First Assignment Report

Tianye Zhao
Zhaoxuan Qin

2019/2/7

Clarification

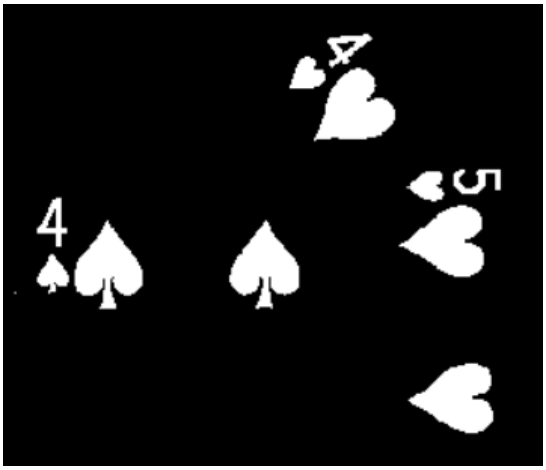
This assignment aims at realizing the experience with components analysis, morphological filters and therefore the use of options computed inside it for recognition of objects. The main objective of the program:

1. Convert the input grayscale image into a binary image.
2. Accomplish preprocessing on image.
3. Through preprocessed image, count objects and get basic features.
4. Collect all required features and find the similar objects.

This program is designed for Binary Image Analysis. Techniques such as connected components analysis, morphological filters and feature extraction are implemented.

Contribution

This program is implemented in Python with OpenCV. In order to convert the original image to binary image, we used the threshold function with Otsu's binary to find optimal threshold for each image. The result is shown in `Figure 1`, it shows the result of all images.



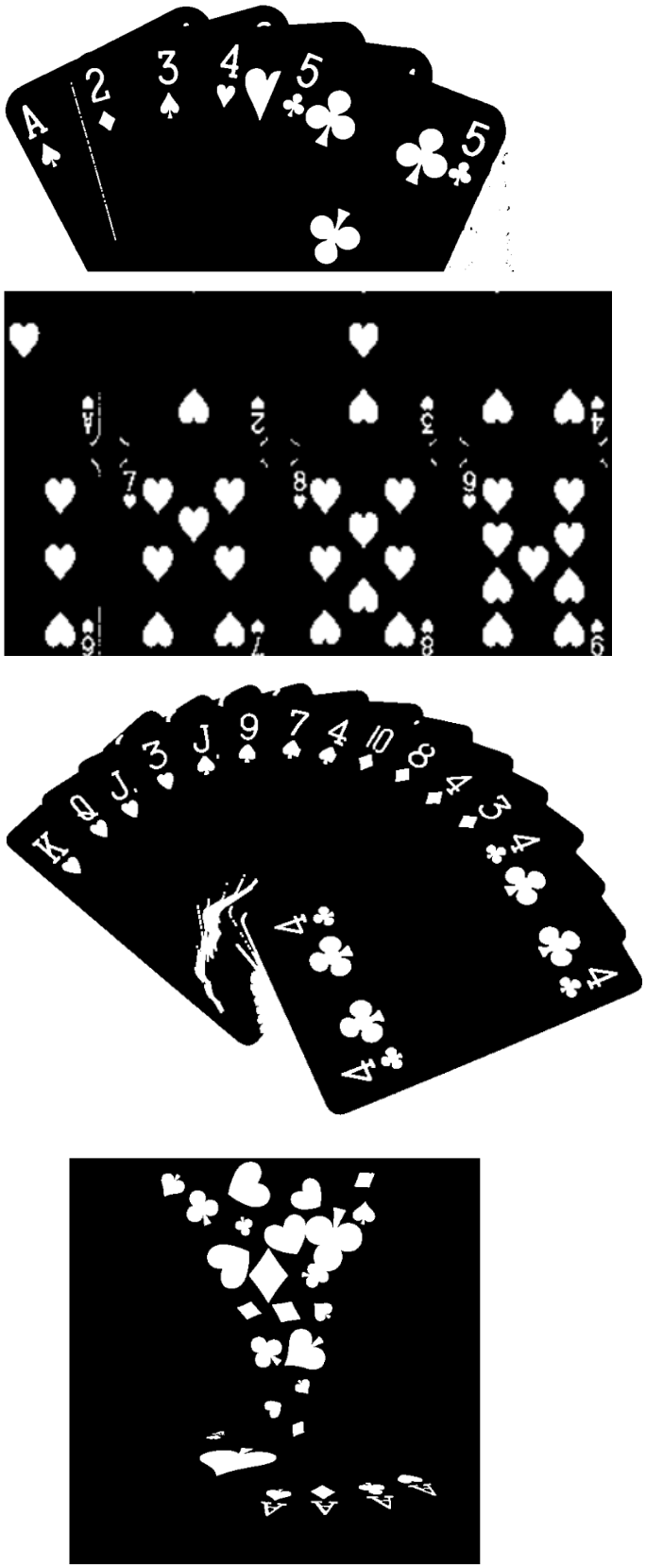


Figure 1 Binary images after preprocessing

Explanation

The workflow follows as original image, binary image, connected components, feature collection and group similar shape.

The Canny algorithm is a significant method to detect the edge of the image, the Figure 2 shows the edge detection of the image 5. With this edge information, it is easy to perform object segmentation with low illumination.

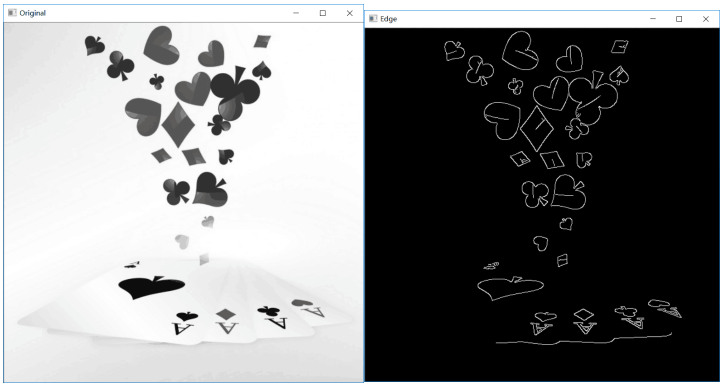
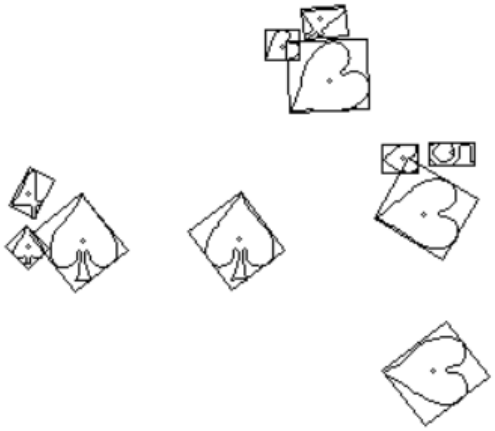
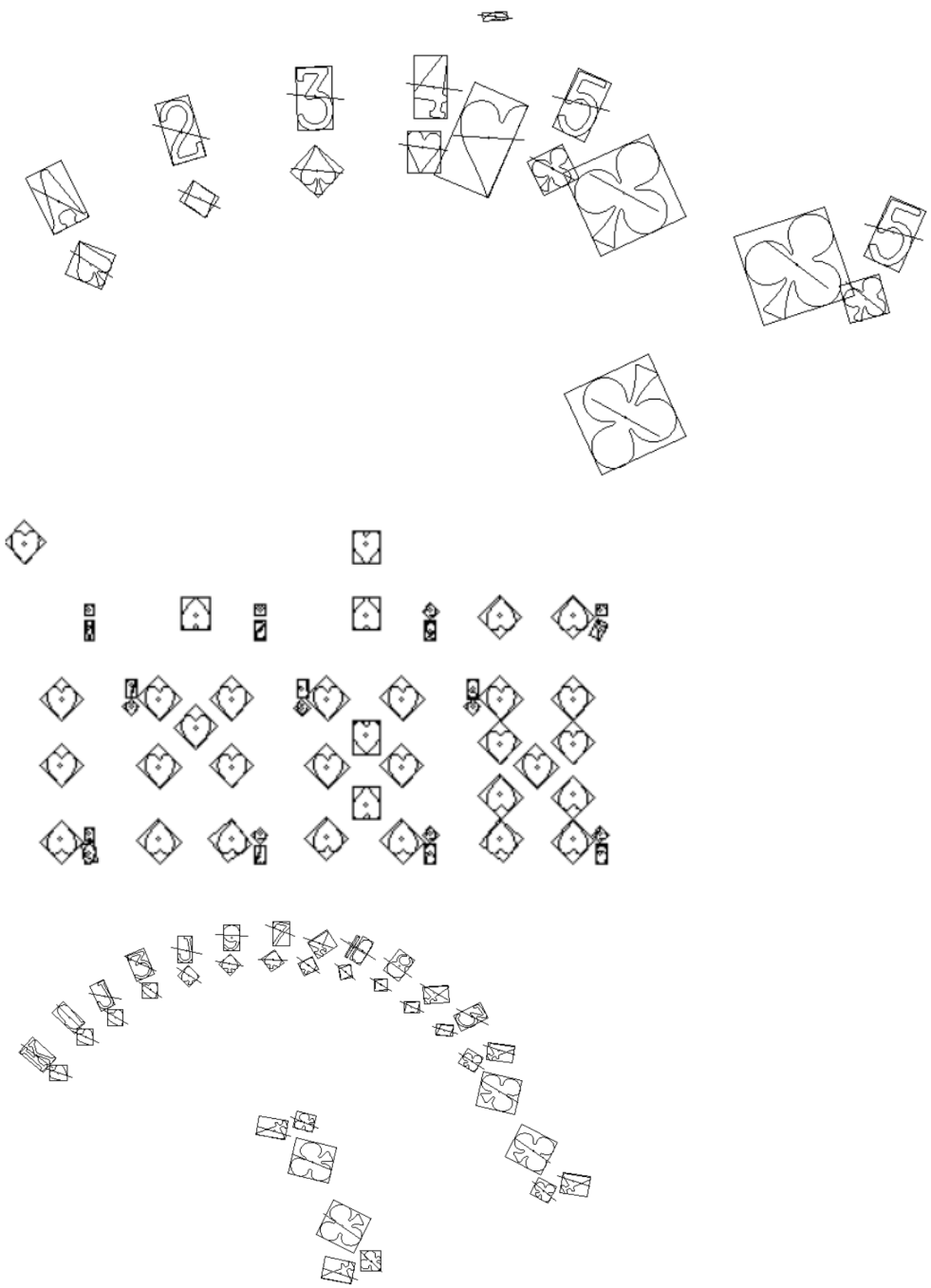


Figure 2 Canny edge detection on image 5

In order to recognize the objects in the image, we used the connected components algorithm, it has been implemented in *test.py* file and has validated on image 1. During implementation stage, it isn't being used since OpenCV has this function. The Figure 3 shows the objects in the images.





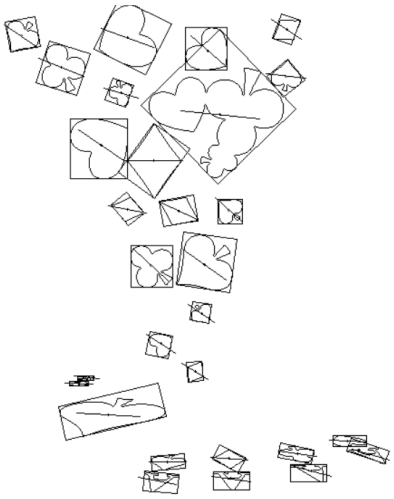


Figure 3 Objects detected on each image

The last step is to report the features as Figure 1.12 written in book and group similar objects according to Hu Moments. The program will give the result as following. Here is all the information of image 5 as an example.

Object	Area	Perimeter	Circularity1	Circularity2	Centroid	mu_rr	mu_rc	mu_cc	Bounding Box	Rotated Bounding Box	Similar Obj
1	2992	227.1	17.2	5.58	(273.6, 41.8)	208.74	40.07	300.5	[(242, 5), (306, 74)]	[[293, 82], [336, 54], [264, -2], [322, 26]]	[4, 9, 11, 17, 18]
2	894	143.6	23.1	5.52	(156.3, 39.7)	77.63	10.95	68.12	[(140, 23), (176, 58)]	[[144, 57], [138, 24], [171, 18], [177, 51]]	[6, 13, 14, 16]
3	510	93.7	17.2	5.84	(449.0, 32.6)	46.46	-12.95	36.43	[(434, 22), (465, 45)]	[[454, 49], [433, 42], [443, 15], [464, 22]]	[10, 11, 12, 18]
4	1601	162.8	16.6	5.64	(362.9, 56.5)	130.27	-32.51	138.28	[(337, 30), (384, 78)]	[[337, 77], [337, 30], [383, 30], [383, 77]]	[1, 9, 11, 17, 18]
5	1629	214.2	28.2	4.71	(200.8, 74.9)	133.22	6.65	146.97	[(179, 49), (227, 102)]	[[219, 104], [174, 92], [186, 45], [231, 57]]	[8, 15]
6	704	119.3	20.2	6.4	(448.5, 84.3)	65.18	1.98	47.74	[(432, 66), (466, 101)]	[[445, 107], [427, 84], [450, 65], [469, 88]]	[2, 13, 14, 16]
7	8559	682	54.3	2.83	(375.9, 126.8)	1292.82	65.77	653.48	[(297, 75), (446, 197)]	[[373, 203], [288, 116], [375, 32], [459, 120]]	[]
8	472	107.3	24.4	5.31	(266.0, 102.7)	36.36	1.15	40.35	[(253, 89), (279, 118)]	[[271, 118], [248, 110], [257, 86], [280, 94]]	[5, 15]
9	3051	226.3	16.8	5.52	(244.5, 160.4)	235.61	59.39	284.57	[(210, 131), (274, 198)]	[[210, 197], [210, 131], [273, 131], [273, 197]]	[1, 4, 11, 17, 18]
10	2423	212.4	18.6	5.52	(302.8, 177.6)	141.29	-0.75	275.53	[(274, 137), (333, 219)]	[[303, 218], [263, 190], [302, 136], [342, 164]]	[3, 11, 12, 18]

In conclusion, we had learned the flows of connected components algorithm and how to manipulate binary image with OpenCV library. And we also found it is difficult to create one general preprocessing method which would work on all images, we hope we can learn new methods to handle this.