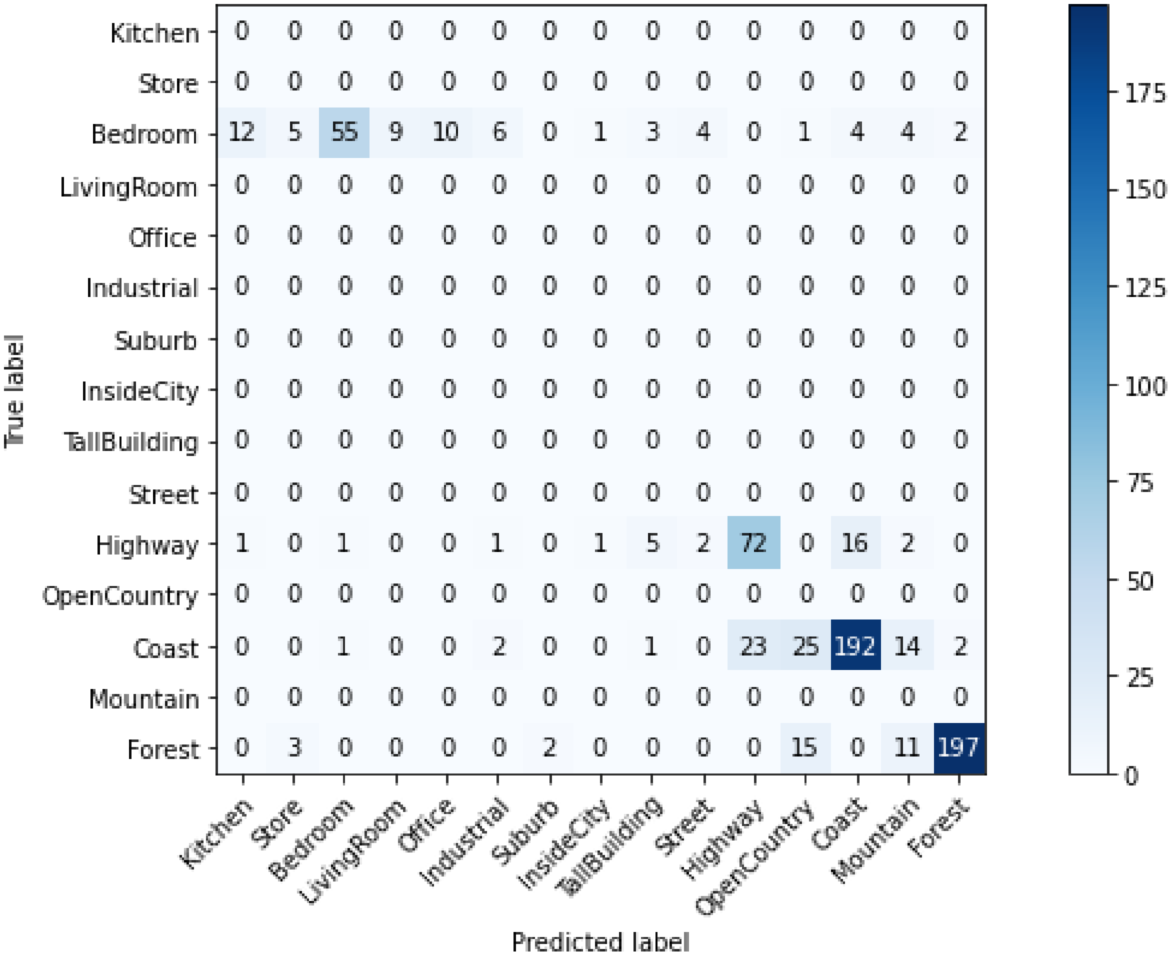
Hyperparamter tunning:

Features	Vocab size	KMeans(# of K)
600	200	200

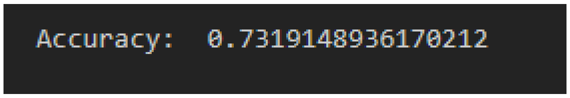


```
The Training Data: 1500 1500
The Testing Data: 705 705
================================================================================
Using SIFT representation for images.
================================================================================
SIFT Features:
Training Data Shape: (1500, 200) (1500,)
Testing Data Shape: (705, 200) (705,)
Best Training Score = 0.509 with parameters {'C': 0.001, 'gamma': 1e-05, 'kernel': 'linear'}
```

Confusion matrix, without normalization



True label \ Predicted label	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	12	5	55	9	10	6	0	1	3	4	0	1	4	4	2
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	1	0	1	0	0	1	0	1	5	2	72	0	16	2	0
OpenCountry	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Coast	0	0	1	0	0	2	0	0	1	0	23	25	192	14	2
Mountain	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Forest	0	3	0	0	0	0	2	0	0	0	0	15	0	11	197

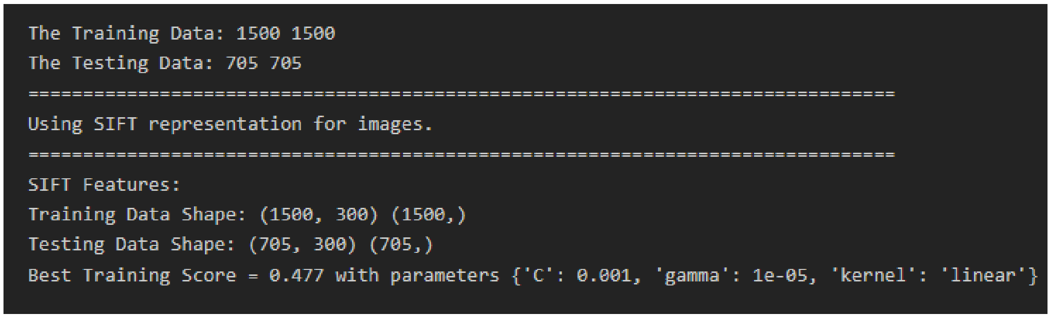


```
Accuracy:  0.7319148936170212
```

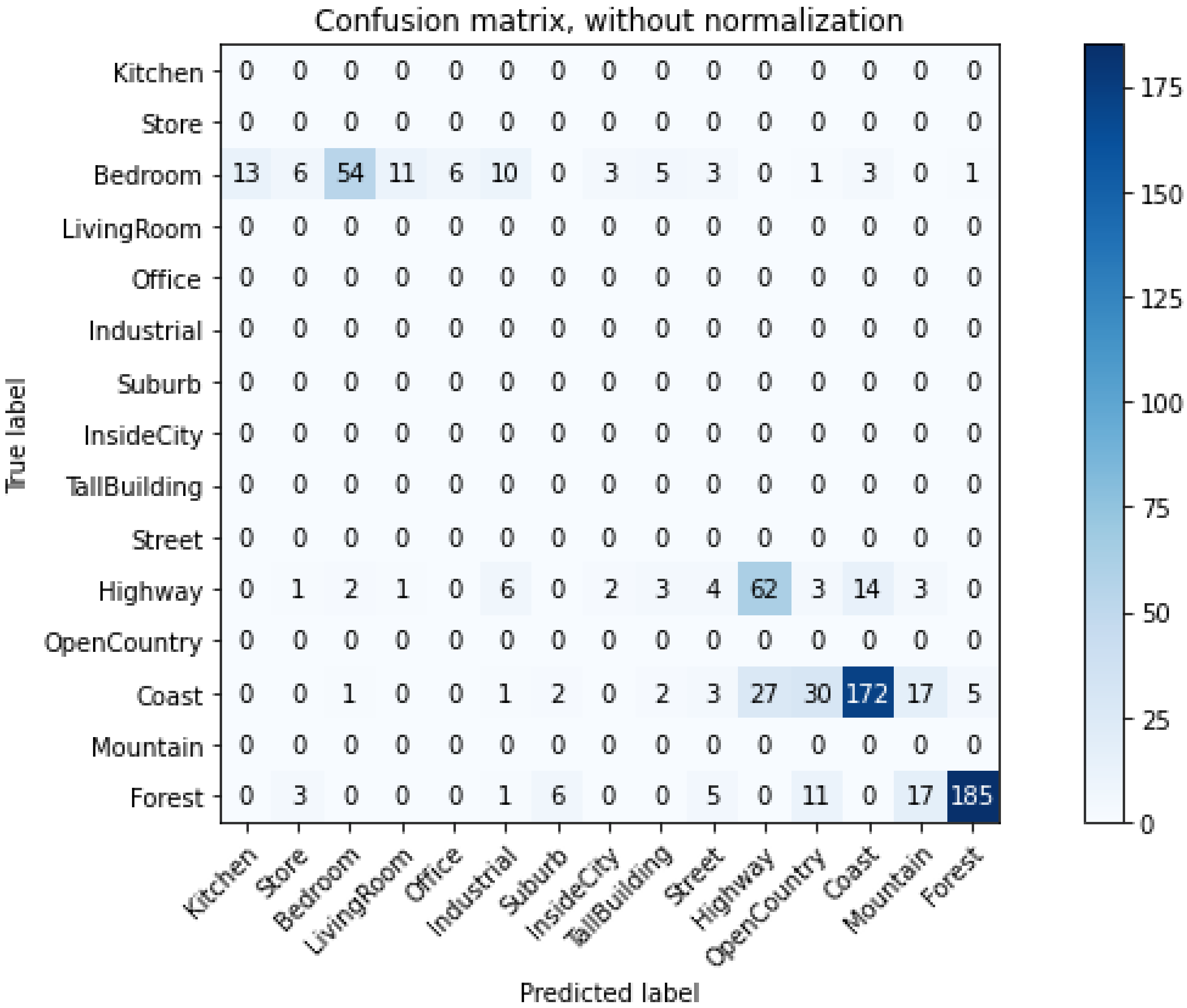
Experiment 4:

Hyperparamter tunning:

Features	Vocab size	KMeans(# of K)
300	300	300

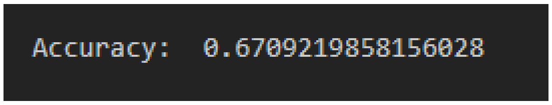


```
The Training Data: 1500 1500
The Testing Data: 705 705
==============================================================================
Using SIFT representation for images.
==============================================================================
SIFT Features:
Training Data Shape: (1500, 300) (1500,)
Testing Data Shape: (705, 300) (705,)
Best Training Score = 0.477 with parameters {'C': 0.001, 'gamma': 1e-05, 'kernel': 'linear'}
```



Confusion matrix, without normalization

	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	13	6	54	11	6	10	0	3	5	3	0	1	3	0	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	0	1	2	1	0	6	0	2	3	4	62	3	14	3	0
OpenCountry	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Coast	0	0	1	0	0	1	2	0	2	3	27	30	172	17	5
Mountain	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Forest	0	3	0	0	0	1	6	0	0	5	0	11	0	17	185

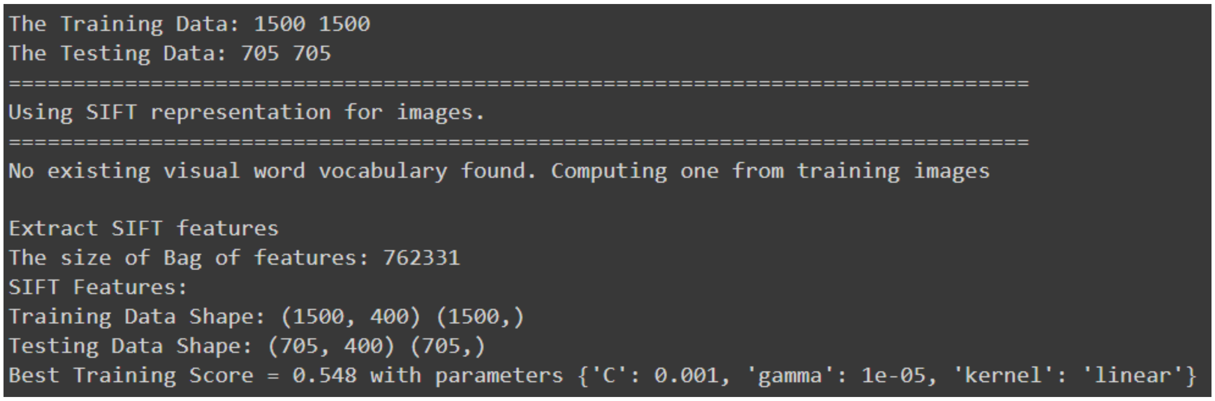


```
Accuracy:  0.6709219858156028
```

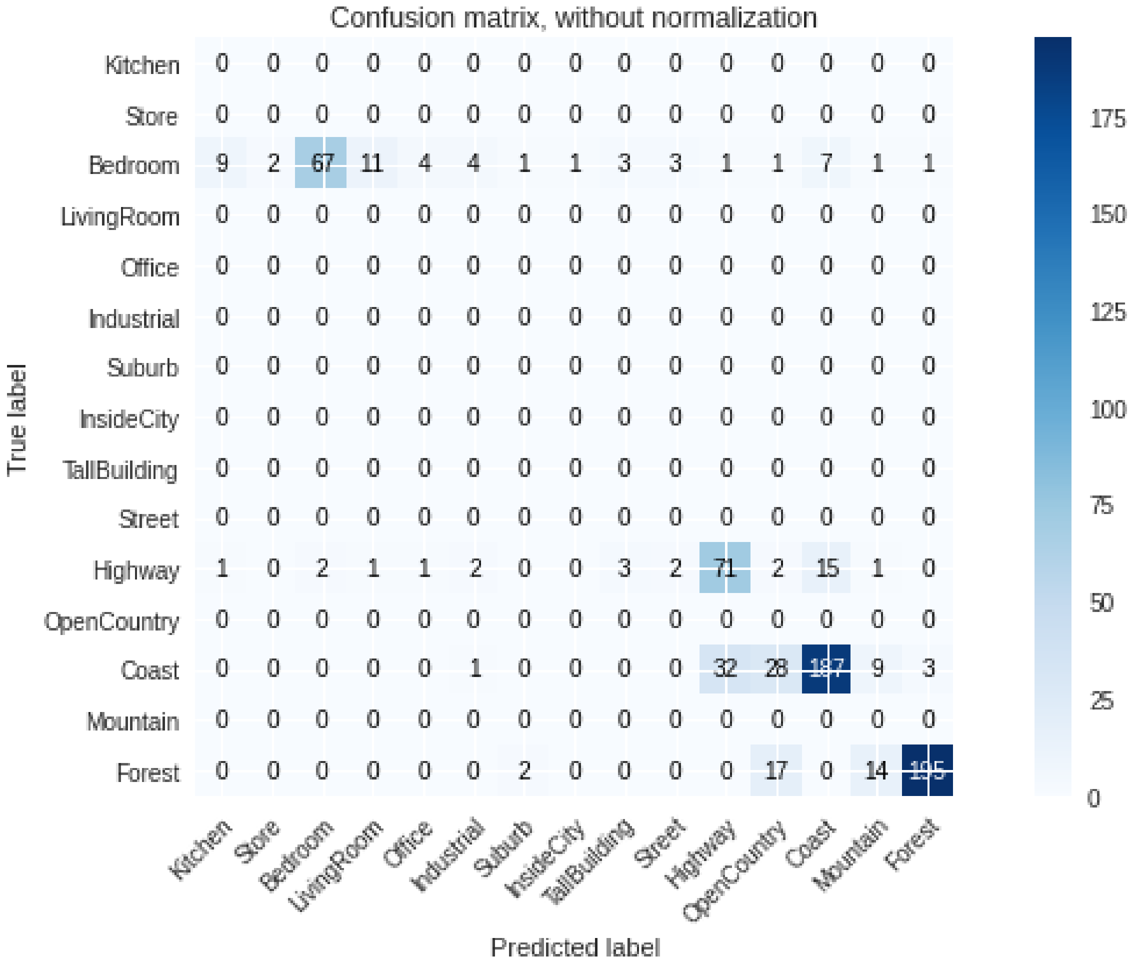
Experiment 5:

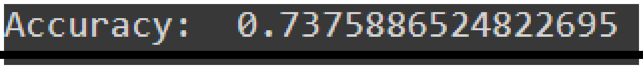
Hyperparamter tunning:

Features	Vocab size	KMeans(# of K)
random	400	400



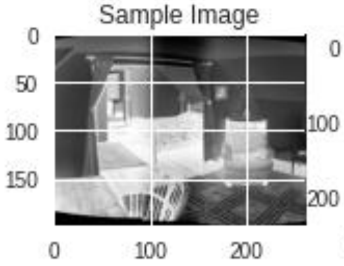
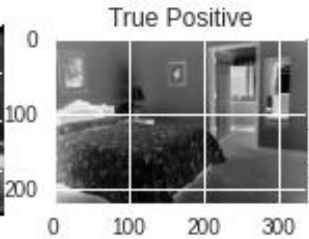
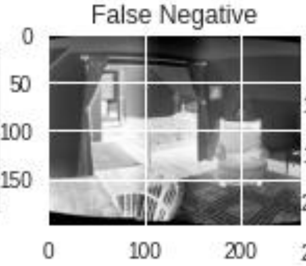
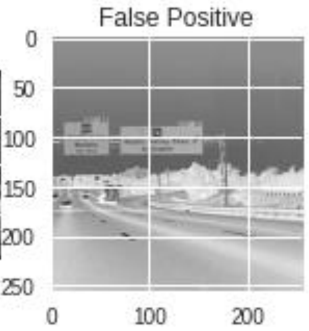
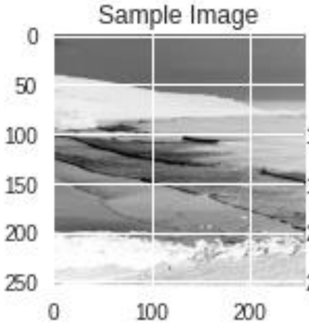
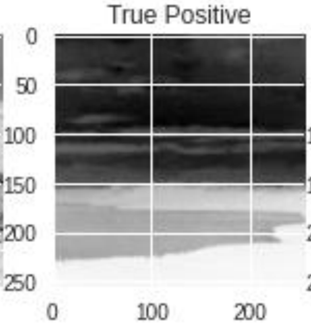
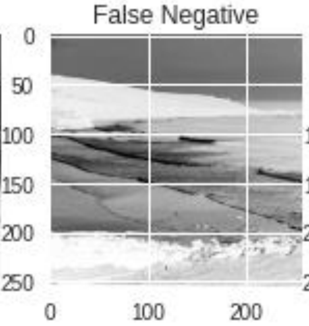
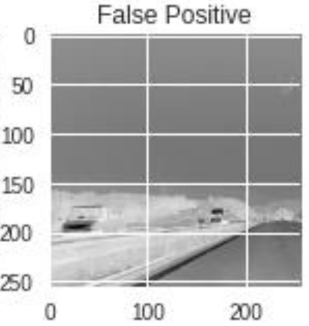
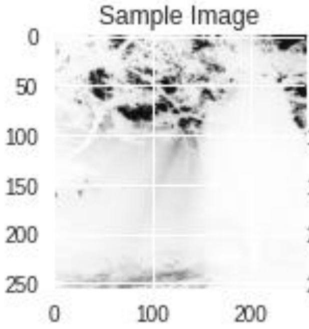
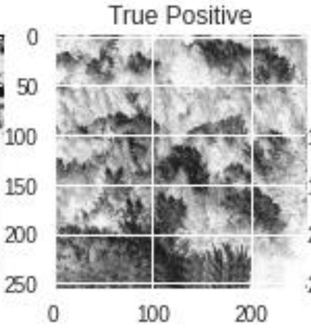
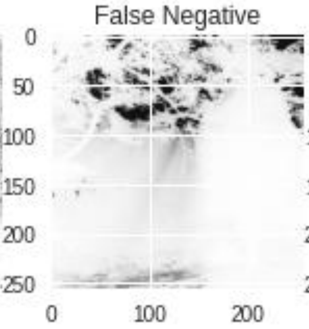
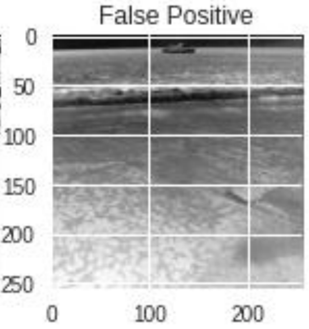
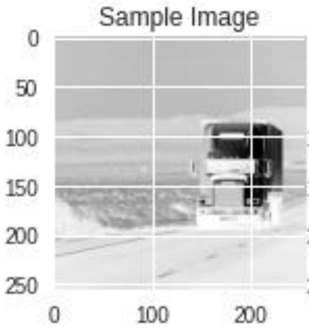
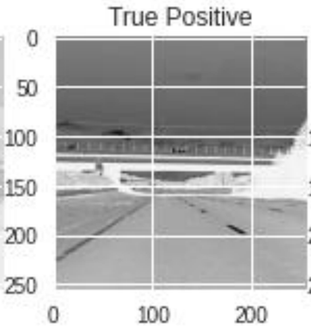
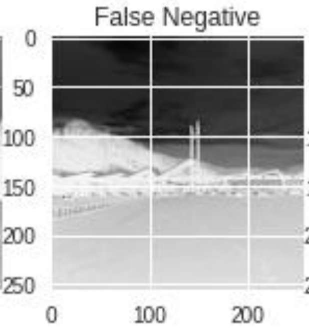
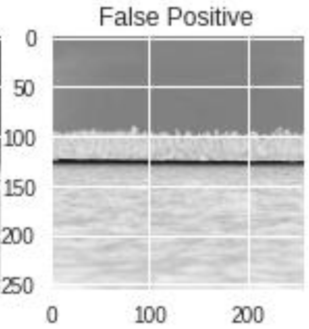
```
The Training Data: 1500 1500
The Testing Data: 705 705
=============================================================================
Using SIFT representation for images.
=============================================================================
No existing visual word vocabulary found. Computing one from training images

Extract SIFT features
The size of Bag of features: 762331
SIFT Features:
Training Data Shape: (1500, 400) (1500,)
Testing Data Shape: (705, 400) (705,)
Best Training Score = 0.548 with parameters {'C': 0.001, 'gamma': 1e-05, 'kernel': 'linear'}
```



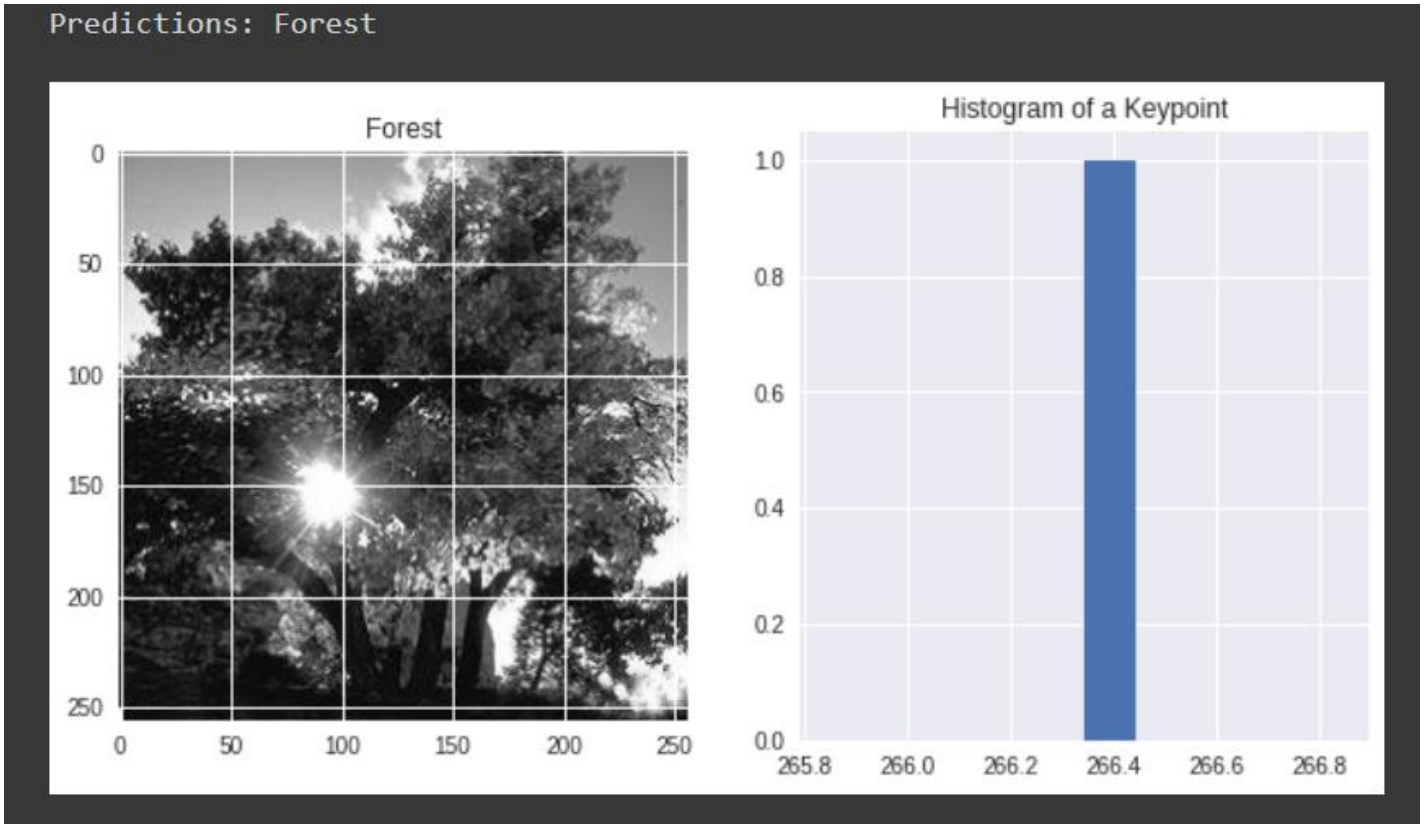


```
Accuracy:  0.7375886524822695
```

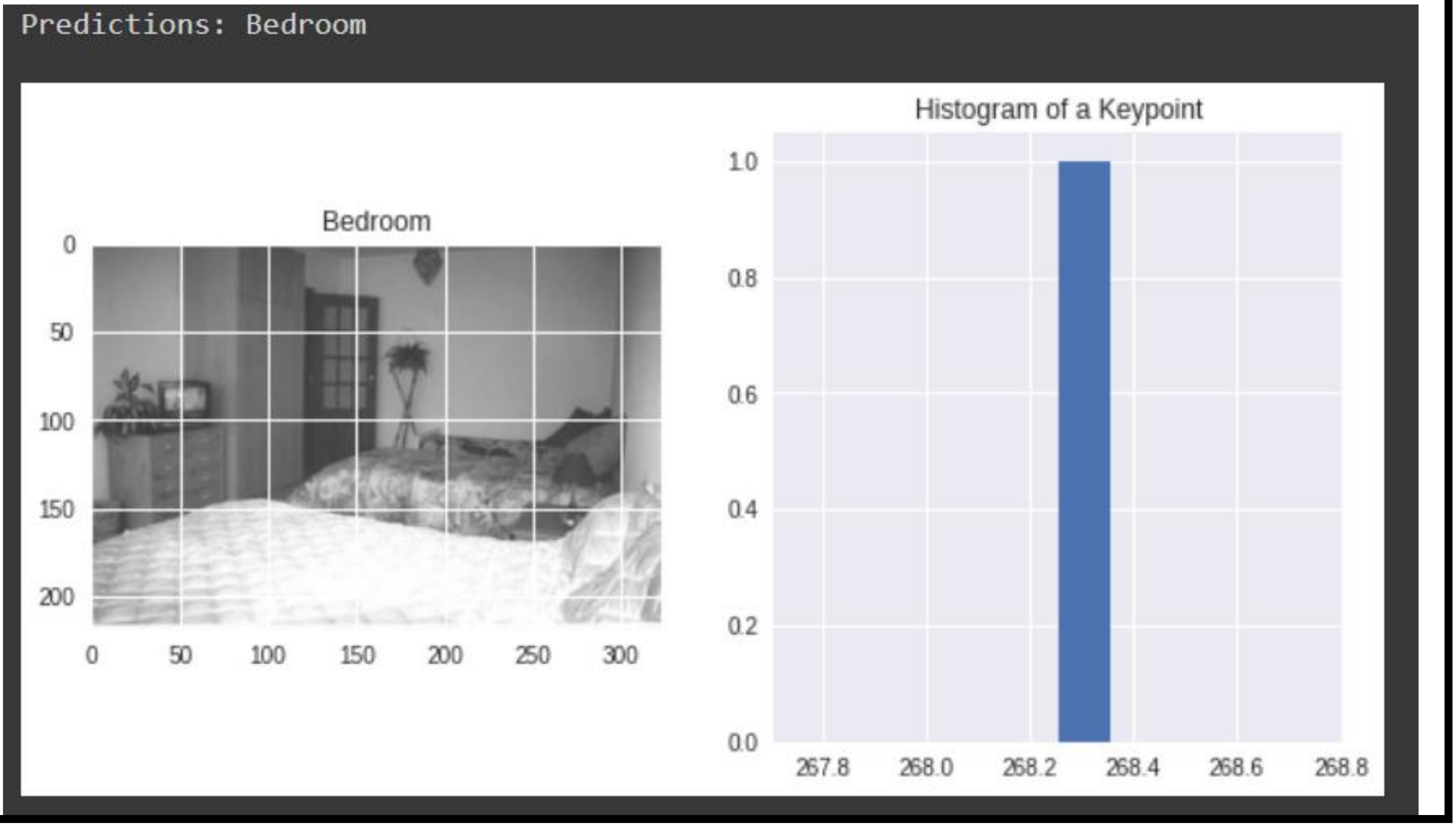
TP,FP,FN Visulaization Table

Category	Sample Image	True Positive	False Negative	False Positive
Bedroom				 False Positive 0 50 100 150 200 250 0 100 200
Coast	 Sample Image 0 50 100 150 200 250 0 100 200	 True Positive 0 50 100 150 200 250 0 100 200	 False Negative 0 50 100 150 200 250 0 100 200	 False Positive 0 50 100 150 200 250 0 100 200
Forest	 Sample Image 0 50 100 150 200 250 0 100 200	 True Positive 0 50 100 150 200 250 0 100 200	 False Negative 0 50 100 150 200 250 0 100 200	 False Positive 0 50 100 150 200 250 0 100 200
Highway	 Sample Image 0 50 100 150 200 250 0 100 200	 True Positive 0 50 100 150 200 250 0 100 200	 False Negative 0 50 100 150 200 250 0 100 200	 False Positive 0 50 100 150 200 250 0 100 200

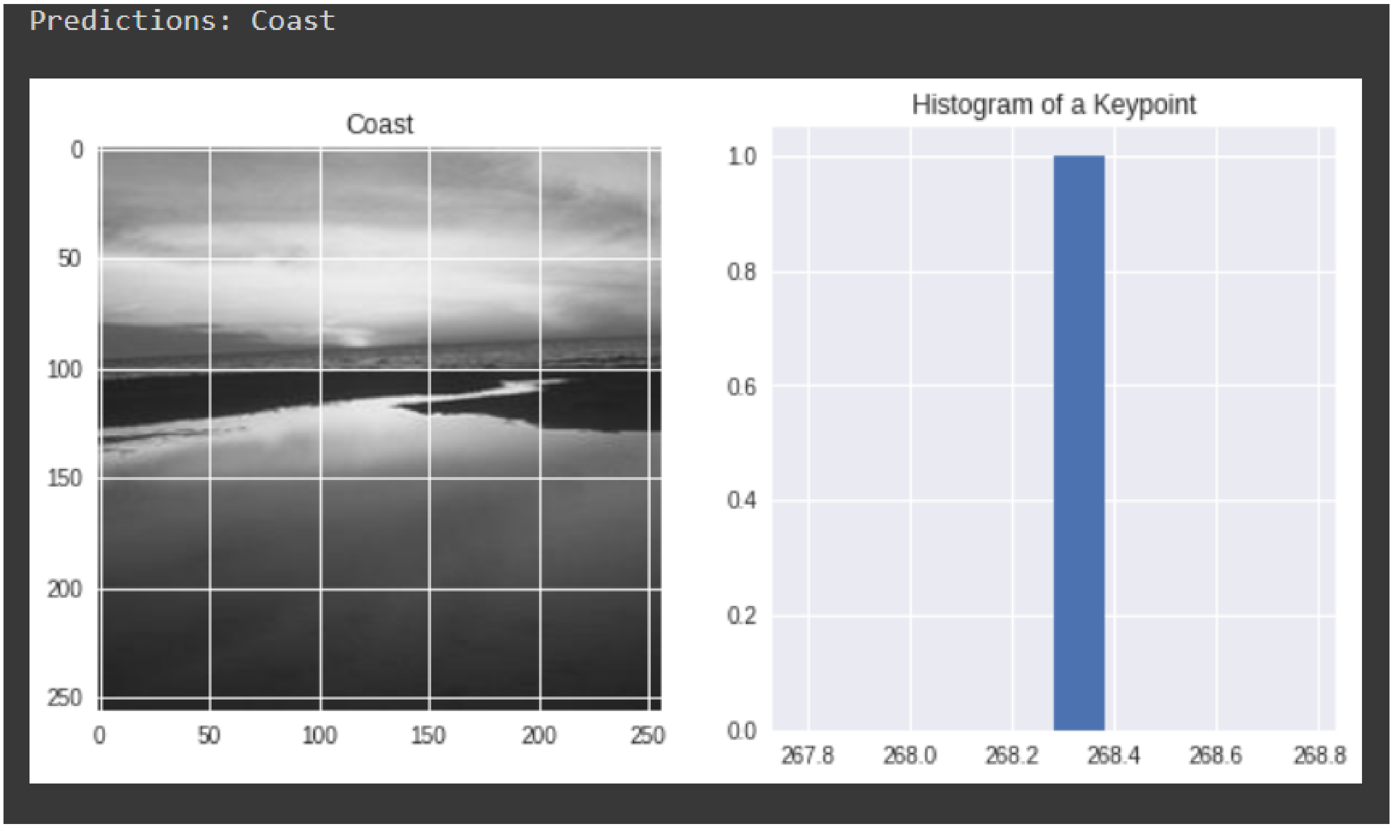
Predictions:



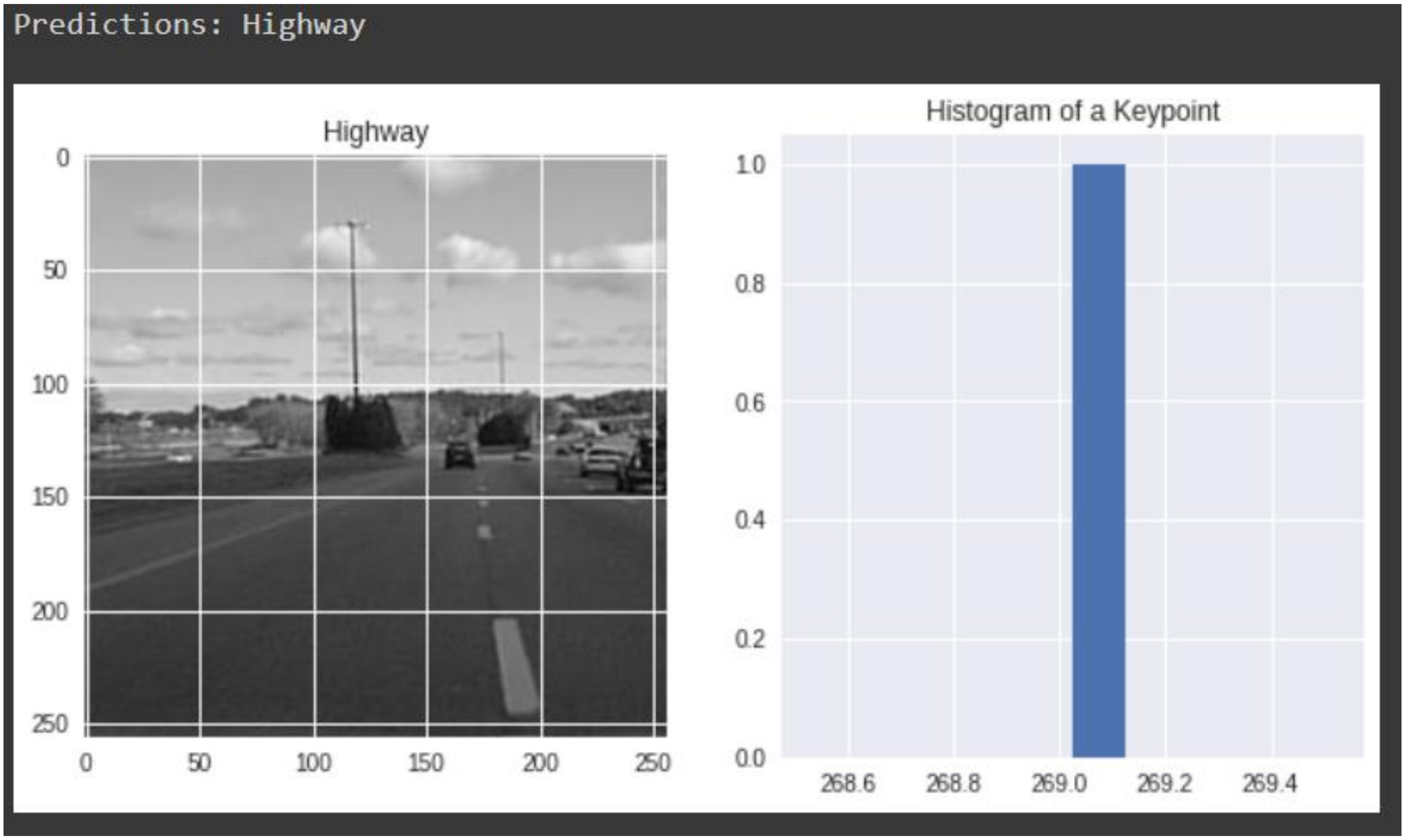
```
Predictions: Forest
```



```
Predictions: Bedroom
```



```
Predictions: Coast
```

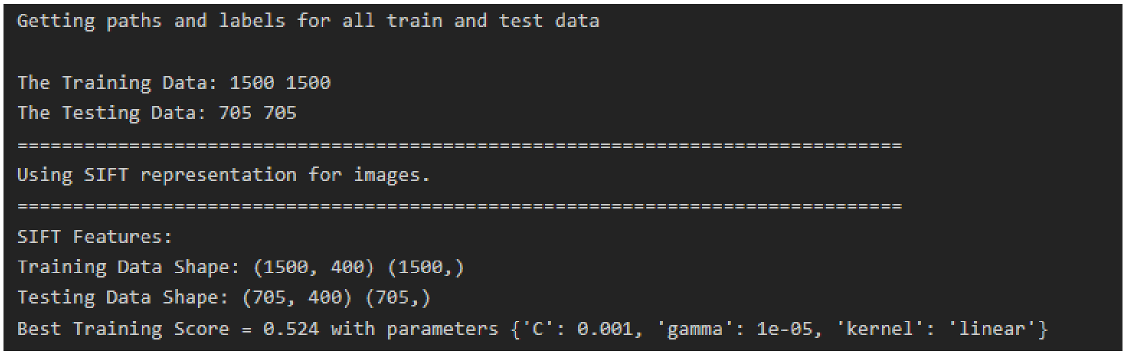


```
Predictions: Highway
```

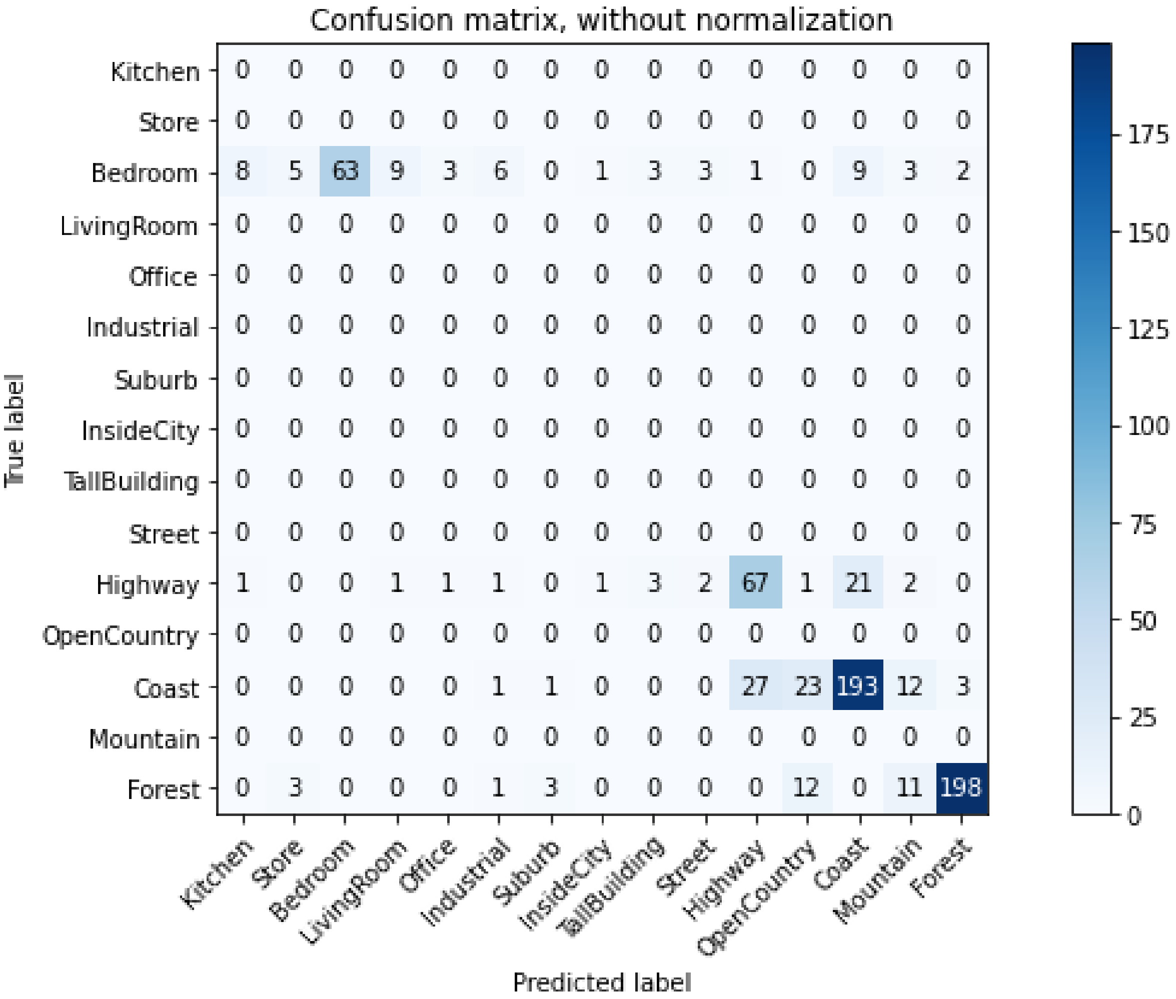
Experiment 6:

Hyperparamter tunning:

Features	Vocab size	KMeans(# of K)
600	400	400

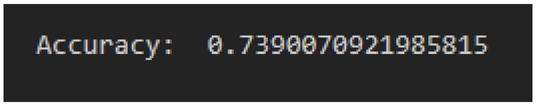


```
Getting paths and labels for all train and test data

The Training Data: 1500 1500
The Testing Data: 705 705
================================================================================
Using SIFT representation for images.
================================================================================
SIFT Features:
Training Data Shape: (1500, 400) (1500,)
Testing Data Shape: (705, 400) (705,)
Best Training Score = 0.524 with parameters {'C': 0.001, 'gamma': 1e-05, 'kernel': 'linear'}
```



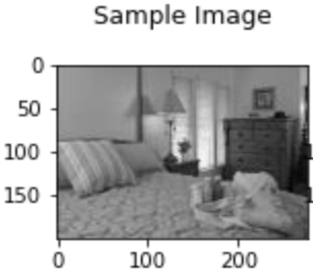
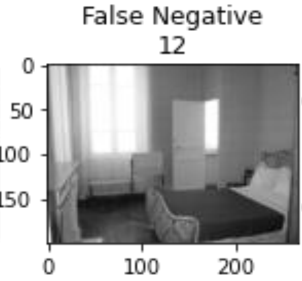
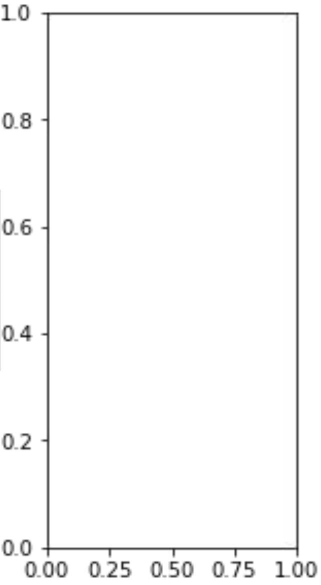
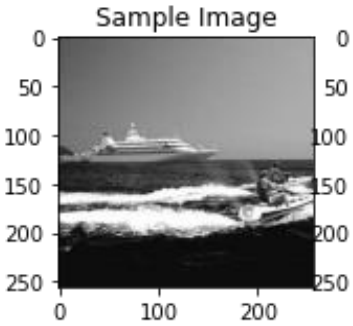
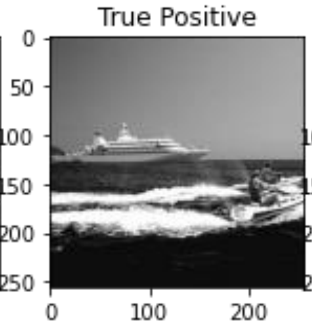
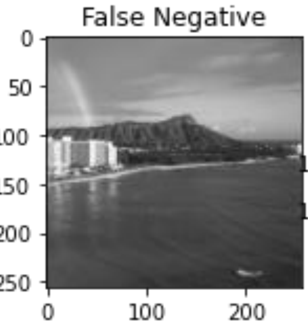
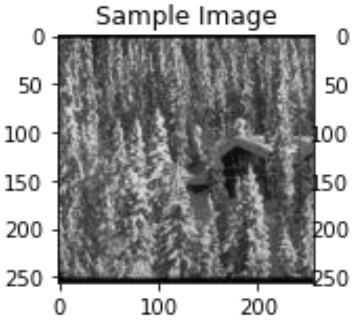
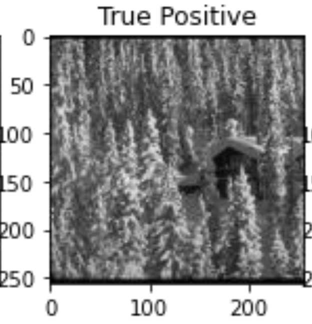
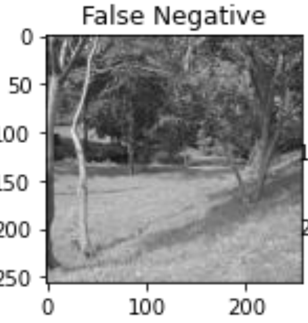
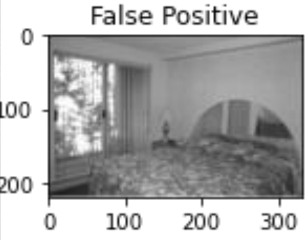
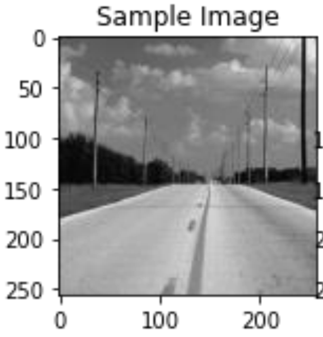
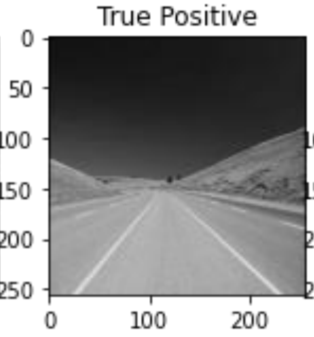
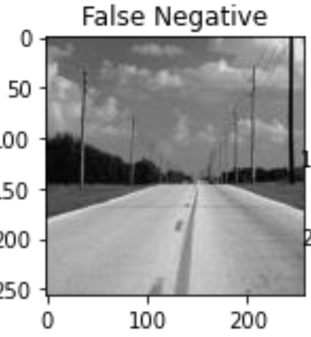
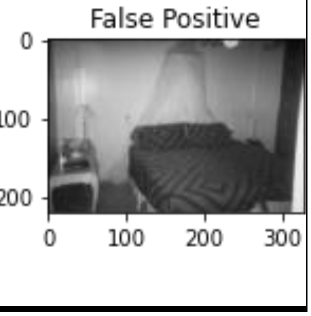
Confusion matrix, without normalization

	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	8	5	63	9	3	6	0	1	3	3	1	0	9	3	2
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	1	0	0	1	1	1	0	1	3	2	67	1	21	2	0
OpenCountry	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Coast	0	0	0	0	0	1	1	0	0	0	27	23	193	12	3
Mountain	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Forest	0	3	0	0	0	1	3	0	0	0	0	12	0	11	198

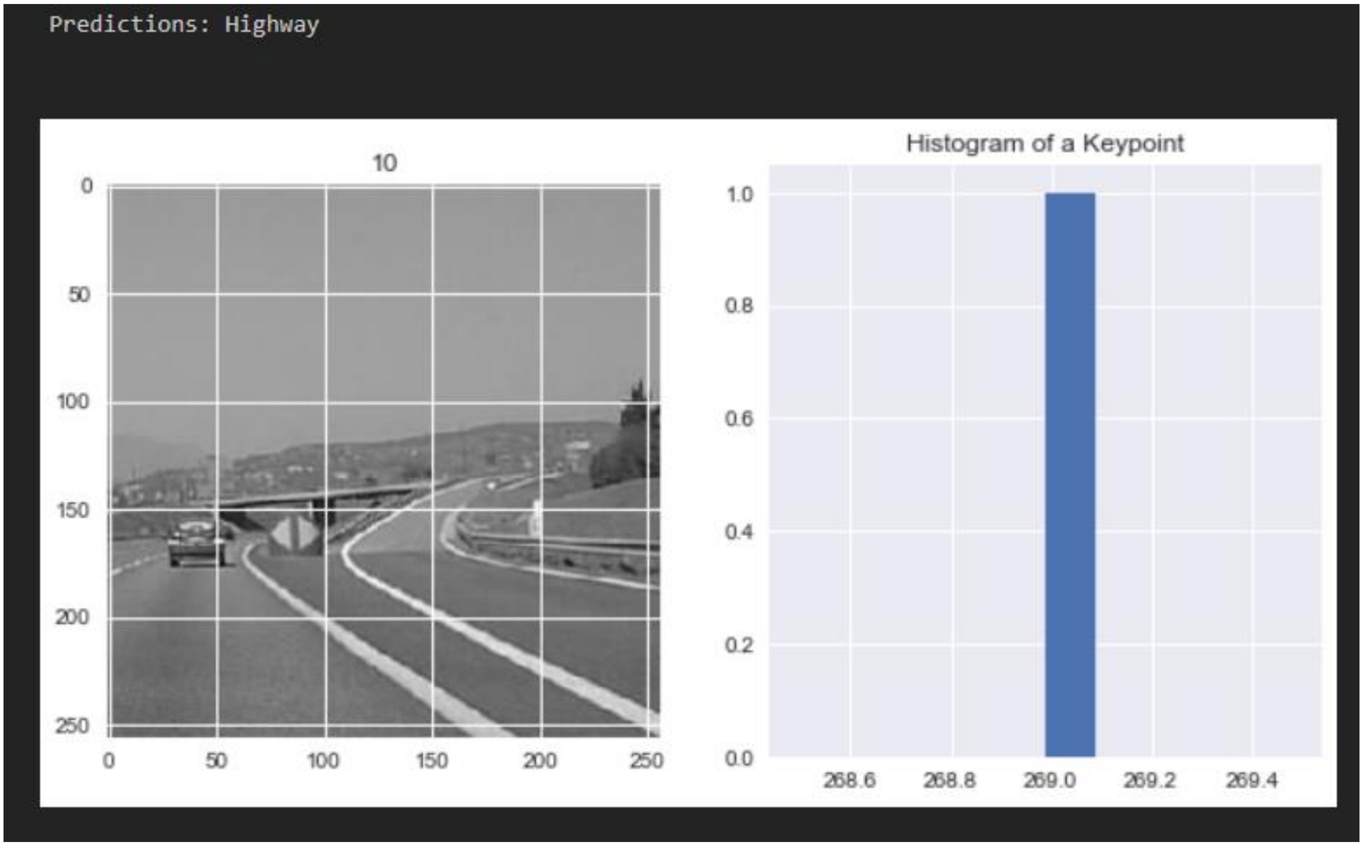


```
Accuracy:  0.7390070921985815
```

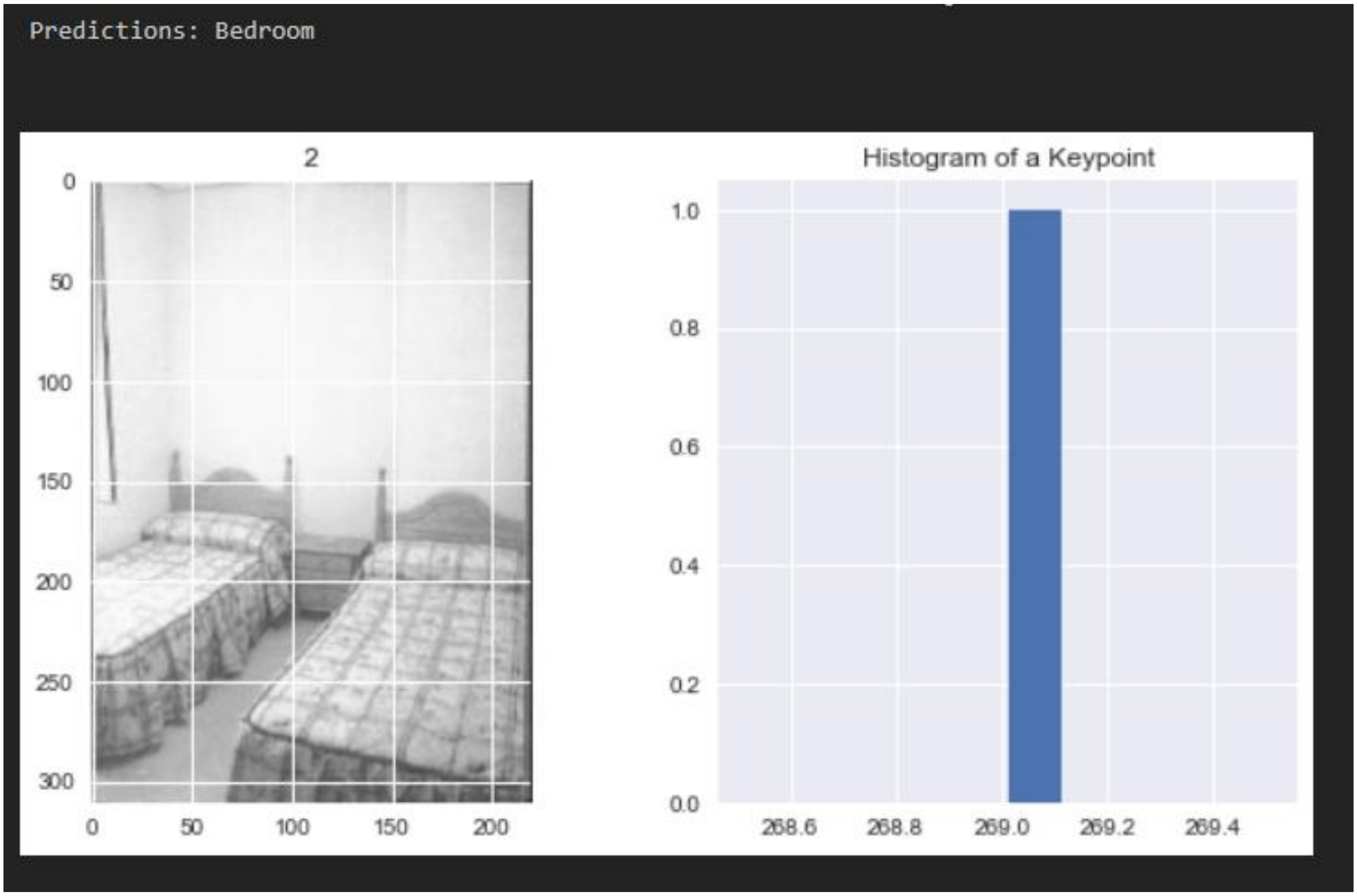
TP,FP,FN Visulaization Table

Category	Sample Image	True Positive	False Negative	False Positive
Bedroom		True Positive 2 0 50 100 150 0 100 200	 False Negative 12 0 50 100 150 0 100 200	
Coast	 Sample Image 0 50 100 150 200 250 0 100 200	 True Positive 0 50 100 150 200 250 0 100 200	 False Negative 0 50 100 150 200 250 0 100 200	
Forest	 Sample Image 0 50 100 150 200 250 0 100 200	 True Positive 0 50 100 150 200 250 0 100 200	 False Negative 0 50 100 150 200 250 0 100 200	
Highway	 Sample Image 0 50 100 150 200 250 0 100 200	 True Positive 0 50 100 150 200 250 0 100 200	 False Negative 0 50 100 150 200 250 0 100 200	

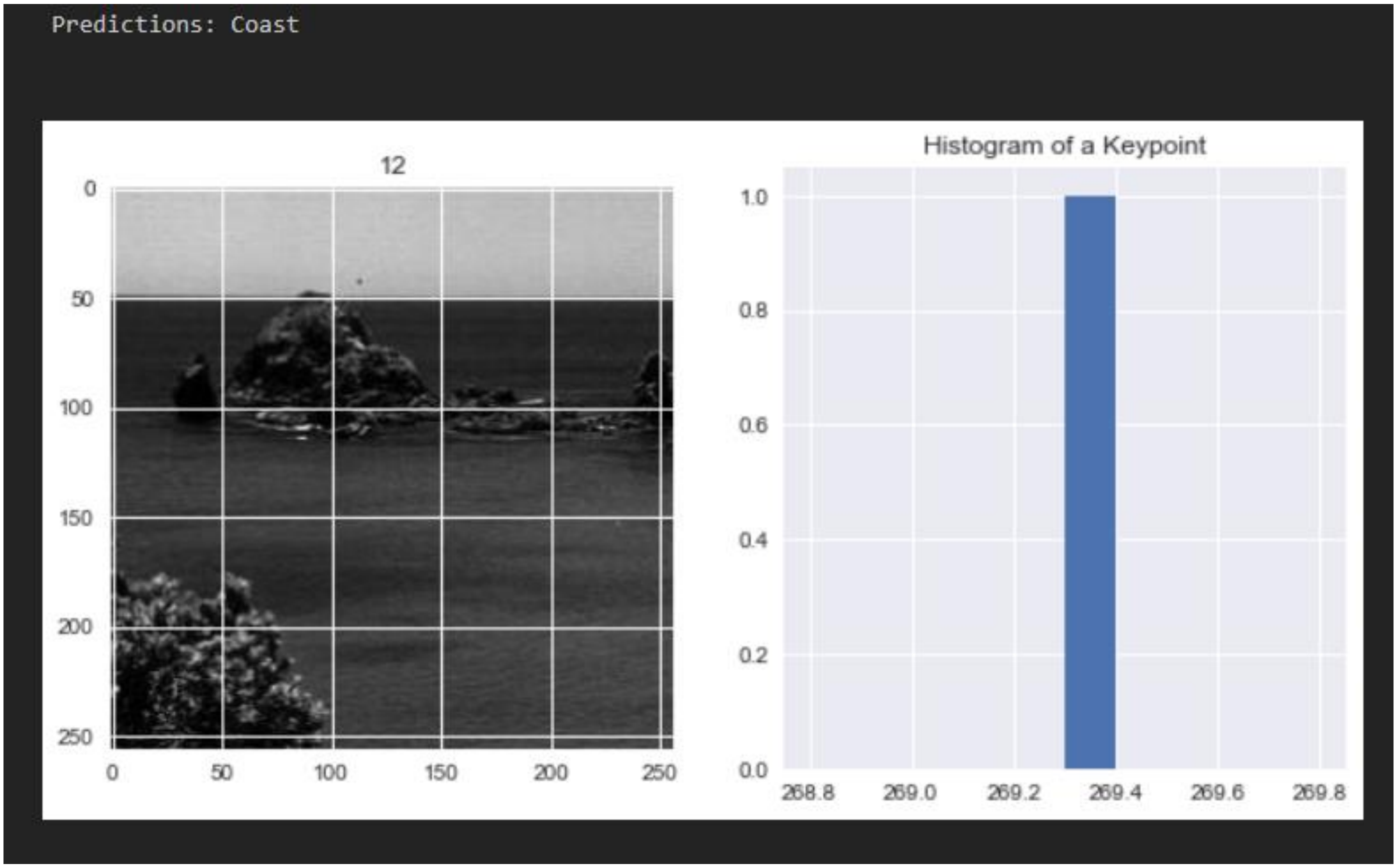
Predictions:



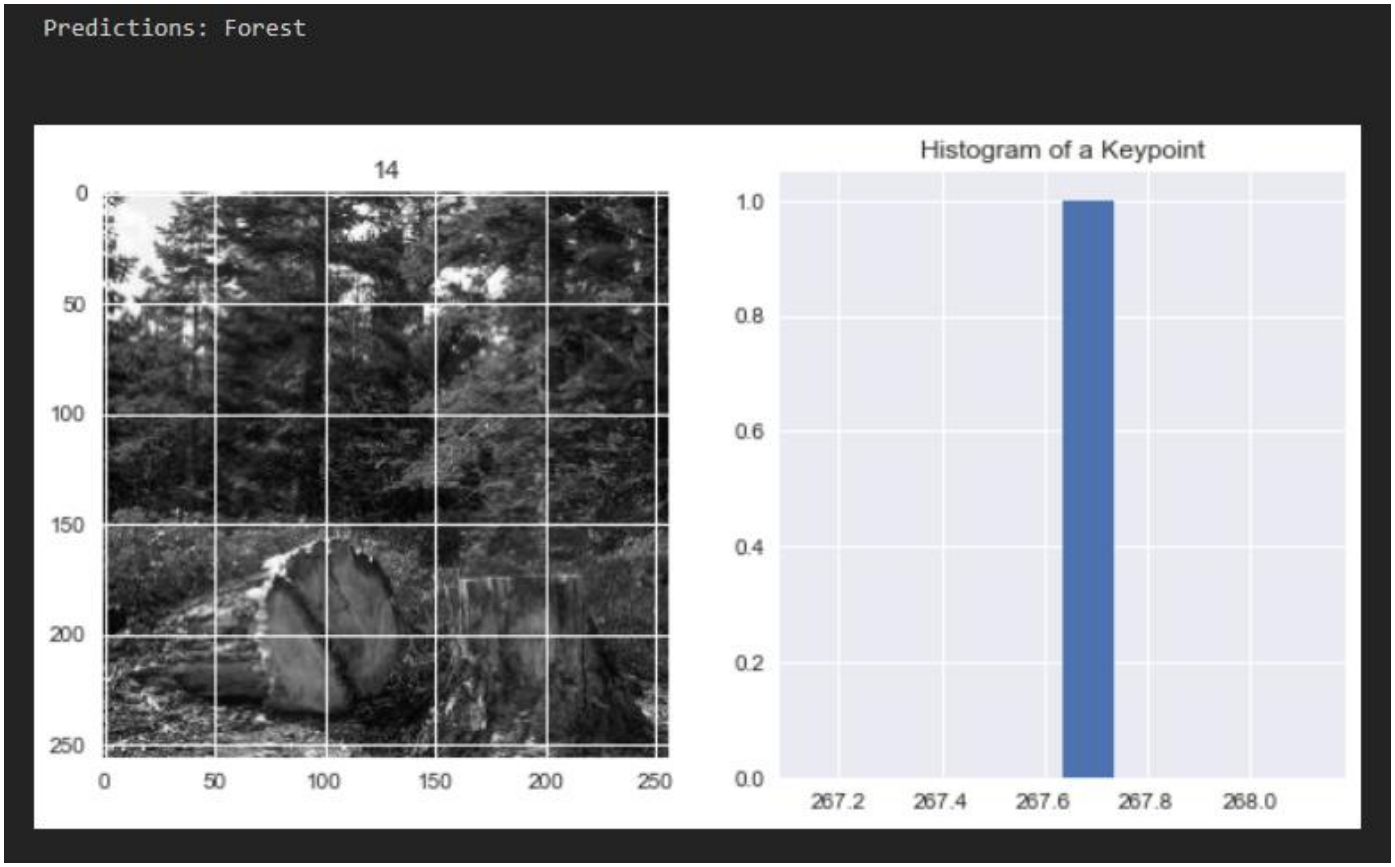
```
Predictions: Highway
```



```
Predictions: Bedroom
```



```
Predictions: Coast
```

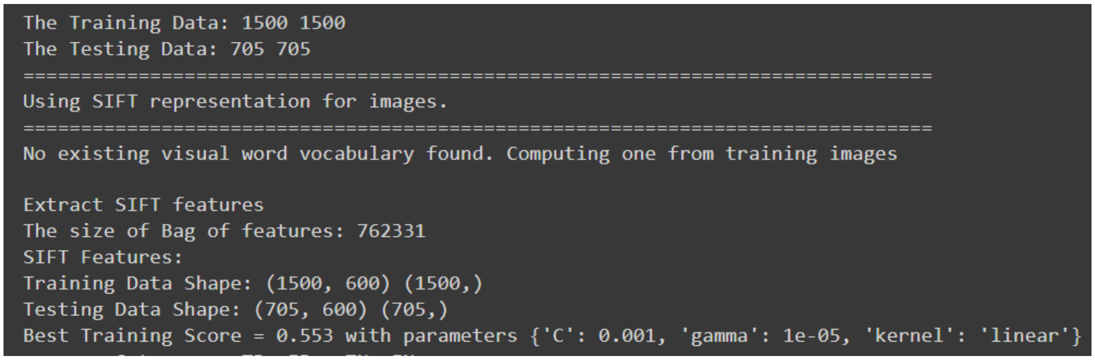


```
Predictions: Forest
```

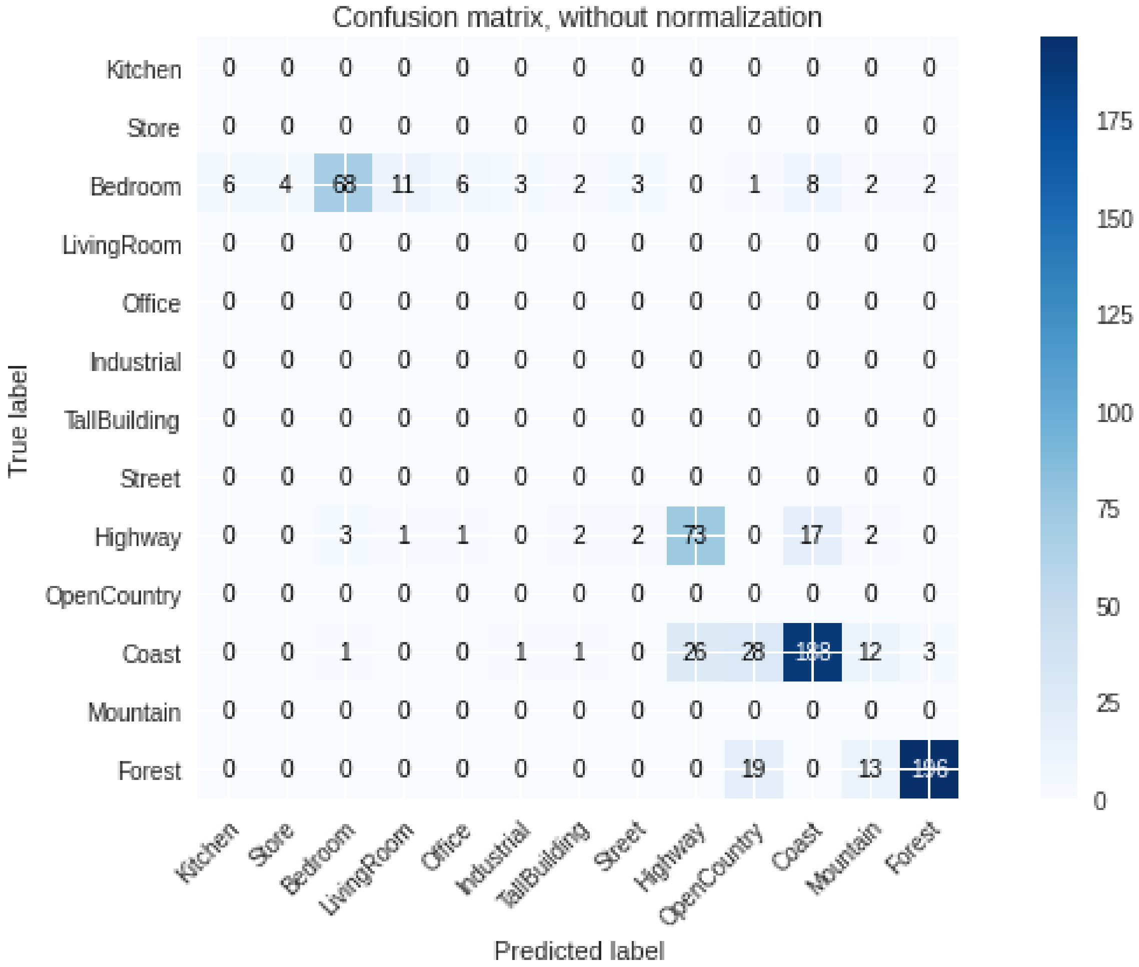
Experiment 7:

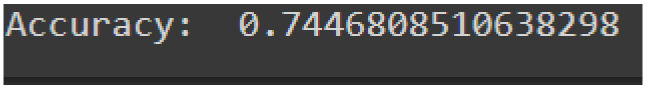
Hyperparamter tunning:

Features	Vocab size	KMeans(# of K)
random	600	600



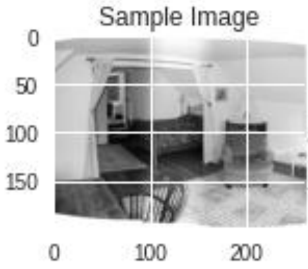
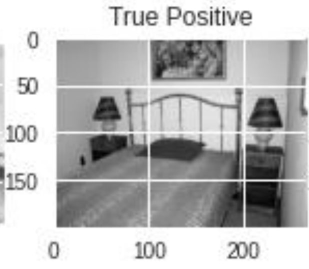
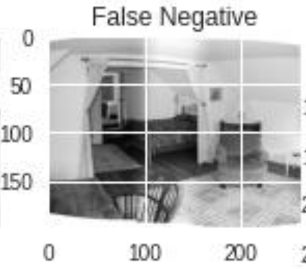
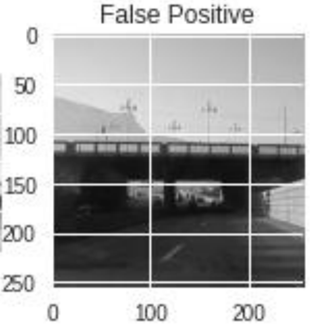
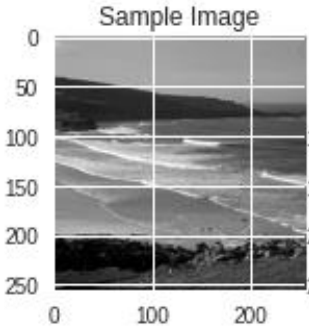
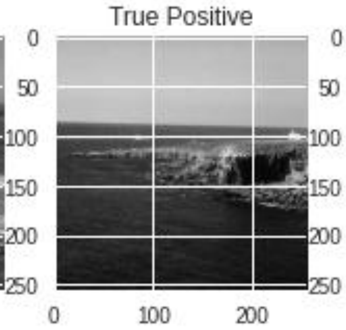
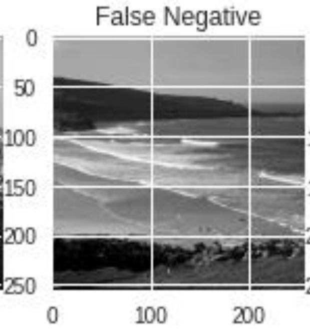
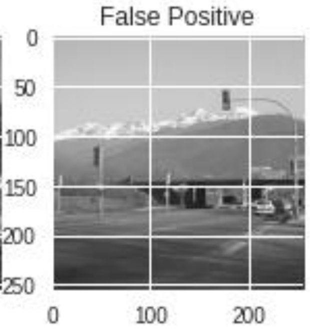
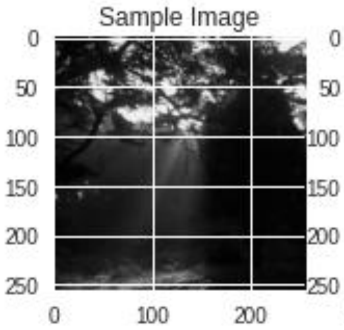
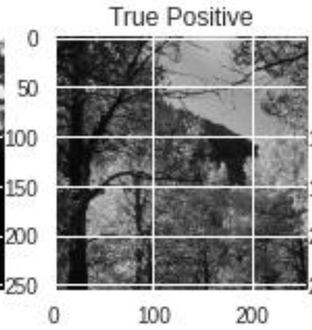
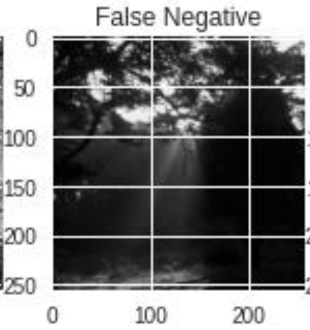
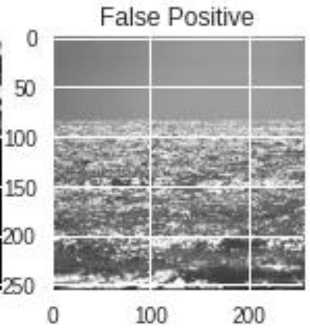
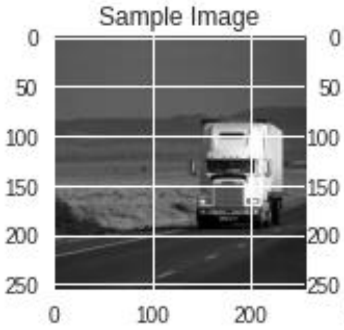
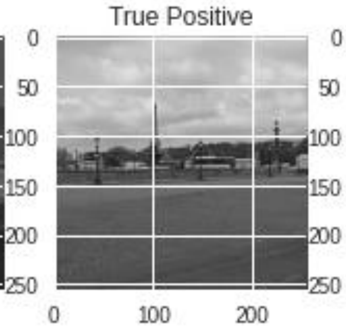
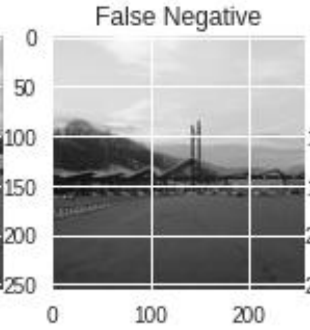
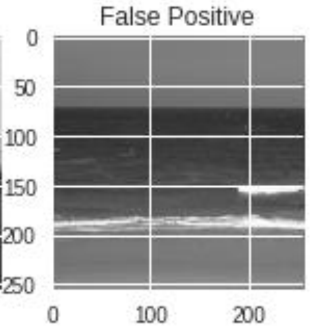
```
The Training Data: 1500 1500
The Testing Data: 705 705
==============================================================================
Using SIFT representation for images.
==============================================================================
No existing visual word vocabulary found. Computing one from training images

Extract SIFT features
The size of Bag of features: 762331
SIFT Features:
Training Data Shape: (1500, 600) (1500,)
Testing Data Shape: (705, 600) (705,)
Best Training Score = 0.553 with parameters {'C': 0.001, 'gamma': 1e-05, 'kernel': 'linear'}
```



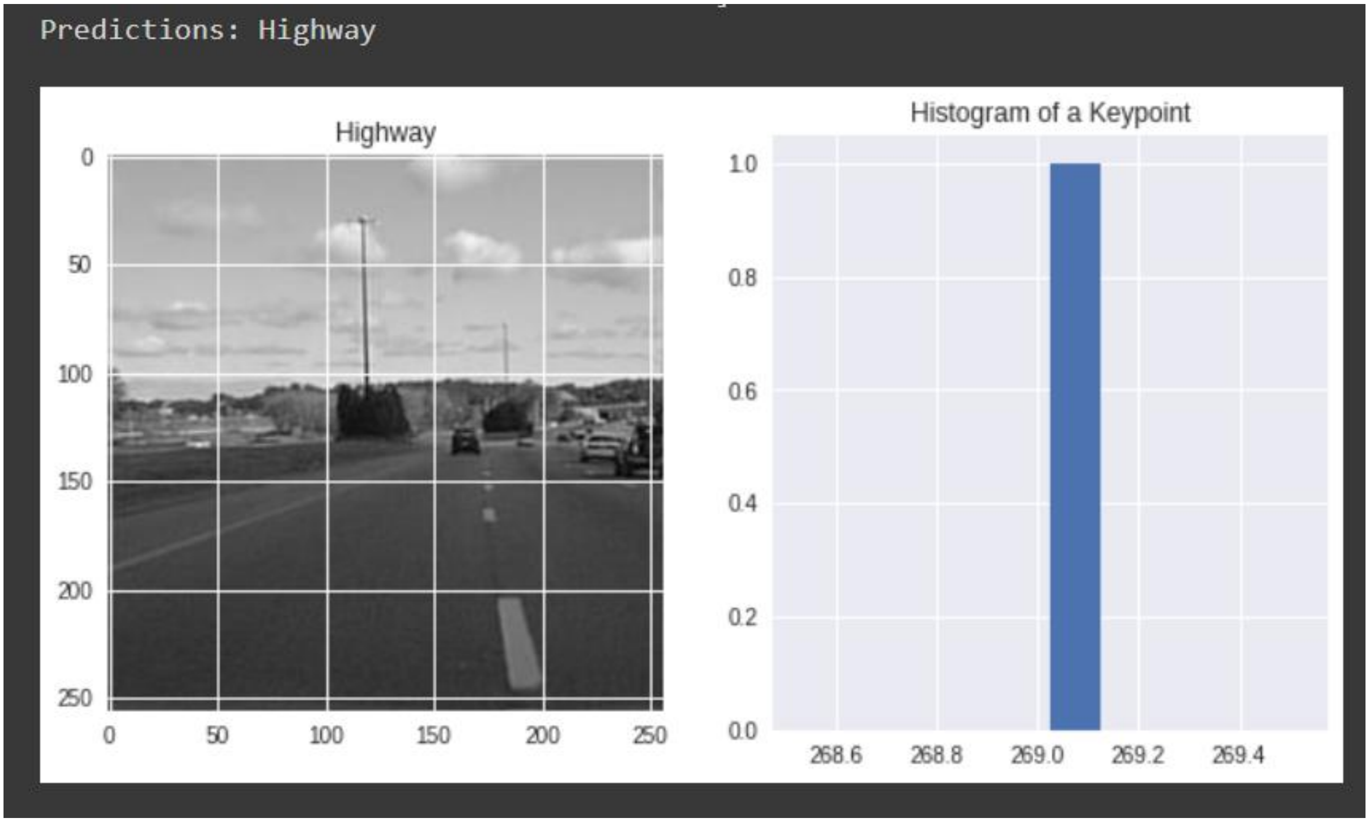


```
Accuracy:  0.7446808510638298
```

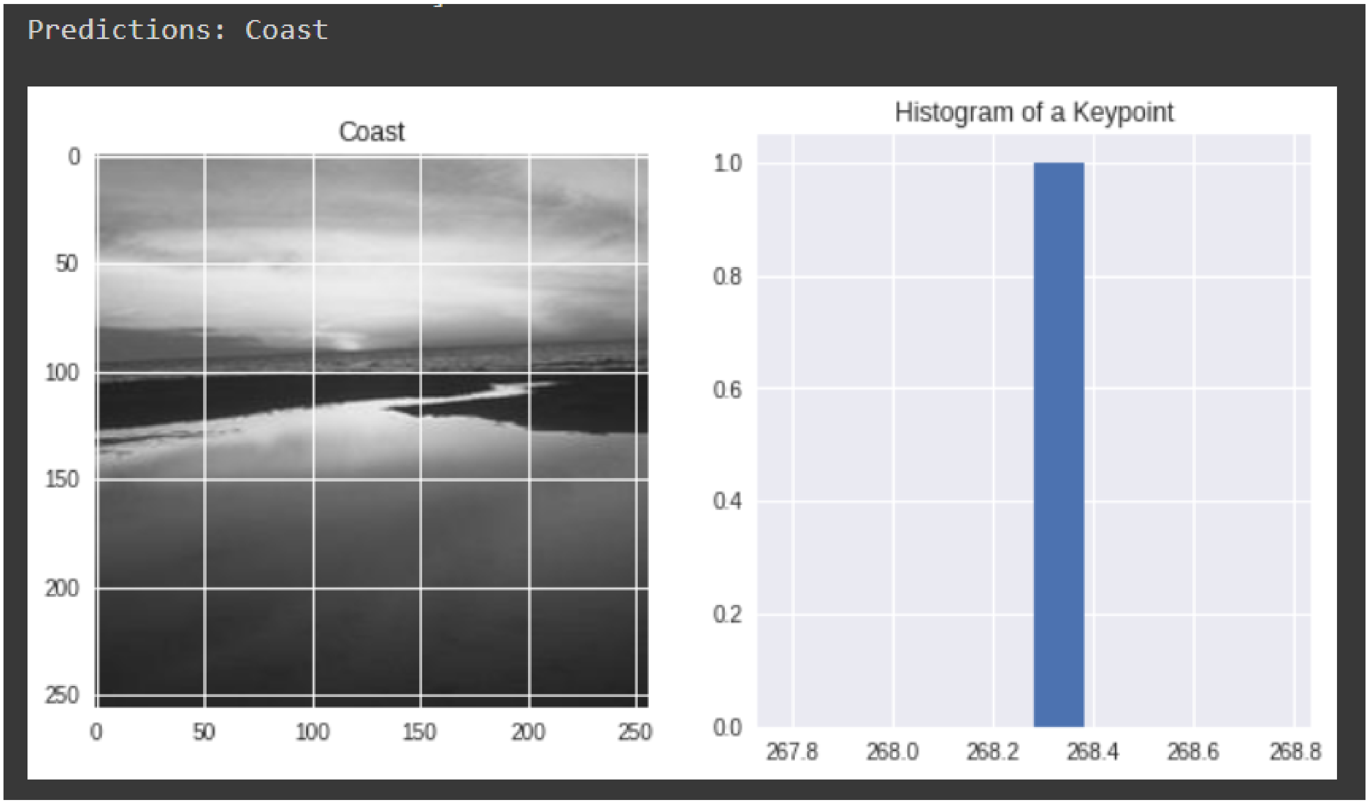
TP,FP,FN Visulaization Table

Category	Sample Image	True Positive	False Negative	False Positive
Bedroom				 False Positive 0 50 100 150 200 250 0 100 200
Coast	 Sample Image 0 50 100 150 200 250 0 100 200	 True Positive 0 50 100 150 200 250 0 100 200	 False Negative 0 50 100 150 200 250 0 100 200	 False Positive 0 50 100 150 200 250 0 100 200
Forest	 Sample Image 0 50 100 150 200 250 0 100 200	 True Positive 0 50 100 150 200 250 0 100 200	 False Negative 0 50 100 150 200 250 0 100 200	 False Positive 0 50 100 150 200 250 0 100 200
Highway	 Sample Image 0 50 100 150 200 250 0 100 200	 True Positive 0 50 100 150 200 250 0 100 200	 False Negative 0 50 100 150 200 250 0 100 200	 False Positive 0 50 100 150 200 250 0 100 200

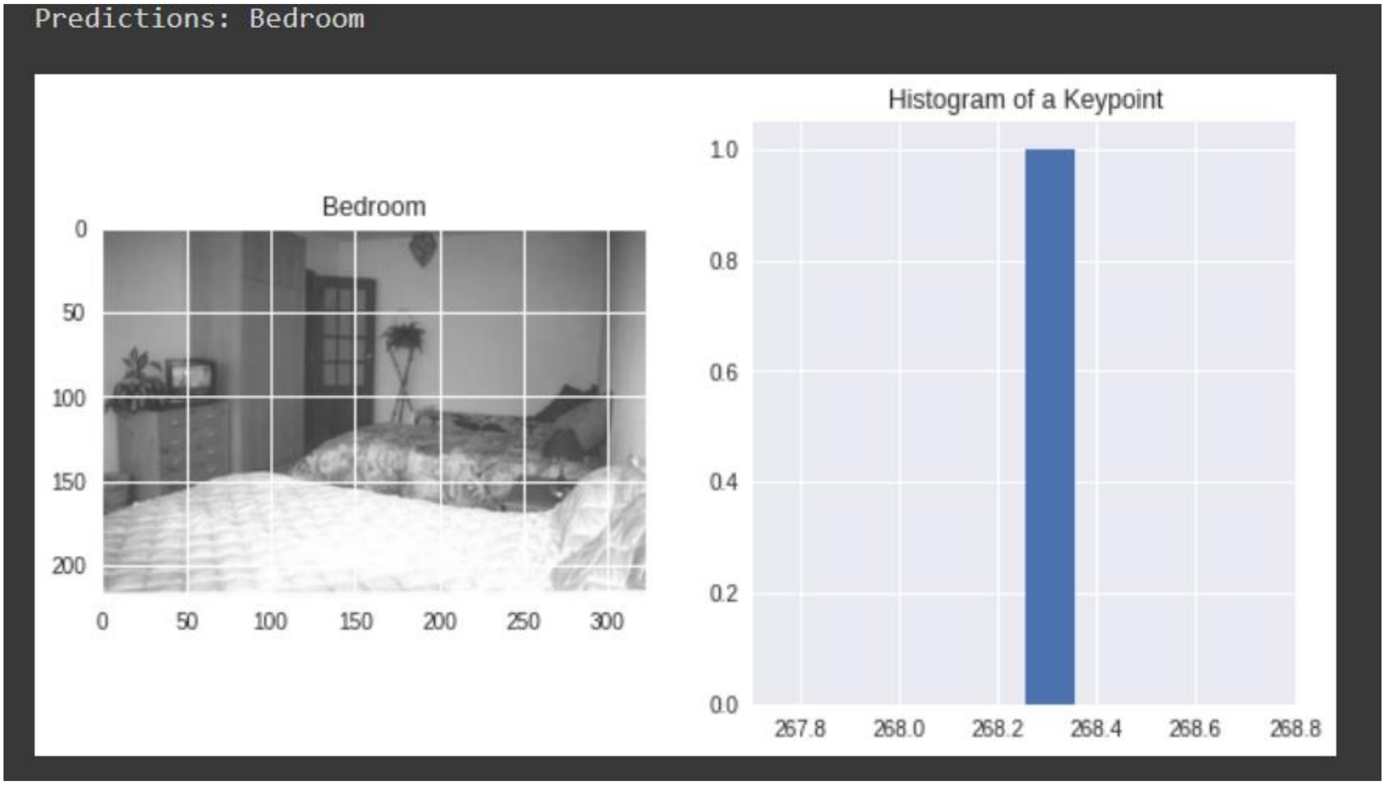
Predictions



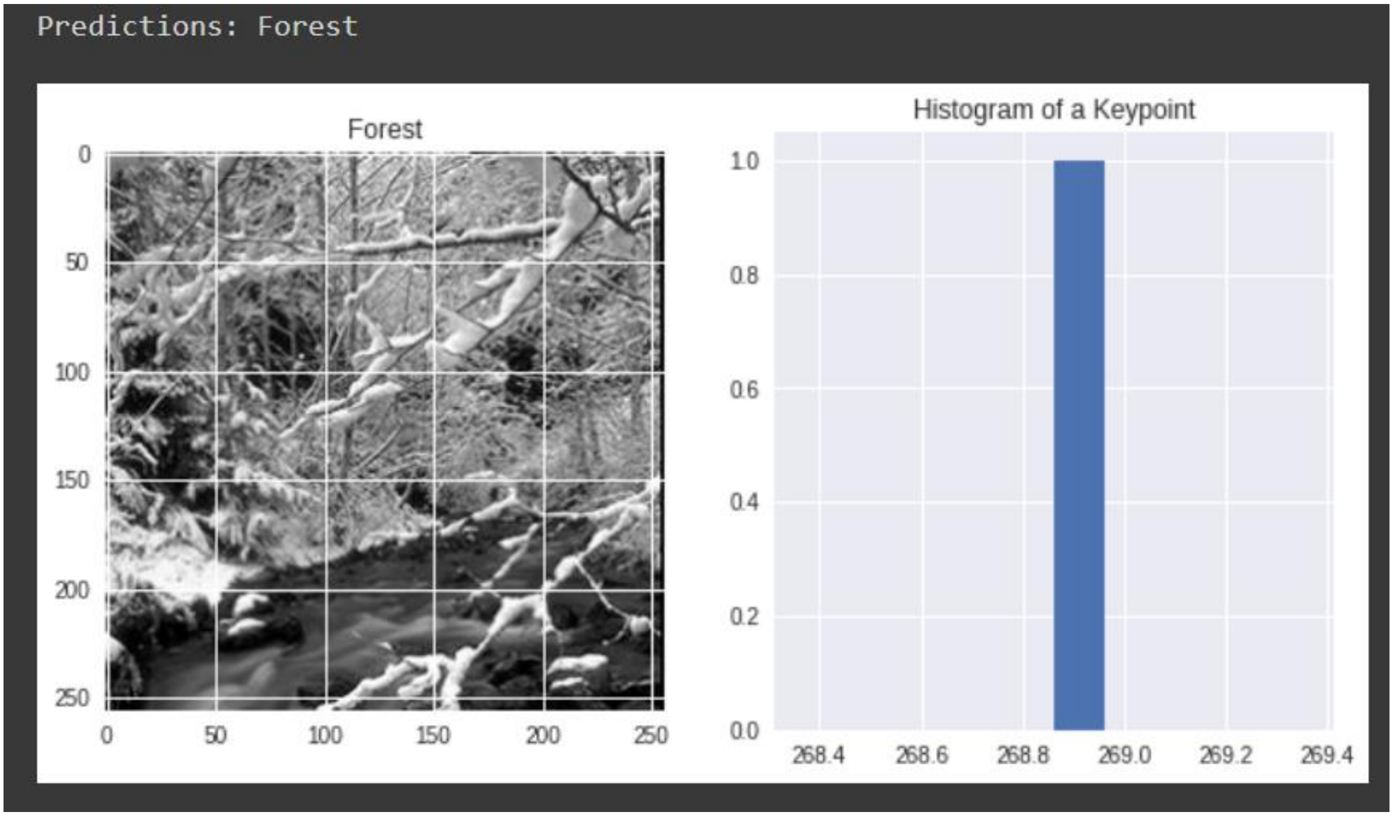
```
Predictions: Highway
```



```
Predictions: Coast
```



```
Predictions: Bedroom
```

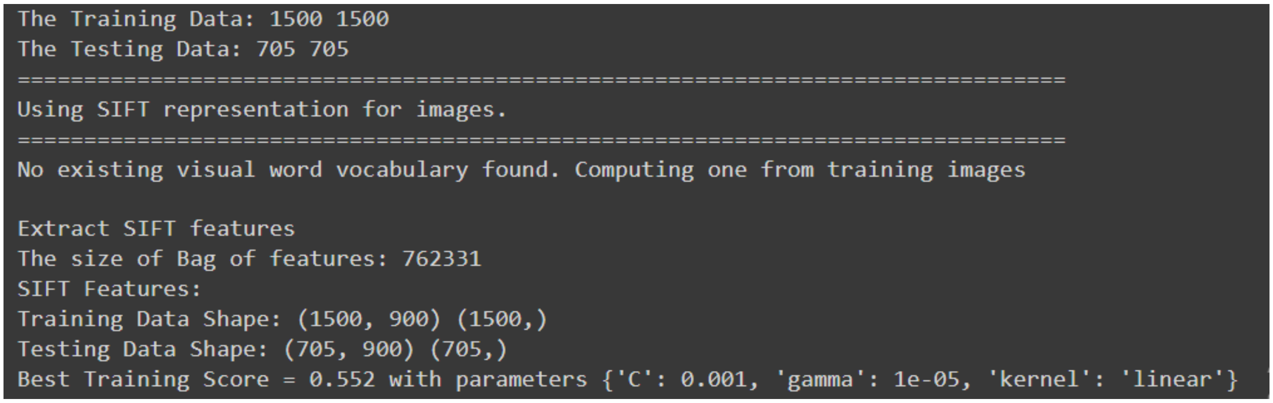


```
Predictions: Forest
```

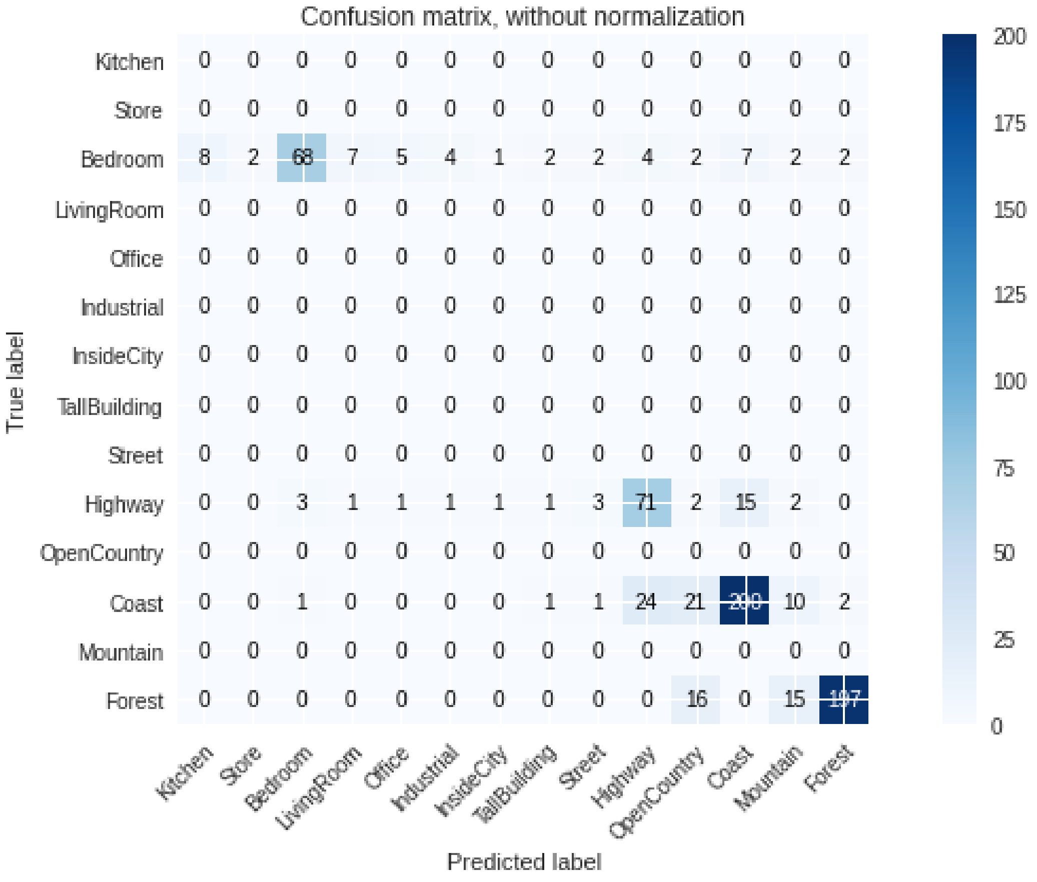
Experiment 8:

Hyperparamter tunning:

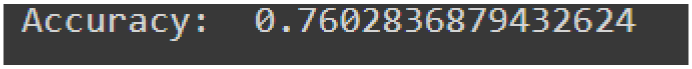
Features	Vocab size	KMeans(# of K)
random	900	900



```
The Training Data: 1500 1500
The Testing Data: 705 705
==============================================================================
Using SIFT representation for images.
==============================================================================
No existing visual word vocabulary found. Computing one from training images

Extract SIFT features
The size of Bag of features: 762331
SIFT Features:
Training Data Shape: (1500, 900) (1500,)
Testing Data Shape: (705, 900) (705,)
Best Training Score = 0.552 with parameters {'C': 0.001, 'gamma': 1e-05, 'kernel': 'linear'}
```

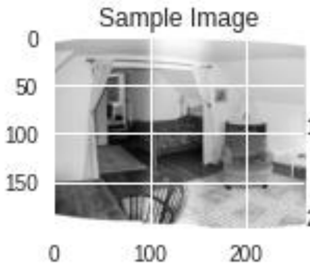
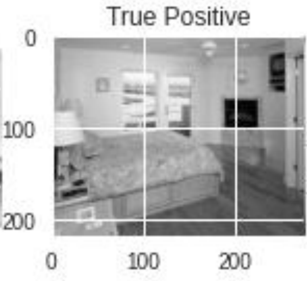
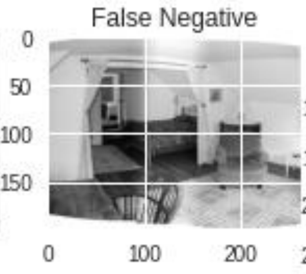
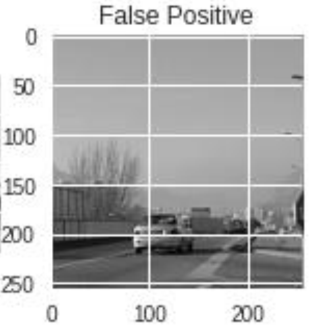
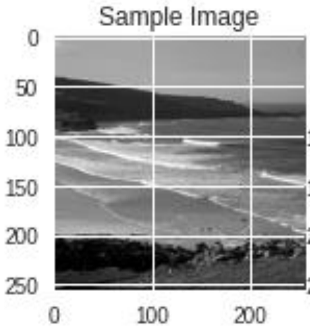
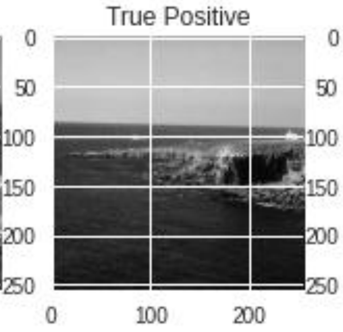
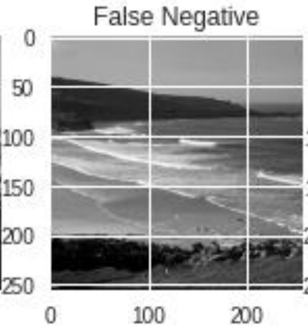
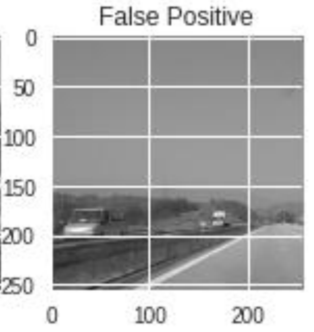
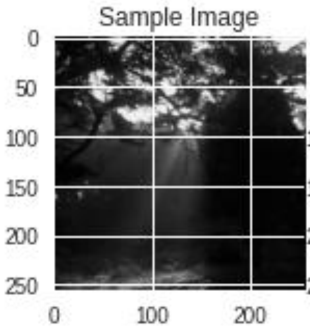
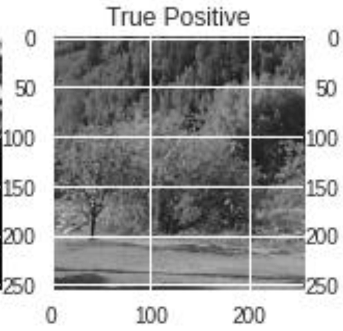
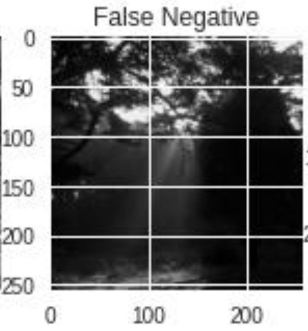
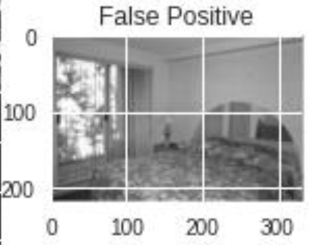
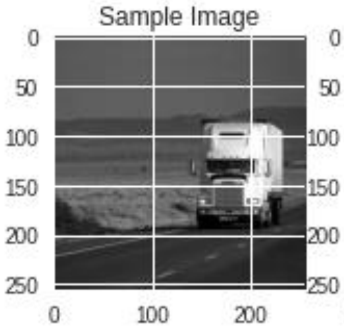
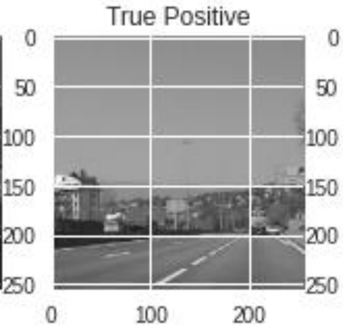
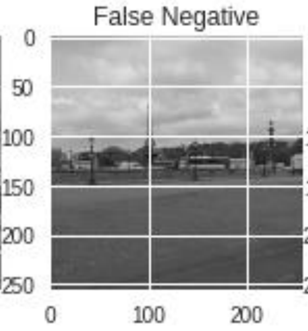
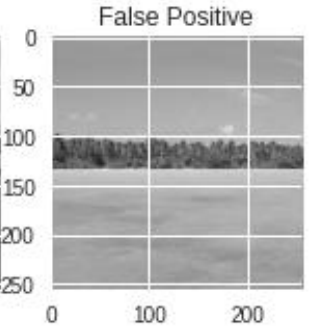


	Kitchen	Store	Bedroom	LivingRoom	Office	Industrial	InsideCity	TallBuilding	Street	Highway	OpenCountry	Coast	Mountain	Forest
Kitchen	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Store	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bedroom	8	2	68	7	5	4	1	2	2	4	2	7	2	2
LivingRoom	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Office	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Industrial	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
TallBuilding	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
Highway	0	0	3	1	1	1	1	1	3	71	2	15	2	0
OpenCountry	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Coast	0	0	1	0	0	0	0	1	1	24	21	200	10	2
Mountain	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Forest	0	0	0	0	0	0	0	0	0	0	16	0	15	197

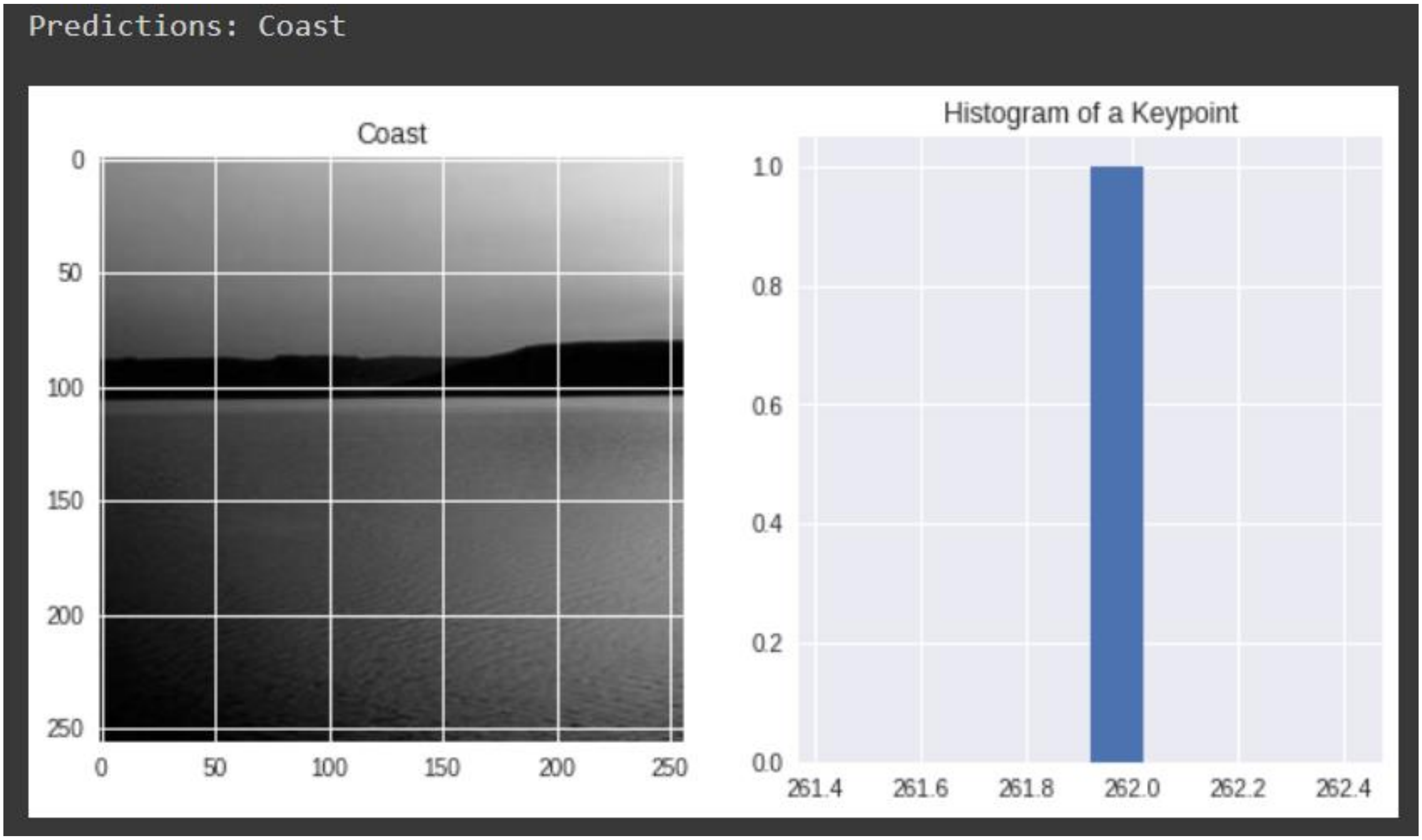


```
Accuracy:  0.7602836879432624
```

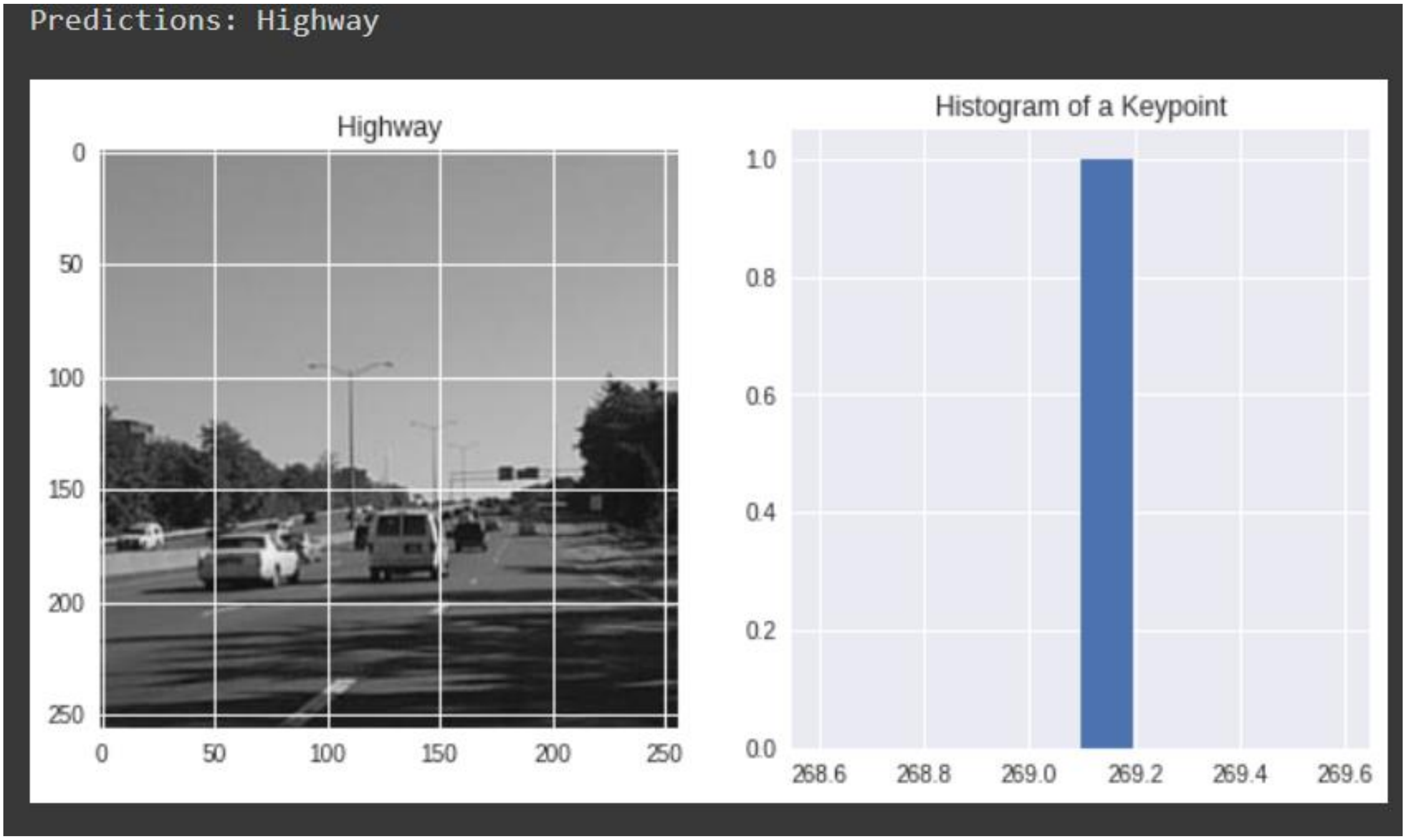
TP,FP,FN Visulaization Table

Category	Sample Image	True Positive	False Negative	False Positive
Bedroom				
Coast				
Forest				
Highway				

Predictions:



```
Predictions: Coast
```



```
Predictions: Highway
```

SVM Discussion:

To discuss the experiments, I am using a table that represents the values of the different parameters used in each experiment. For example number of features, vocab size, and the number of clusters. The next figure shows the distribution of data and shape before feature extraction and after feature extraction. Confusion matrix to display results for each class and then the accuracy of the model within the experiment. The implementation of code is the same as it was given in the starter code.

Analysis of experiments:

Discussing the Bag of SIFT with SVM, After doing experiments on features size and vocab size in KNN now familiar with the values to set for features and vocab size. Therefore here I have done some random experiments on SVM and compared its performance with KNN. Following are highlights of experiments:

- Experiment 1 and 2 is performed with simple SVM without hyperparameter tuning in SVM.
- The accuracy of experiment 1 with KNN is better than SVM. With the same parameters in the KNN model accuracy is 0.70 while SVM is at 0.60. Increasing the features and vocab size did not much affect the model. The model accuracy dropped at experiment 2 with a value of 0.58. Other than 1 and 2 all experiments are done with hyperparameter tunning in SVM using GridSearchCV.
- The results of 3rd experiment are different for both SVM and KNN. SVM performance on this experiment is better than KNN with an accuracy of 0.73.
- For the 4th experiment, I tried to keep the same size for features and vocab. The experiment results in 0.67 accuracy.
- In experiment 6, I did the hyperparameter tunning of experiment 2. The results were great model accuracy increased by 15%.
- In experiments 5,7 and 8, I set the features to random and tried to change the vocab values by (400,600,900) the model accuracy improved at 900 by 0.76.
- The Predictions and Results table for experiments 5,6,7 and 8 are given. The Model Best Performance is at experiment 3.

Reading Part

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

   Talking about the recognizing semantic category of images the simple Bag of Visual Words method does not give imposing results. For this kind of task, the Bag of Visual words feature method does not preserve the spatial information of image features. It represents the image as an orderless collection of local features. Therefore can not capture the shape and semantic information of an object. Finding effective structural information of an object is really important for the tasks like object recognition in heavy occlusion like disturbance. Here comes Spatial Pyramid Matching that uses a kernel-based method to compute global correspondence. Th method subdivides the image and computes the histograms of local features by increasing scale at fine resolution. Using global information as an indication of the existence of an object the model outperforms the simple Bag of Visual features. The Spatial pyramid Matching model also provides useful discriminative results even on a high Variable dataset.

2. **Interpret the "Pyramid Match Kernel" described in section 3.1 of the paper.**

   Pyramid match kernel measures the similarity between any two sets of d-dimensional feature space. Consider the following steps to approximate correspondence:

   - Place multi-resolution grid over point sets in feature space.
   - At each level of resolution compute a weighted sum of several points that match and from match means over here are those points that fall on the same cell of the grid.
   - Points at the finer resolution are highly weighted than coarser resolution.

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

Given figure show the Histogram intersection to find the number of matches at each level of resolution given by L where H_X^ℓ and $\bar{H}_Y^\ell$ are the two histograms of given any two sets of vectors

X and Y in any d-dimensional space. Here I represent the number of points from the X and Y that lie in the ith cell of the grid.

One thing to be noticed over here is that it also includes the number of matches found at finer level L+1 except level L. By representing $\mathcal{I}(H_X^\ell, H_Y^\ell)$ to $\mathcal{I}^\ell$. The scenario here is to deal with that larger cells where matches are found but may also contain non-matching features. The way it works penalizes them by assigning weights to each level. The weight is inversely proportional to the cell width at each level. Therefore now both histogram intersection pyramid-match –kernel is given below.

$$\begin{aligned}\kappa^L(X,Y) &= \mathcal{I}^L + \sum_{\ell=0}^{L-1} \frac{1}{2^{L-\ell}}\left(\mathcal{I}^\ell - \mathcal{I}^{\ell+1}\right) \\ &= \frac{1}{2^L}\mathcal{I}^0 + \sum_{\ell=1}^{L} \frac{1}{2^{L-\ell+1}}\mathcal{I}^\ell \,.\end{aligned}$$

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

Two kinds of feature extraction are performed in this paper. One is named as a weak feature and the other as a strong feature.

- Weak features are the edge points which are the magnitude of the gradient in a given direction collected by using a minimum threshold value. To extract edge points 2 scales and 8 orientations have been used.
- For strong feature extraction SIFT descriptor is used in this paper. SIFT descriptors of 16*16 pixel patches. The model tends to perform well on dense features than the weak ones.