EECE 5644: Assignment 4

Jeffrey Chen

December 3, 2024

Repository Link:
https://github.com/jeffreychen159/EECE-5644-Machine-Learning-and-Pattern-Recognition

---

1 Question 1

1.1 Data Generation

The data was generated by using the given functions and parameters. For the x values, depending on their labels, a radii value was calculated. This allowed the circle to be formed. On top of this, noise was added to this signal to cause more randomness. After this was done, the data was processed and was ready to train.

1.2 Training and Results

The code used was given in the demo but I replaced it with my data. On top of this, the rest stayed the same and it calculated the correct and incorrect values. By inputting our data and running the code, we get the following results:

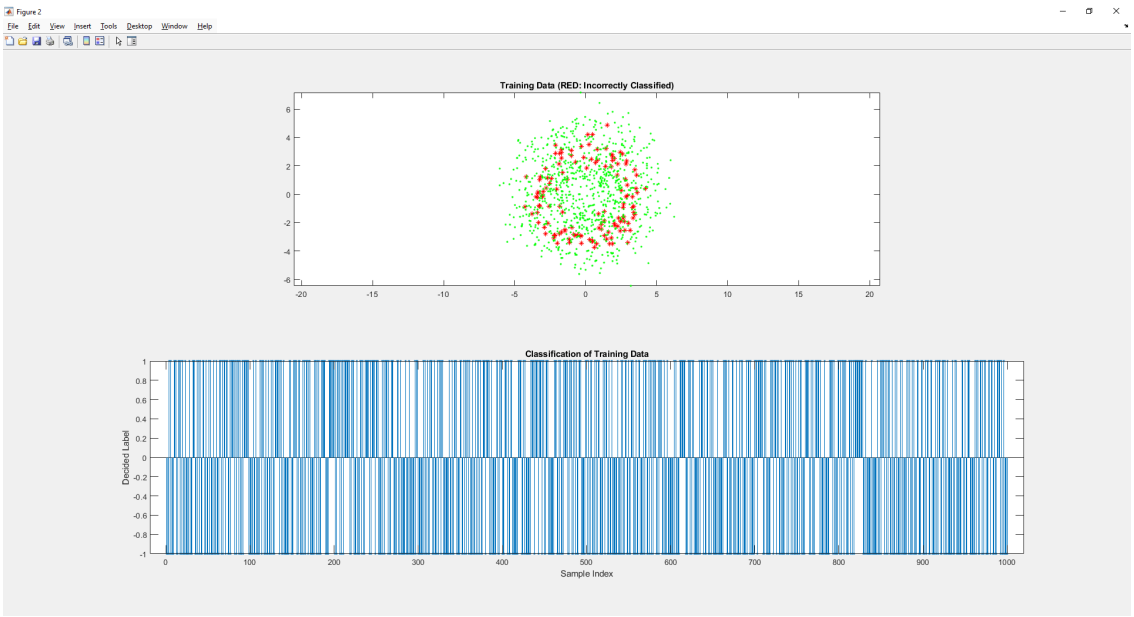

Figure 1: Training result

From the training, we can see that this forms the disks where the data is classified. There seems to be a lot of error due to the fact that a $\sigma = 1$ noise was introduced which was a lot of noise.

This data also gave a $P_{error} = 0.1310$ which can clearly be shown by the red spots in the graph.

From both of the data, we can say that in general, this model is able to predict with about an 86% accuracy for each label. With this accuracy, we can see the disks formations even though there is a lot of noise in the data.

2 Question 2

2.1 Processing Data

This is the image that was used and processed:

Figure 1: Original image being processed

To process this image, we took each pixel and we gave it 5 features. 3 features describing the color on an RGB scale and two describing the location of the pixel in x, y coordinates.

2.2 Generating Clusters

We plan to generate a total of $k = 100$ clusters. To generate this, we generated $k_n = 10$ clusters $n = 10$ times which then we combined to generate the final $k = 100$ clusters. This is the clustering result:

Figure 2: Clustering of the images

From the k clusters, we can see each cluster pick out elements of the original image. Only by combining these clusters do we get a more in-depth view of the full cluster. Below is the final result:

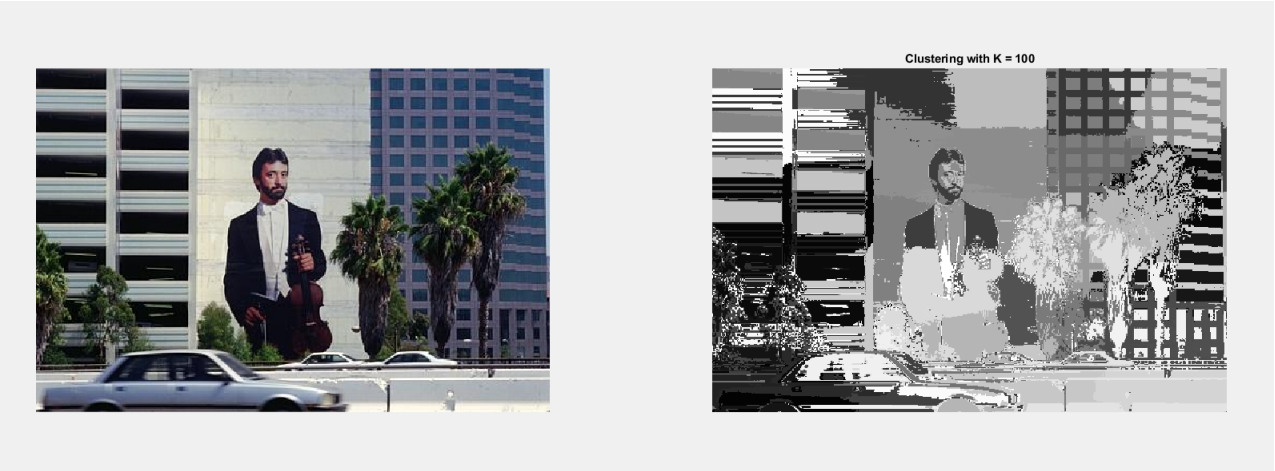

Figure 3: Final cluster of the full image

From what we can see here, the clustered images seem to align well with the original image. For solid colors, we can clearly see the clustering distinguish between them. But for the trees, it seems a little harder.

Just to test, I decided to try a $k = 1000$ cluster. This is the result:

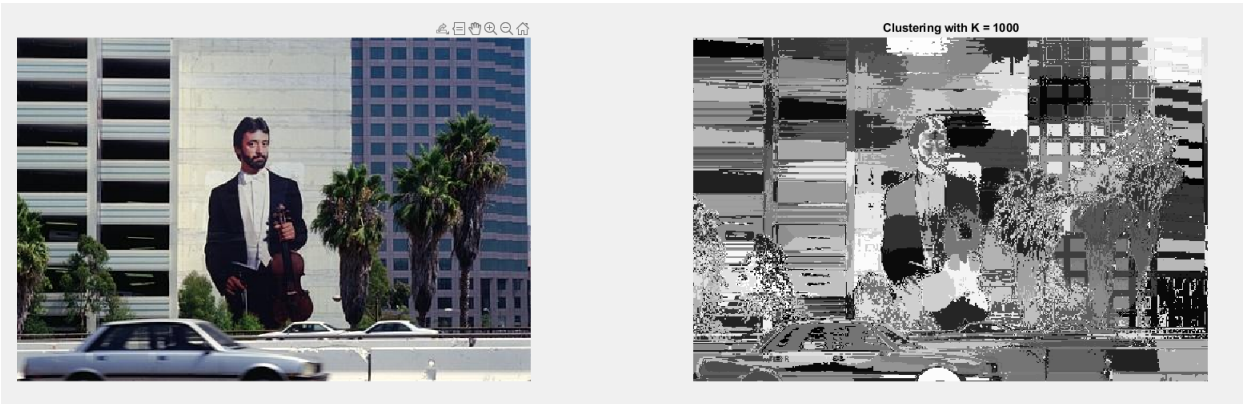

Figure 3: Testing a $k = 1000$ cluster

We can see that in this image, there is a lot more detail in the clustering. It seems to be separating out many more shades. While this works better for more detailed parts of the image like the trees, it isn't as good for the buildings as it kind of clusters colors that shouldn't be there.