Chapter 1

Handwriting Recognition Engine

1.1 Radical Recognition Process

1.1.1 Radical Recognition Features

The recognition of a Radical is a search process. The data format of Radicals is described in section **??**. Essentially, a Radical is an ordered collection of strokes. In order to recognise a radical, the sequence of strokes from the database is compared to an input sequence. The measure of similarity is a feature vector that consists of the number of strokes and the similarities between them. Furthermore, permutations of the stroke sequence will be penalised. In order to match two radicals by their feature vectors F_{db} and F_{exp}, the following values are conidered:

$$F := \begin{pmatrix} \text{Total number of strokes} & n_s \\ \text{Position of bounding box} & P \\ \text{Size of bounding box} & S \\ \text{Stroke matching values} & m_{i,j} \\ \text{Stroke sequence} & < s_1, \ldots, s_n > \end{pmatrix}$$

Some of the features need little explanation. The total number of strokes n_s is an integer. In the matching process - if the $n_{s,db}$ and $n_{s,exp}$ differ, the confidence value of the radical recognition value will be lowered. The position of the bounding box $P$is determined by the handle point of the box. The euclidian distance between the P_{db} and P_{exp} serves as a distance measure. The size of the bounding box S can support or refute assumptions about the general shape and size of a radical.

1.1.2 Exploitation of the Stroke Recognition Process

Stroke matching forms the core of the Radical recognition process. Each input stroke is matched against all the strokes of the Radical. A matrix of matching values will be generated by comparing each input stroke to each database stroke for a Radical. The confidence values in the stroke matching matrix M are used to analyse if the input stroke sequence signifies the currently analysed Radical.

$$M_{m,n} = \begin{pmatrix} \mu_{1,1} & \mu_{1,2} & \cdots & \mu_{1,n} \\ \mu_{2,1} & \mu_{2,2} & \cdots & \mu_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ \mu_{m,1} & \mu_{m,2} & \cdots & \mu_{m,n} \end{pmatrix}$$

The matrix M forms the basis of the Radical recognition. The highest confidence value for a Radical will be achieved if

- $m = n$
- the highest matching value $\mu_{i,j}$ for each stroke is above a threshold in order to conclude that it matched with its corresponding stored stroke from the database.
- the highest matching values $\mu_{i,j}$ for all strokes form a diagonal path through the matrix, with $i = j$ for each highest matching value, starting at $\mu_{1,1}$, ending at $\mu_{n,n}$ (and $m = n$).

If two identical stroke sequences are analised, the ideal is fulfilled. Any deviance from these constraints will be penalised and leads to a lower confidence value for the Radical match. The matrix is the source for both the features of the stroke matching values $m_{i,j}$) and the stroke sequence $< s_1, \ldots, s_n >$. A deviance from the diagonal path means deviance from the ideal stroke sequence.

1.2 Character Recognition Process

In the usual case, the strokes do not interfere in their sequence, but there are a view exceptions.

In order to scale to the normalised size, the length of an edge of the bounding box is compared to the length of the normalised character.

1.3 Error Handling

see section **??** in chapter **??** for possible sources of error

1.3.1 Error Recognition

why this section? to demonstrate own achievements of error recognition. the reader should know how it is done technically.

what goes into this section? the aspects of finding errors. finding errors is not a straightforward trivial task - whenever something does not match it is an error - doesn't work like that. instead, firstly, it needs to be made sure that it actually is an error. meaning - not a recognition error, but a user error. secondly, the type of error needs be identified. see section **??** (or handwritten page 58) for sources of error.

how will this section be written? technical - first describe how the error recognition integrates into the recognition process, then how errors are identified.

1.3.2 Error Processing

why this section? actually the 'handling' or 'processing' aspect could be described in the recognition section 1.3.1 as well. so this section is only for a better overview, for document structure, thematically they are the same section. thus they are put together under Error Handling 1.3.

what goes into this section?