Candidate: og14775

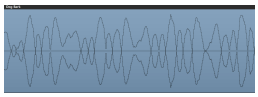
SPS Coursework 2: Report: Introduction and Background

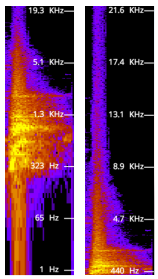
For over a century, we have known that any n-dimensional aperiodic function can be decomposed into a set of sin and cos waves (or complex exponentials as shown below) with a Fourier transformation. This can be applied to images (two dimensional matrices). Images can be decomposed into a set of waves, each with frequencies where peaks of the respective waves are represented by white areas (ignoring RGB) and troughs representing black areas where each value in the matrix represents a discrete value. Waves of different phases and magnitudes (individually represented using Argand diagrams) can be combined and subtracted to form constructive and destructive interference patterns, which eventually form the image. Higher frequency waves represent finer detail within an image and lower frequency waves represent image contrast. In Fourier space, after transformation by properties of conjugate symmetry (see right) where the top left and bottom right sections are switched (and vice versa for top right and bottom left), each point represents a frequency, with points further away from the centre representing higher frequencies and those towards the centre, lower frequencies. Often these frequencies can be close to zero and so taking the logarithm allows for better visual inspection, in order to see them. There was no need to do that in this coursework. In addition, horizontal lines in Fourier space reflect vertical lines in the inverse Fourier space (the image), and vice versa. Diagonal lines represent diagonal lines in the Fourier space by multiplying the gradient of the line in the image by -m where m is the gradient of that line. As tablets become more prevalent, optical character recognition increases in importance and can be very effective with limited training data.

Conjugate Symmetry

$$F(u,v) = F^*(-u,-v) \rightarrow |F(u,v)| = |F^*(-u,-v)|$$

Bark waveform





Fourier space of bark

The image, first on the right is the bark of my dog in the time domain, which refers to variation of amplitude of the signal with time. The middle (log representation) and far right figures show the spectrogram. This is the frequency domain which shows in a given time period, the quantity of a particular frequency in the one dimensional audio signal, where a brighter colour shows higher decibels at a particular frequency. The process of conversion is called a Fourier transformation.

$$F(k) = \int_{-\infty}^{\infty} f(x) e^{-2\pi i k x} dx$$

Fourier Transform Equations

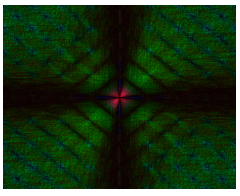
$$f(x) = \int_{-\infty}^{\infty} F(k) e^{2\pi i k x} dk$$

Euler's Theorem

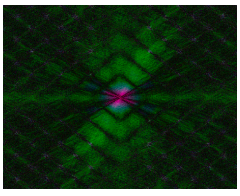
$$= \int_{-\infty}^{\infty} f(t)(\cos(2\pi\nu t) - i\,\sin(2\pi\nu t))\,dt$$

This process is described by the following set of equations, where F(x) is the mapping from the time to the frequency domain and f(x) is the lossless inverse Fourier transform which converts to the time domain. The complex exponentials can be converted to a set of sin and cos waves through the application of Euler's theorem **(see right above)**.

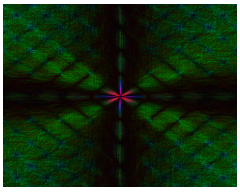
S & T difference mask



S & V difference mask



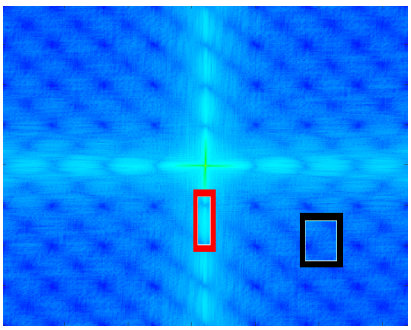
S & T difference mask



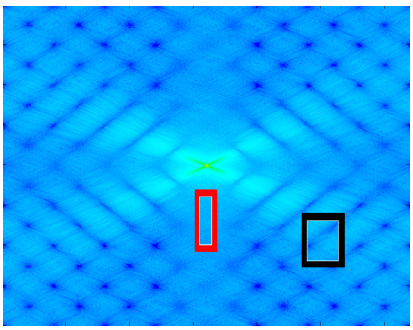
Analysis of the Fourier domain & Feature Selection

I selected my features by visual inspection of the Fourier space & trial and error. I tried to keep the number of features as small as possible and to keep features orthogonal to each other. Before feature extraction, I **considered** an enhancement.

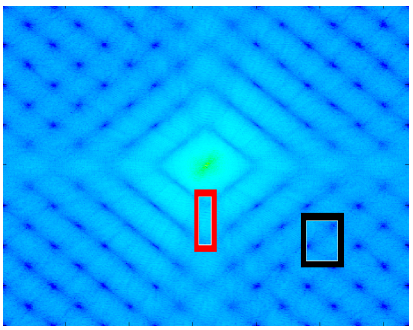
Mean T Image



Mean V Image



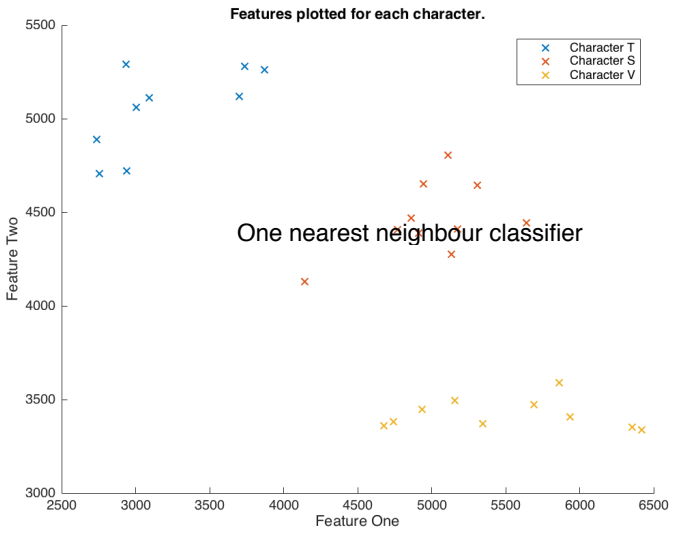
Mean S Image



This enhancement was to remove salt and pepper noise from the image or to correct any defective pixels using a 3px median filter, which preserves edges, something that is important for the Fourier analysis. However, after analysing the difference in Fourier space of a non noisy and noisy image there was little difference and the feature values did not change much.

After this, for each character, I took the mean of the Fourier values across all character images. This allowed me to see the shared maxima and minima easily. I then applied a difference mask on the mean images in Photoshop to allow me to see the areas of maxima even more clearly, **see above**. On the difference mask the darker areas are where the image is most similar, the lighter coloured areas are where there is a greater difference. I noticed that the **red rectangular feature (feature two)** in T differed greatly from the red box in V, which clearly has a lower magnitude.

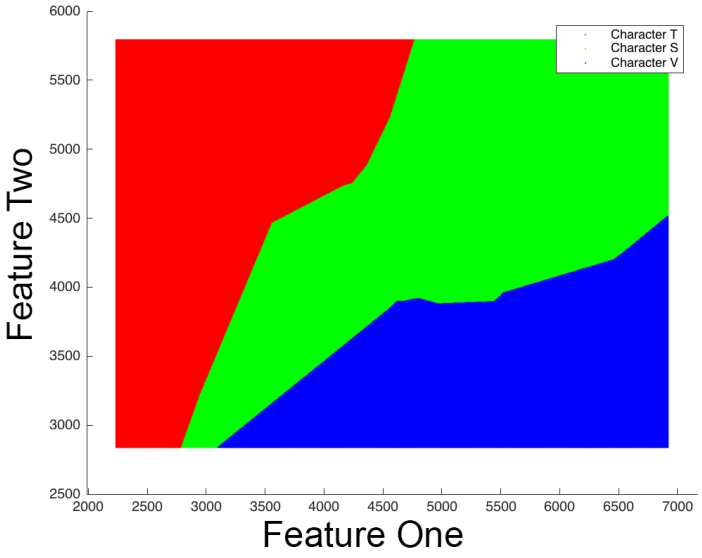
In S, the values for feature two tended to be brighter than V also, which due to the shape of the character, V does not have many as horizontal lines in the spatial domain (represented by the vertical lines in the Fourier space) and different from T which had a vertical stripe running through the middle which represents the horizontal bar associated with the head of the "T" shape in the spatial domain. In the **black box (feature one)** to the lower right hand side, S and V are relatively similar, with V having a slightly higher magnitude than , but both very different from T which has far fewer prominent diagonal lines representing the frequencies that occupy the lower left corner of the Fourier spaces. V has a greater magnitude than S, though, in this region from the diagonal lines it has in the spatial domain which represent diagonal lines in the Fourier space. This, I hypothesised, was enough to distinctly classify all three as a clear inequality relation can be constructed, for feature one V >= S >= T and for feature two T >= S >= V. One may ask why two features are needed and this is because often the amount of difference is small and two features prevent overlap. I considered also using ratios of values from one region over another to perhaps add more robustness, but decided against it after seeing that the method worked effectively. I therefore saw no reason to add complexity to the system. All Fourier space images have adjusted contrast/saturation/ exposure changed to help see features clearly.



Feature two is less, but not completely, invariant to rotation, important for tablet input, and also more reflective of the contrast and less responsive to changes in detail in the image. In the centre, for every degree moved the power spectrum of the rectangle value will not change as much, however it is thin. I did not choose a ring mask to allow for complete rotation invariance as if I receive an upside down character, I see that as a completely different character and I would say that classifying a 180 degree rotated T shape is under-fitting. This is important when considering how to best take input from a tablet.

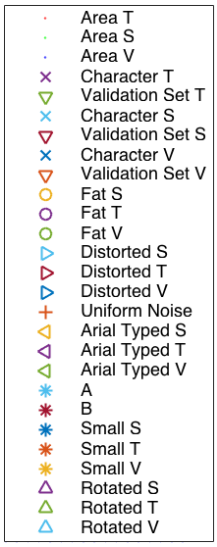
Feature Plots

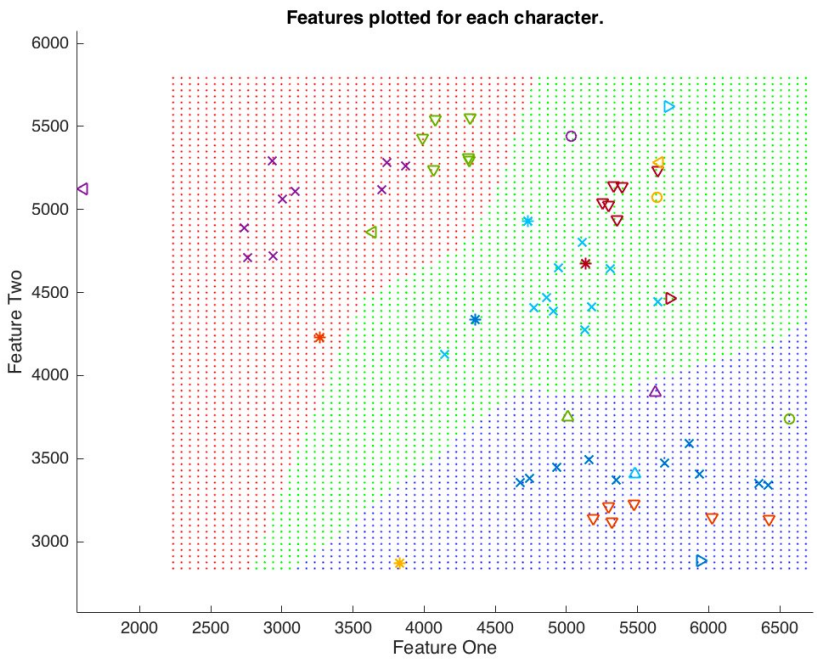
After calculating the power spectrum of the regions described, I plotted the two features for each character against each other, **see above**. This achieved a compact and well clustered grouping, shown here. Upon visual inspection, although the single linkage between any two clusters is slightly smaller than the class variance, given the right number of neighbours these can be classified well.

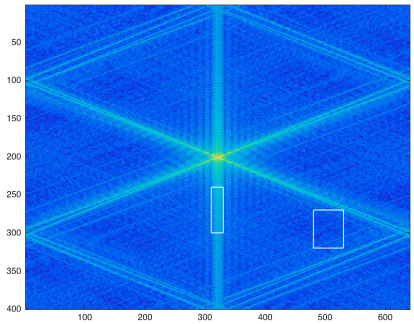


There appears to be one single anomalous result in category S at (~4100,~4100), This is still closest to the

cluster of S data points. We can see the feature set is orthogonal. Additional feature selection or feature extraction such as PCA of any kind would be inappropriate. One could argue, however, that with more features, a tighter clustering could be achieved.







Arial V Fourier Space (Above typedV.gif)

Nearest Neighbour Classifier

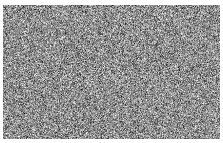
The nearest neighbour classifier, given a set of **sample** points looks at the k nearest neighbour **data** points in terms of euclidean distance and outputs a classification depending on what the majority class of the k neighbouring data points are. If there is an equal number, the algorithm randomly chooses.

In my case I generated a regularly spaced grid of points. This gave very clear and well defined groups with no irregular enclaves as can be seen on the right.

Nearest Neighbour Analysis

The classifier generalised well with the feature set, in the sense that each character in the validation set was correctly classified. The validation set consisted of new images that were hand drawn by a variety of friends, from both the UK and outside of the UK in order to keep the results independent and identically distributed, and to test the robustness of the system in dealing with variation. In addition, I generated monochromatic uniform noise **(see right)**, and found the classifier placed the coordinate, corresponding to the noise an order of magnitude away, which lends towards the ability of the feature set to determine irrelevant features.

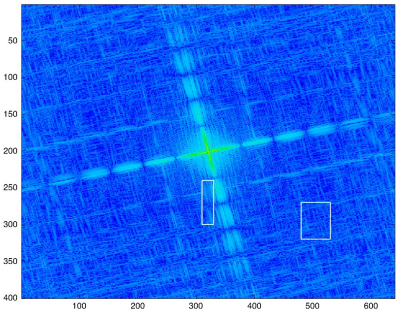
I tested images drawn with a far larger 'pen', labelled fat images (see right), which represented perhaps a larger finger on a tablet display. The fat T image was incorrectly classified as an S. I also distorted many images with drastic warping effects and found that once an
again the T image was incorrectly classified as an S character.

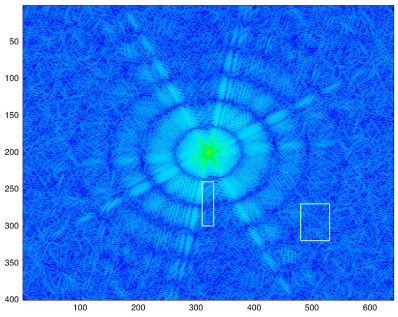


noise.gif

I also tested arial font to see how well it could take alternative unseen data. The feature selection classified all correctly (including T upon obvious extrapolation on the graph) apart from V, which it classified as a T. This is due to the horizontal bar in the image at the bottom of the V which creates a strong vertical presence in the Fourier space, similar to what every T has. I also tested small characters, which children may have typed, for instance, which were all classified correctly as they had similar Fourier

Fat T Distorted T





spaces and respective frequencies, given the images had the same structure.

Rotation had an interesting effect on the images. The feature set and resulting classifier did not perform well, classifying two out of the three incorrectly; specifically S and T in the V area. Visual analysis quickly identifies why.

The T Fourier space looks incredibly similar to the V Fourier space. The features do not capture high magnitude frequency regions, instead they take two relatively medium magnitude regions which is characteristic of a v shape, because the vertical lines in the spatial domain were offset. The contrast in all images represented in the centre of the Fourier space, is very similar for all images.

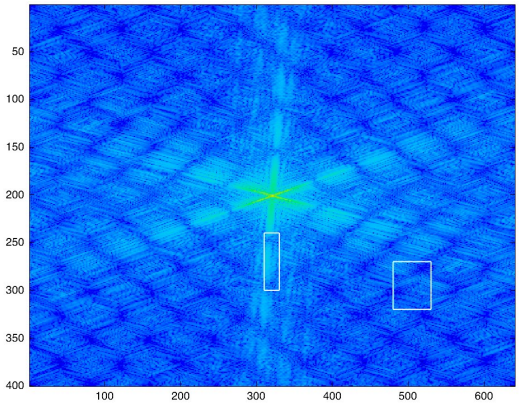
Classification of A & B

Both images were classified as an S. This can be seen on the graph above, where A and B are labelled with stars. The features I used remained unchanged for A1.GIF and B1.GIF.

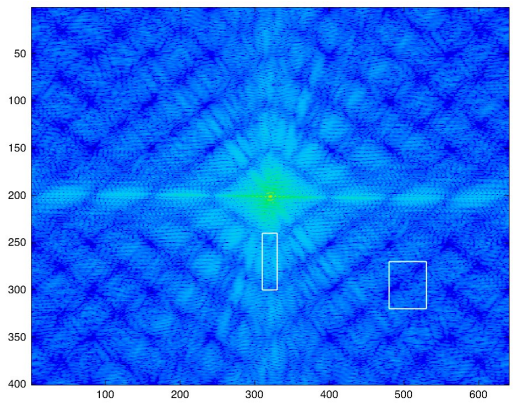
Classification of A

A had a relatively high value for both feature one and feature two, the two white boxes in the image above. This gives it the properties to be classified as character S. It has strong vertical components and strong diagonal components in the Fourier space from the presence of diagonal and horizontal lines within the spatial domain of the image, which derive from the two diagonal 'arches' in the spatial domain of the image A1.GIF. This creates a star like shape in the Fourier space.

A Fourier Space



B Fourier Space



The contrast on both images is also similar shown by similar central areas. A is closer to the Fourier space of T, than B because it has a slightly higher feature two value from the stronger horizontal components than B, with the prominent central stripe that T has and has slightly less diagonal frequencies, so feature one is lower pushing it even more towards T. One may have thought that the diagonal arches of the letter in the spatial domain would have pushed it to be classified as a V, but the arches are at different angles and therefore have different frequencies in the Fourier domain.

Classification of B

B also had a relatively high value for both of the features. B has distinct 'pulsating' diamond shapes emanating from the centre of the Fourier image, like S. From visual inspection this may have a relation to the curved characteristic of the B shape which is very similar to the shape of an S, as B almost contains the letter S as a subset of its shape. It has very strong horizontal components in the frequency domain which derive from the strong vertical lines in the spatial domain of B1.GIF i.e. the backbone which supports the letter. It indicates that more features, or even better features would have to be created in order to support the recognition of A and B on a tablet device.

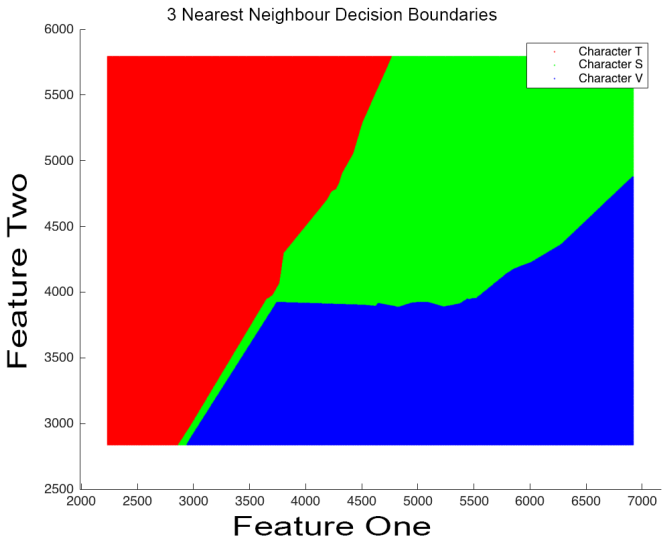
Alternative classifier: Decision Tree

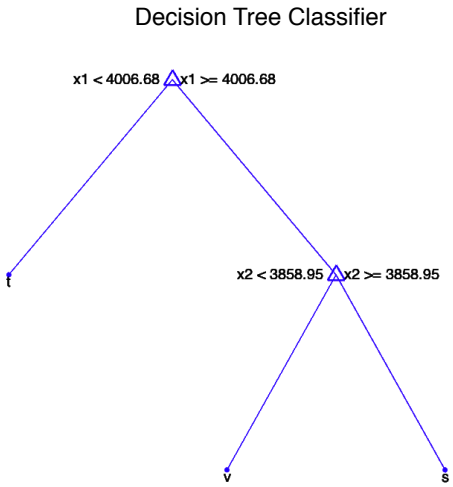
After playing with k nearest neighbours, using different values, the results were not interesting e.g. k = 3 **(see below)** is very similar to a value of k = 1. Increasing the value of k further started to produce erratic results, as the classes interfered with each other as some points were closer to other classes for certain values of k resulting in misclassification.

I decided to implement a decision tree **(see below)**. The tree did not need to be pruned as MatLab decided it found an optimal split with only two levels. The tree partitions the instance space into regions of uniform class membership where internal nodes are labelled with features and leaves are labelled with classes. It divides the instance space into axis parallel rectangles and attempts to maximise the decrease of impurity from parent to child node. The impurity index that MatLab uses is the Gini Index. This is very similar in shape to entropy **(see left for two event entropy)**. It measures the amount of information a particular split provides and maximises this by setting a particular threshold at each split which it must be over, where p_j is the probability of an event j occurring.

$$-p_1 \log_2 p_1 - (1-p_1) \log_2 (1-p_1)$$

$$\sum_{j=1}^{c} -p_j \log_2 p_j$$



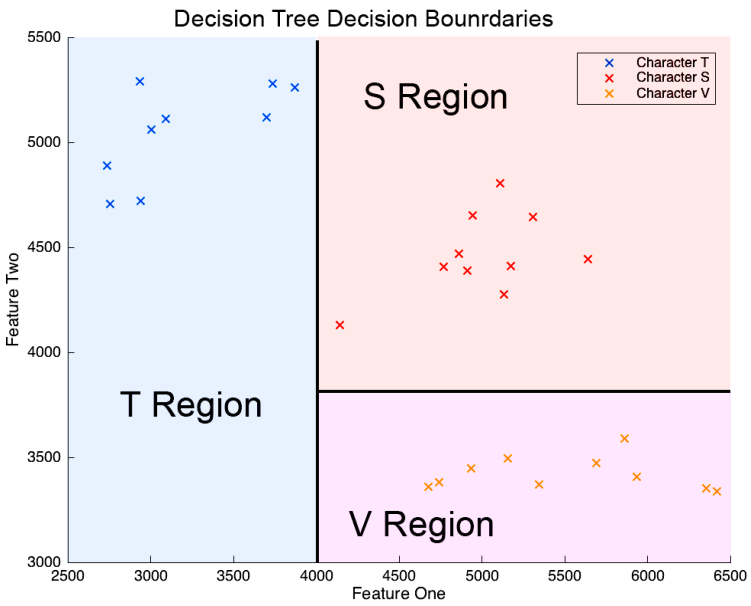


The motivation to use this was in order to compare different classifiers and inspect the different accuracy levels, given the relatively small amount of data that we were presented with.

Results of the Decision Tree Classifier

It managed to predict the classes S and V completely correctly with just two splits in the decision tree. However, it failed when classifying the t validation set, only managing to classify only one of the six correctly (normT3.gif). This suggests that the model is under-fitting and failing to generalise well. Despite turning pruning off, to my best efforts, Matlab was unable to improve on this, and did not increase the number of splits in order to give more detail to the model, but one must consider that the risk of overfitting also looms. The tree is on the right. It classified A and B in the S region. The decision

tree generally classified the rotated, fat, small, typed and distorted characters well, but often misclassified the letter T. Compared to the nearest neighbour classifier it is far simpler and doesn't have the complexity of the decision boundary, as it approximates it with axis parallel rectangles **(see left)**. The reason why the tree stopped at the point it did is that it had reached maximum entropy gain and split the space into regions where in each region every instance was only one class, lending itself to minimum impurity. There is an inductive bias of the decision learner in that it prefers trees with informative attributes close to the root, and it also has a preference bias, as it does not search the complete space. This is different to the nearest neighbour classifier which will only search a local area, exclusively. The nearest neighbour performs far better with respect to classifying the validation set, and each of the other characters. Both of these classifiers have done surprisingly well on very limited training data. Comparing the massive amount of data an equivalent neural network may need, what we have with only 10 images is effective.



References: https://upload.wikimedia.org/math/e/4/2/e4285d1d4bd15710afcc2753c9ed3095.png