NTU DIP 2020 Spring HW3 Report

B06902002　資工三　沈郁鈞

Problem 1: TEXTURE ANALYSIS

The original image is shown as below:

sample1.jpg

First, perform Law's method with the following 9 micro-structure 3×3 arrays:

$$\frac{1}{36}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$
Laws 1

$$\frac{1}{12}\begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix}$$
Laws 2

$$\frac{1}{12}\begin{bmatrix} -1 & 2 & -1 \\ -2 & 4 & -2 \\ -1 & 2 & -1 \end{bmatrix}$$
Laws 3

$$\frac{1}{12}\begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$
Laws 4

$$\frac{1}{4}\begin{bmatrix} 1 & 0 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & 1 \end{bmatrix}$$
Laws 5

$$\frac{1}{4}\begin{bmatrix} -1 & 2 & -1 \\ 0 & 0 & 0 \\ 1 & -2 & 1 \end{bmatrix}$$
Laws 6

$$\frac{1}{12}\begin{bmatrix} -1 & -2 & -1 \\ 2 & 4 & 2 \\ -1 & -2 & -1 \end{bmatrix}$$
Laws 7

$$\frac{1}{4}\begin{bmatrix} -1 & 0 & 1 \\ 2 & 0 & -2 \\ -1 & 0 & 1 \end{bmatrix}$$
Laws 8

$$\frac{1}{4}\begin{bmatrix} 1 & -2 & 1 \\ -2 & 4 & -2 \\ 1 & -2 & 1 \end{bmatrix}$$
Laws 9

Then, compute 9-dimension energy with original image convolving with each arrays above using the following formula (window size = 19):

$T_i(j,k) = \sum\sum \left|M_i(j+m,k+n)\right|^2$ $(0 \leq m, n \leq w$, $w =$window size)

Next, choose 3 self-specified initial point $(352, 500)$, $(653, 500)$, $(51, 500)$, which represents the zebra, grassland and sky.

After setting initial points as 3 centers, perform k-means clustering with **Euclidean distance**, update 3 centers and record labels for each iteration. Keep performing 10 iteration, and the result images is as follows:

`sample1_label_final.jpg`

> Note: Black represents zebra, gray represents grassland, and white represents sky.

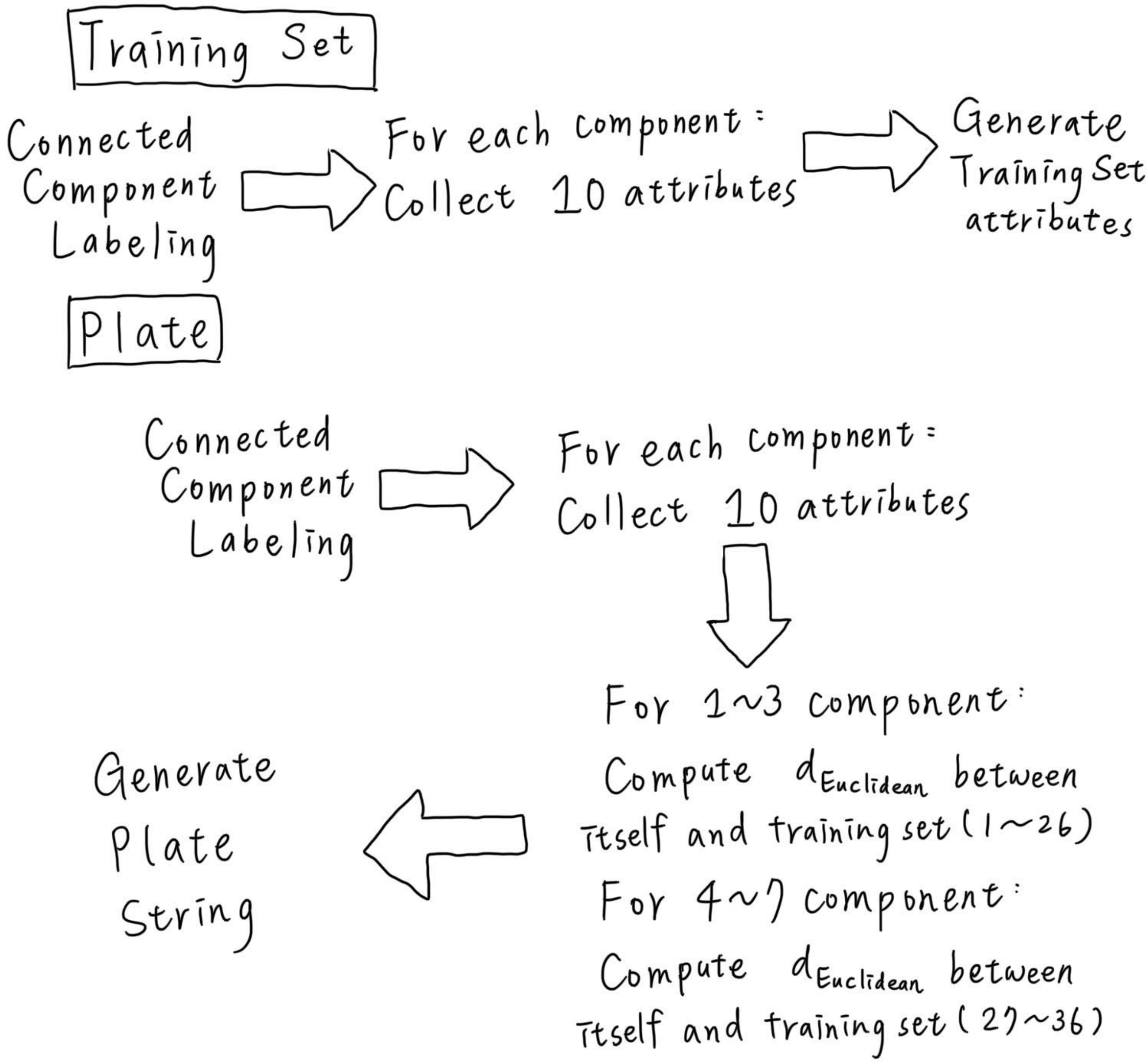
Here I tried 3 different window size $w = 13, 19, 31$, with other conditions fixed:

		
$w = 13$	$w = 19$	$w = 31$

As results above, when windows size gets larger, less white stripes on zebra would be misinterpreted. On the other hand, larger area of grass (especially the right-bottom corner) would be recognized as part of zebra.

Problem 2: SHAPE ANALYSIS

Flowchart



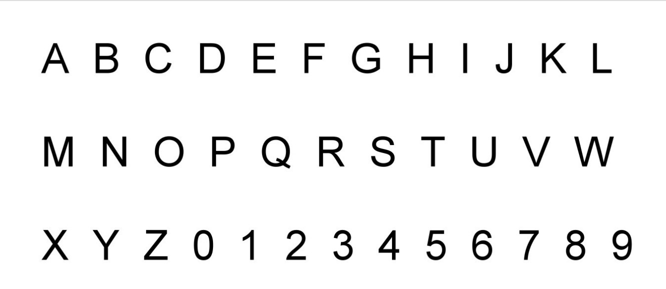
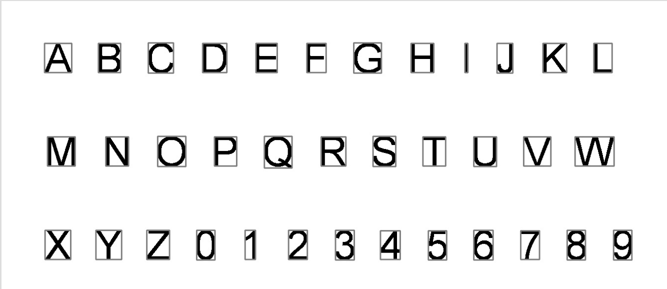
Training Set

Steps:

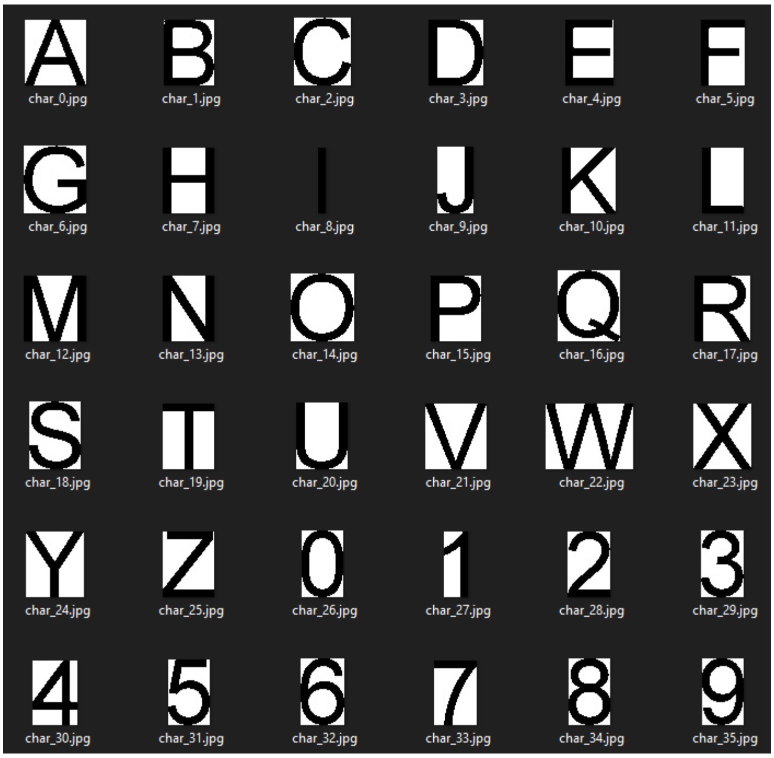
1. **Binarize the** `TrainingSet.jpg` **.**
2. **Find the connected components in result of step 1 using 4-connectivity.**

> **Note: There are 36 characters in** `TrainingSet.jpg` **, thus 36 segmented images are saved in directory** `train_segments` **. (A~Z and 0~9 in order)**

The following shows the original `TrainingSet` **and labeled with bounding box.**

	
`TrainingSet.jpg`	`TrainingSet_boxes.jpg`

And the following is the list of segments:



3. **Compute 10 attributes of each segments.**

> **Note: The first 5 attributes are counting number of convexes with 5 vertical line $i = 0$, $i = \frac{2w}{5}$, $i = \frac{w}{2}$, $i = \frac{3w}{5}$, $i = w$, and the remaining 5 attributes are counting number of convexes with 5 horizontal line $j = 0$, $j = \frac{2h}{5}$, $j = \frac{h}{2}$, $j = \frac{3h}{5}$, $j = h$ (h, w are height, width respectively.)**

Plate

Steps:

1. Binarize the `Sample{index}.jpg` . If input image has white words with black background ($P(black) > P(white)$, occurs in `sample4.jpg`), then invert the binarized image.

2. Find the connected components in result of step 1 (using 4-connectivity). Here we set threshold of pixel intensity to filter unnecessary components ($T_L = 300, T_H = 5000$) and check the boundary of boxes is reasonable. (If the area of the bounding box is larger than 25% of image area, then discard it.)

The following shows the original plate and the labeled binarized plate.

`Sample2.jpg`

`Sample2_binary.jpg`

`Sample3.jpg`

`Sample3_binary.jpg`

`Sample4.jpg`

`Sample4_binary.jpg`

3. Compute 10 attributes of each components.

> The definition is similar as above. However, for vertical line, I choose $i = 3$ and $i = w - 3$ instead of $i = 0$ and $i = w$, and for horizontal line, I choose $j = 3$ and $j = h - 3$ instead of $j = 0$ and $j = h$. The Reason is that each components is slightly tilted, so I decided to ignored the bezels (thickness = 3)

4. Referred to the latest standard plate format, for the first 3 components, compute the **Euclidean distance** to the training set 1~26, and for the first 3 components, compute the **Euclidean distance** to the training set 27~36. After computing distance to each segments, find the $argmin(d_{Euc})$ so that we could determine whether the character is.

Execution Result

Visually, the three plates are “AGB8888”, “EMB0588”, “EWA5588” respectively. Below is the execution result:

```
[Problem 2]
        Processing TrainingSet.jpg

        Processing sample2.jpg
                      RGB8888
        Processing sample3.jpg
                      EHB0589
        Processing sample4.jpg
                      ZUK5588
```

Here is a table in comparison with execution result and reality:

Success/All	`sample2.jpg`	`sample3.jpg`	`sample4.jpg`	**Overall Accuracy**
Alphabet	$2/3$	$2/3$	$0/3$	44.4%
Digit	$4/4$	$3/4$	$4/4$	91.7%
Overall Accuracy	86.7%	71.4%	57.1%	71.4%

Conclusion

From the table above, we find that it could recognize dit effectively with high accuracy. On the other hand, the alphabet could not be well-recognized. I think one of the critical reason is that the training set uses proportional font, which the width of character may vary. And, the plate uses monospaced font (https://en.wikipedia.org/wiki/Monospaced_font). Besides, the font on the plate is thick to be easily visually-recognized from long distance (For instance, M might be similar to H). In my opinion, to improve accuracy, changing training set with other monospaced font might be a choice (though it is sometimes impossible). In addition, some preprocessing to each component could be applied to have better accuracy.