Statistical Pattern Recognition
Assignment-2

Kapil Kumar Bhardwaj – CS24MT012
Manish Bisht – CS24MT022
Mridul Chandrawanshi –CS24MT002

TABLE OF CONTENT

DATASET-1: Nonlinearly separable classes

1.1 Performance Matrices for different number of GMM Components

TABLE 1: Performance Matrices for Different Number of GMM Components(K)

K	ACCURACY	MEAN PRECISION	MEAN RECALL	MEAN F1-SCORE
1	0.6373	0.6420	0.6374	0.6381
2	0.6955	0.6992	0.6954	0.6952
4	0.7155	0.7174	0.7153	0.7122
8	0.7238	0.7186	0.7235	0.7032
16	0.7255	0.7288	0.7252	0.7161
32	0.3328	0.1109	0.3333	0.1665
64	0.6622	0.6629	0.6623	0.6621

OBSERVATION

The performance matrices is improve as the value of k increases, from k=1 to k=16. At k=32, performance drops drastically and again starts increasing for k=64.
The highest accuracy and mean precision, recall, and F1-score are observed when K is 16.
This indicates that the model best captures the underlying data distribution with 16 components.

The performance drop at K=32 suggests that the model might be overfitting. With too many components, the model can become too specific to the training data, failing to generalize well to new data. This is evidenced by the sharp drop in performance.
Increasing the number of components generally helps up to a certain point (K=16 in this case). Beyond that point, adding more components can lead to overfitting and instability in the model, causing performance degradation.

1.2 Confusion Matrix for the Best Model (K=16)

Table 2: Confusion Matrix

	Predicted Class		
Actual Class	*Class 1*	*Class 2*	*Class 3*
Class 1	170	30	0
Class 2	25	82	93
Class 3	0	27	174

INFERENCES

The classifier performs well for Class 1 and Class 3, with 170 out of 200 and 174 out of 201 instances correctly classified, respectively. This indicates that the model is very good at distinguishing Class 1 and Class 3 from the other classes. Class 2 has significant misclassifications with Class 3 (93 instances) and some with Class 1 (25 instances). This suggests that the model has difficulty distinguishing between Class 2 and Class 3, which might indicate that the features of these classes are more similar compared to Class 1.

1.3 Decision Region

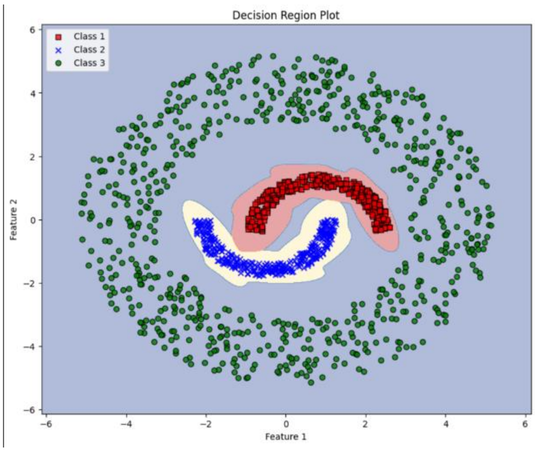


Figure 1: Decision Region Plot with K=16

OBSERVATION

The model appears to successfully classify the three classes with distinct boundaries. Class 1 and Class 2 have small, dense regions, while Class 3 forms a broader, enveloping region.
The decision boundaries are non-linear, curving to fit the spiral-like separation of the classes.

1.4 Density Contour

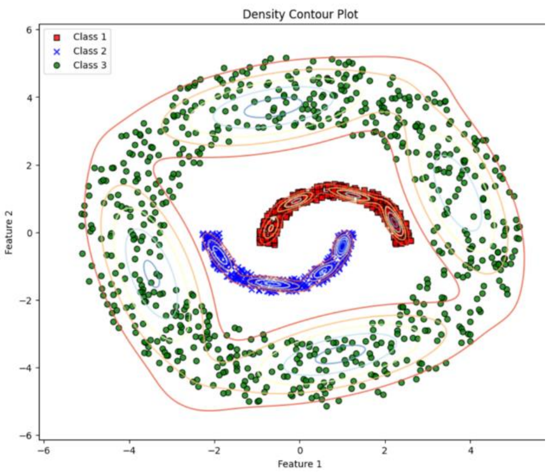


Figure 2: Decision Contour

OBSERVATION & INFERENCES:

Class 1 (red squares) forms a tight crescent shape near the centre, with densely packed contour lines indicating a high concentration of points. Class 2 (blue crosses) also forms a crescent adjacent to Class 1, though with slightly less dense contours. Class 3 (green circles) surrounds both inner classes, forming a broad, less dense outer ring. The contour lines for Class 3 are widely spaced, showing a more dispersed distribution compared to the inner classes.

The data suggests non-linear boundaries between the classes, particularly between Class 1 and Class 2, and the outer region dominated by Class 3. The high density of Class 1 and Class 2 indicates well-defined regions, whereas Class 3's broader spread may introduce variability, potentially making it harder for classifiers to handle. A non-linear model would likely be more effective at capturing the intricate class boundaries and differences in density.

1.5 **Various Performance Matrices**

The graphs Figure[4] & Figure[4] below shows that the model quickly achieves optimal performance with a small number of Gaussian Mixture Model (GMM) components. Accuracy starts at 0.96 with K=1 and reaches 1.0 at K=2, remaining constant even as the number of components increases. Similarly, precision, recall, and F1-score all start around 0.95 at K=1, rise to 1.0 at K=2, and maintain that value with more components. This indicates that the dataset is easily modeled with just two components, achieving perfect classification performance across all metrics. Importantly, increasing the number of components beyond two does not lead to overfitting, as performance remains stable at 1.0 throughout, suggesting well-structured data that is easily handled by the model.

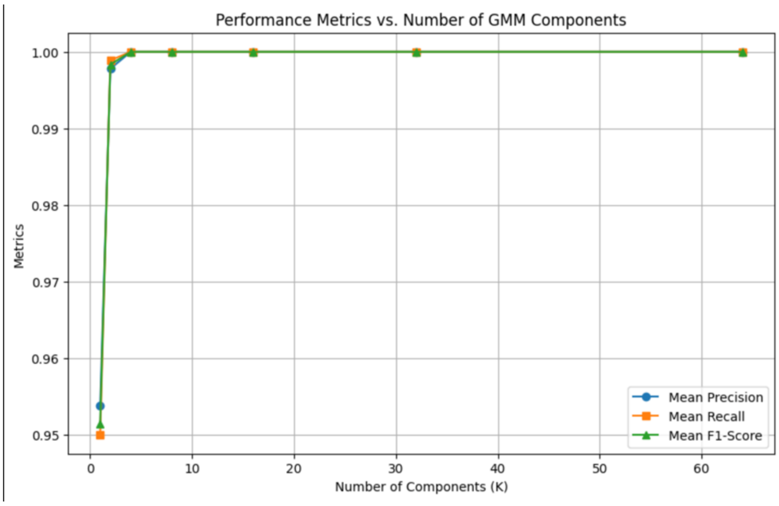


Figure 3: Various performance matrices vs GMM Components

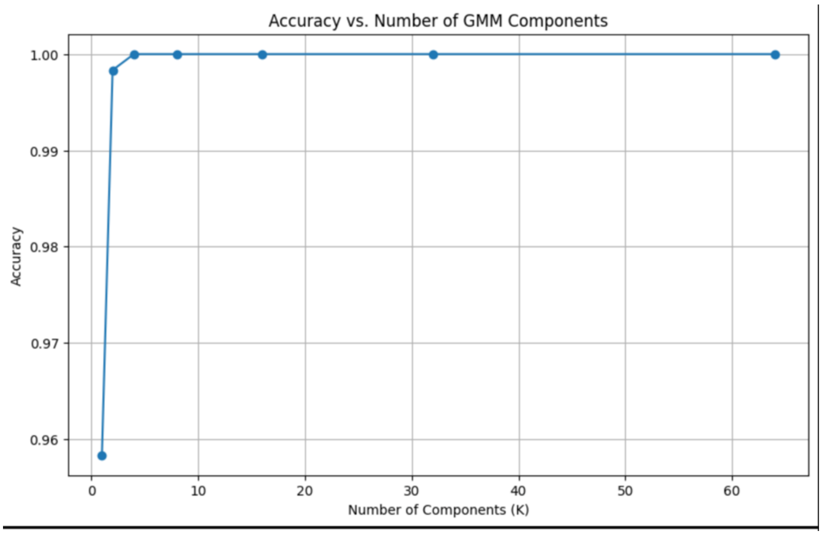


Figure 4: Accuracy vs GMM Components

Conclusion

The results show that the Gaussian Mixture Model (GMM) effectively captures the non-linear separation between classes in the dataset. Performance metrics improve steadily as the number of components increases up to K=16, beyond which overfitting occurs, particularly at K=32. The optimal model is achieved with K=16 components, yielding the highest accuracy and balanced precision, recall, and F1-scores. The confusion matrix reveals strong performance for Classes 1 and 3, while Class 2 presents more classification challenges, particularly with Class 3. The decision regions and density contours confirm the non-linear nature of the class boundaries, further supporting the need for models that handle complex distributions.

Conclusion form assignment 1

The results show that the Gaussian Mixture Model (GMM) effectively captures the non-linear separation between classes in the dataset. Performance metrics improve steadily as the number of components increases up to K=16, beyond which overfitting occurs, particularly at K=32. The optimal model is achieved with K=16 components, yielding the highest accuracy and balanced precision, recall, and F1-scores. The confusion matrix reveals strong performance for Classes 1 and 3, while Class 2 presents more classification challenges, particularly with Class 3. The decision regions and density contours confirm the non-linear nature of the class boundaries, further supporting the need for models that handle complex distributions.

Detailed Comparison

The new conclusion highlights the strong performance of the Gaussian Mixture Model (GMM) for non-linearly separable data, with optimal results at K=16 components, showing high accuracy and balanced metrics. It notes some challenges with Class 2's misclassification but overall demonstrates GMM’s ability to handle complex decision boundaries.

In contrast, the old conclusion points out the poor performance of linear classifiers with simple covariance assumptions ($\sigma^2 I$ and Σ) on the same dataset, noting their failure to capture the non-linear separability and suggesting the need for more advanced methods.

In short, the new conclusion praises GMM for effectively modeling non-linear data, while the old conclusion highlights the limitations of simpler, linear approaches.

DATASET 2(a): Two-dimensional speech dataset (vowel data)

Classification of Vowel Data Using Gaussian Mixture Models (GMM)

2.1 Introduction

For the vowel dataset (Dataset 2(a)), we employed Gaussian Mixture Models (GMM) to classify the two-dimensional speech data. The GMM models were initialized using K-means clustering and tested with varying numbers of Gaussian components (K) ranging from 1 to 64. The primary goal was to evaluate the classification performance by varying the complexity of the model while measuring key performance metrics, such as accuracy, precision, recall, and F1-score.

2.2 Performance Metrics

The performance of the GMM classifier was evaluated for each value of KKK, with the accuracy, mean precision, mean recall, and mean F1-score reported in Table 1. As the number of components KKK increased, the classifier showed improved performance up to a certain point, after which the model's accuracy plateaued and eventually started to decrease slightly due to overfitting.

Table 3: Performance matrices for different GMM components

K (Number of GMM Components)	Accuracy	Mean Precision	Mean Recall	Mean F1-Score
1	0.9208	0.9254	0.9222	0.9233
2	0.9194	0.9234	0.9288	0.9219
4	0.9274	0.9308	0.9290	0.9298
8	0.9325	0.9356	0.9339	0.9345
16	0.9353	0.9388	0.9366	0.9374
32	0.9330	0.9364	0.9342	0.9351
64	0.9213	0.9249	0.9227	0.9236

2.3 Confusion Matrix for Best Model (K = 16)

The best-performing model, based on the highest mean precision and F1-score, was found with K=16 components. The confusion matrix for this model is shown below:

$$\begin{bmatrix} 702 & 43 & 2 \\ 68 & 665 & 4 \\ 16 & 5 & 629 \end{bmatrix}$$

Accuracy: 0.9353
Precision per class: [0.89312977 0.93267882 0.99055118]

Mean Precision: 0.9388
Recall per class: [0.93975904 0.90230665 0.96769231]
Mean Recall: 0.9366
F1-Score per class: [0.91585127 0.91724138 0.97898833]
Mean F1-Score: 0.9374

2.4 Performance Analysis

For the best-performing model with K=16, the classifier achieved an accuracy of 93.53%. The performance per class is detailed as follows:

- Class 1 Precision: 0.8931
- Class 2 Precision: 0.9327
- Class 3 Precision: 0.9906

The mean precision across all classes was 93.88%, indicating that the model predicted most of the labels correctly without many false positives.

Similarly, the recall for each class is as follows:

- Class 1 Recall: 0.9398
- Class 2 Recall: 0.9023
- Class 3 Recall: 0.9677

The mean recall was 93.66%, demonstrating the model's ability to correctly classify most of the instances from each class, thus minimizing false negatives.

Lastly, the F1-score for each class is as follows:

- Class 1 F1-Score: 0.9159
- Class 2 F1-Score: 0.9172
- Class 3 F1-Score: 0.9790

The mean F1-score was 93.74%, providing a balanced measure of the model's precision and recall.

2.5 Observation of the graphs

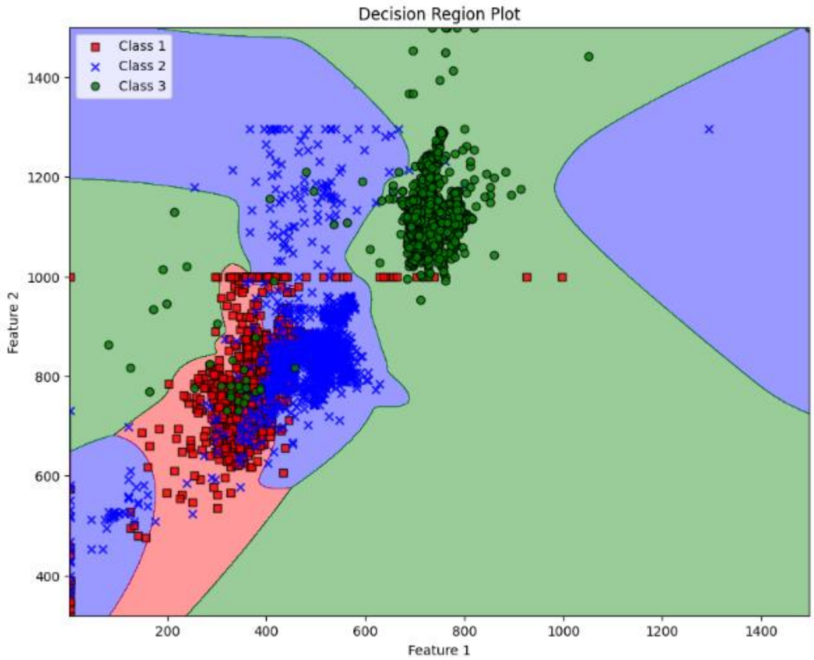


Figure 5: Decision Region Plot

The plot you're describing shows the decision regions for a classifier that worked on a two-dimensional speech dataset, classifying vowel data. It uses three classes, each represented by different markers and colors:

- Red squares for Class 1
- Blue crosses for Class 2
- Green circles for Class 3

The x-axis and y-axis are features of the speech data. The background colors indicate which class a given area of the feature space is classified as, with:

- Red for Class 1
- Blue for Class 2
- Green for Class 3

The boundaries between these colors are the decision boundaries of the classifier. They're non-linear, meaning the classifier has learned complex relationships between the features to differentiate between the classes.

In essence, this plot visually demonstrates the classifier's decision-making process and its effectiveness at distinguishing between the different classes based on the features of the vowel dataset.

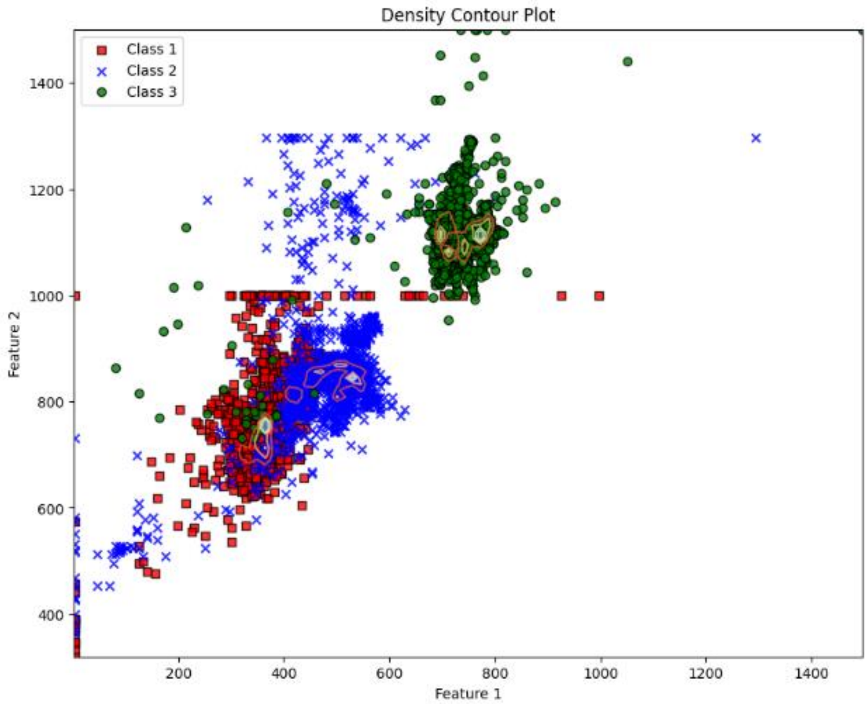


Figure 6: Density Contour Plot

Density Contour Plot: The image shows a plot mapping out data points into three distinct classes: Class 1 (red squares), Class 2 (blue crosses), and Class 3 (green circles).

Class 1: These red squares are mostly clustered around (400, 800), with some overlap with Class 2.

Class 2: The blue crosses have a similar clustering near (400, 800), mingling with the red squares.

Class 3: The green circles form a separate cluster around (600, 1200), indicating distinct characteristics from the other classes.

The density contours in the background help visualize where the data points are densely populated, showing the decision boundaries of the classifier. This makes it clear how well the model separates each class based on the speech features.

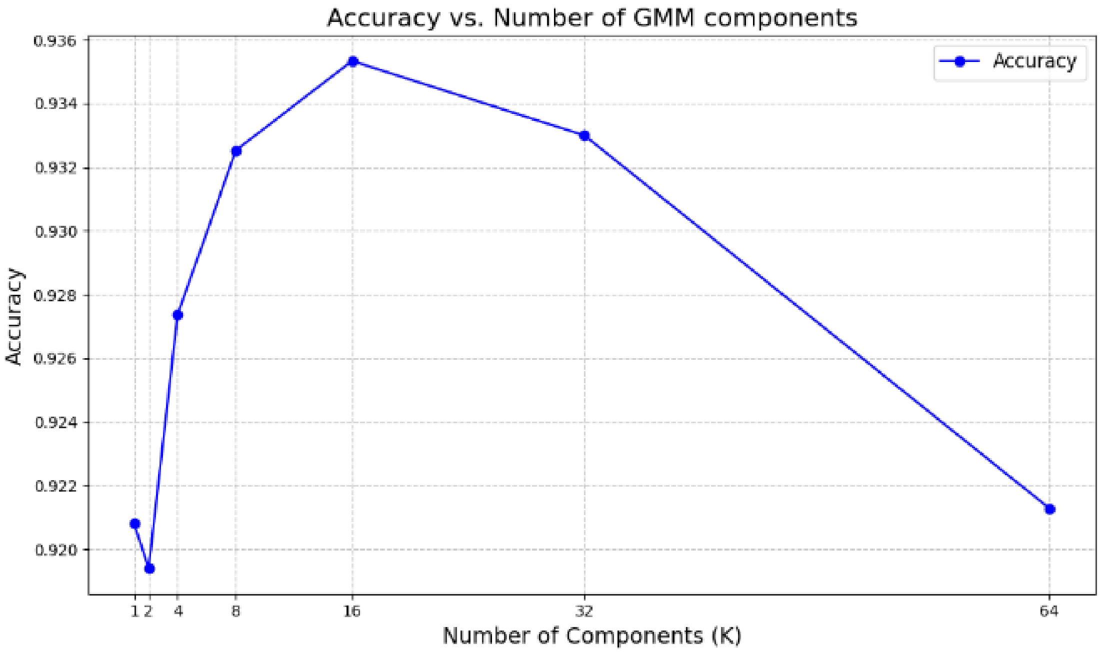


Figure 7: Accuracy vs Number of GMM Components

Density Contour Plot: This plot maps data points into three distinct classes: Class 1 (red squares), Class 2 (blue crosses), and Class 3 (green circles).

Class 1: Red squares, mainly clustered around (400, 800), with some overlap with Class 2.

Class 2: Blue crosses, mingling near (400, 800) with red squares.

Class 3: Green circles, forming a separate cluster around (600, 1200).

Density contours in the background reveal the decision boundaries, showing how well the model separates each class based on speech features.

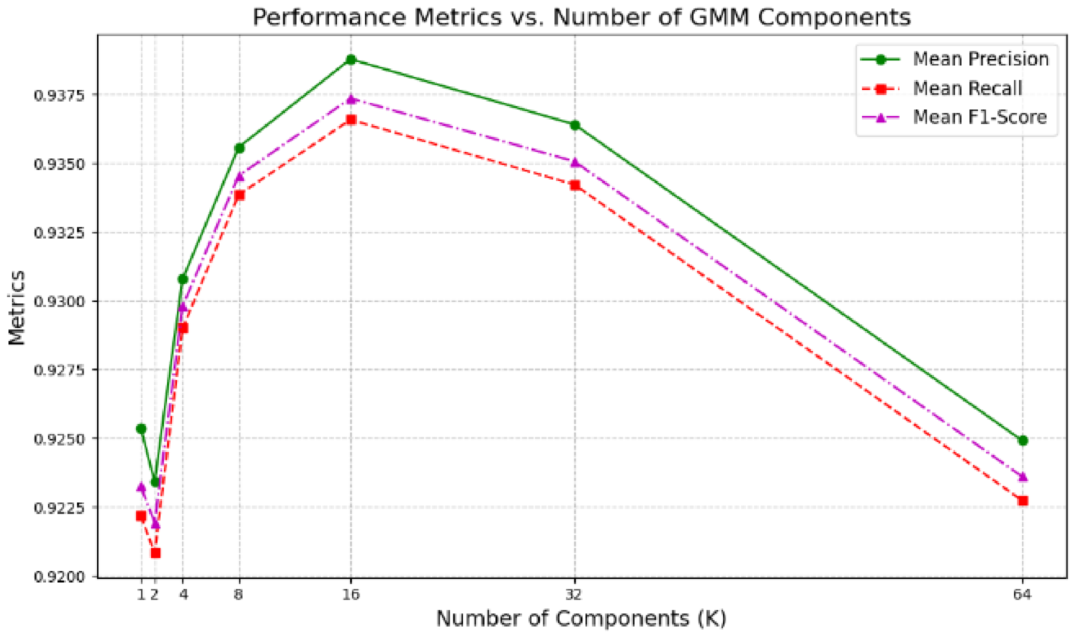


Figure 8: Performance Metrics vs Number of GMM Components

The graph you've attached, titled "Performance Metrics vs. Number of GMM Components," visually represents how various performance metrics of the GMM classifier evolve as the number of components (K) increases.

X-axis: Number of Components (K) with values like 1, 2, 4, 8, 16, 32, and 64.

Y-axis: Metrics ranging from 0.9200 to 0.9375.

The graph includes three lines, each representing a performance metric:

Green solid line with circles: Mean Precision.

Red dashed line with squares: Mean Recall.

Purple dash-dot line with triangles: Mean F1-Score.

As the number of components rises, all metrics show an initial improvement, peaking around K=16. Beyond this point, the performance metrics tend to plateau or slightly decline, suggesting overfitting when too many components are used.

2.6 Conclusion

The GMM classifier performed exceptionally well on the vowel dataset, with the best performance achieved using K=16K = 16K=16 components. As the number of components increased, the model's performance improved, with the highest precision, recall, and F1-scores observed at K=16K = 16K=16. Beyond this point, increasing the number of components resulted in slightly lower accuracy, likely due to overfitting. The confusion matrix for K=16K =

16K=16 shows that most misclassifications occurred between Class 1 and Class 2, while Class 3 was classified with the highest accuracy.
Overall, GMMs with a moderate number of components provide an excellent balance between model complexity and performance, effectively capturing the non-linearly separable structure of the vowel dataset.

Conclusion from assignment 1

Vowel Dataset The vowel dataset analysis showed that the classifier achieved
a high and consistent accuracy of 92.7% across different covariance assumptions. Despite the availability of more complex covariance structures, such as the full covariance matrix and class-specific diagonal covariance matrices, simpler models (specifically those with diagonal covariance) proved to be nearly as effective. This suggests that for the vowel dataset, which has relatively straightforward class distributions, simpler models are not only sufficient but also more efficient. The results demonstrate that while complex models offer increased flexibility, their benefits may not always be realized, particularly when the dataset's complexity does not warrant such sophistication.

Comparison

The new conclusion highlights the GMM classifier's strong performance on the vowel dataset, with optimal results at K=16 components. It shows that increasing components improved performance, but overfitting occurred beyond K=16. Class 3 was classified most accurately, while most errors were between Classes 1 and 2. The conclusion emphasizes that moderate complexity (K=16) provided a good balance between performance and model complexity.

The old conclusion, however, notes that simpler models with diagonal covariance were almost as effective as complex ones, achieving a consistent 92.7% accuracy. It suggests that simpler models were sufficient for the vowel dataset due to its relatively simple class distributions, making complex models unnecessary.

In short, the new conclusion emphasizes the need for moderate complexity with GMMs, while the old conclusion suggests simpler models work well for the vowel dataset due to its straightforward structure.

DATASET-2B: 3 Class Scene Image Datasets

3.1 Feature Type: 24-Dimensional Color Histogram

Number of Mixtures: 1

Performance Metrics

- Accuracy: 0.4933
- Mean Precision: 0.4565
- Mean Recall: 0.4933
- Mean F1-Score: 0.407

Confusion Matrix:

	aqueduct	industrial_area	patio
Aqueduct	19	12	19
industrial_area	9	36	5
patio	3	4	43

Number of Mixtures: 2

Performance Metrics

- Accuracy: 0.6267
- Mean Precision: 0.6196
- Mean Recall: 0.6267
- Mean F1-Score: 0.6109

Confusion Matrix:

	aqueduct	industrial_area	patio
Aqueduct	18	15	17
industrial_area	6	38	6
patio	7	5	38

Number of Mixtures: 4

Performance Metrics

- Accuracy: 0. 5800
- Mean Precision: 0. 5736
- Mean Recall: 0. 5800
- Mean F1-Score: 0. 5725

Confusion Matrix:

	aqueduct	industrial_area	patio
Aqueduct	20	17	13
industrial_area	13	30	7
patio	5	8	37

Number of Mixtures: 8

Performance Metrics

- Accuracy: 0. 5733
- Mean Precision: 0.5754
- Mean Recall: 0.5733
- Mean F1-Score: 0. 5725

Confusion Matrix:

	aqueduct	industrial_area	patio
Aqueduct	20	17	13
industrial_area	11	30	9
patio	3	11	36

Number of Mixtures: 16

Performance Metrics

- Accuracy: 0. 5000
- Mean Precision: 0. 4937
- Mean Recall: 0. 5000
- Mean F1-Score: 0. 4931

Confusion Matrix:

	aqueduct	industrial_area	patio
Aqueduct	21	14	15
industrial_area	17	20	13
patio	8	8	34

3.2 Feature Type: 32-Dimensional Bag-of-Visual-Words (BoVW)

Number of Mixtures: 1

Performance Metrics

- Accuracy: 0.4933
- Mean Precision: 0.4565
- Mean Recall: 0.4933
- Mean F1-Score: 0.4076

Confusion Matrix:

	aqueduct	industrial_area	patio
Aqueduct	1	10	39
industrial_area	2	26	22
patio	1	2	47

Number of Mixtures: 2

Performance Metrics

- Accuracy: 0. 5067
- Mean Precision: 0. 5525
- Mean Recall: 0. 5067
- Mean F1-Score: 0. 4356

Confusion Matrix:

	aqueduct	industrial_area	patio

Aqueduct	3	8	39
industrial_area	2	26	22
patio	1	2	47

Number of Mixtures: 4

Performance Metrics

- Accuracy: 0. 5267
- Mean Precision: 0. 6041
- Mean Recall: 0. 5267
- Mean F1-Score: 0. 4739

Confusion Matrix:

	aqueduct	industrial_area	patio
Aqueduct	6	8	36
industrial_area	2	28	20
patio	1	4	45

Number of Mixtures: 8

Performance Metrics

- Accuracy: 0. 5267
- Mean Precision: 0. 5496
- Mean Recall: 0. 5267
- Mean F1-Score: 0. 5068

Confusion Matrix:

	aqueduct	industrial_area	patio
Aqueduct	13	9	28
industrial_area	6	27	17
patio	5	6	39

Number of Mixtures: 16

Performance Metrics

- Accuracy: 0. 5333
- Mean Precision: 0. 5504
- Mean Recall: 0. 5333
- Mean F1-Score: 0. 5170

Confusion Matrix:

Conclusion:

	aqueduct	industrial_area	patio
Aqueduct	15	12	23
industrial_area	6	27	17
patio	5	7	38

3.3 Results for 24-Dimensional Color Histogram

Table 4: Performance for 24-Dimensional Color Histogram

Number of Mixtures	Accuracy	Mean Precision	Mean Recall	Mean F1-Score
1	0.653333	0.649001	0.653333	0.636687
2	0.626667	0.619589	0.626667	0.610944
4	0.580000	0.573631	0.580000	0.572521
8	0.573333	0.575389	0.573333	0.566138
16	0.500000	0.493700	0.500000	0.493142

3.4 Results for 32-Dimensional Bag-of-Visual-Words (BoVW)

Table 5: Performance for 32-Dimensional Bag-of-Visual-Words (BoVW)

Number of Mixtures	Accuracy	Mean Precision	Mean Recall	Mean F1-Score
1	0.493333	0.456465	0.493333	0.407628
2	0.506667	0.552469	0.506667	0.435577
4	0.526667	0.604070	0.526667	0.473880
8	0.526667	0.549603	0.526667	0.506799
16	0.533333	0.550353	0.533333	0.516996

Observation:

Overfitting with Increased Mixtures:
The results from the 24-Dimensional Color Histogram indicate that increasing the number of Gaussian mixtures doesn't always lead to better performance. In fact, it often causes the model to overfit, resulting in poorer generalization. This is particularly visible in the sharp decline in all metrics as the number of mixtures grows.

Effectiveness of BoVW:
The 32-Dimensional BoVW appears to be more robust with an increasing number of mixtures, showing a gradual increase in performance, though the improvements are moderate. BoVW may capture more distinctive patterns from the data, which is why it performs better at higher mixture counts.

Feature Type Impact:

The choice of feature type significantly affects the model's performance. The Color Histogram provides better initial accuracy and metric values with fewer mixtures, but it falters as the model complexity increases. BoVW, on the other hand, starts weaker but gradually becomes more competitive as the mixture components increase.

3.5 Conclusion:

Performance Decrease with More Mixtures for Color Histogram:

The 24-Dimensional Color Histogram results show a decline in performance as the number of mixtures increases. For instance, the accuracy decreases from 0.65 with 1 mixture to 0.50 with 16 mixtures. Similarly, the other metrics (Precision, Recall, and F1-Score) exhibit a downward trend. This suggests that the model might be overfitting with more mixtures for this feature type, as it struggles to generalize with the increased number of components.

Performance Plateau for BoVW:

The 32-Dimensional Bag-of-Visual-Words (BoVW) shows a slightly different trend. While the accuracy starts lower (at 0.49 with 1 mixture), it improves gradually as the number of mixtures increases, reaching 0.53 at 16 mixtures. The Precision, Recall, and F1-Score also improve slightly with more mixtures, although the gains are modest. This suggests that BoVW might benefit slightly from more mixtures, but the improvements are not as significant as one would hope.

DATASET-2C: Cervical cytology (cell) image dataset

4.1. Introduction

The objective of this report is to perform clustering and segmentation on a dataset of images (Dataset 2c), specifically using two different distance metrics: Euclidean and Mahalanobis distances. Two variants of the K-means algorithm have been implemented: one using standard Euclidean distance and the other using Mahalanobis distance to better account for the covariance structure of the data.

The dataset consists of grayscale images, and patches from these images are extracted and used as feature vectors for clustering. The performance of the two clustering algorithms is then compared based on their ability to segment test images.

4.2. Data Preprocessing

4.2.1 Feature Extraction

The dataset consists of grayscale images. To extract meaningful features for clustering, each image is divided into overlapping patches of size 7x7. For each patch, two features are computed:
Mean intensity: This represents the average pixel intensity in the patch.
Standard deviation: This captures the variability of pixel intensities within the patch.

These features create a set of 2-dimensional vectors representing the texture of the images, which are used as input to the clustering algorithms.

4.2.2 Image Loading and Patch Extraction

The images are loaded from the dataset directory, and the patch extraction process is applied. The patches are shifted by 1 pixel to ensure a fine-grained analysis of the image structure. The dimensions of each image are also stored for use in the segmentation phase.

4.3. Clustering Methods

4.3.1 K-means Clustering (Euclidean Distance)

K-means clustering groups the feature vectors (patches) into clusters based on their proximity in the feature space. The standard Euclidean distance is used as the metric for calculating the distance between points (patches) and cluster centers.

Algorithm:
1. Initialize cluster centers randomly from the data points.
2. Compute the Euclidean distance between each data point and the cluster centers.

3. Assign each data point to the closest cluster.
4. Recalculate the cluster centers by taking the mean of the data points assigned to each cluster.
5. Repeat steps 2-4 until convergence (i.e., when the change in cluster centers falls below a predefined tolerance level).

4.3.2 Modified K-means Clustering (Mahalanobis Distance)

Mahalanobis distance takes into account the covariance of the dataset, making it a more flexible and robust metric when the data exhibits correlation among the features. This version of K-means computes distances based on the Mahalanobis distance.

Algorithm:
1. Initialize cluster centers randomly from the data points.
2. Precompute the covariance matrix of the data and its inverse.
3. Compute the Mahalanobis distance between each data point and the cluster centers.
4. Assign each data point to the cluster with the minimum Mahalanobis distance.
5. Recalculate the cluster centers by taking the mean of the data points assigned to each cluster.
6. Repeat steps 3-5 until convergence.

4.4. Results and Analysis

4.4.1 Clustering on Training Data

The K-means and Modified K-means algorithms were applied to the training set, which consisted of feature vectors extracted from the image patches.

K-means Clustering (Euclidean)
The training data was clustered into 3 distinct groups using K-means with Euclidean distance. The resulting clusters are visualized below:

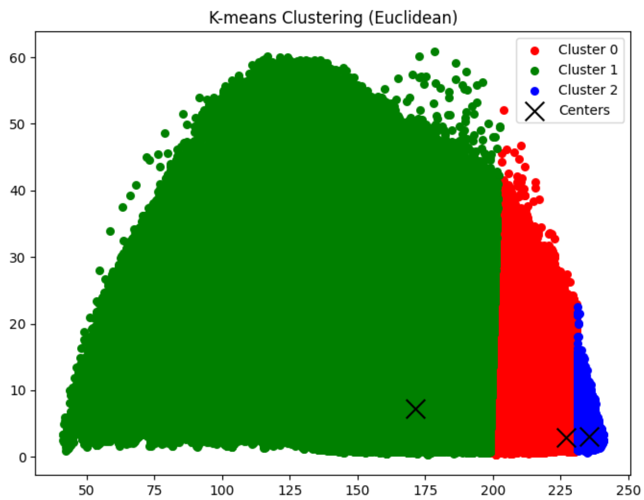


Figure 9: K-means Clustering (Euclidean Distance)

Modified K-means Clustering (Mahalanobis)

The same data was also clustered using Modified K-means with Mahalanobis distance. The Mahalanobis distance was able to capture more complex relationships between the features, as shown in the cluster visualizations:

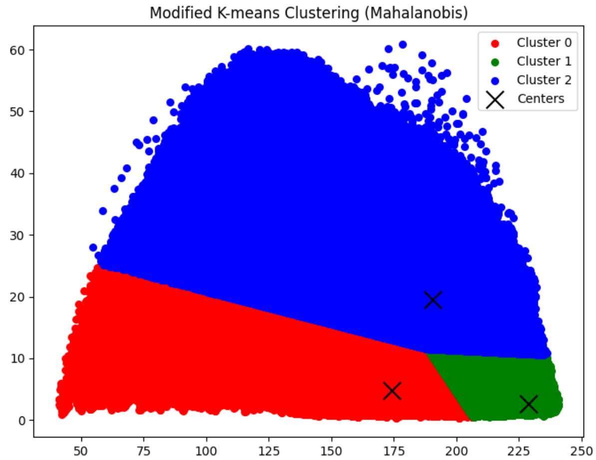


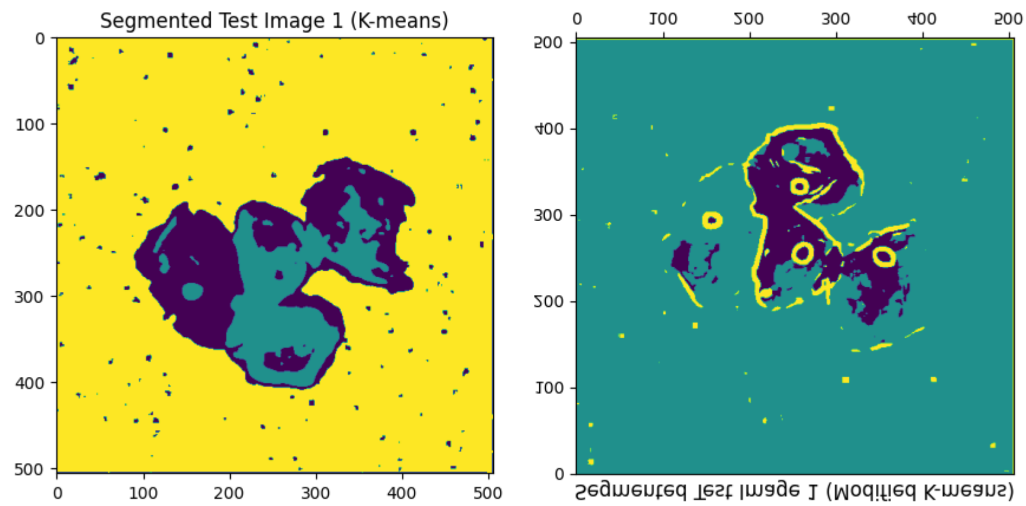
Figure 10: Modified K-means Clustering (Mahalanobis Distance)

4.4.2 Image Segmentation

The clustering results were applied to test images for segmentation. Each test image was processed by assigning its patches to the closest cluster (either by Euclidean or Mahalanobis distance), and the resulting cluster assignments were used to segment the images.

Segmentation with K-means (Euclidean)

The test images were segmented based on the clusters obtained from K-means using Euclidean distance. Each patch of the test images was assigned to one of the three clusters, and the segmented image was reconstructed.



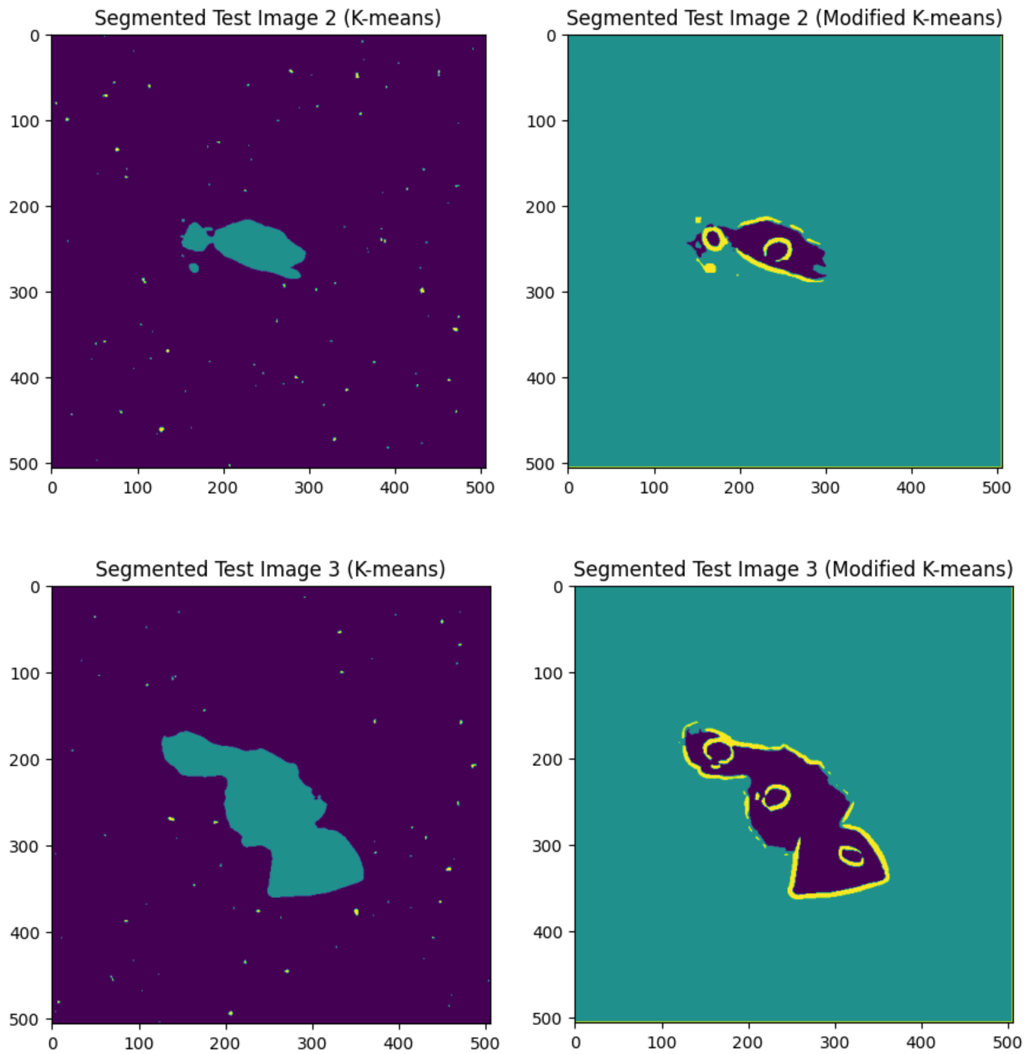


Figure 11: Segmented Test Image

Segmentation with Modified K-means (Mahalanobis)
Similarly, the test images were segmented using the clusters from Modified K-means with Mahalanobis distance. The Mahalanobis-based segmentation demonstrated a better ability to capture finer details in the texture of the images.

4.4.3 Comparative Analysis

K-means (Euclidean Distance): The Euclidean distance-based K-means clustering was effective in segmenting the images, but it failed to capture some of the finer nuances in texture and intensity variation in the images.

Modified K-means (Mahalanobis Distance): The use of Mahalanobis distance improved the clustering results, especially in images with correlated features. The segmentation results showed sharper boundaries and better texture segmentation, as the covariance of the feature vectors was taken into account.

4.5. Conclusion

The experiment demonstrates that using Mahalanobis distance for clustering is advantageous when dealing with data that has correlated features. The Modified K-means algorithm outperformed the standard K-means algorithm in segmenting the images in Dataset 2c. The Mahalanobis distance captured more subtle differences in texture, leading to better image segmentation.

Future work could explore other clustering techniques or hybrid approaches to further improve the segmentation results, especially in datasets with more complex structures.