
Feature Extractor for Character Recognition

Pattern Recognition - EQ2340 Assignment 2

Bram Leenders
leenders@kth.se

Deepa Krishnamurthy
deepa@kth.se

October 9, 2015

1 DESIGN

The feature extractor extracts the following features from the drawn character:

- *Scaled X-position*: The X- axis of the handwritten character is scaled to the maximum value of the x-y coordinate of the character figure drawn.
- *Scaled Y-position*: The Y- axis of the handwritten character is also scaled in the same way. Scaling helps in making the character recognition independent of: size of the character and position of the character on the grid.
- *Number of lines*: This feature is crucial to identify characters as it helps in differentiating single stroke and multi stroke characters.
- *Direction*: To differentiate the characters having same number of lines, we consider the direction between the current and previous point in radians. This will help in distinguishing different alignments of various characters. This also enables to take into account the relative position between the end of one pen stroke and the start of the next.
- *Number of lines drawn*: The number of lines drawn so far. The first line has 1, the second 2, etc. So, for a capital T there are some vectors with a 1 as feature value and then some with 2 as value. This might help to get an even better distinction between different lines.

The information given by this feature could also be derived from a sudden change in X/Y positions. Maybe we will remove this feature if experiments with the HMM setup show it doesn't contribute.

Most of the features used are continuous. With the above features, we believe that the HMM should be able to recognize the character patterns quite efficiently.

1.1 VERIFICATION OF THE FEATURE EXTRACTOR WITH DIFFERENT SIZES

In the figures below (1.1, 1.2, 1.3), 3 plots of the features extracted of the character 'C' of different sizes are shown. We can observe that the feature extractor extracts features with significantly similar patterns, irrespective of the sizes of the character drawn. This is mainly checked by scaling the X-Y coordinates.

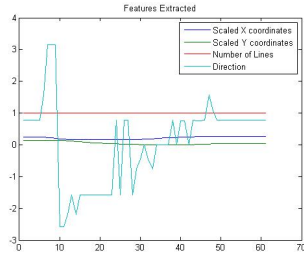


Figure 1.1: C drawn big

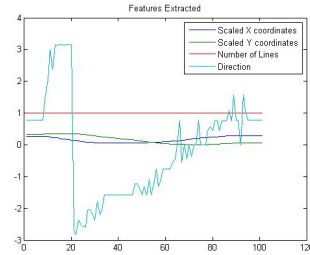


Figure 1.2: C drawn small

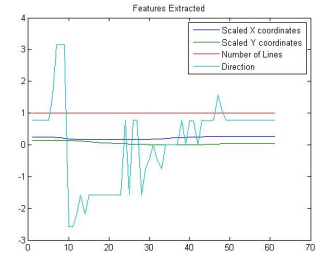


Figure 1.3: C drawn very small

Figure 1.4 below shows the plot of the features extracted for the character 'O'. We can observe that though it is mostly similar to the previous plots, it can be seen that there is a difference in the direction vector at the end where it goes beyond the value 1. In the previous plot for character 'C', it can be seen that this direction vector at the end does not mostly go beyond the value 1. Hence, this attributes to the similarities and differences of the characters 'C' and 'O'.

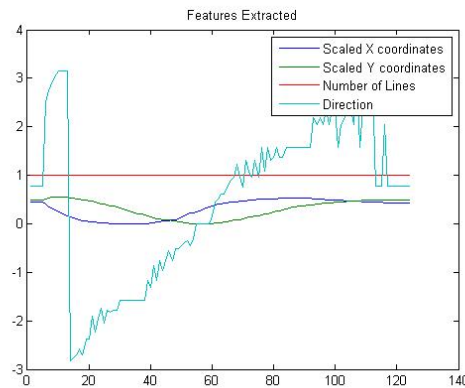


Figure 1.4: Plot of feature vectors for character 'O'

1.2 COMPARISON OF SAME CHARACTER WRITTEN WITH DIFFERENT PEN MOVEMENTS

The output feature vectors should not depend on the movements made while the mouse was not pressed, much like a pen that doesn't touch paper shouldn't have an influence on what character is being written.

To test this, we drew the same character ('T') multiple times, while making random mouse movements before drawing the first line, between drawing the lines and after having drawn the second line. The resulting feature vectors are plotted in figure 1.5 and 1.6.

The difference between these figures is minimal, from which we can conclude that mouse movements while the mouse is not pressed do not affect the resulting feature vectors.

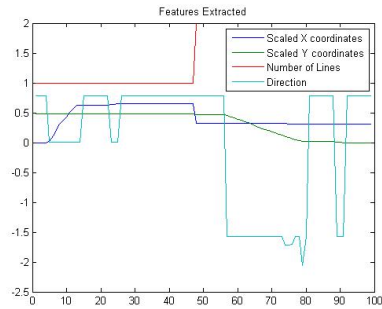


Figure 1.5: First drawing of T

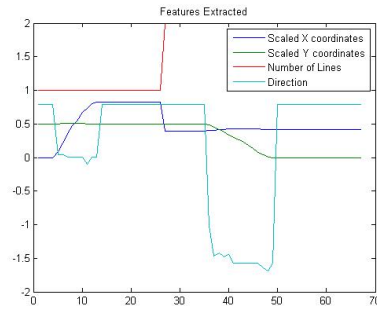


Figure 1.6: Another drawing of T

1.3 DIFFERENCES BETWEEN 'PLUS' AND 'T'

The differences between the character 'plus' and 'T' can be seen in the figures below. There is a variation in the magnitude of the value of y coordinates for 'plus' and 'T', as shown in figure 1.7 and 1.8

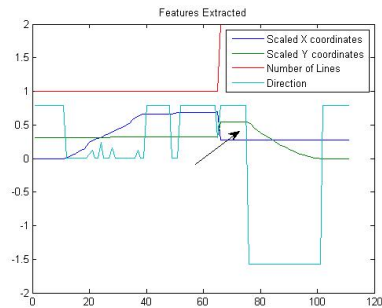


Figure 1.7: Features extracted for '+'

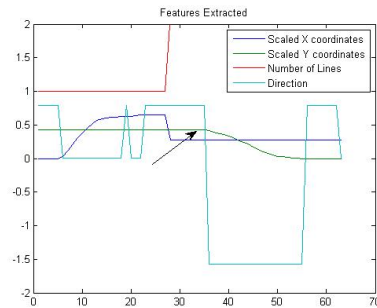


Figure 1.8: Features extracted for 'T'

1.4 ASPECTS THAT ARE NOT COVERED

The feature extractor that is designed is primarily designed for block letters of English alphabets. This could possibly be extended to cursive handwriting. However, for cursive handwriting,

ing recognition, curve fitting could be used to extract curvy features that is predominant in cursive handwriting. The feature extractor could further be made more robust for random strokes generated between 2 pen movements. The extension could possibly contain some pre-processing where the randomness of the strokes can be filtered out.