CENG 466 Digital Image Processing
Assignment 4
Prepared by:
Asli Gulen 112887-5
Kayhan ince 116246-0

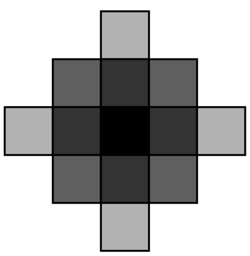
Submitted to:
Turan Yuksel
15.01.2001

REPORT

1. Thinning by morphological operators

Since the backgrounds of our sample images are white (256 in gray-scale), we used dilation operator, which dilates the background and thins the image.
$(f \oplus b)(s,t) = \max\{f(s-x,t-y)+b(x,y) \mid (s-x),(t-y) \in D_f ; (x,y) \in D_b\}$, where b is the structuring element.
To prevent values greater than 255, which appear as the result of dilation operator above, we made a kind of normalization by multiplying each gray value by the coefficient (255/maximum gray value) if maximum gray value is greater than 255.

Here is the structuring element :



Pivot element is the one in the center (darkest one).

2. Feature extraction for classification

We used chain coding for feature extraction because all characters that we are interested in consist of a single line. We thought that chain coding could express such images better than polygonal approximations. However chain coding is very sensitive to noise. Therefore we used median filtering (with a square mask of 3x3) as a pre-processing step in order to reduce the noise.

A knowledge database was necessary for classification. To facilitate a feature database, we extracted chain codes of each class and embedded them in our code.

Here is some implementation detail of chain coding process:

- Enlarge the image by 8 pixels from each side and fill the enlarged parts with 255's (white color). Aim of this step is to prevent getting out of image bounds while searching for next chain.
- Extract the skeleton of the image using Hilditch Algorithm. We used this algorithm because of its property that it does not break the image while extracting the skeleton.
- For each connected component in the image, we found the end points. As we know that each connected component is a single line, it has only two end points except the negligible noises. If one of these starting points is above the other, we selected the lower one. If they are nearly on the same height (height difference is smaller than a threshold), we selected the one on the left. This process selects the end points consistently.
- Find chain code by performing breath first search on the connected component starting from the end point found in the previous step. While searching the pixels, we checked the distance between that pixel and the reference point where first reference point is starting point. If the distance is greater than CHAIN_STEP which is 8 in our program, we calculated the positive angle between the x axis and the line passing through (x,y) and (xRef,yRef). Using this angle, we calculated the corresponding chain code value and assign that point (x,y) as the reference point. We tried 8 and 16 directions for chain codes and obtained better result with 16 directions.
- As chain code of each shape is different in length, we extended chain codes to a fixed length of 100 (CHAIN_LENGTH).

After obtaining a chain code from the image, we compared it with our knowledge database. We calculated the difference between the chain code and the chain code of each class in the database. We did not use the classifier given on the web. We used two approaches for calculating difference:

1. $$\sum_{i=0}^{CHAIN\_LENGTH} \min(absolute(chain_1(i) - chain_2(i)), 16 - absolute(chain_1(i) - chain_2(i)))$$
2. $$\sum_{i=0}^{15} absolute(ratioOfChainValue_1(i) - ratioOfChainValue_2(i))$$

$$ratioOfChainValue(i) = \frac{\text{number of occurrences of chain value i in chain code}}{\text{number of elements in chain code}}$$

We weighted both difference values to obtain a single difference value. We took the class that is closest to our chain code with respect to this difference.