# 1 Introduction

This report aims to demonstrate the classification of test data using Fourier Analysis. The data being classified is in the form of a 2D image and represents a text character that has been handwritten. Initially, 10 images for each of the characters 'S','T' and 'V' were used for training the classifier. Test data was then generated for each of these characters in order to test the performance of the classifier. Lastly, the report explores the classification of characters outside the domain of just 'S','T' and 'V'.

# 2 Approach to analysis in the Fourier domain

The Fourier transform expresses an image given by $f(x, y)$ where $x$ and $y$ are the coordinates of a pixel, as a function $F(u, v)$ which is defined by a series of sinusoidal functions of different frequencies. Before extracting or selecting features, it is important to understand what exactly different regions of the Fourier transform represent. By filtering different regions, it is easy to see the effect that a region of the Fourier transform has on the image that it represents. In the following section it is only the magnitude of the Fourier transform that is involved. Using a filter, $H(u, v)$, a new Fourier transform can be created: $G(u, v) = F(u, v)H(u, v)$. A low or high pass filter is a function that holds a value of 1 for some values of $u$ and $v$, and then 0 for others. A low pass filter only allows the lower frequency components of the Fourier Transform to be present in the resulting Fourier Transform. A high pass filter does the same but for the higher frequency components. Figure 2 and 3 show the process of applying a low pass and high pass filter, respectively, to Figure 1.

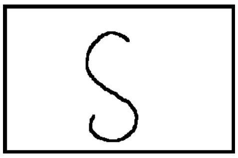

Figure 1: *An image of the character 'S'*

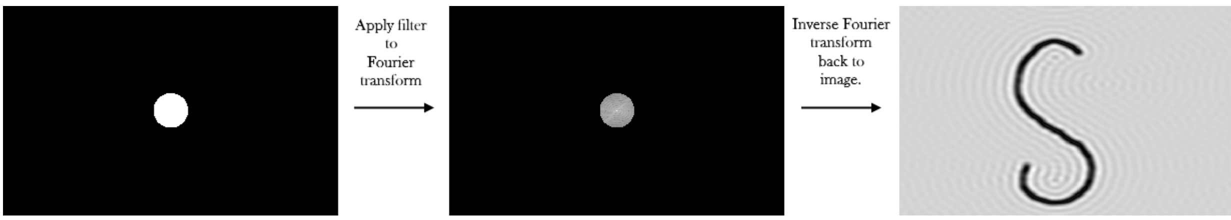


Figure 2: *Image of applying a low-pass filter to Figure 1*

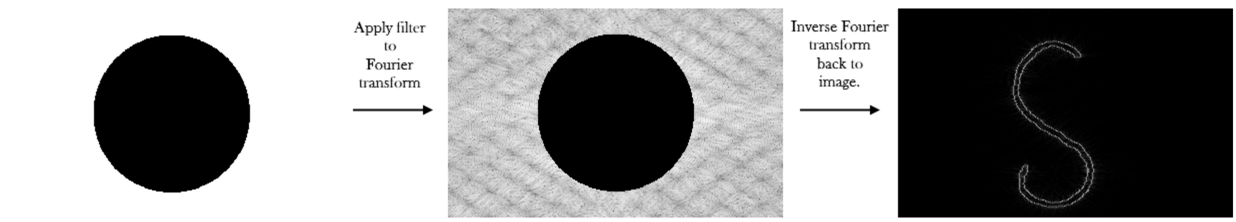


Figure 3: *Image of applying a high-pass filter to figure 1*

Two important observations can be made from this:

- Lower frequency components of the Fourier Transform represent the more fundamental or basic structure of an image - it focuses on the main contrast in the image. This is evident in the black body formed by the 'S' in contrast to the light gray background in Figure 2.
- Higher frequency components of the Fourier Transform contribute more to the bespoke or fine details of an image. This is evident in the fine white outline produced by Figure 3, which shows detail that is very specific to the given image.

If there is a focus on the higher frequency components of the Fourier transform, then there's a good chance that images of the same class will have little similarity in these values, because finer details differ more between characters of the same class than basic details. Alternatively, if there is a focus on the lower frequency components of the Fourier transform, then it's quite probable that images of the same class will have similar values as they share the same basic structure (ideally). Therefore, lower frequency components are key in differentiating between images of different characters, and features within the region of the low pass filter are good candidates for the classifier.

## 3 Feature Selection

Seeing as the Fourier Transform is separable, its axes can be dealt with independently. A low pass filter was applied to the Fourier transform of each image to separate the lowest frequency components from the rest. We took the 32 lowest frequency values from the u axis and v axis to form a feature vector with 64 components. In the case of using a classifier that depends on 2 of these features, there are $c(64, 2) = 2016$ possible combinations. This is obviously too much to sort through manually and so we took advantage of a property of linear plots; if a 1-dimensional plot of a feature shows separation between classes then it is a good candidate for clustering. The idea behind this is that a 2-dimensional plot of 2 features which both show separation will also show separation between the same classes. Features which show good separation in linear plots will form separation along a single axis in higher-dimensional plots.

In this case, the classes of the data are known, so these are plotted as different colours (S,T,V correspond to red, green, blue). Looking at 1 dimensional plots means we only need to consider 64 different plots. To help find "better" features i.e. ones which show a larger degree of separation between classes, we calculated the overlap between the range of each class for each feature. E.g. Figure 4 has an overlap of 0 between the red and blue classes because the space occupied by the set of red classes does not overlap with the set of blue classes. Figures 4 and 5 all show features which have an overlap of 0 between the red and blue classes:

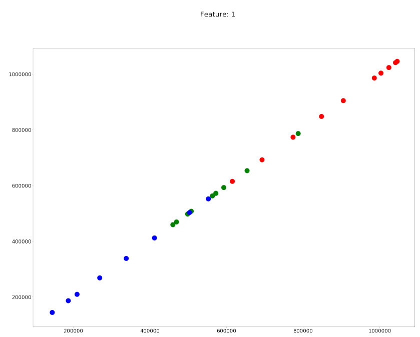


Figure 4: *Linear plot of Feature 1*

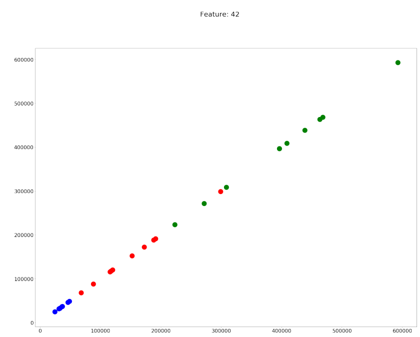


Figure 5: *Linear plot of feature 42*

While feature 1 displays a good separation between the red and blue classes, there is almost no separation between the green class and any others. Feature 42 on the other hand separates all three classes with clear boundaries apart from a red value which deviates at around 30,000. This means that it is a good candidate for clustering.

We decided to sort the set of features in ascending order, using the value of the overlap to sort. This means that features with an overlap of 0 (no overlap) are shown first, and so it is easy to select the features with minimal overlap between classes. Following this we made two lists, one for features which have an overlap of 0 between red and blue classes and another for features which have an overlap of 0 between red and green: List 1 : [1, 42, 46, 50, 53, 56, 59, 61, 58, 62], List 2 : [10, 11, 12, 13, 15, 43]. We manually observed the clustering for each combination of feature values from each list as it was quick to see which feature combinations were suitable and which weren't. Figure 6 shows feature 15 and figure 7 shows our best candidate for clustering which uses feature 15 and feature 42.

It should be noted that green and blue classes are separated by both feature 42 and 15. Thus, a 2D plot of feature 15 against feature 42 provides clear separation between each of our classes:

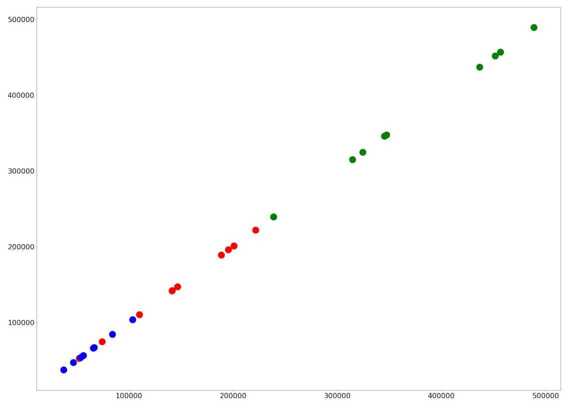


Figure 6: *Linear plot of Feature 15*

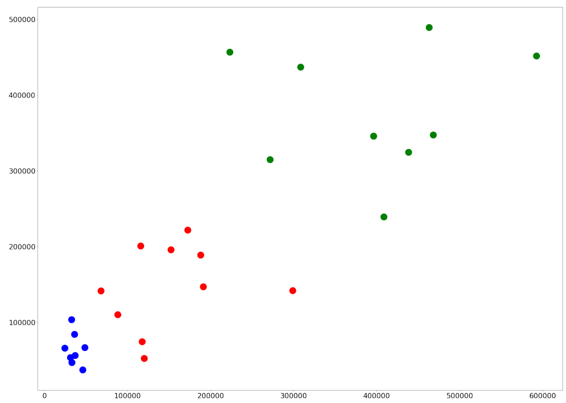


Figure 7: *Plot of feature 42 against 15*

## 4 Analysis and Evaluation of the Classifier

The test data was created using an 8 pixel-wide paintbrush and no anti-aliasing in the same fashion as the training data. 2 sets of test data were produced, standard characters and irregular characters which are poorly drawn and potentially more difficult to classify.

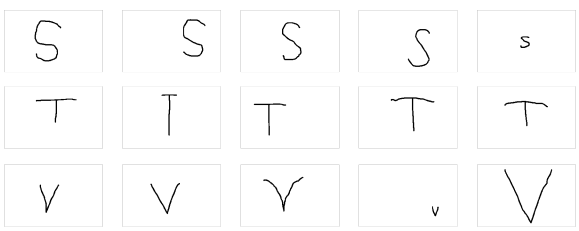

Figure 8: *The standard test data (1 to 5 from left to right)*

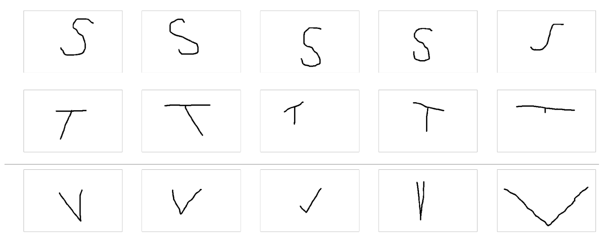

Figure 9: *The irregular test data (1 to 5 from left to right)*

Applying a nearest neighbour classifier to a test point finds the training data point with the smallest euclidean distance from it and assigns the test point the same class as the training point:

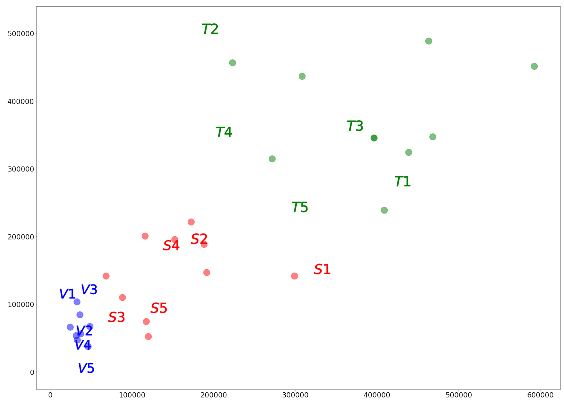


Figure 10: *Classification of standard data*

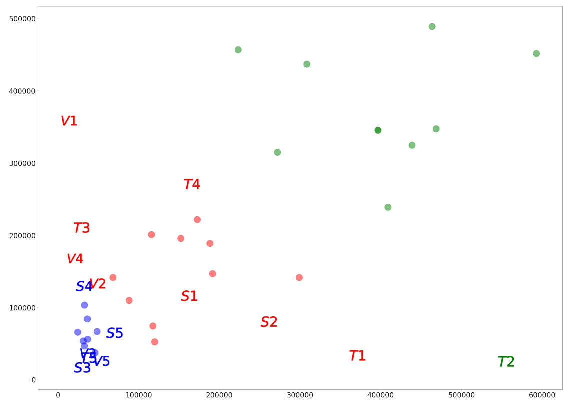


Figure 11: *Classification of irregular data*

The classifier has an accuracy of 15/15 on the standard test data. However, on the irregular test data it has an accuracy of 5/15. T3, T4, T1 and V2 on figure 11 are incorrectly classified and they lie outside the convex hull of the training data, suggesting that more training data is needed to improve the accuracy of the method. The fact these points lie outside the convex hull shows that there is a capacity for new training data to correctly classify these test points. For such a small quantity of training data, a nearest centroid approach could be more suitable, as it provides a more accurate result for regions which are unpopulated by training data. The difference between the accuracy of classification highlights a key quality about the classifier; it performs very well on standard data but it is highly sensitive to data which deviates from the norm. This is perhaps because the features only focus on low frequency components. Additionally, it may be because the feature values do not take into account enough features of the image; results may be more consistent when taking the average of a region in the Fourier Transform, as opposed to using specific values as was done here.

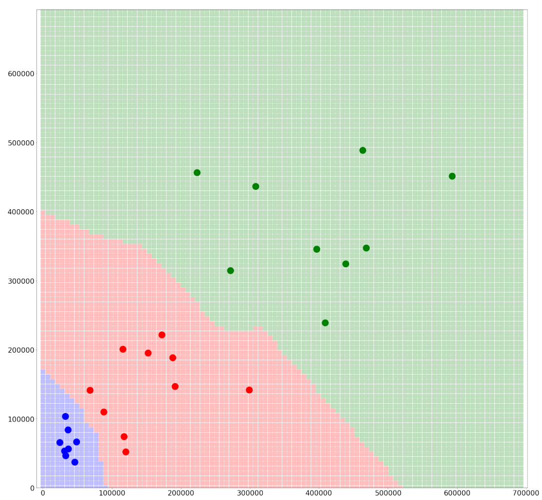

Figure 12: *Decision Boundaries for the Classifier*

The decision boundaries form approximate rings around the origin. The classifier is stronger at differentiating Ts from non-Ts than it is at differentiating 'S's from 'V's, as evident in the irregular classification. The close proximity between 'S' and 'V' values shows that the feature values bear significant similarity. A potential way of improving the classifier could be to add further feature values to make the classifier n-dimensional instead of two dimensional. Euclidean distance is straight forward to define for n-dimensions and so the nearest neighbour calculations would be simple but potentially intensive. Using nearest centroid would make boundary classification less locally sensitive and more globally sensitive.

# 5 AB Classification

Classification of characters which aren't S, T or V yield unexpected results with this classifier. The position of the plot depends purely on the values for two low-frequency components of the Fourier transform; in this case the low frequency values for S are what most closely match the values for both A and B.

It should be easy to see from figure 13 that B lies very roughly equidistant to points of all classes. It also lies in unpopulated territory, distant from any of the training points. This shows that the classification is weak. It could be the case that the empty region where B lies is a region for classifying 'B' characters. All training data points lie within a sector of the plot; B definitely deviates from this trend, and perhaps just signifies that this classifier is unsuitable for classifying 'B's.

On the other hand, 'A' lies in close proximity to the training points for S. This immediately indicates that this classifier is unsuitable for classifying 'A's because the feature value for this 'A' is too similar to that of the typical 'S'. Therefore the classifier has no way of differentiating between an 'S' and an 'A'. Equally, it could be the case that this 'A' is an outlier, and that the typical 'A' would occupy the empty region in the bottom right corner. It is simply inconclusive when only dealing with a single test point. The positioning of this 'A' would suggest that the lower frequency components of the Fourier transform for 'S' has a similar basic structure to that of 'A'.

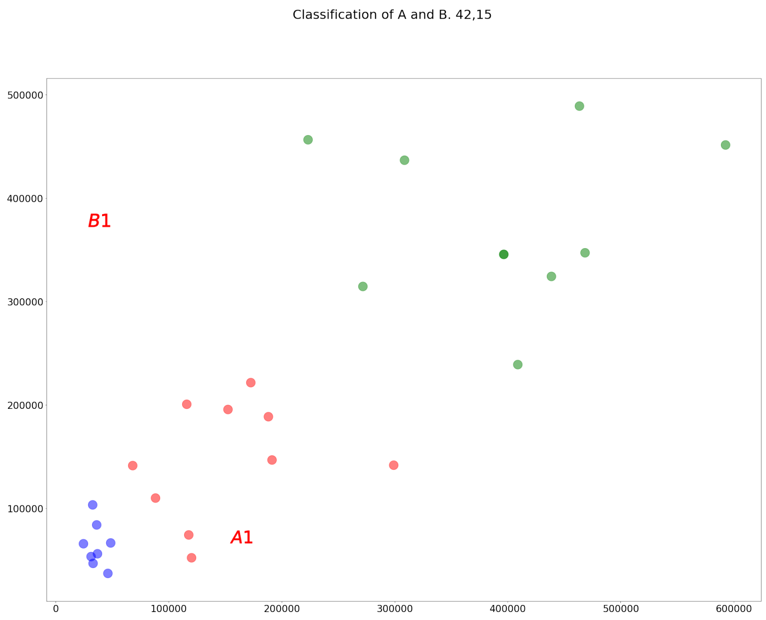


Figure 13: *Classification of A and B*