Fourier Transform and Feature Selection

1 INTRODUCTION

Nowadays, with the ever growing importance of scientific visualization, image processing is gaining more and more popularity in a large number fields, medicine, genetics, and robotics being the mains ones. Image processing involves applying specific standard techniques that use mathematical operations in order to transform and manipulate images. The output of these processes could be a set of characteristics or parameters related to the image or even a new image derived from the original one.

Fourier Transform represents an image processing tool which is used to decompose an image into its sine and cosine components. The result of the decomposition represents the image in the frequency domain, the initial image being referred to as the spatial domain. Each point in the frequency domain represents a value from the spatial domain. The frequency domain is also called Fourier space.

Now we can make use of the Fourier space to search and describe certain geometric characteristics of the input image. This is really useful when we want, for example, to perform feature selection on an image as it is far easier for a computer to process information from the Fourier space than it is from the image itself. In this report, I will present how I made use of the frequency domain in order to perform feature selection for 3 characters: ‘S’, ‘V’ and ‘T’ and how I have used the features to create decision boundaries for future data.

2 APPROACH TO ANALYSIS IN THE FOURIER DOMAIN

We are given the initial input which is constituted of thirty images: ten for the letter ‘S’, ten for the letter ‘V’ and ten for the letter ‘T’. The images contain slight changes in the position and size of the letters, changes that influence the frequencies in the Fourier space. We apply the fourier transform to the given images which will decompose them into sinusoidal components, giving us the frequency domain. It is a common practice to shift the Fourier space such that the image mean – F(0,0) - is in the centre of the image. After these operations, the results will represent the magnitude as viewed in Figure 1. The magnitude encodes the brightness from the spatial domain. All the magnitudes will initially look as in Figure 1 as the dynamic range of the Fourier coefficients is too large to be displayed. Applying a logarithmic transofrmation to the magnitude spectrum helps bringing out the details of the Fourier transform regions where F(x,y) is close to 0, thus being able to analyze the frequencies.

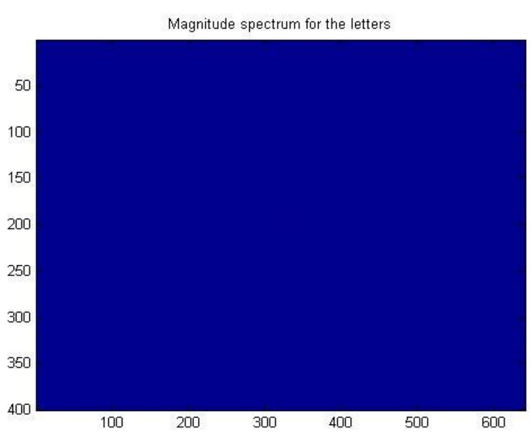
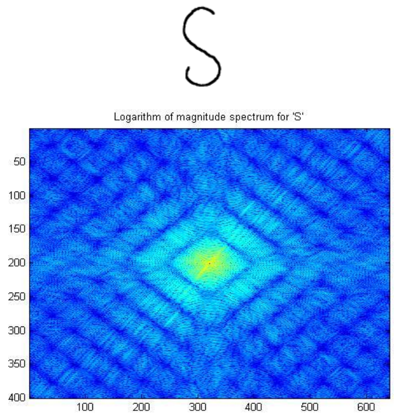
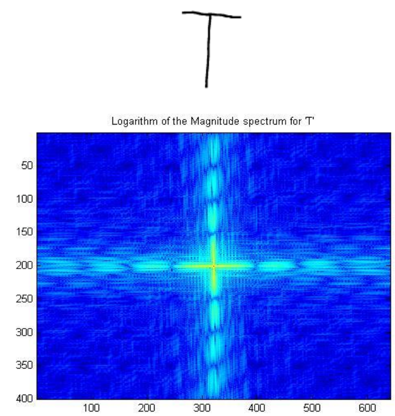


Figure 1.(Magnitude spectrum)





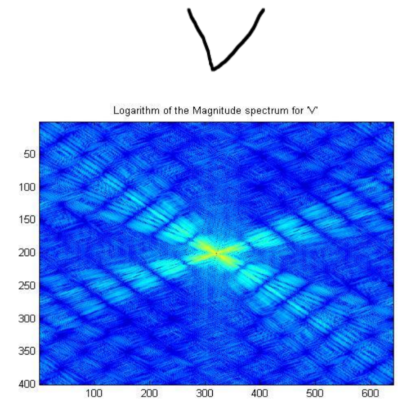
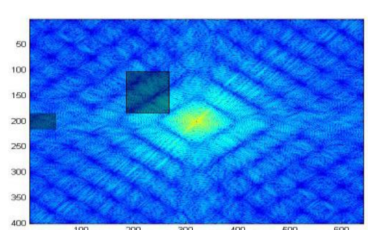


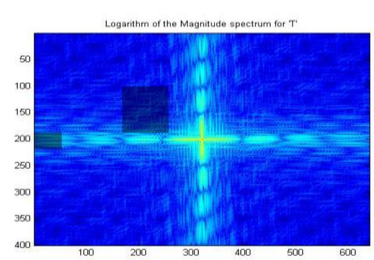
Figure 2.(Fourier space – S) *Figure 3.(Fourier space – T)* *Figure 4.(Fourier space – V)*

In Figures 2,3 and 4 we visualize the logarithm transformations of the magnitude spectrum for the three characters We can observe that we can predict the shape of the letter from the Fourier space. For example, in the spatial domain, 'T' has an horizontal line and a vertical line, which is depicted by two directions in the frequency domain: one passing horizontally through the center and one passing vertically through the center . On the other hand, 'V' has two oblique lines which can also be interpreted from the Fourier space. Moreover, the magnitude spectrum for 'V' varies more than the one for 'T' as the given 'V' character is very slightly curved. We can observe this clearly if we a look at Figure 2, which is the frequency domain for the letter 'S' which has values in all directions.

3 FEATURE SELECTION

Having the Fourier space for our three characters, the next step is feature selection. Feature selection is a very common practice in Data Science that acheives dimensilnality reduction. In other words, we only keep the features that best separate our three characters and use the selected features to classify them accordingly. We strive for compact representation of the properties of the data, representation that removes redundancy and irrelevancy. This procedure is essential for making reliable predictions for future data, as well ass increasing efficeny when working with large amounts of data. Exploring the frequency domain, I tried to select different spectral regions by aplying regtangular masks for finding the spectral features that best separate our data. A mask lets us ignore the rest of the spectral space and concentrate on the selected region. I followed a trial and error approach by selecting rectangles in sensible regions of the spectrum.





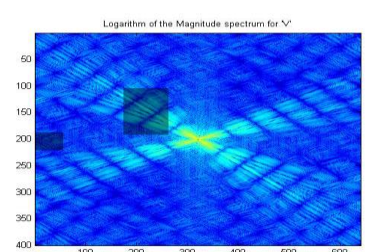


Figure 5. *Figure 6.* *Figure 7.*

After experimentation, I have chosen the area that provided the best separation between the characters and then started to experiment with different sizes of the regtangular masks in order to find the optimal size.

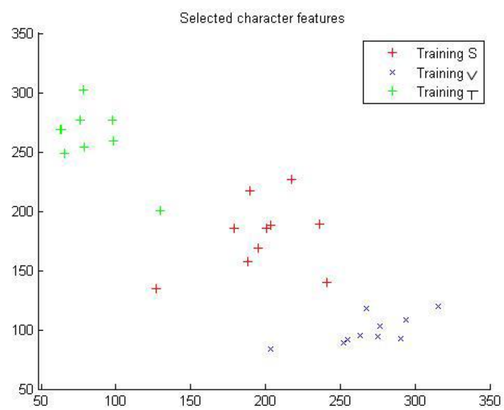


Figure 8.(the selected features represented graphically)

The chosen rectangles are faded out in the figures 5, 6 and 7 above. The logic behind it is based on the values of the fequencies in those rectangles. For example, looking at the large square, the values are very high for ‘V’, very low for ‘T’ and slowly decreasing for ‘S’. Next, we pick the magnitudes values that are contained in these regions and compute a mean for each of them. By plotting the two mean values, our features will be depicted graphicaly like in the figure below. If the chosen regions contain features that achieve good class separation, then we should observe a small within-class distance. We can see that the three classes are well-distinguishable, having small variances on both X and Y axes. It is highly probable that by epxerimenting further, better features could be chosen as there are numerous combinations of regions. Better feature selection gives training data that can derive a more powerful classifier. This is crucial when working with large amounts of data as it can save a lot of work.

4 K-NN CLASSIFIER

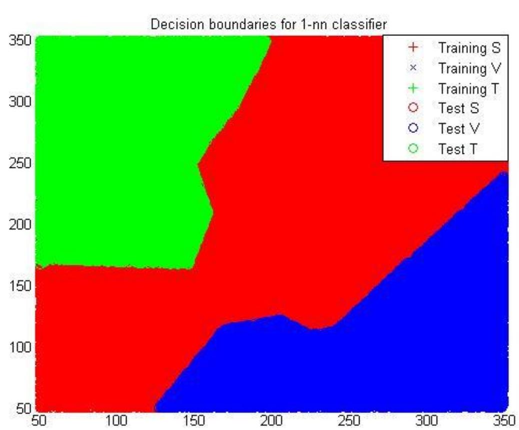


Figure 9.(decision boudaries for 1-nn classifier)

Having our training data selected, we can now make use of it and derive our classifier. This time, we are going to use the Nearest-neighbour classification method which consists of assigning the future data to the class of its nearest exemplar. By increasing the number of neighbours to k: k-Nearest-neighbour classification, we assign a test instance to the majority class among its k-nearest neighbours. By using this approach, we get smoother decision boudary than with k = 1. In order to visualize the decision regions, I generated a regular grid of test points which are colored according to the class assigned by the 1-Nearest-neighbour classifier as you can see in Figure 9.

K-nearest neighbour methods take all features into account. Thus, with large number of features this causes problems as irrelevant features dominate distance calculations. Also, the training set covers only a fraction of the instance space, which causes overfitting. Later, we are going to look at a more efficient classifier in terms of computations: Nearest-centroid classifier.

5 GENERATING NEW TEST DATA

In order to test the classifier, I have generated two test characters for each letter and applied 1-nearest neighbour classifier in in order to add the corresponding class label. For this purpose, I have used one set of characters that was drawn a bit shaky and another one where the letters had a slight rotation. Both of these slight changes cause modifications in the Fourier space that can put our classifier at test.The classifier seemed to perform well in these cases. However, the results are highly influenced by the chosen number of neighbours taken into account when doing the classification. We can observe this in the examples bew:

Figure 10 illustrates the classification of the six characters using 1-Nearest-neighbour classifier. One of the characters that should have been classified as a ‘T’ has been classified as an ‘S’. This is because the character had a very slight curve in the vertical line which increased the diversity of the frequency values in the large rectangle mask. However, if we use 5-Nearest-neighbour classifier, the character is classified well, as more neighbors are taking into account which smoothens the decision boundaries of the classifier. We can see the new decision boundaries for the 5-Nearest-neighbour classifier in Figure 12 and the data point having the right label in Figure 11.

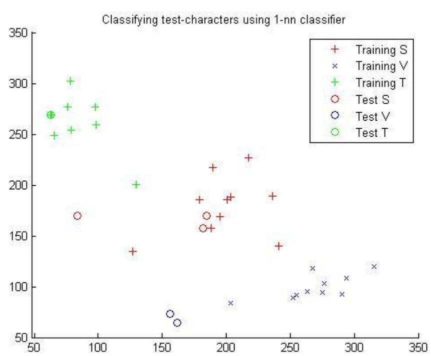


Figure 10. (1-nn classifier)

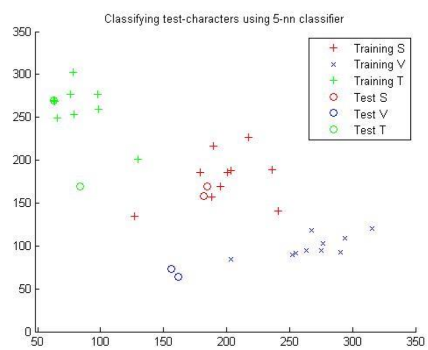


Figure 11.(5-nn classifier)

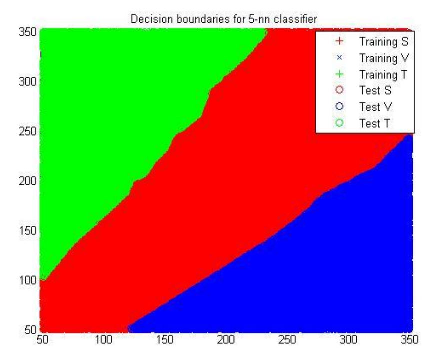


Figure 12.(5-nn classifier regions)

6 CONCLUSION

Fourier Transform is a very powerful mathematical tool that allows us to view our signal in a different domain. Thus, complex problems become much simpler to analyze using relevant regions of the Fourier space. In this project, we have seen one of the many applications of Fourier Transform, involving characters recognition which may seem complex, but simplifies when changing the problem into the frequency domain.

7 CLASSIFYING ‘A’ AND ‘B’

In this part I am going to analyze and present how two ‘A’ and ‘B’ characters are classified using the training data obtained from extracting the features for ‘S’, ‘V’ and ‘T’ characters. The labels are assigned using the 5-Nearest-neighbor classifier.

In Figure 13, the data points from ‘A’ and ‘B’ have been represented with the ‘o’ symbol. As we can see, ‘A’ has been classified as a ‘V’ and ‘B’ has been classified as a ‘T’. Looking at the Fourier spaces for ‘A’ and ‘B’ we understand why the characters have been labeled in this way.

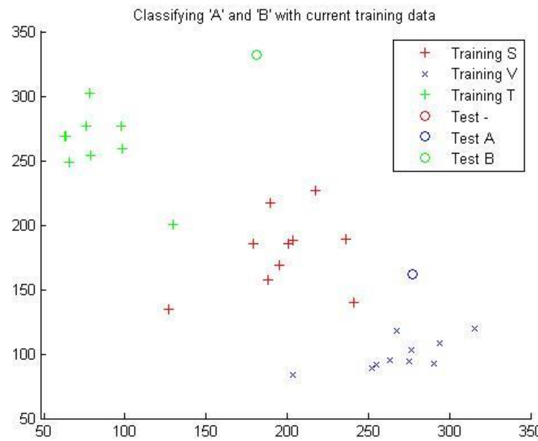


Figure 13. (classifying ‘A’ and ‘B’)

Figure 14 represents the frequency domain for the letter ‘A’, which is constituted of three lines: two oblique ones and one horizontal one, depicted by a vertical line respectively two oblique lines passing through the center of the Fourier Space. If we look back at the chosen spectral regions (Figures 5-7), character ‘A’ has very similar values with both ‘S’ and ‘T’, this being the reason why the test point representing the character ‘A’ is close to the decision boundary between the two letters.

Figure 15 is the Fourier Space of ‘B’. The strong horizontal line in the frequency domain is given by the vertical line of the letter ‘B’. This is the main reason why the ‘B’ character is classified as a ‘T’ as they have a very strong similarity in the small rectangle.

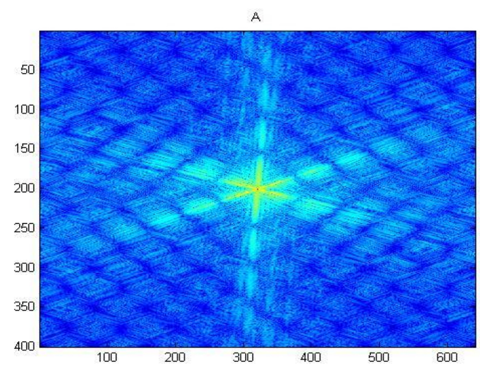


Figure 14. (Fourier space – ‘A’)

However, ‘B’ is close to the decision boundary between ‘V’ and ‘S’ as the frequency values in the bigger spectral region are similar between ‘B’ and ‘S’. The values in the Fourier space for the smaller rectangle are much bigger than the ones in the bigger rectangle, this being the main reason why ‘B’ is classified as a ’V’.

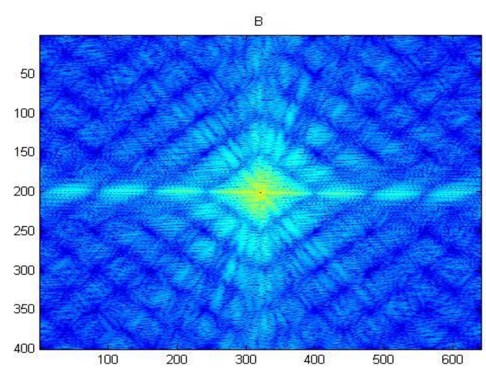


Figure 15. (Fourier space – ‘B’)

8 NEAREST-CENTROID CLASSIFIER

So far, we have used the k-Nearest-neighbor classifier in order to label our future data. In this part, I am going to demonstrate the classification of the same test characters with a different classifier: Nearest – centroid, while analyzing and comparing the results of the two labeling methods.

The k-Nearest-neighbor classification helps us predict the label of a new data point by assigning it to the class that has the most instances amongst the k-nearest points to the test data. On the other hand, the Nearest-centroid classifier assigns the test data to its nearest centroid. The centroids are obtained from calculating the mean value for each feature classes. For example, if we have three classes, we are going to obtain three centroids, each representing the mean value of the instances in the class it has been assigned to.

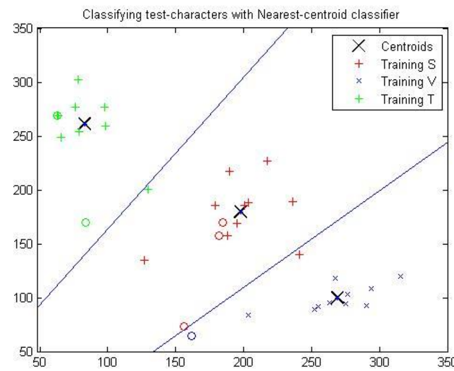
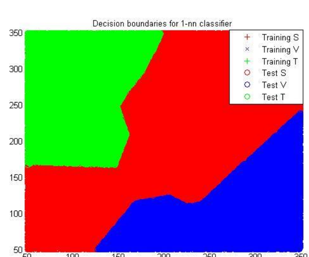
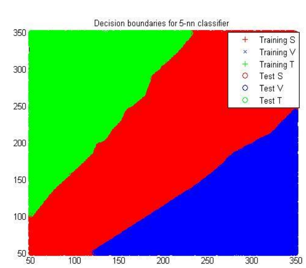


Figure 16. (Nearest-centroid boundaries and test data)

In Figure 16, I have chosen the same features as for the k-Nearest-neighbor classifier and then computed their means by using the method described above. If we plot the results, we should see three points representing the centroids of each class, which are marked with 'X' in Figure 16. To help us compare the two classification methods, we plot the decision boundaries of the Nearest-centroid classifier as we can see in Figure 16. To this end, I have made use of a Voronoi diagram which represents a partitioning of the plane where every region consists of all points closer to the corresponding centroid than to the other two. We can already see a big difference between the decision boundaries in Figure 16 and Figure 9. As the 1-Nearest-neighbour classifier ignores all the other points but the closest one, the outliers can make a larger impact, this explaining the curved decision boundaries in Figure 9. The outliers have a small contribution to the class mean, implicitly to the decision boundaries in Figure 16, so we can observe that two points of the training data would have been labeled wrong using the Nearest-centroid classifier. Another observation would be that the decision regions of the Nearest-centroid classifier are very similar to the ones of the 5-Nearest-neighbour classifier. This is due to the fact that more data points are taken into account which converges into similar decision boundaries. The decision regions for the Nearest-centroid classifier will always be the same, but they will change depending on the 'k' in the k-nn classifier as we can see below. Thus, the Nearest-centroid classifier is more general and more efficient in terms of computations. In Figure 16, we can observe that the test characters have been classified similarly.



5-neighbours



15-
neighbours