Andreas Georgiou - ag14774

Symbols, Patterns and Signals

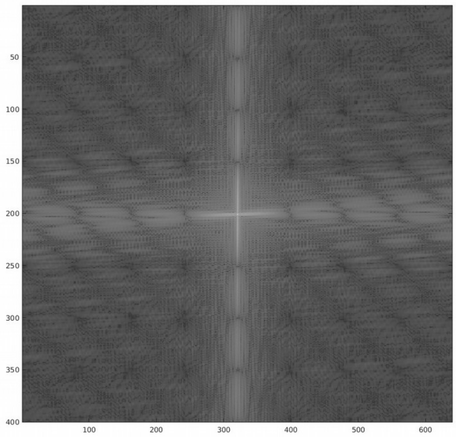
Coursework #2, Report

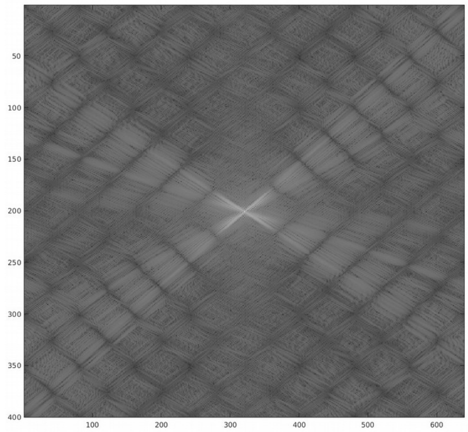
Introduction

The objective of this assignment was to design a classifier for a set of characters that given an image of a handwritten character(S, T or V) classifies the image and produces as output the letter shown in the image. This involved using Fourier transform to decompose each training image into the frequencies that make it up and carefully selecting features from the Fourier space that will help us distinguish between letters. In the next sections, I will analyze the criteria with which features were selected, how the “strength” of each one was measured and how the k-nearest neighbour classifier was used to decide on a closest match. Moreover, a number of images will be tested and classified using the methods mentioned above and the results will be shown here. Throughout the report, the term 'feature' will be used to refer to the 2 areas selected from the Fourier domain and the term 'characteristic' to refer to a particular part of a letter.

Feature Selection

The first step was to analyse the training data and decide which features to use. For that, each image from the training data was converted to its Frequency domain using Fast Fourier Transform. Fast Fourier Transform is able to express any 2D signal as a sum of a series of sinusoids. In the case of images, these are sinusoidal variations in brightness across the image. The image returned encodes the spatial frequency(each pixel represents a different frequency), the magnitude(encoded as the intensity of each pixel) and the phase. However, the frequency domain returned by FFT is represented in complex numbers and also contains negative numbers which are not very useful. Therefore, we convert it to a polar representation where the magnitude is the length of the vector in the real/imaginary plane and phase is the angle from the real axis. Since we cannot extract any useful information from the phase, we discard it and isolate the magnitude by taking the absolute value. By using the frequency domain of the image, all the unnecessary details from the picture were discarded, such as the position of the letter in the canvas. This is because the frequencies in the Fourier image always lie in the same position, with frequency 0(also known as the DC term) in the centre, the lower frequencies around the centre and the higher frequencies further away from the centre. Looking at the frequency domain of T, it is clear that there are very strong lines that go through the centre, one horizontal and one vertical, caused by the vertical part and horizontal part of T respectively(**figure 1**). Similarly, for V, there are two lines that go through the centre, but this time diagonally, caused by the diagonal lines of V(**figure 2**) and finally, S contains both horizontal and diagonal lines. In fact, if you imagine drawing a tangent line at every point throughout the letter S, it can be seen that there are lines in various angles and therefore the frequency domain ends up being strong in almost all frequencies close to the centre(**figure 3**). For this classifier, 2 features were selected. The first feature(**figure 4**), includes 4 sectors from the low frequencies around the centre. These are located in a way such that they include the horizontal and vertical lines of T but not the diagonal lines of V. The angle was chosen to be much wider than it was needed to select T's characteristic lines mainly for two reasons: if T is slightly rotated, it would still capture its characteristic lines and secondly, since S is strong in all around the centre, it will increase the reading of S without significantly affecting the reading of T which is more concentrated in a line. The second feature(**figure 5**) is again a sector of a circle around the centre but this time it is located in a way that selects the parts of the image in which V displays a strong reading. As before, although a T is weak in that particular area, an S is still going to be relatively strong but spread throughout that sector. In contrast, V is concentrated in that diagonal line and therefore increasing the angle and therefore the area of the sector does not have a significant effect on V but it does have an effect on S. Using this method, an S will have a strong reading in both features despite the fact that it does not exhibit the characteristic lines as are present in the images of V or T.





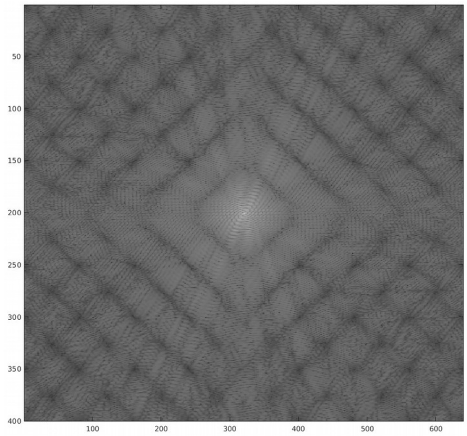
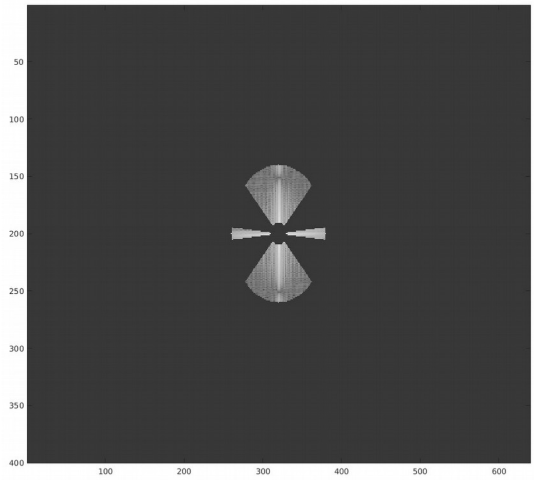


figure 1(left), **figure 2**(centre) and **figure 3**(right) show the frequency domain of an example T, V and S images respectively. By scaling the magnitude and applying a log transform of its intensity values we are able to bring out any visual detail. The resulting 'log' magnitude image is known as the image's 'spectrum'.

It is worth noting that for the first feature, the 2 horizontal sectors were only added later, to further separate T from the rest of the letters. The combination of those 2 features will help differentiate between the 3 letters because, T is going to be strong in the first feature but weak in the other, V is going to be strong in the second feature and S is going to be relatively strong in both features. To validate that both features are contributing towards the classification, the covariance matrix was calculated per class and concluded that the features are indeed independent since the covariances were very close to 0 for all of the three classes.

Feature Extraction and Rescaling

In the previous section, I referred to the “strength” of the feature without going into details. The strength of a particular feature is calculated by the power of its area. In particular, given a matrix F representing the Fourier domain of an image, the power of a feature is $\sum_u \sum_v |F(u,v)|^2$, i.e. the sum of all the elements (u,v) squared that fall in the area that corresponds to that feature. The reason for that is mainly because we need a way to transform an area of points that represent the magnitudes of the frequencies that lie in that area to a single number that we are able to manipulate and use it to make calculations and comparisons. The next step is to loop over the whole image and determine which pixels should be counted towards the first or second feature. To calculate whether a point should be included in a feature, first, its distance from the centre was measured. This way, the system is able to select pixels that are only part of a ring around the centre because that is the area in which the characteristics that help distinguish between letters are stronger. The reason behind choosing a ring instead of a circle is to avoid selecting the very centre of the image that contains the DC term. The DC term is significantly stronger than the rest of the frequencies in all images and therefore when included, it “conceals” the contributions of the rest of the pixels towards the power of a feature. The ring was selected by limiting the distance to be within two values which were decided by trial and error. Then the system further filters the points in the ring by selecting only those that fall in one of the sections mentioned above(**figure 4,5**). This is done by calculating the angle of each point relative to the centre using arctan, allowing the angles to be between certain limits and filtering out those that do not satisfy those constraints. By calculating the power for both features for all 30 train images, a 30x2 matrix is obtained which can be used later on with the nearest neighbour classifier. Keep in mind, that since this is supervised training, we need to label each image as being a member of S, T or V beforehand. On this end, a 30x1 matrix was created containing the class of each one of the 30 training images. Moreover, because different images can have different intensities and hence a much higher or lower power than the training data, normalisation is needed to be applied to each image before extracting the



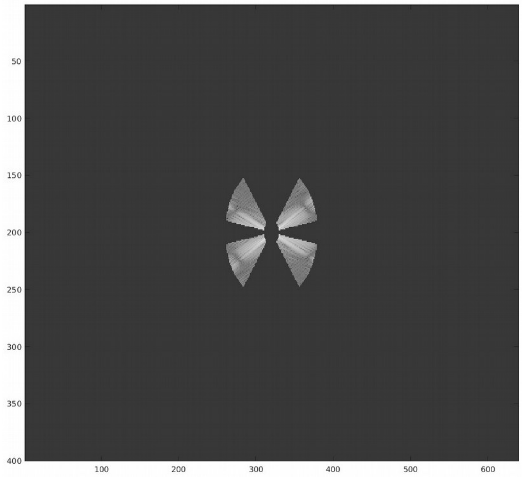


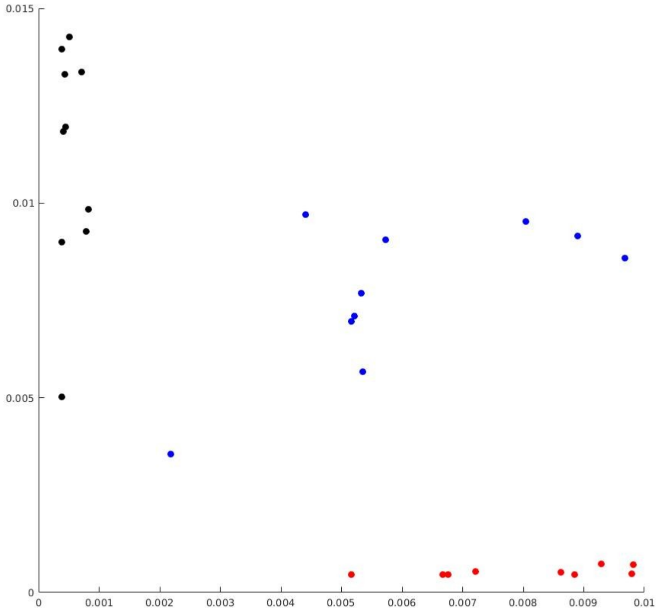
figure 4(left) and **figure 5**(right) show the features 1 and 2 respectively. All the points except the points included in the two areas have been zeroed out to make it easier to visualise the parts of the image that have been chosen as features.

---

features. This is achieved by rescaling the whole magnitude matrix i.e. calculating the minimum and maximum value and then setting all $Mag(u,v)=\frac{|F(u,v)|-min}{max-min}$. This rescales the magnitudes of the whole image to be in the range [0,1] because the point with the lowest value will have $Mag(u,v)=\frac{min-min}{max-min}=0$ and conversely the point with the maximum value will have $Mag(u,v)=\frac{max-min}{max-min}=1$. The resulting plot of the train images with respect to the selected features is shown in (**figure 6**). The clear separation of the points into 3 classes is apparent with different colours used to demonstrate the labelling. The red points(T) get a strong reading in only the first feature while scoring almost 0 in the second feature. Vice versa, the black points score almost 0 in the first feature but get higher marks in the second. Finally, the blue points get high marks in both features.

Nearest Neighbour Classifier

Now that we have our training data and each exemplar is a member of one of the three classes, a classifier can be built that is going to be used to recognise future test images. The first step is to apply FFT to the image to be classified, to get the frequency domain and isolate the magnitude. As before, the magnitude matrix is rescaled so that all F(u,v) are in the range of [0,1]. Following the same procedure, the power of the same areas is calculated and the feature vector is extracted. To classify this image, all the distances(Euclidean) between the test image feature vector and each of the training images has to be calculated and the closest neighbour, i.e. the one with the smallest distance, is selected. The class of that neighbour is the class that is also assigned to the test image. The class of the neighbour can be efficiently found by looking it up in the 30x1 matrix calculated before. Moreover, it is possible, in making a decision about the class of a test image, to consider the assigned classes of more neighbours other than just the first one. In the process of choosing the parameter k for the k-nearest neighbour classifier, various options had been tested and concluded that k=1 performed the best when tested on a separate "tuning" data set specifically created for this reason(the performance on the training data is not important) since measuring performance on the test data as a means of tuning the classifier leads to overfitting and therefore the use of a separate data set, helps towards the goal of generalization. (**Figure 7**) shows the resulting decision boundaries produced by this classifier. As it was suggested, the boundaries were plotted by generating a regular grid of test points and colouring them using the classifier. The grid only covers up to 0.02 in each direction instead of taking the whole range [0,1] mainly because neither the



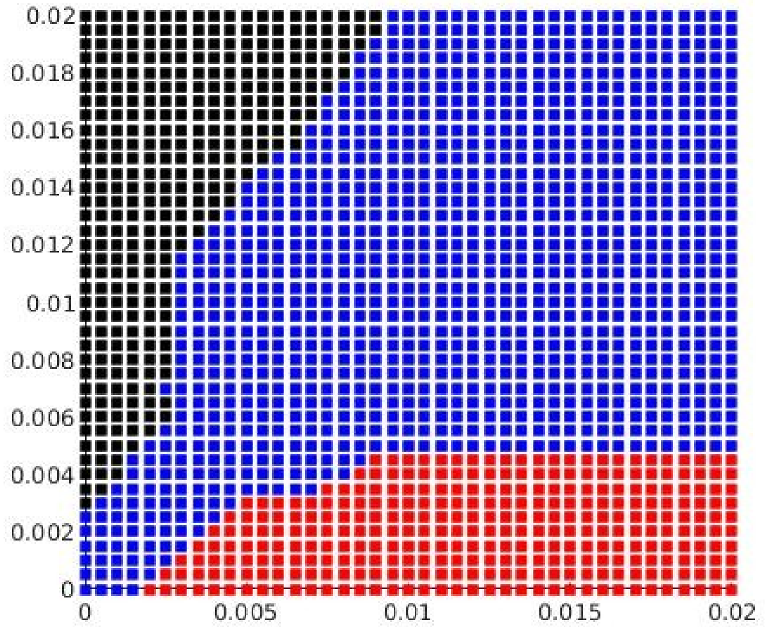


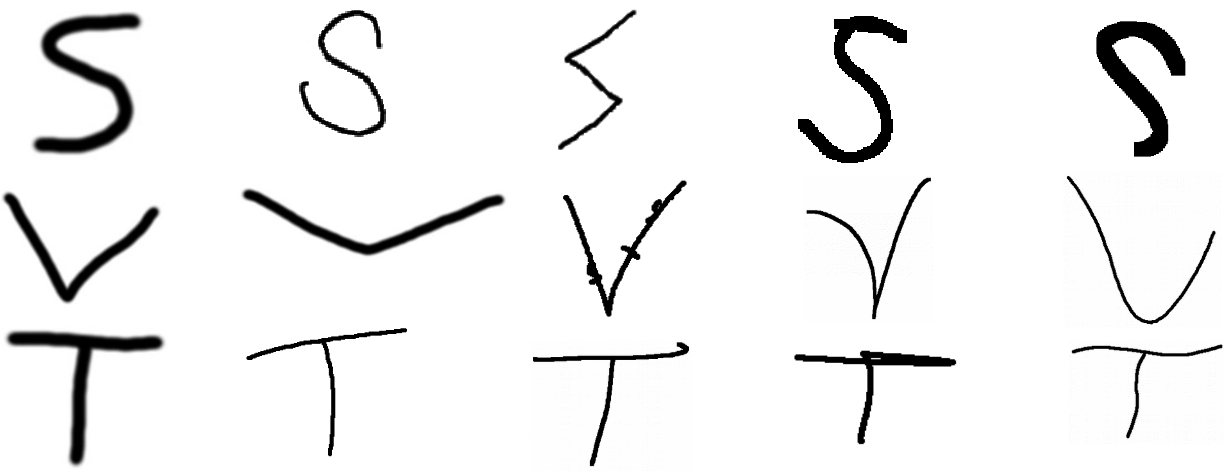
figure 6(left) is the resulting scatter diagram of the training data with respect to the two features. T(**red**) is only strong horizontally, V(**black**) is strong vertically and S(**blue**) in both directions. **figure 7**(right) is the decision boundaries of the 1-nearest neighbour classifier. The colours are the same as figure 6. Despite the fact that S covers the majority of the plot, the classifier is still accurate because the other 2 letters have more distinct characteristics and can more easily score high in either of the features and achieve their respective boundaries.

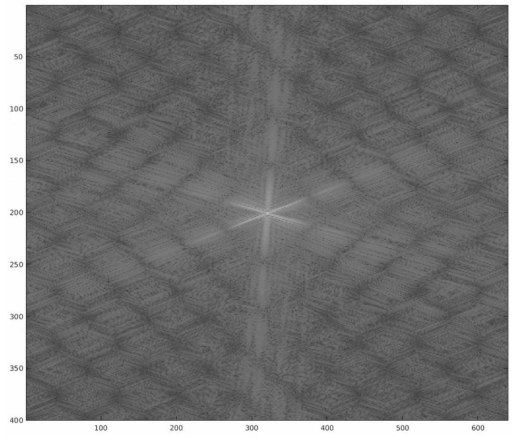
---

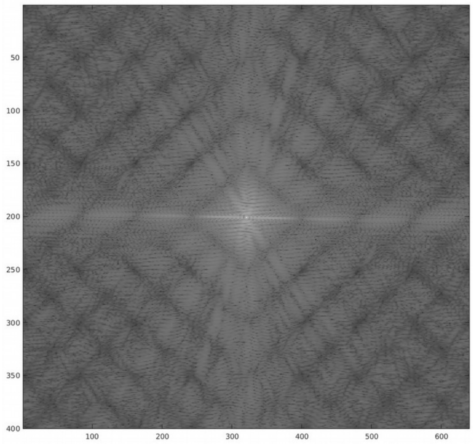
training data nor the test data managed to score higher than that and therefore the figure better visualises the decision boundaries of that area. A grid that covers the whole [0,1] range pushes the black and red boundaries to the very left and bottom of the image respectively and does not clearly demonstrate the area covered by V and T.

Testing the classifier

To test the accuracy of the classifier, new test data were produced from which some were deliberately designed to trick the classifier by mixing characteristics in small amounts that are usually present in other letters. The idea was to emulate how the letters would look like in a sentence, where some letters can sometimes overlap others. 29 out of 30 of the test images were correctly identified. The second test image for V(as shown below) was mistakenly identified as a T due to its very wide angle. More surprisingly, the third test image for S was correctly identified despite the strong diagonal lines which move it towards the area of V. Some of the test images are shown below:

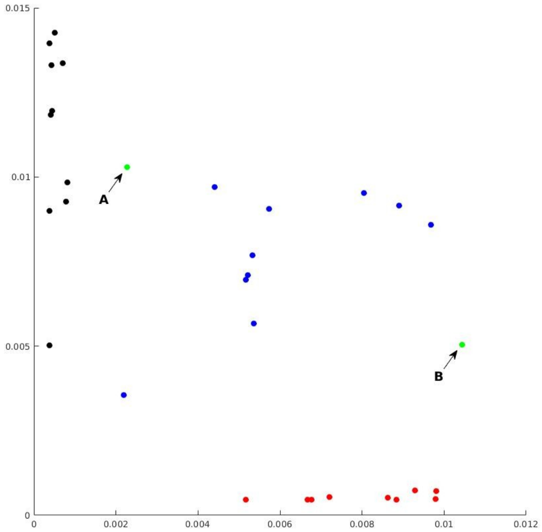




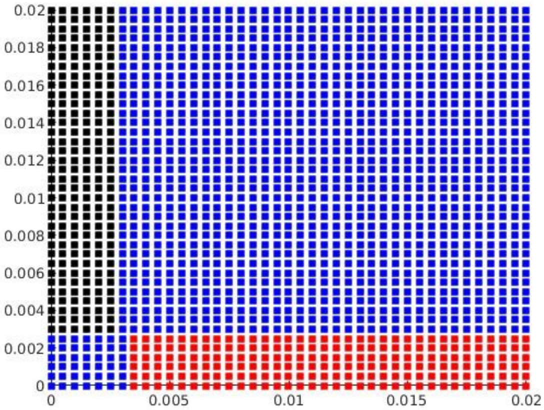


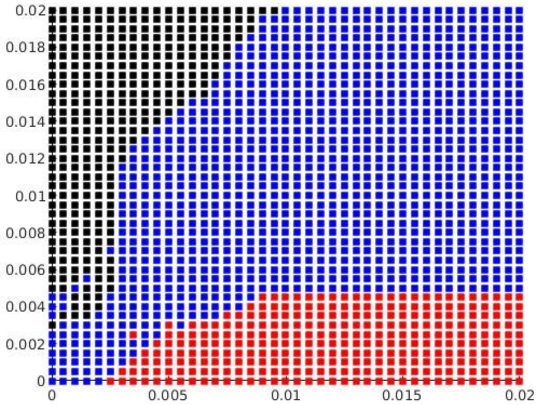
(**figure 8**) The Fourier domains of A and B(left and right respectively). In the first image, it is clear that both features 1 and 2 are strong but due to the absence of a strong horizontal line the image classifies as V. Similarly for the second image, the first feature is strong because of the horizontal line present, but not enough to pull it towards T.

Classifying the letters A and B:



As expected, the letters A and B are classified by the system as V and S respectively. After decomposing A into its frequencies using FFT, it is clear that the two diagonal lines of A are the main characteristic that influenced this result. This is because the same 2 lines are present in all V images caused again by V's diagonal lines. The two diagonal lines significantly increase the total power of the second feature and therefore pull the data point towards V's section of the diagram. Moreover, by observing the Fourier domain of A, it can be seen that the horizontal line of A has also caused a vertical line in its frequency domain signifying the variation in brightness along that axis. This increases the power of feature 1 in which T usually displays a strong reading. However, because feature 1 also takes into account the vertical part of T(horizontal line in Fourier domain), the influence is not strong enough to change the decision since this is not present in A and hence that particular feature scores lower than what it would have scored if it was a T. The presence of the horizontal line in the spatial domain, explains why the data point of A is slightly to the right of the set of V characters. Similarly, B has similar characteristics to S. In particular, because of the smooth angles and constantly changing gradient of its lines, a strong reading is observed all around the centre of its Fourier domain, as it was observed with S. Another important characteristic of B is its straight vertical line which produces a horizontal line in its Fourier domain. As it was the case with A, this characteristic contributes towards T as horizontal lines contribute towards feature one in which images of T usually have a strong reading, but it is not strong enough to pull the decision away from S(this can be clearly observed in the scatter plot, since the data point for B is to the far right of the feature 1/horizonal axis). The frequency domains of each of the images are shown in(**figure 8**).





(**figure 9**):(left)The hard-coded boundaries classifier would have performed better if the boundaries chosen had a slight slope. e.g. the more towards the right(stronger in feature 1) a data point is, it should require more power towards the second feature(upwards) to pull it away from T. (right) The kNN-Average classifier is very similar to the kNN classifier, but it also takes into account the distance of each neighbour.

Alternative classifier:

As it was analysed in the previous sections, the features were selected in a way such that T is strong in one feature but weak in the other; V is strong in the other feature and S is strong in both features. With this in mind, this alternative classifier has hard-coded boundaries/checks that determine whether an image is strong in one of the two features. If the image is strong in the first feature, then it is classified as T, if it is strong in the second feature it is classified as V and otherwise, it is classified as S. The classifier was initially fine tuned to perform as well as possible against the "tuning" data set, but still was not able to achieve the same result as the nearest neighbour classifier when tested against the same test data. This classifier was only able to correctly identify 24 out of 30 test images. For example, as it was the case with the nearest neighbour classifier before, this classifier failed to recognise correctly the second test image for V with the wider angle and additionally also failed to recognise the 4th test image for T which was classified as an S. The reason for the failure is the fact that this classifier is more "strict" in the sense that it requires the power of a feature to pass a certain threshold without taking into account its power in the other feature. Therefore, boundaries with a slight slope are more appropriate than the plain horizontal or vertical boundaries used here. The advantage of this classifier compared to the 1-nearest neighbour is in terms of speed and efficiency. Because there is no need to compare with any training points and search for the closest, it is able to make a decision immediately after the features have been extracted. Because on the fact that 1-NN performed better, another alternative was created with this concept in mind. This classifier firsts takes the 3 nearest neighbours in terms of euclidean distance, calculates the votes for each class and then finds the average distance of the point to be classified to each class based only on the 3 nearest neighbours(a class with no votes is rejected). This classifier achieved a great result with 27 out of 30 test images correctly identified. The parameter k was decided again by trial and error on the "tuning" data set. As parameter k increases, the classifier approaches the boundaries of the nearest centroid classifier, hence this can be though of as the middle ground between kNN and the Nearest centroid classifier. The major advantage of this is that the distance to a neighbour plays an important role in the decision(i.e. The vote of a single neighbour can be stronger than the votes of the other two together). By considering more than 1 neighbour, the classifier is able to generalize slightly better than the 1-NN classifier. However, taking into account the distances can make the classifier more susceptible to outliers because a single outlier can now outvote the other two neighbours and hence work against this goal. The best performing classifier was by far the 1-nearest neighbour since it only failed in a very extreme case of letter V. The decision boundaries for both alternative classifiers are presented in (**figure 9**).