

Assignment#2 Report

Question 1:

It took 17.2sec with N=64

```
original_image_path = 'Q1 Dataset/Mona-lisa.png'
stroke_image_paths = ['Q1 Dataset/Mona-lisa stroke 1.png', 'Q1 Dataset/Mona-lisa stroke 2.png']
process_image(original_image_path, stroke_image_paths)

#result with N=64
✓ 17.2s
```

The screenshot shows a Jupyter Notebook cell with Python code. The code imports the image, defines stroke paths, and processes the image with N=64. It also prints the result and the execution time (17.2s). Below the code are two images: the original Mona Lisa painting and its segmented version, which appears as a blue and white binary mask.

It took 16secs with N=32

```
original_image_path = 'Q1 Dataset/Mona-lisa.png'
stroke_image_paths = ['Q1 Dataset/Mona-lisa stroke 1.png', 'Q1 Dataset/Mona-lisa stroke 2.png']
process_image(original_image_path, stroke_image_paths)

#result with N=32
✓ 16.0s
```

The screenshot shows a Jupyter Notebook cell with Python code. The code imports the image, defines stroke paths, and processes the image with N=32. It also prints the result and the execution time (16.0s). Below the code are two images: the original Mona Lisa painting and its segmented version, which appears as a blue and white binary mask.

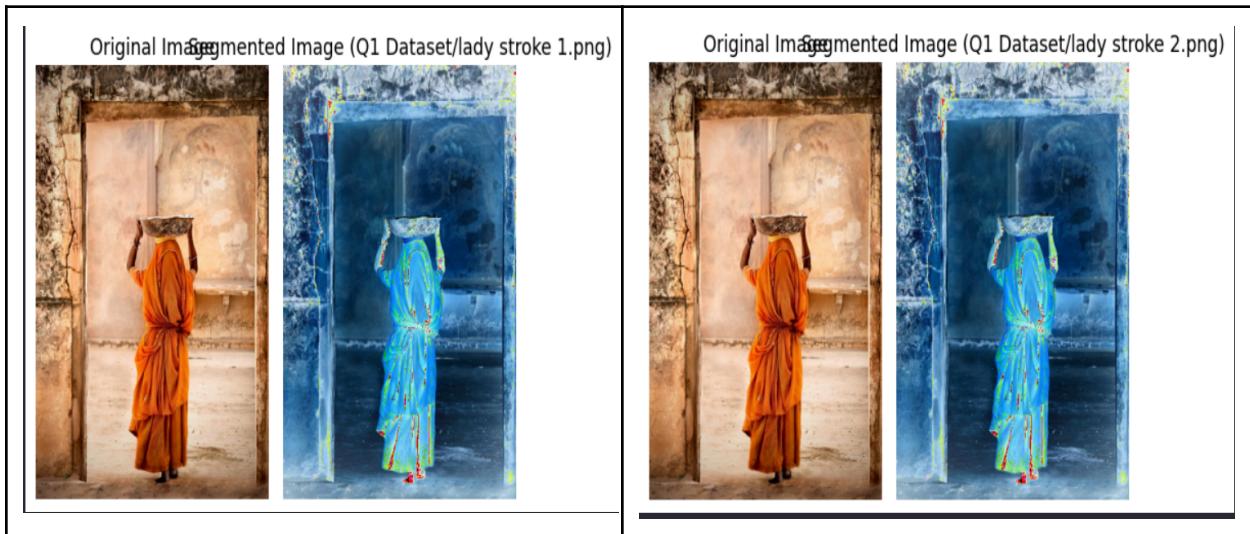
It took 14.6secs with N=16

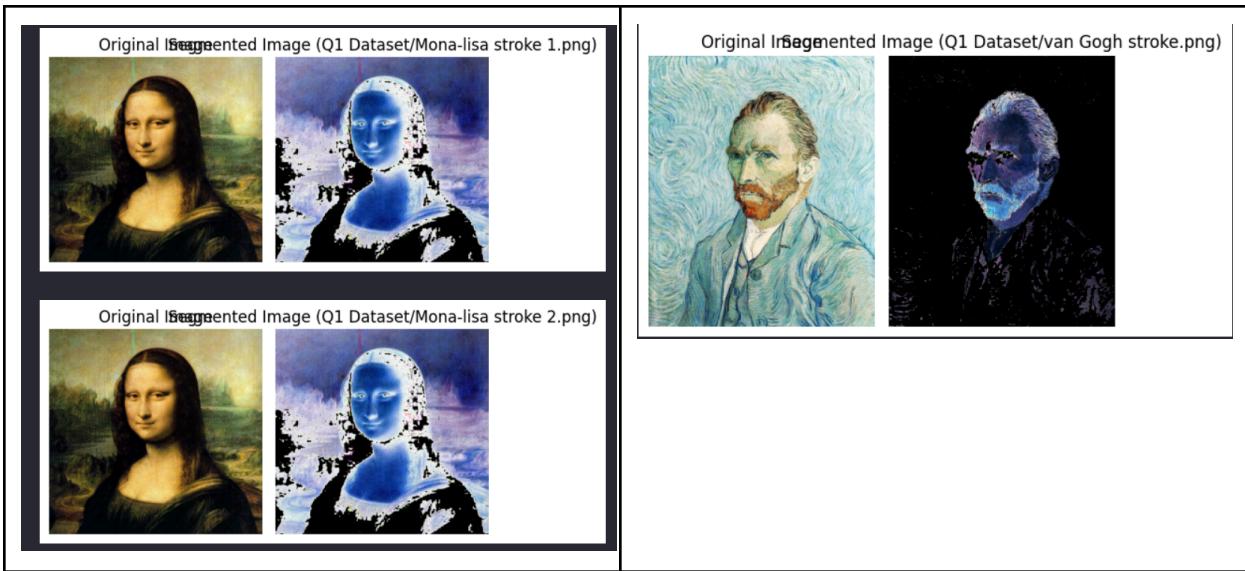
```
original_image_path = 'Q1 Dataset/Mona-lisa.png'  
stroke_image_paths = ['Q1 Dataset/Mona-lisa stroke 1.png', 'Q1 Dataset/Mona-lisa stroke 2.png']  
process_image(original_image_path, stroke_image_paths)  
  
#result with N=16  
14.6s
```

Original Image Segmented Image (Q1 Dataset/Mona-lisa stroke 1.png)



We can conclude that by increasing the number of N the time taken by Kmean cluster increases
Results of this questions :





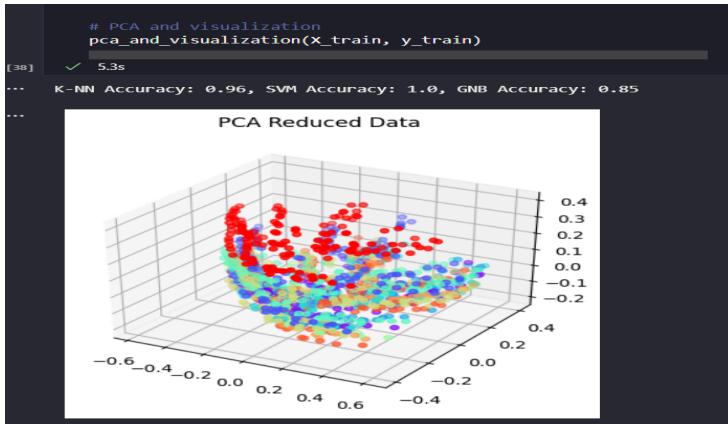
Question 2:

Observations :

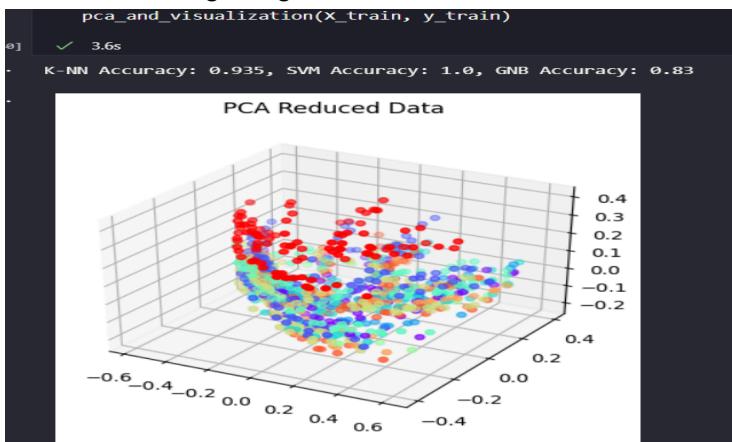
- k=2 KNN accuracy was 0.97 and it took 5.3 secs
- k=5 KNN accuracy was 0.98 and it took 5.2 secs
- k=7 KNN accuracy was 0.96 and it took 4.8 secs
- k=11 KNN accuracy was 0.97 and it took 5.3 secs

Here k values, the algorithm regularly displayed great accuracy, getting scores ranging from 0.96 to 0.98. However, there were modest variations in execution time, with the method requiring roughly 5.3 seconds for k=2 and k=11, and somewhat less time for k=5 and k=7, at around 5.2 and 4.8 seconds, respectively. While there was no significant difference in accuracy between different k values, k=5 emerged as a feasible alternative due to its balanced performance in terms of accuracy and execution time.

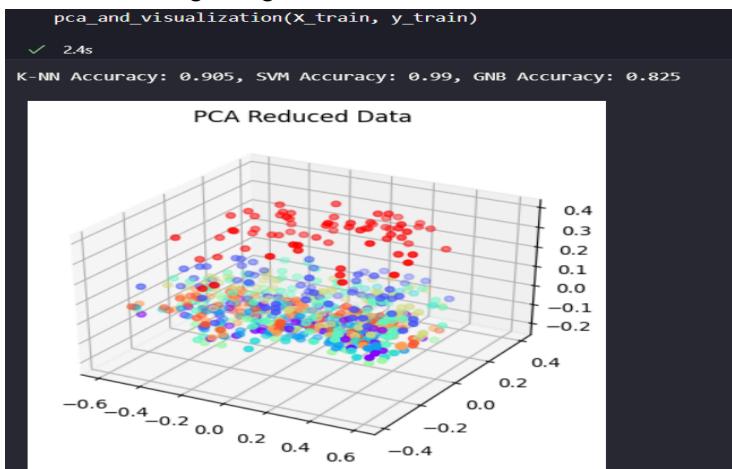
- With 150 training images the results were



- With 100 training images the results were



- With 70 training images the results were



- K-NN Accuracy: 0.96, SVM Accuracy: 1.0, GNB Accuracy: 0.85 this means that Support Vector Machines (SVM) had the greatest accuracy (1.0), followed by K-Nearest Neighbours (K-NN) at 0.96 and Gaussian Naive Bayes (GNB) at 0.85. SVM gave the best classification performance, while K-NN delivered comparable accuracy. GNB, while less accurate, may nevertheless be beneficial in cases requiring computing efficiency or when working with datasets that meet its assumptions. Finally, the choice of algorithms is determined by the individual job requirements, accuracy, computing complexity, and modelling assumptions.

K-NN accuracy: 0.96, SVM Accuracy: 1.0, GNB Accuracy: 0.85
 K-NN Accuracy: 0.96, SVM Accuracy: 1.0, GNB Accuracy: 0.85