CW2 Character Recognition

# Fourier Space Analysis

In order to select features that would form the basis of my classifiers, I began by visually inspecting the log of the Fourier space of some of the different characters individually. This allowed me to get an overview of how the different shapes in each of the letters related to the output of the Fourier transform.

In an attempt to get more definitive output, I decided to try using Sobel edge detection. When viewed as the raw image output, this appeared to be very effective, and very clearly showed the defining lines of each character. However, when passed through the Fourier transform and viewed in the Fourier domain, the pixelated nature of the output produced very strong horizontal and vertical lines through the origin. Therefore, I decided against using the Sobel operator.

In order to quickly get an idea of how the different variations of each character produced similar patterns in the Fourier domain, I decided to average all the images of each symbol. To achieve this, I read in all images into a 3 dimensional matrix, then averaged this back down to a 2 dimensional image. I then performed the 2D FFT on these averaged S, V, and T inputs. This made it much easier to compare the three different classes side-by-side, incorporating all the variations.

# Spectral Feature Extraction

Looking at the three Fourier domains, I saw the three patterns created by the characters were as follows:

|  |  |  |
| --- | --- | --- |
|  |  |  |
| **Figure 1 (S):** A concentric square rotated 45 degrees, with a fairly regular spread of intensity. | **Figure 2 (T):** Two very strong areas of magnitude along the horizontal and vertical axes. | **Figure 3 (V):** Similar to T, albeit with the intense regions rotated and more diffused. |

From looking at these, I chose to use a pair of box features, where I would take the summation of frequency magnitudes in that area of the Fourier domain.

|  |  |
| --- | --- |
|  | **Figure 4:** The red rectangles show the box features I chose to differentiate between the three characters, superimposed on the Fourier domain for V. I decided to take a long vertical strip along the centre, as this area was a strong differentiator between T and V characters. (Due to the presence of long vertical lines in T characters not as prominent elsewhere.) I also decided to take a rectangle from the diagonal strip of intensity that was consistently prominent in the Fourier domain of V characters. (Due to the diagonal lines of V.) |

# Fourier Domain Analysis

I used k-Nearest-Neighbour classification in order to classify character data based on the features I had selected. The first step I took to test the accuracy of my classifier was to use it to classify the data I used to train it. I had to use a k value of greater than 2 – this is because I was using the nearest point to decide on classification in the event of equal occurrences of two (or more) classes. Therefore, in either case of k = 1 or k =2, I was relying solely on the single nearest neighbour. Intuitively, this would simply return the classification I manually gave each point, as they would be identical values.

# k-NN Classifier Analysis

To further test my classifier, I drew 2 of each character to use as test data, and passed them through my classifier. (These images can be found in the archive test\_chars.zip.) This gave me the following results:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S1** | v | **T1** | t | **V1** | s |
| **S2** | s | **T2** | t | **V2** | v |

As you can see, my classifier is accurate for T, but has some problems differentiating between S and V in 50% of cases.

Another method I used to analyse the efficacy of my classifier was to produce a visual representation of the decision boundaries it produced. I achieved this by generating a regular grid of all possible values for my features around the range of magnitudes displayed in the data I had seen so far. I then plotted these coloured in correspondence to their classification by my k-NN classifier. You can see the outcome of this in figure 5 below.

|  |  |
| --- | --- |
| D:\Local\Dev\Courseworks\COMS21202\cw2\decisionBoundaries.png | **Figure 5:** This plot shows the decision boundaries created by my classifier. The x axis represents the sum of absolute magnitude in the box feature that captures diagonals. The y axis represents that of the vertical bar.  This clearly explains the pattern shown here – the Fourier space for T symbols is very intense in the vertical box, yet very weak in the diagonal area. Therefore, it occupies the left hand side of the plot. |