Introduction

Our collaborative project focuses on creating an image recognition application by harnessing the power of Scale-Invariant Feature Transform (SIFT) features and a Bag-of-Words (BoW) model on the Describable Textures Dataset (DTD).

Dataset

The Describable Textures Dataset (DTD) is a comprehensive collection of textured images designed to aid in the analysis, modelling, and synthesis of textures. The DTD comprises 47 different texture classes, each capturing a distinct type of visual texture. Each class has 120 instances in the dataset. These classes encompass a wide range of textures found in the real world, such as fabrics, surfaces, natural materials, and artistic patterns.

We chose the following 4 classes for our model:

- Wrinkled
- Perforated
- Waffled
- Studded

We imported the dataset from TensorFlow.

The total number of images in the dataset used for our analysis is 480.

Within each texture class, the DTD contains a collection of high-quality images that capture variations in different aspects of the texture. These variations include changes in scale, lighting conditions, orientations, and spatial distributions. This diversity enabled us to develop models that are robust to these variations.

We then randomly split our data into train, validation and test sets.

EXAMPLE IMAGES

Train Set:

WRINKLED wrinkled_049.jpg

PERFORATED perforated_0111.jpg

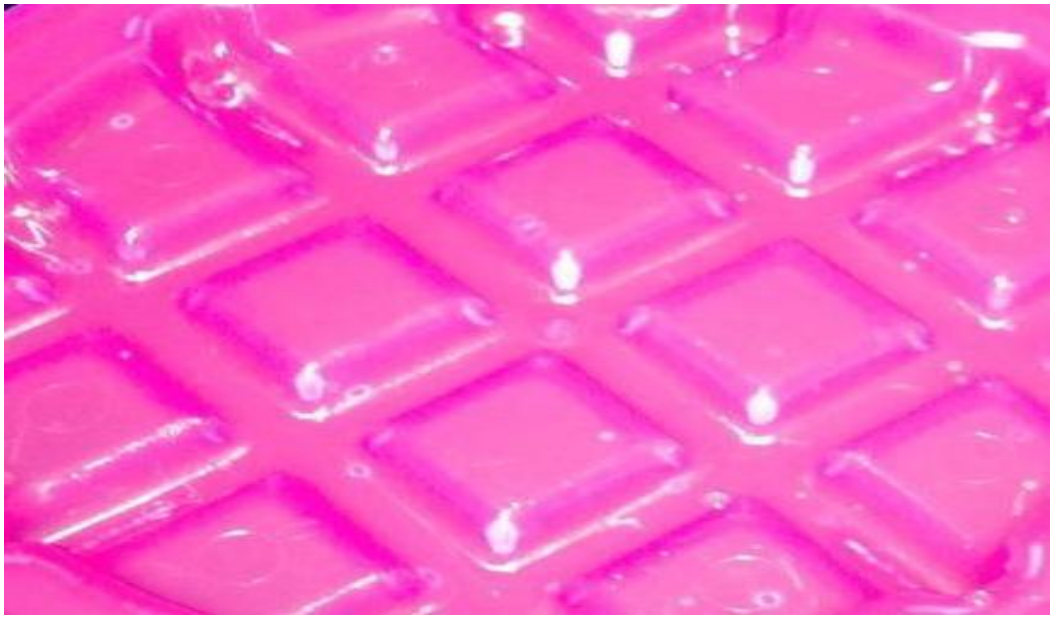
WAFFLED waffled_0074.jpg

STUDDED studded_0177.jpg

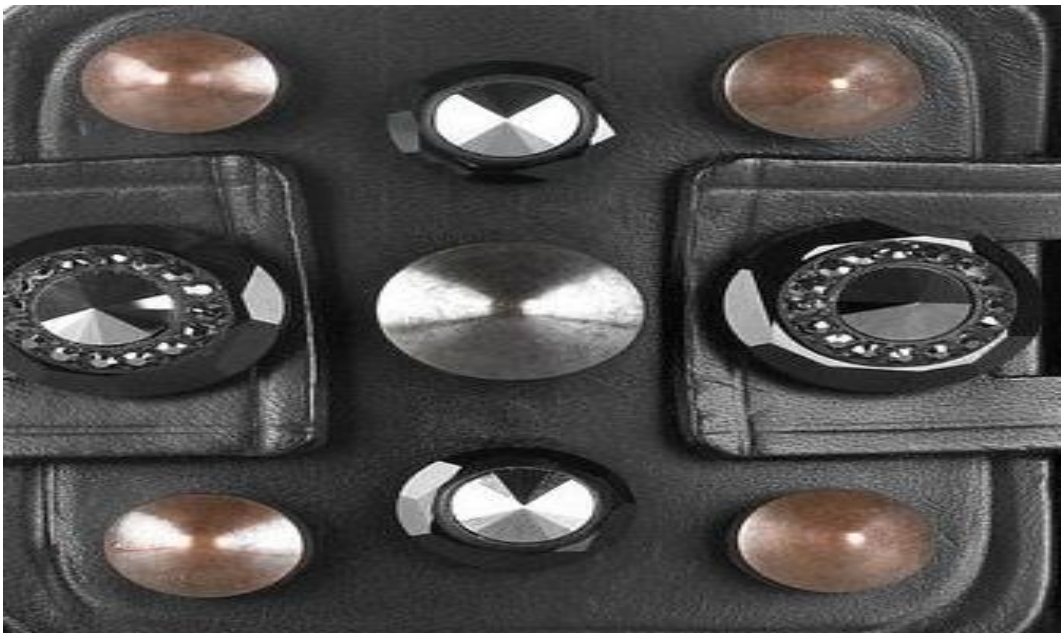
Validation Set:

WRINKLED wrinkled_0027.jpg

PERFORATED perforated_0040.jpg



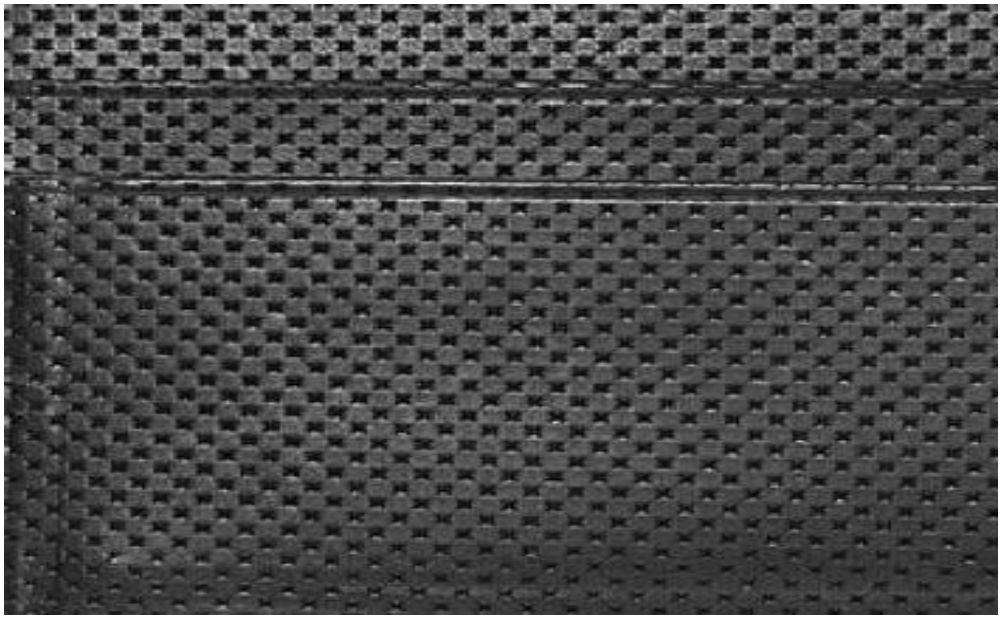
WAFFLED waffled_0120.jpg



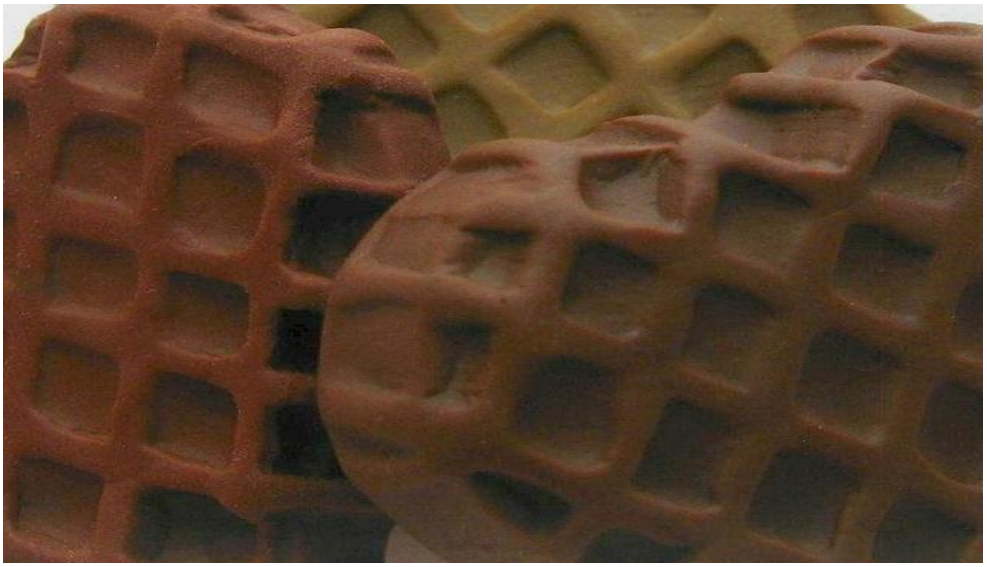
STUDDED studded_0054.jpg

Test Set:

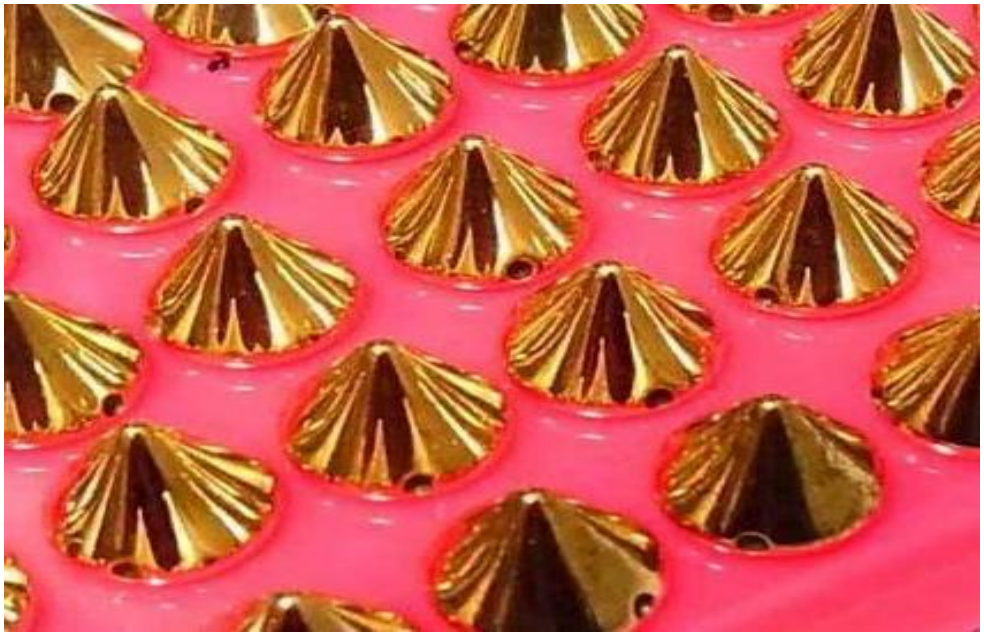
WRINKLED wrinkled_55.jpg



PERFORATED perforated_0138.jpg



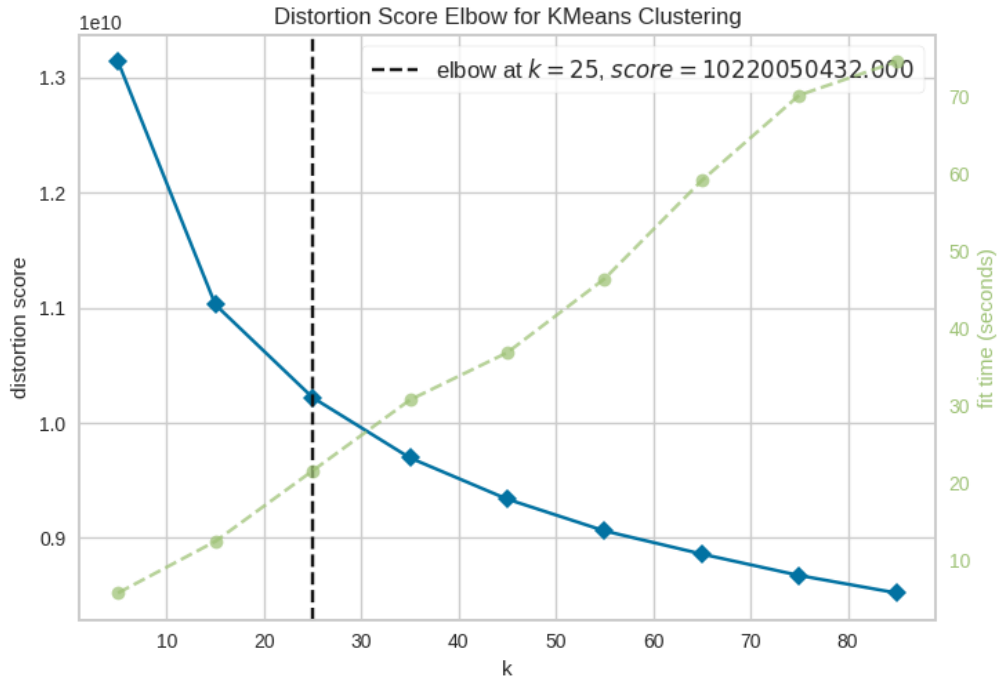
WAFFLED waffled_0128.jpg



STUDDED studded_0118.jpg

Choosing Optimal K

From plot 1 we can see that optimal K = 25 according to the elbow method.

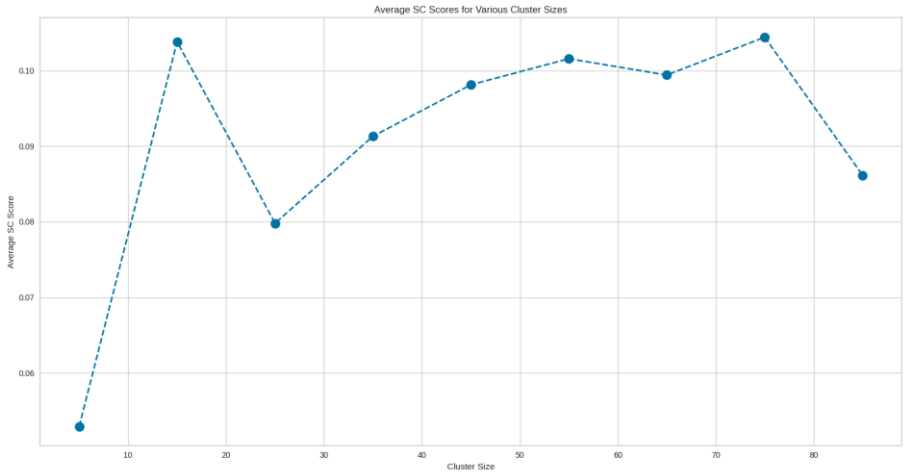


Plot 1: Elbow Method Analysis (random seed = 6)

Average Silhouette Coefficient (SC) Scores for All Cluster Sizes:

K	SC Score
5	0.05293300002813339
15	0.10380099713802338
25	0.07979699969291687
35	0.09130699932575226
45	0.09811600297689438
55	0.10155899822711945
65	0.09940499812364578
75	0.10440900176763535
85	0.0861390009522438

From plot 2 we can see that the optimal cluster size according to the average silhouette coefficient score is 75



Plot 2: Silhouette Score Analysis (random seed = 6)

For K = 25 (Elbow Method)

Model Validation

KNN:

K	Accuracy
1	70.0%
2	69.17%
3	70.0%
4	74.17%
5	72.5%
6	70.83%
7	69.17%
8	71.67%
9	70.0%
10	67.5%
15	67.5%
20	68.33%
25	67.5%
30	65.0%

Table 1: Accuracy for different values of K in KNN classifier on the validation set

Confusion Matrices and Accuracy

Classification Performance with KNN(1)
Accuracy: 70.0%
Confusion Matrix:
[[22 1 5 2]
[2 22 2 4]
[5 1 19 5]
[5 2 2 21]]
Training score: 100.0

Classification Performance with KNN(2)
Accuracy: 69.17%
Confusion Matrix:
[[27 0 3 0]
[3 22 4 1]
[9 3 16 2]
[7 4 1 18]]
Training score: 78.75

Classification Performance with KNN(3)
Accuracy: 70.0%
Confusion Matrix:
[[22 1 6 1]
[2 22 2 4]
[5 2 19 4]
[5 2 2 21]]
Training score: 78.12

Classification Performance with KNN(4)
Accuracy: 74.17%
Confusion Matrix:
[[24 1 4 1]
[3 22 2 3]
[4 2 21 3]
[4 2 2 22]]
Training score: 73.75

Classification Performance with KNN(5)
Accuracy: 72.5%
Confusion Matrix:
[[23 0 6 1]
 [2 22 2 4]
 [5 2 19 4]
 [4 1 2 23]]
Training score: 75.0

Classification Performance with KNN(6)
Accuracy: 70.83%
Confusion Matrix:
[[20 0 7 3]
 [2 22 2 4]
 [6 1 19 4]
 [4 0 2 24]]
Training score: 73.12

Classification Performance with KNN(7)
Accuracy: 69.17%
Confusion Matrix:
[[20 0 8 2]
 [2 22 2 4]
 [7 1 18 4]
 [5 1 1 23]]
Training score: 72.5

Classification Performance with KNN(8)
Accuracy: 71.67%
Confusion Matrix:
[[21 0 7 2]
 [2 20 4 4]
 [5 0 21 4]
 [5 0 1 24]]
Training score: 72.5

Classification Performance with KNN(9)
Accuracy: 70.0%
Confusion Matrix:
[[20 0 8 2]
 [2 19 5 4]
 [5 1 21 3]
 [5 0 1 24]]
Training score: 70.0

Classification Performance with KNN(10)
Accuracy: 67.5%
Confusion Matrix:
[[20 0 8 2]
 [2 18 6 4]
 [8 1 20 1]
 [4 0 3 23]]
Training score: 68.12

Classification Performance with KNN(15)
Accuracy: 67.5%
Confusion Matrix:
[[21 0 8 1]
 [2 17 7 4]
 [6 1 20 3]
 [5 0 2 23]]
Training score: 65.62

Classification Performance with KNN(20)
Accuracy: 68.33%
Confusion Matrix:
[[20 0 9 1]
 [2 17 8 3]
 [5 0 22 3]
 [4 0 3 23]]
Training score: 64.38

Classification Performance with KNN(25)
Accuracy: 67.5%
Confusion Matrix:
[[21 0 8 1]
 [2 16 9 3]
 [6 0 22 2]
 [4 1 3 22]]
Training score: 62.5

Classification Performance with KNN(30)
Accuracy: 65.0%
Confusion Matrix:
[[20 0 9 1]
 [2 14 11 3]
 [5 0 23 2]
 [4 1 4 21]]
Training score: 60.62

Conclusion

As can be seen from table 1 and the above-mentioned results we get the best accuracy for KNN at K = 4; Accuracy = 74.17%

SVM:

C	**Accuracy**
10	75.0%
20	74.17%
30	75.0%
40	75.83%
50	76.67%
60	77.5%
70	79.17%
80	79.17%
90	76.67%
100	75.0%

Table 2: Accuracy for different values of C in SVM classifier on the validation set

Confusion Matrices and Accuracy

Classification Performance with SVM(10)
Accuracy: 75.0%
Confusion Matrix:
[[21 0 8 1]
[2 24 0 4]
[3 2 24 1]
[3 1 5 21]]
Training score: 75.0

Classification Performance with SVM(20)
Accuracy: 74.17%
Confusion Matrix:
[[24 0 5 1]
[2 23 2 3]
[6 2 21 1]
[3 1 5 21]]
Training score: 80.62
Classification Performance with SVM(30)
Accuracy: 75.0%
Confusion Matrix:

[[23 0 5 2]
 [1 24 1 4]
 [4 1 22 3]
 [3 1 5 21]]
Training score: 81.25

Classification Performance with SVM(40)
Accuracy: 75.83%
Confusion Matrix:
[[25 0 4 1]
 [1 24 1 4]
 [4 1 21 4]
 [2 2 5 21]]
Training score: 82.5

Classification Performance with SVM(50)
Accuracy: 76.67%
Confusion Matrix:
[[24 0 4 2]
 [1 24 1 4]
 [4 1 21 4]
 [3 0 4 23]]
Training score: 83.12

Classification Performance with SVM(60)
Accuracy: 77.5%
Confusion Matrix:
[[25 0 4 1]
 [1 24 1 4]
 [5 1 20 4]
 [3 0 3 24]]
Training score: 85.0

Classification Performance with SVM(70)
Accuracy: 79.17%
Confusion Matrix:
[[26 0 3 1]
 [1 24 1 4]
 [4 1 21 4]
 [3 0 3 24]]
Training score: 85.0

Classification Performance with SVM(80)
Accuracy: 79.17%
Confusion Matrix:

[[25 0 3 2]
 [1 24 1 4]
 [5 2 20 3]
 [2 0 2 26]]
Training score: 85.0

Classification Performance with SVM(90)
Accuracy: 76.67%
Confusion Matrix:
[[25 0 3 2]
 [1 24 1 4]
 [5 2 19 4]
 [3 1 2 24]]
Training score: 85.62

Classification Performance with SVM(100)
Accuracy: 75.0%
Confusion Matrix:
[[24 0 4 2]
 [1 24 1 4]
 [5 3 18 4]
 [3 1 2 24]]
Training score: 85.62

Conclusion

As can be seen from table 2 and the above-mentioned results we get the best accuracy for SVM at C = 70; Accuracy = 79.17%

AdaBoost:

num_estimators	**Accuracy**
10	64.17%
50	67.5%
100	71.67%
150	70.83%
200	71.67%
250	70.83%

Table 3: Accuracy for different values of n in AdaBoost classifier on the validation set

Confusion Matrices and Accuracy

Classification Performance with AdaBoost(10)
Accuracy: 64.17%
Confusion Matrix:
[[19 0 10 1]
[0 20 6 4]
[6 2 19 3]
[0 4 7 19]]
Training score: 76.88

Classification Performance with AdaBoost(50)
Accuracy: 67.5%
Confusion Matrix:
[[20 0 9 1]
[0 23 5 2]
[6 2 19 3]
[1 4 6 19]]
Training score: 81.25

Classification Performance with AdaBoost(100)
Accuracy: 71.67%
Confusion Matrix:
[[19 0 10 1]
[0 23 5 2]
[4 3 22 1]
[1 3 4 22]]
Training score: 88.75

Classification Performance with AdaBoost(150)
Accuracy: 70.83%
Confusion Matrix:
[[19 0 10 1]
[0 26 3 1]
[4 3 21 2]
[2 4 5 19]]
Training score: 90.0

Classification Performance with AdaBoost(200)
Accuracy: 71.67%
Confusion Matrix:
[[19 1 9 1]
[0 28 1 1]
[5 5 18 2]
[1 4 4 21]]
Training score: 88.75

Classification Performance with AdaBoost(250)
Accuracy: 70.83%
Confusion Matrix:
[[18 1 10 1]
[0 26 3 1]
[5 4 20 1]
[2 4 3 21]]
Training score: 90.62

Conclusion

As can be seen from table 3 and the above-mentioned results we get the best accuracy for AdaBoost at num_estimators = 100; Accuracy = 71.67%

Model Testing

Classifier	**Accuracy**
KNN(4)	74.17%
SVM(70)	78.33%
AdaBoost(100)	76.67%

Table 4: Accuracy for different tuned classifiers on the test set

Confusion Matrices and Accuracy

Classification Performance with KNN(4)
Accuracy: 74.17%
Confusion Matrix:
[[28 1 0 1]
[5 18 5 2]
[7 0 23 0]
[7 2 1 20]]
Training score: 73.75

Classification Performance with SVM(70)
Accuracy: 78.33%
Confusion Matrix:
[[26 0 2 2]
 [3 23 3 1]
 [6 0 24 0]
 [5 1 3 21]]
Training score: 85.0

Classification Performance with AdaBoost(100)
Accuracy: 76.67%
Confusion Matrix:
[[25 0 4 1]
 [0 25 3 2]
 [3 5 22 0]
 [2 2 6 20]]
Training score: 88.75

Conclusion

As can be seen from table 4 and the above-mentioned results we get the best accuracy for SVM;
Accuracy = 78.33%

For K = 75 (Silhouette Score)

Model Validation

KNN:

K	Accuracy
1	70.83%
2	70.0%
3	70.83%
4	71.67%
5	70.83%
6	72.5%
7	70.83%
8	70.83%
9	70.83%
10	70.83%
15	68.33%
20	70.0%
25	66.67%
30	64.17%

Table 5: Accuracy for different values of K in KNN classifier on the validation set

Confusion Matrices and Accuracy

Classification Performance with KNN(1)
Accuracy: 70.83%
Confusion Matrix:
[[23 0 5 2]
[1 23 3 3]
[5 1 21 3]
[5 5 2 18]]
Training score: 100.0

Classification Performance with KNN(2)
Accuracy: 70.0%
Confusion Matrix:
[[25 0 4 1]
 [2 25 2 1]
 [9 1 19 1]
 [7 5 3 15]]
Training score: 82.5

Classification Performance with KNN(3)
Accuracy: 70.83%
Confusion Matrix:
[[22 0 5 3]
 [2 24 2 2]
 [6 2 20 2]
 [8 1 2 19]]
Training score: 78.12

Classification Performance with KNN(4)
Accuracy: 71.67%
Confusion Matrix:
[[23 0 5 2]
 [3 21 4 2]
 [5 1 20 4]
 [5 1 2 22]]
Training score: 81.88

Classification Performance with KNN(5)
Accuracy: 70.83%
Confusion Matrix:
[[21 0 8 1]
 [3 21 4 2]
 [7 1 21 1]
 [5 1 2 22]]
Training score: 78.12

Classification Performance with KNN(6)
Accuracy: 72.5%
Confusion Matrix:
[[23 0 5 2]
 [3 21 4 2]
 [6 1 20 3]
 [5 0 2 23]]
Training score: 74.38

Classification Performance with KNN(7)
Accuracy: 70.83%
Confusion Matrix:
[[21 0 7 2]
 [2 21 4 3]
 [5 2 20 3]
 [5 0 2 23]]
Training score: 76.25

Classification Performance with KNN(8)
Accuracy: 70.83%
Confusion Matrix:
[[23 0 5 2]
 [2 19 4 5]
 [6 2 20 2]
 [5 0 2 23]]
Training score: 71.88

Classification Performance with KNN(9)
Accuracy: 70.83%
Confusion Matrix:
[[22 0 6 2]
 [2 18 6 4]
 [6 1 21 2]
 [5 0 1 24]]
Training score: 70.62

Classification Performance with KNN(10)
Accuracy: 69.17%
Confusion Matrix:
[[22 0 7 1]
 [2 18 5 5]
 [7 1 20 2]
 [6 0 1 23]]
Training score: 68.12

Classification Performance with KNN(15)
Accuracy: 68.33%
Confusion Matrix:
[[23 0 6 1]
 [2 16 7 5]
 [8 0 20 2]
 [6 0 1 23]]
Training score: 65.62

Classification Performance with KNN(20)
Accuracy: 70.0%
Confusion Matrix:
[[23 0 5 2]
 [2 15 8 5]
 [6 0 22 2]
 [5 0 1 24]]
Training score: 61.88

Classification Performance with KNN(25)
Accuracy: 66.67%
Confusion Matrix:
[[23 0 5 2]
 [2 12 12 4]
 [6 0 22 2]
 [6 0 1 23]]
Training score: 61.25

Classification Performance with KNN(30)
Accuracy: 64.17%
Confusion Matrix:
[[22 0 6 2]
 [2 11 13 4]
 [6 0 22 2]
 [4 0 4 22]]
Training score: 58.75

Conclusion

As can be seen from table 5 and the above-mentioned results we get the best accuracy for AdaBoost at K = 6; Accuracy = 72.5%

SVM:

C	Accuracy
10	75.0%
20	74.17%
30	74.17%
40	75.0%
50	75.0%
60	75.0%
70	75.83%
80	76.67%
90	76.67%
100	76.67%

Table 6: Accuracy for different values of C in SVM classifier on the validation set

Confusion Matrices and Accuracy

Classification Performance with SVM(10)
Accuracy: 75.0%
Confusion Matrix:
[[21 0 8 1]
[1 24 1 4]
[4 1 24 1]
[2 0 7 21]]
Training score: 78.12

Classification Performance with SVM(20)
Accuracy: 74.17%
Confusion Matrix:
[[20 0 9 1]
[1 25 1 3]
[6 1 22 1]
[2 0 6 22]]
Training score: 85.62

Classification Performance with SVM(30)
Accuracy: 74.17%
Confusion Matrix:
[[20 0 9 1]
 [1 25 1 3]
 [4 1 22 3]
 [2 1 5 22]]
Training score: 86.88

Classification Performance with SVM(40)
Accuracy: 75.0%
Confusion Matrix:
[[20 0 8 2]
 [1 25 1 3]
 [4 1 22 3]
 [4 1 2 23]]
Training score: 89.38

Classification Performance with SVM(50)
Accuracy: 75.0%
Confusion Matrix:
[[21 0 7 2]
 [1 25 1 3]
 [5 1 21 3]
 [2 3 2 23]]
Training score: 92.5

Classification Performance with SVM(60)
Accuracy: 75.0%
Confusion Matrix:
[[21 0 7 2]
 [1 25 0 4]
 [4 2 21 3]
 [2 3 2 23]]
Training score: 92.5

Classification Performance with SVM(70)
Accuracy: 75.83%
Confusion Matrix:
[[22 0 6 2]
 [1 25 0 4]
 [4 2 21 3]
 [2 3 2 23]]
Training score: 93.12

Classification Performance with SVM(80)
Accuracy: 76.67%
Confusion Matrix:
[[22 0 6 2]
[1 25 0 4]
[3 2 22 3]
[2 3 2 23]]
Training score: 93.12

Classification Performance with SVM(90)
Accuracy: 76.67%
Confusion Matrix:
[[22 0 6 2]
[1 25 0 4]
[3 2 22 3]
[2 3 2 23]]
Training score: 93.12

Classification Performance with SVM(100)
Accuracy: 76.67%
Confusion Matrix:
[[22 0 6 2]
[1 25 0 4]
[3 2 22 3]
[2 3 2 23]]
Training score: 93.75

Conclusion

As can be seen from table 2 and the above-mentioned results we get the best accuracy for SVM at C = 80; Accuracy = 76.17%

AdaBoost:

num_estimators	**Accuracy**
10	54.17%
50	57.5%
100	65.83%
150	66.67%
200	67.5%
250	69.17%

Table 7: Accuracy for different values of n in AdaBoost classifier on the validation set

Confusion Matrices and Accuracy

Classification Performance with AdaBoost(10)
Accuracy: 54.17%
Confusion Matrix:
[[11 1 15 3]
 [0 18 8 4]
 [9 1 18 2]
 [0 3 9 18]]
Training score: 69.38

Classification Performance with AdaBoost(50)
Accuracy: 57.5%
Confusion Matrix:
[[12 0 15 3]
 [0 18 10 2]
 [5 1 20 4]
 [1 3 7 19]]
Training score: 76.88

Classification Performance with AdaBoost(100)
Accuracy: 65.83%
Confusion Matrix:
[[18 0 10 2]
 [0 21 6 3]
 [4 1 21 4]
 [1 3 7 19]]
Training score: 87.5

Classification Performance with AdaBoost(150)
Accuracy: 66.67%
Confusion Matrix:
[[18 0 8 4]
 [0 22 4 4]
 [4 1 20 5]
 [1 3 6 20]]
Training score: 90.0

Classification Performance with AdaBoost(200)
Accuracy: 67.5%
Confusion Matrix:
[[19 0 10 1]
 [0 24 3 3]
 [5 1 19 5]
 [1 3 7 19]]
Training score: 93.75

Classification Performance with AdaBoost(250)
Accuracy: 69.17%
Confusion Matrix:
[[21 0 7 2]
 [0 24 3 3]
 [5 1 19 5]
 [1 3 7 19]]
Training score: 94.38
Best AdaBoost Accuracy: 69.17 found for num_estimators = 250

Conclusion

As can be seen from table 7 and the above-mentioned results we get the best accuracy for AdaBoost at num_estimators = 250; Accuracy = 69.17%

Model Testing

Classifier	**Accuracy**
KNN(6)	76.67%
SVM(80)	80.83%
AdaBoost(250)	79.17%

Table 8: Accuracy for different tuned classifiers on the test set

Confusion Matrices and Accuracy

Classification Performance with KNN(6)
Accuracy: 76.67%
Confusion Matrix:
[[30 0 0 0]
 [6 18 6 0]
 [9 0 21 0]
 [6 0 1 23]]
Training score: 74.38

Classification Performance with SVM(80)
Accuracy: 80.83%
Confusion Matrix:
[[26 0 2 2]
[0 27 2 1]
[5 1 22 2]
[6 1 1 22]]
Training score: 93.12

Classification Performance with AdaBoost(250)
Accuracy: 79.17%
Confusion Matrix:
[[26 0 4 0]
[0 22 6 2]
[2 2 25 1]
[3 0 5 22]]
Training score: 94.38

Conclusion

As can be seen from table 8 and the above-mentioned results we get the best accuracy for SVM; Accuracy = 80.83%

Conclusion

The best classifier is trained on a BOW model with a codebook consisting of 75 codewords, the best performance is achieved by the SVM classifier. This could be attributed to the fact that a more diverse codebook would be able to handle more variable data and capture some more trends that might be missed otherwise. Good performance with the SVM classifier signifies that the data is linearly separable. At C=80, we risk overfitting our classifier but we still get good testing performance which signifies that the training instances give a good generalisation of the real world.

Contribution

- **Lakshya Mittal (222521408)**
  Project management: Took lead to initialise and schedule meetings, helped with the distribution of the work and assigning tasks.
  Report: Complied all the results, conducted research and drafted the report for submission.

- **Joel Paul Varghese (222006748)**
  Model - Incorporated the elbow method to determine optimal K value for the classification models.
  Analysis – Tested the KNN, AdaBoost and SVM classifiers and performed analysis to understand the quality of the clustering algorithms using inter cluster distance maps.

- **Tim Harris**
  Model - Implemented the KNN, AdaBoost and SVM classifiers.
  Project management: Took lead to initialise and schedule meetings, helped with the distribution of the work and assigning tasks.