An Image Processing Application on QuickCog and Matlab

"Door-Key Recognition System"

Lei Yang

Oct, 2009

Prof. Dr.-Ing. Andreas König

Overview

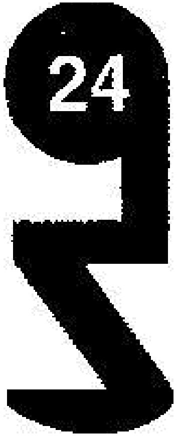
1. Introduction
   - *Motivation*

2. Parts of the Project
   - *Methods used in the project*
   - *Results and comparison*

3. Conclusion

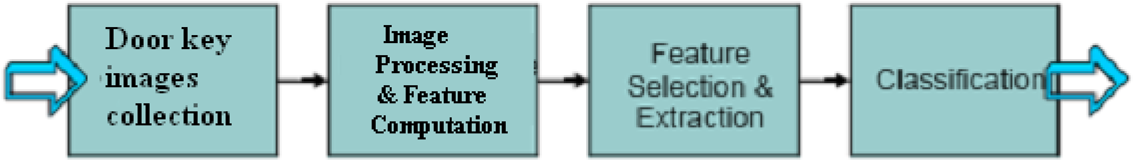
Motivation

- Nowadays in the service stations for key purchase, keys are still identified by human operators with the help of data sheets/catalogues. My project targets on automating this problem, designing and implementing an intelligent vision system.



Parts of Project

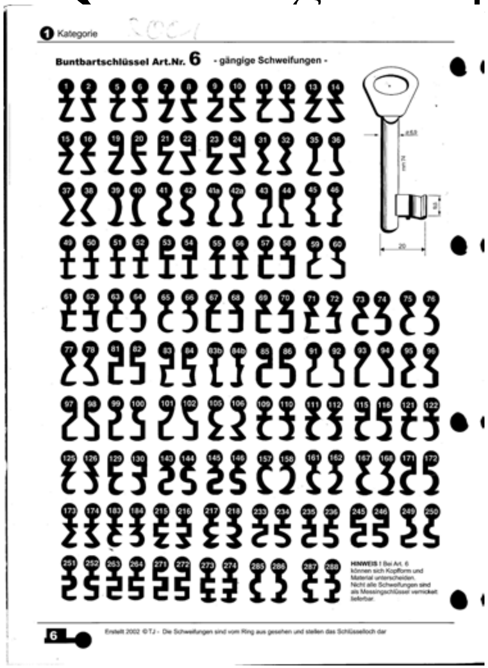
- My project is key recognition by image processing using QuickCog.
- In this project, I built a program according to the following steps. It will train the system and do the test job.


Door key images collection
Image Processing & Feature Computation
Feature Selection & Extraction
Classification

- Since the large data base, a template matching method is also used.

1. Image Acquisition (1)

- First, I scanned the brochure of the key-models, cut them into single models (91*234 pixels)and classified them in QuickCog Stichprobeneditor.



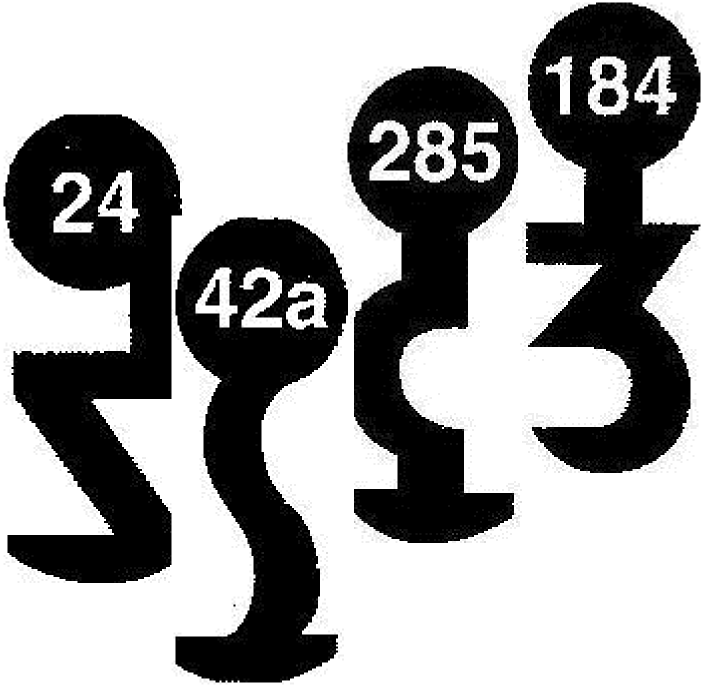
1 Kategorie

Buntbartschlüssel Art.Nr. 6 - gängige Schweifungen -

HINWEIS ! Bei Art. 6 können sich Kopfform und Material unterscheiden. Nicht alle Schweifungen sind als Messingschlüssel vernickelt lieferbar.

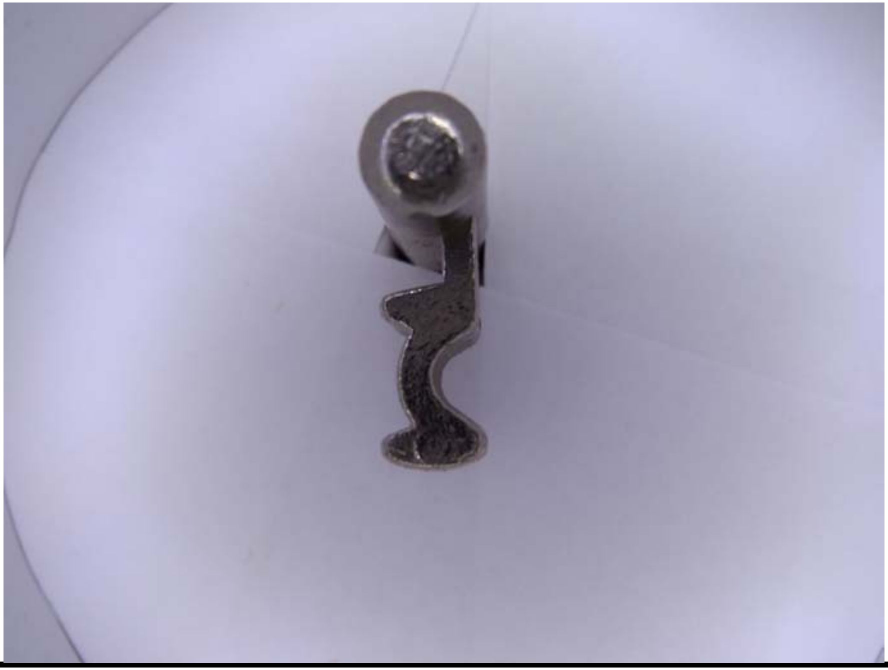
6

Erstellt 2002 ©TJ - Die Schweifungen sind vom Ring aus gesehen und stellen das Schlüsselloch dar



1. Image Acquisition (2)

- For the real keys, the origin pictures are taken under the LED lights. I made a whole in a white paper, and got the key through the hole, and then put the camera inside the LED lights to take the pictures.



2. Image Processing (1)

- **Reason:**

  *The key models have the number on them which may be taken as the wrong feature.

  *The key pictures taken by camera will bring some noise

- **Method:**

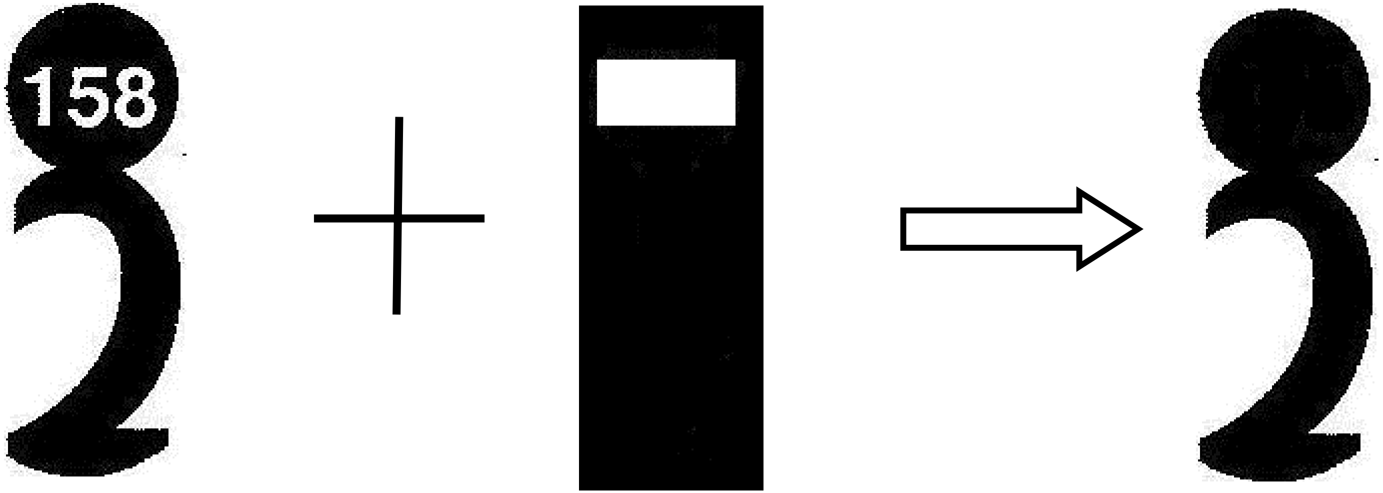
  *Let the original pictures subtract a mask to get rid of the numbers.

  *Apply "opening" block with a 3*3 morph to reduce the noise.

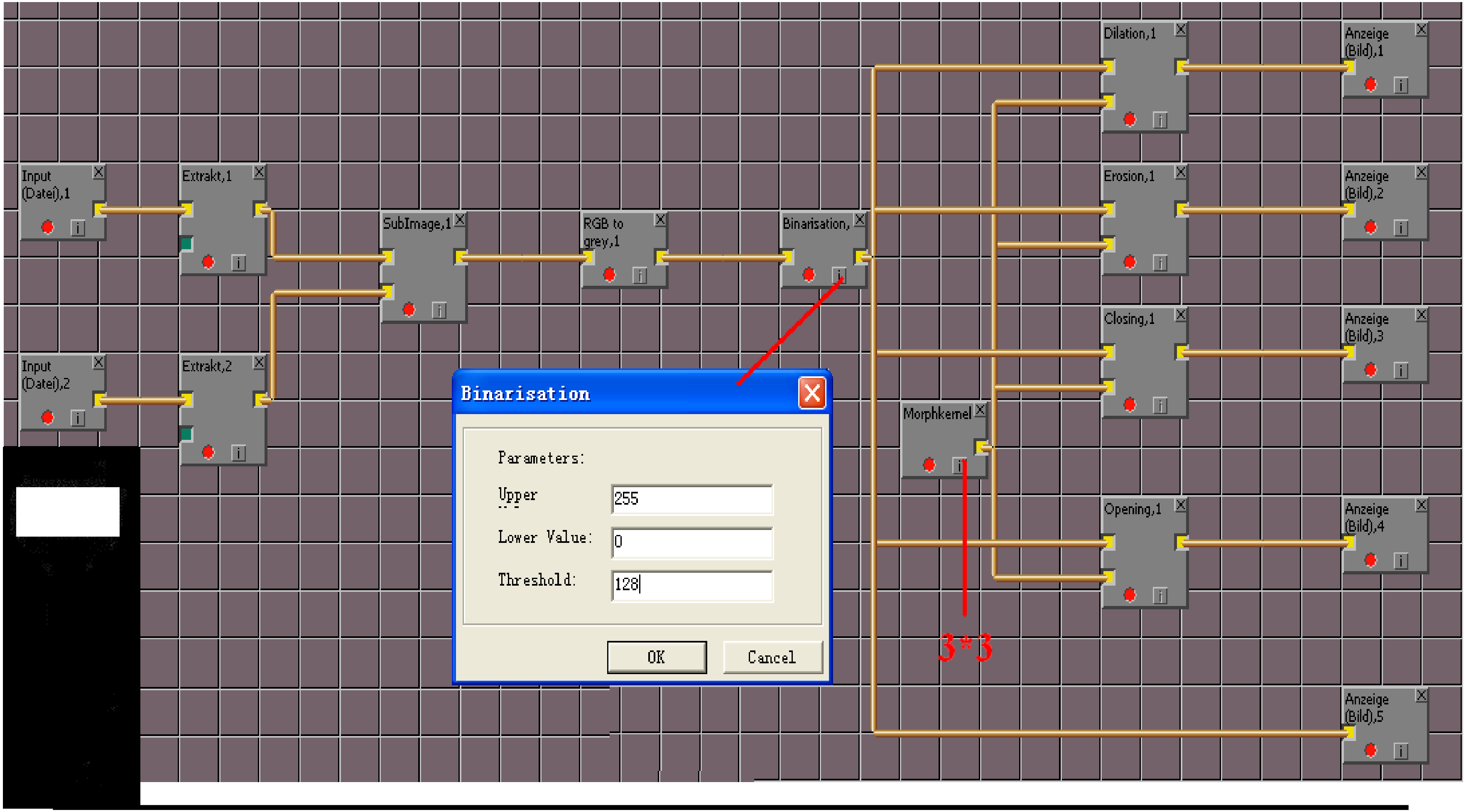
2. Image Processing (2)

- Modify the copy models

  In order to get rid of the numbers, I added a mask to envelop the number area and use some image processing algorithm to get a perfect picture without the numbers.



2. Image Processing (3)



Binarisation

Parameters:

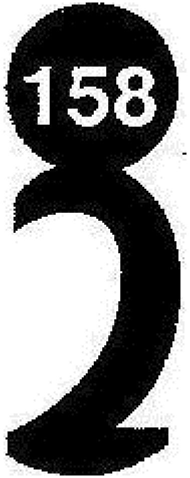
Upper: 255

Lower Value: 0

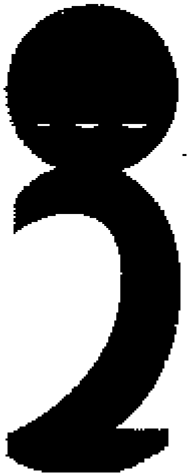
Threshold: 128

2. Image Processing (4)

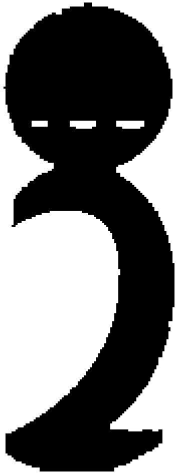
- Pictures Test for copy model



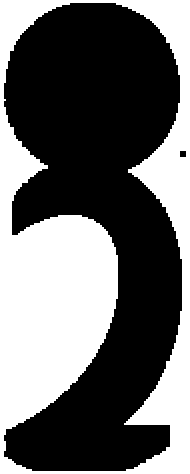
Original



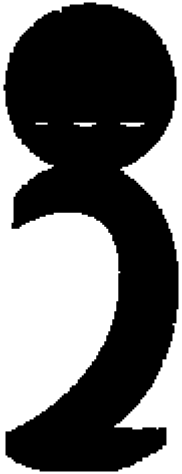
Binary



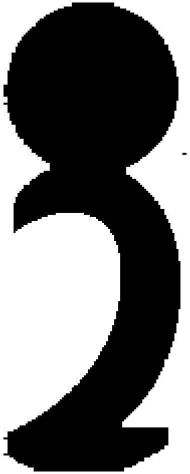
Dilation



Erosion



Closing

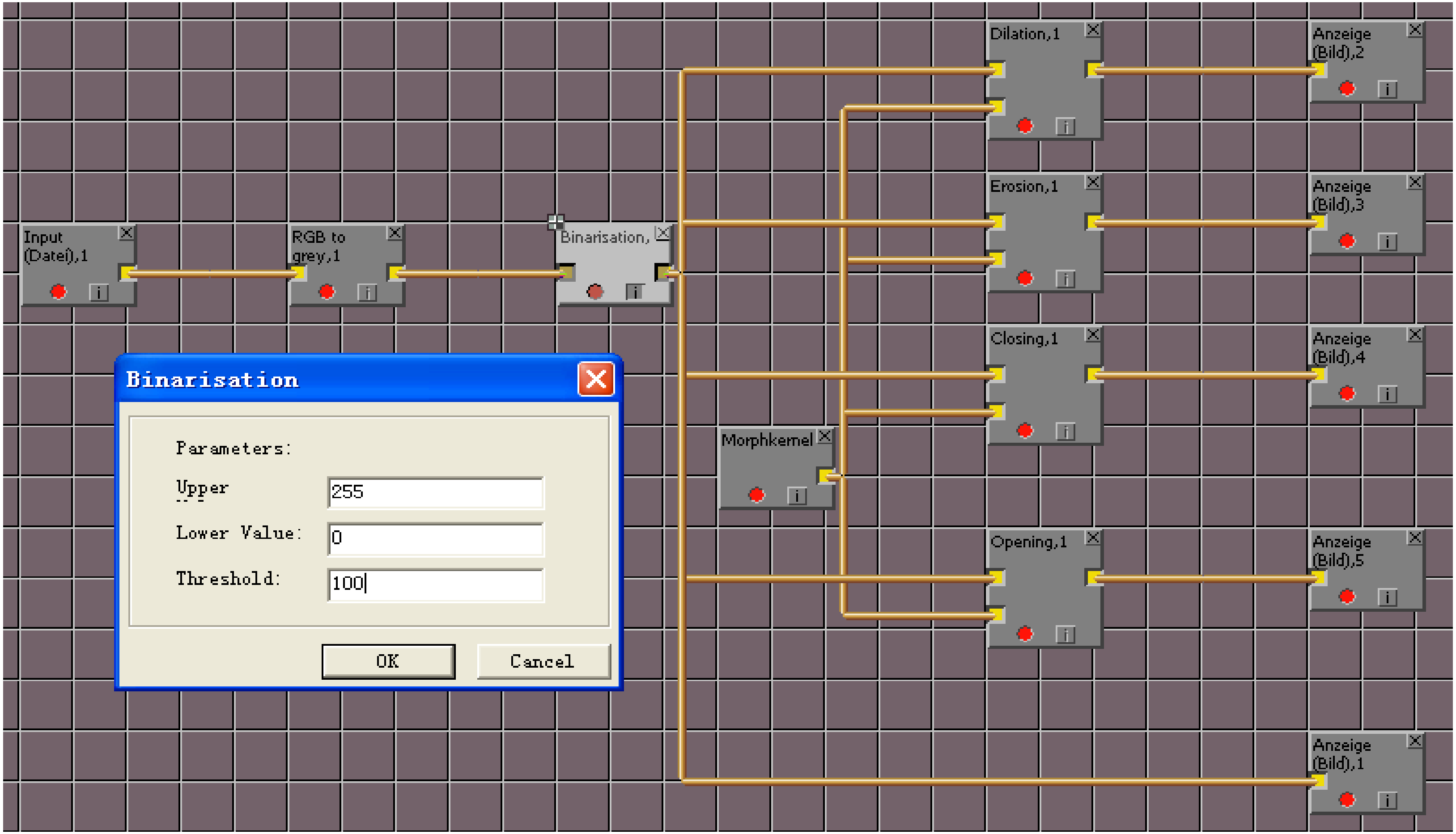


Opening

2. Image Processing (5)

- Modify the key pictures taken by camera
  - * Previous picture modify by hand (scale changed)
  - * More noise
  - * Without numbers
  - * The threshold value is choosen between 70-120 depends on the test results.
  - * Different algorithm according to the picture test result.

2. Image Processing (6)



Binarisation

Parameters:

Upper: 255

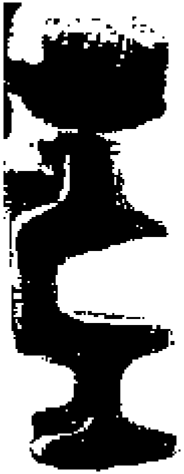
Lower Value: 0

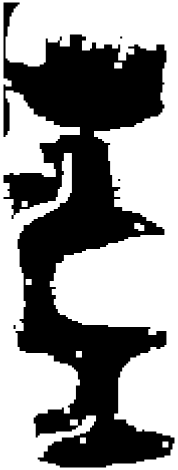
Threshold: 100

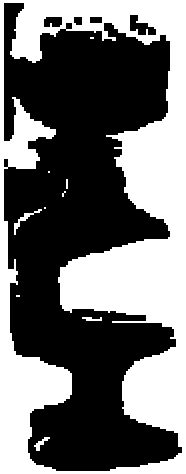
2. Image Processing (7)

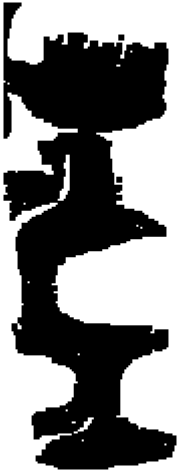
- Pictures Test for real keys

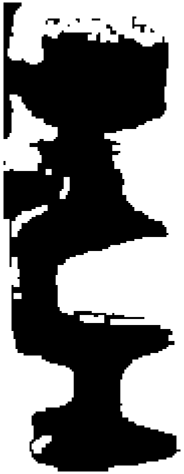
Original


Binary


Dilation


Erosion

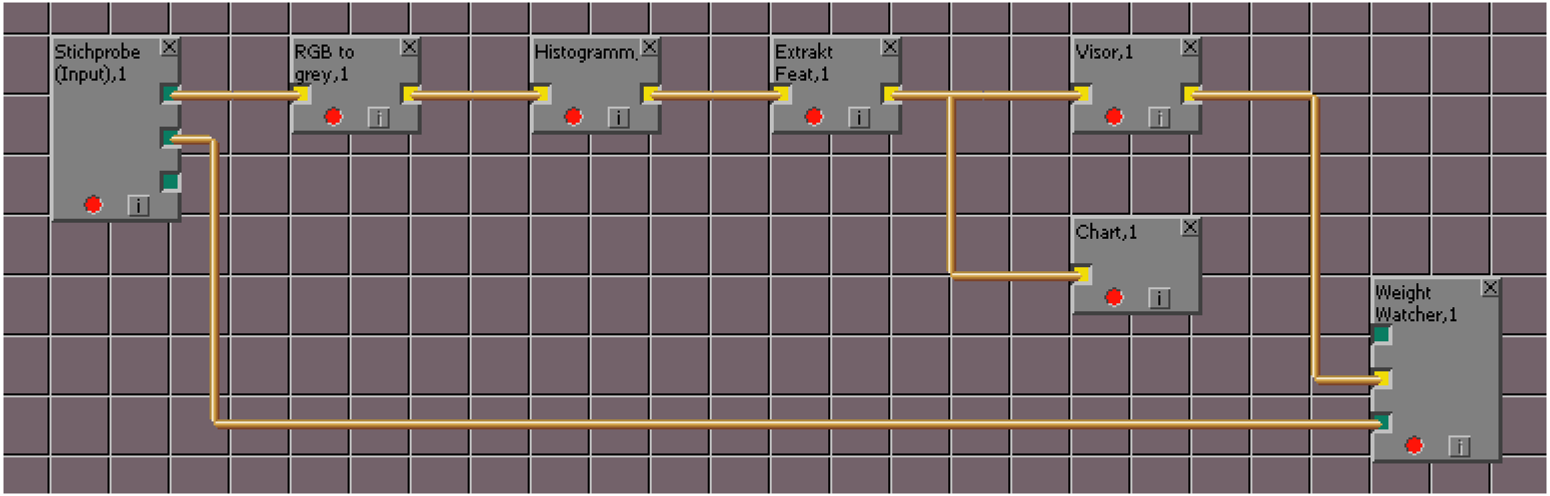

Closing


Opening

3. Feature Extract and Classification (1)

- Feature Extract

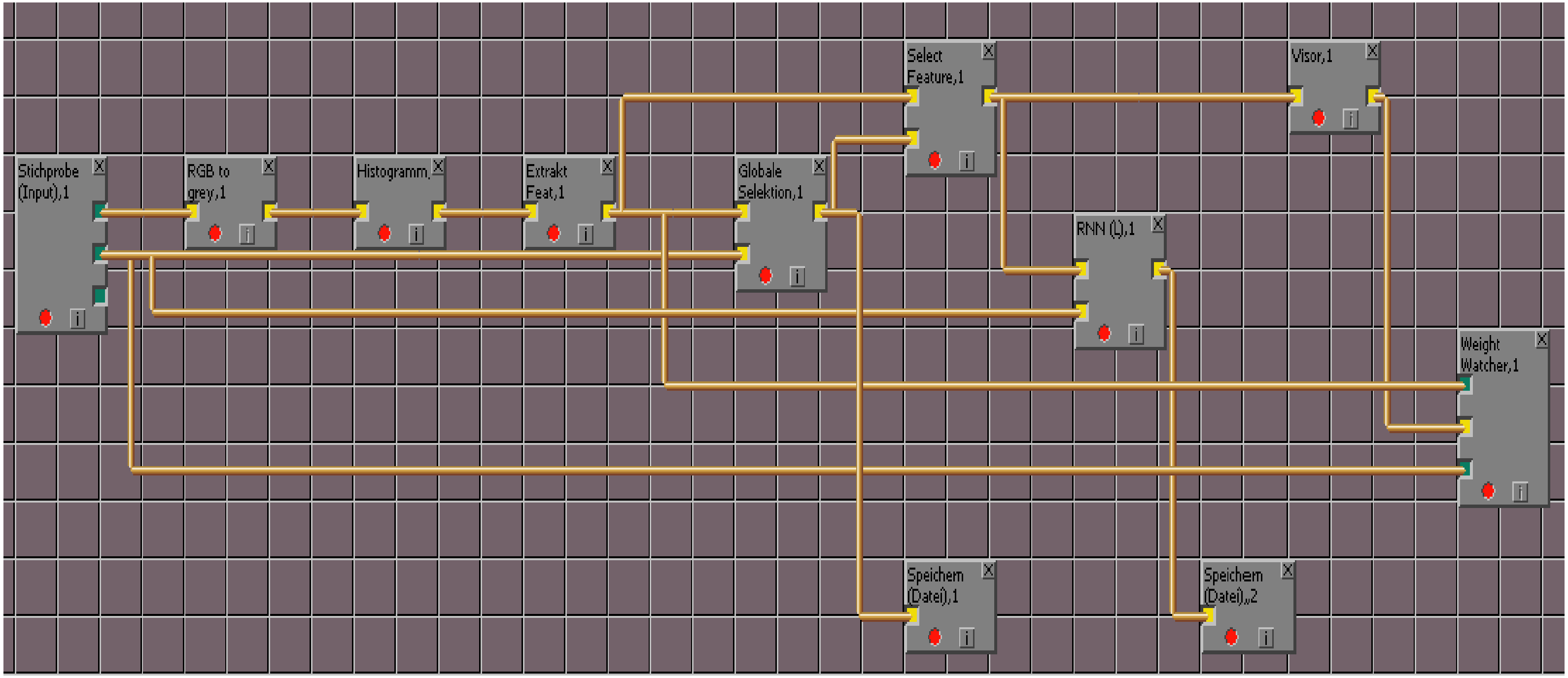
  After data aquisition, we used histogram in order to get some statistics about the features of the data and selected the features in the range of 1-19 and 245-253



3. Feature Extract and Classification (2)

- Feature selection

In this part, dimension reduction methods is used to select the best feature which can separate all the classes.

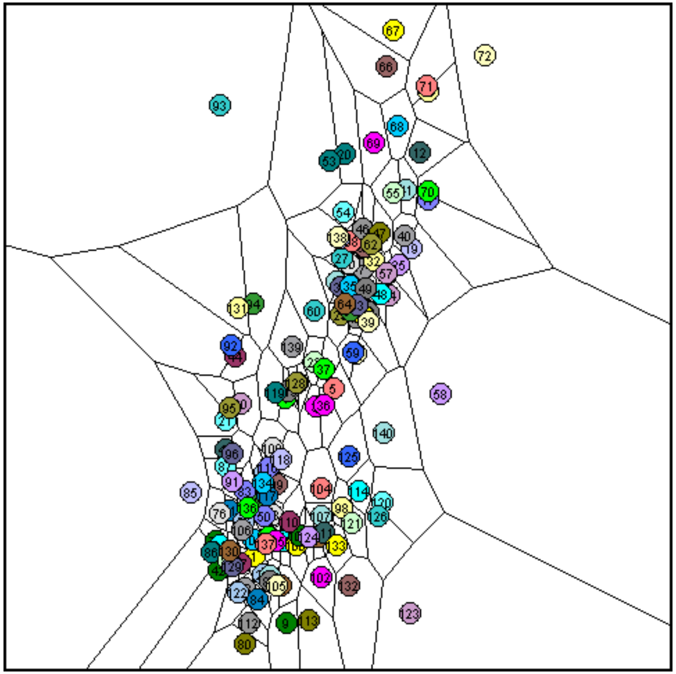


3. Feature Extract and Classification (3)

- Classification (1)

Three methods are used in this part: RNN, k-NN and PNN, the results are compared in order to choose the best method.

Included the image processing part, they compose the hole training part.

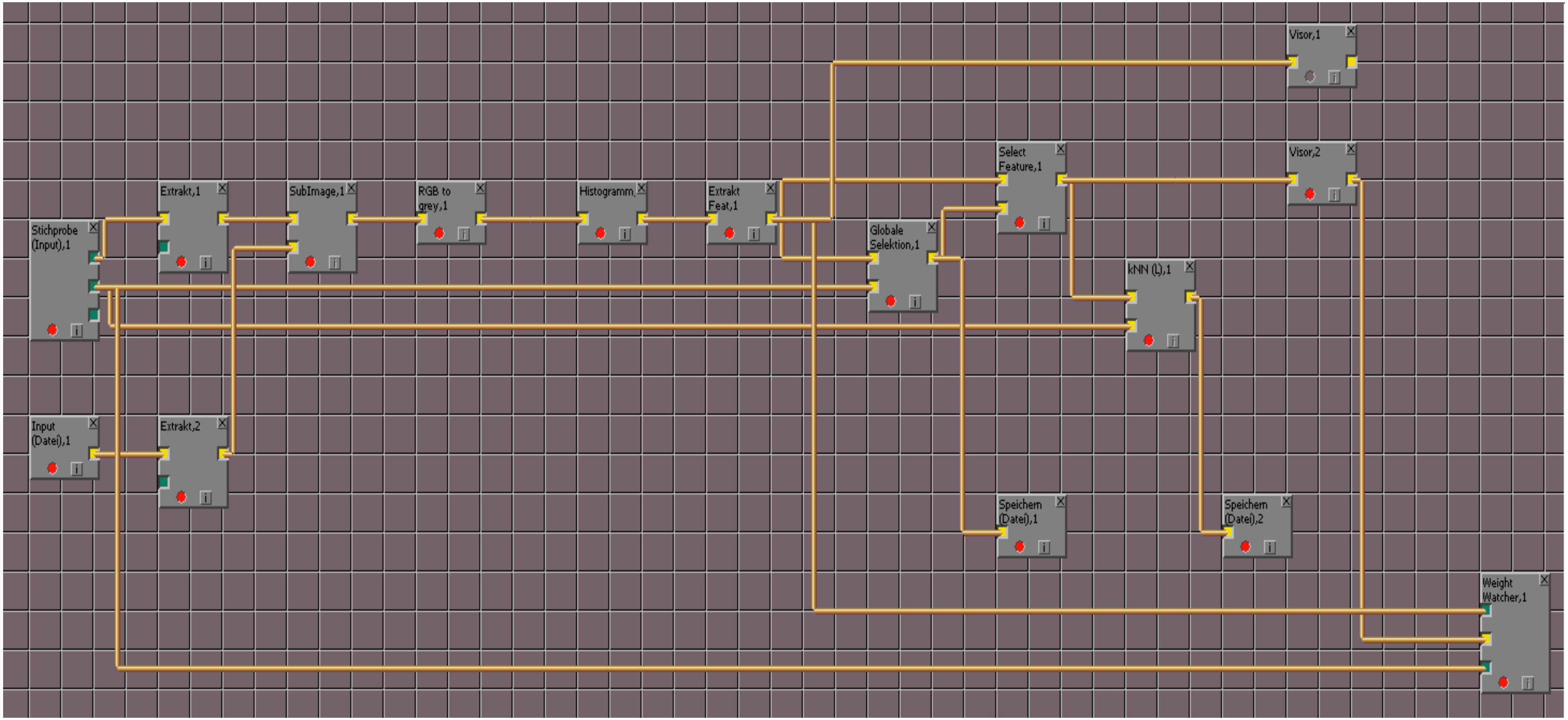


The training result is as follows:

- Train:
- Best Quality: 1.00000
- Best Features: 13 14
- keine Fehler, 0 Warnung(en)

3. Feature Extract and Classification (4)

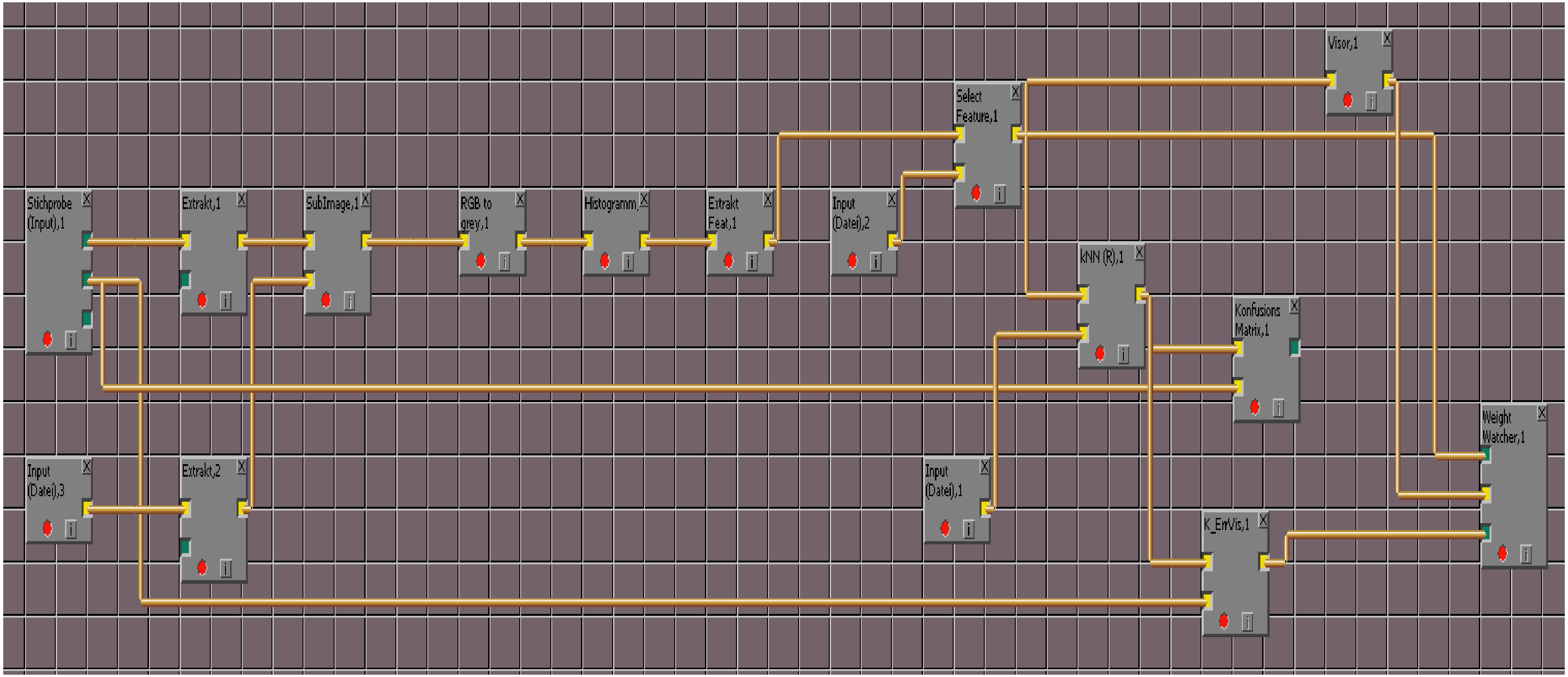
- Classification (2)



3. Feature Extract and Classification (5)

- Test result (Quickcog)

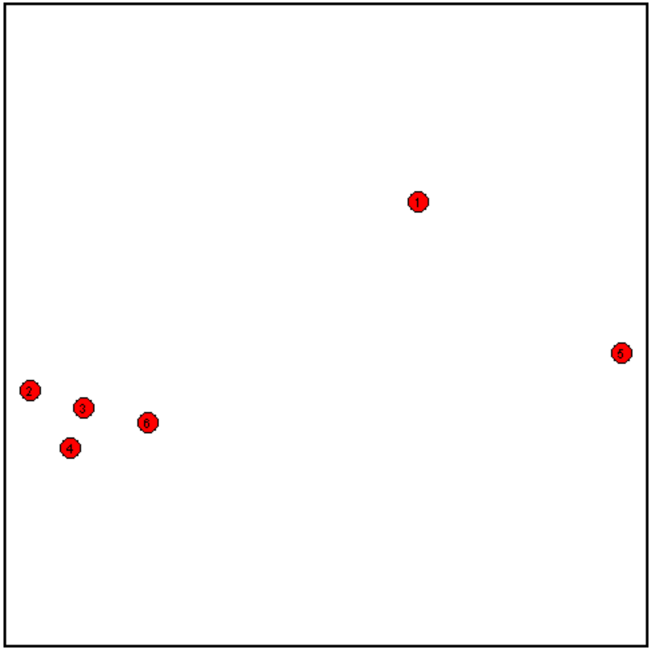
Several test are made with the help of the training data

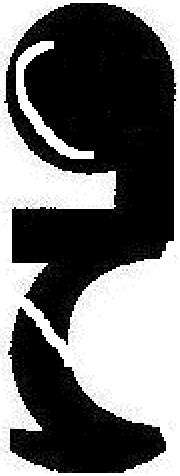


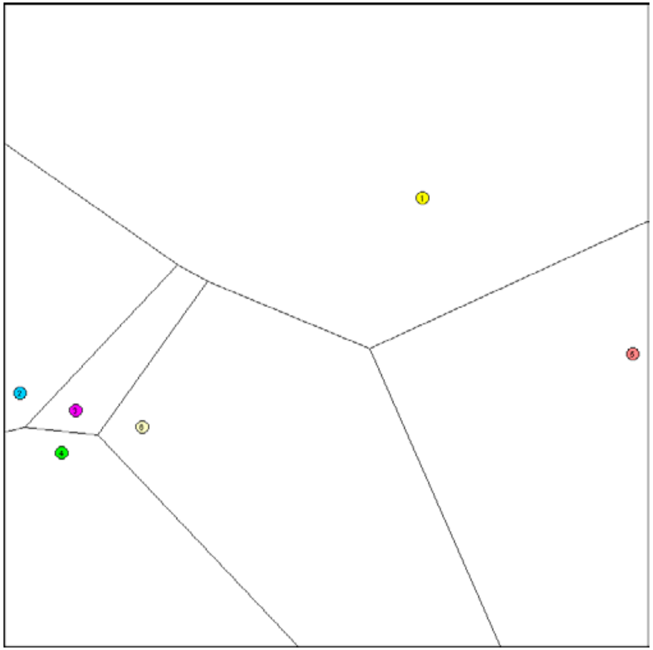
3. Feature Extract and Classification (6)

- Test result (with test pictures)

Since we have too many classes, the test result is not so good. I also reduced the test base, but the result is also not satisfied.





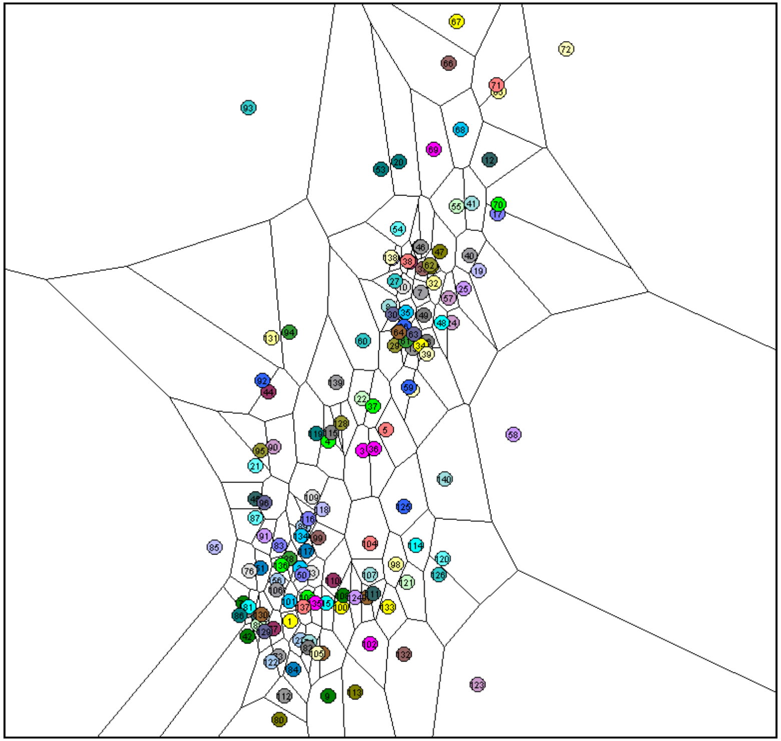


3. Feature Extract and Classification (7)

- Test result (with the train data)

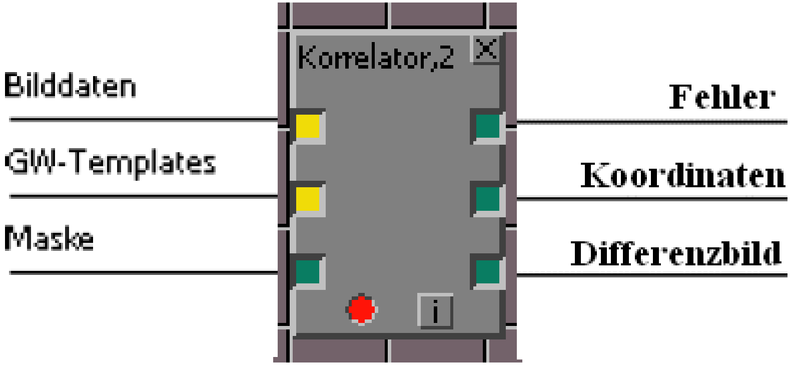
Only if the exactly train data are used, the result is good.

- Erkennungsrate: 100.000 %
- keine Fehler, 0 Warnung(en)



4. Template Matching (1)

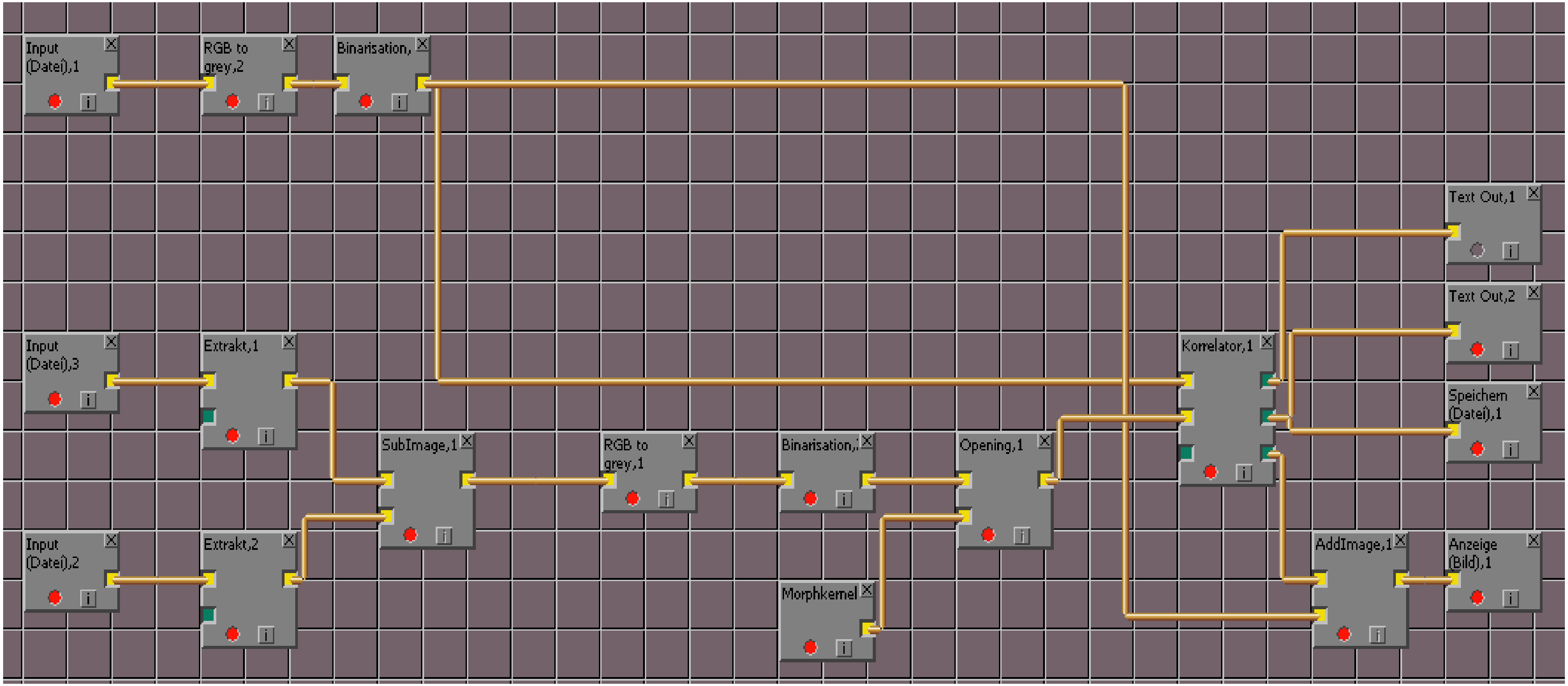
- Template matching is taking the test picture to compare with a template, in order to find the most likely picture.
- Here I use the module in QuickCog ‘Korrelator’



- The Korrelator will match the picture matrix by the template matrix and find the most similar matrix block.

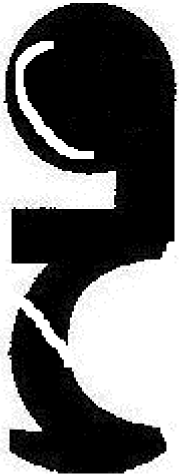
4. Template Matching (2)

- Program of template matching

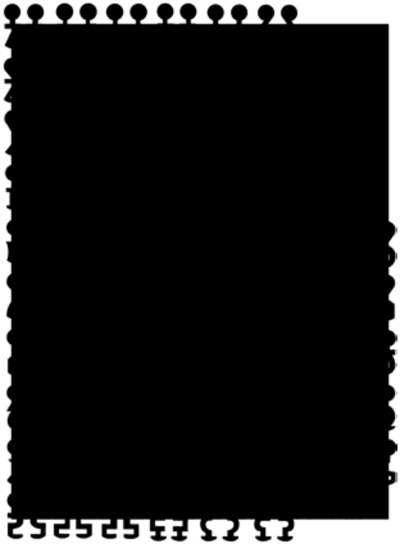


4. Template Matching (3)

- In order to test whether this method works or not, I take a single picture from the templates. Some artificial noise are added to test the robust of the system.



Test picture



Position picture

- Objekte: 1, Elemente pro Objekt: 4
- x * 1.0E+000
- 1 Objekt 1 1575,0 1699,0 91,0 235,0

Position information

4. Template Matching (4)

- Result Visualize

  From the direct result we can not got the image information quickly. So I also write a matlab function to show the result.

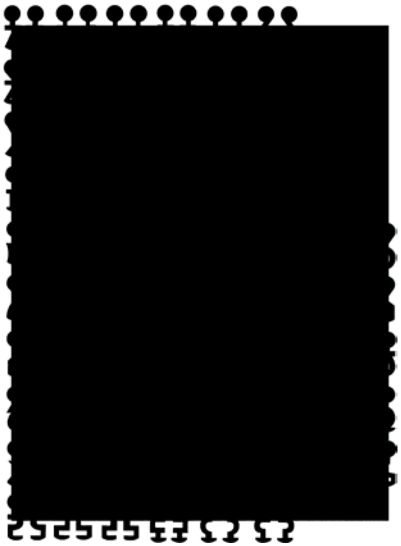
```
function result = showpic(startx,starty,width,higth)
templete=imread('yang0111.jpg');
x=starty;
y=startx;
for i=1:higth
      for j=1:width

         result(i,j)=templete(x,y);
         y=y+1;
          j=j+1;
      end
      y=startx;
      x=x+1;
      i=i+1;
end
imshow(result)
```

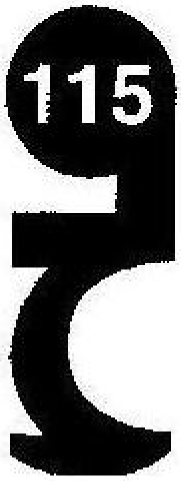
4. Template Matching (5)

- Visible result of test picture

Test picture



Position picture



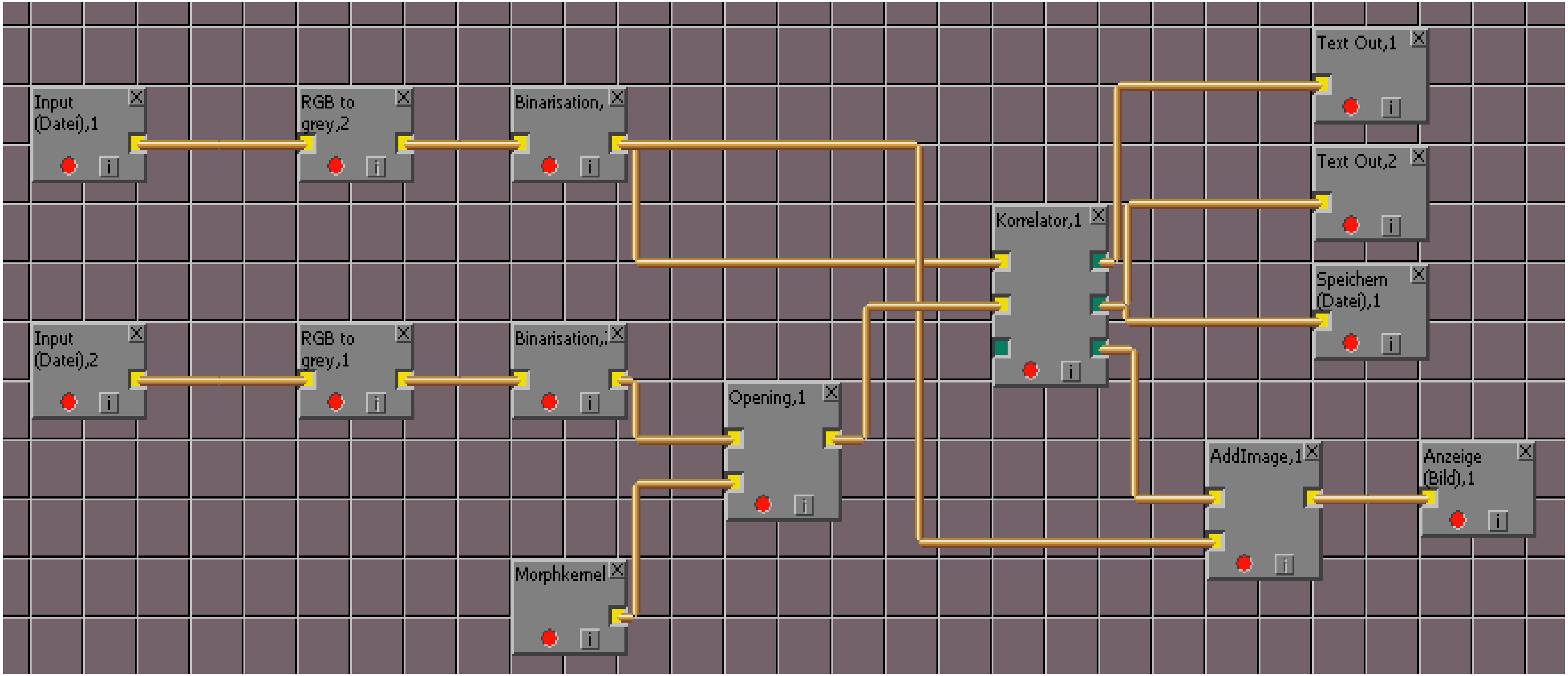
Position information

4. Template Matching (9)

- After the test of the catalogue picture, I begin to test all the three real keys.
- Since the original picture are too large for the program. In order to decrease the running time, the pictures are cut by hand and change by scale.
- Before every matching a picture test is necessary. The processing algorithm and binary threshold value is also different from each other.

4. Template Matching (7)

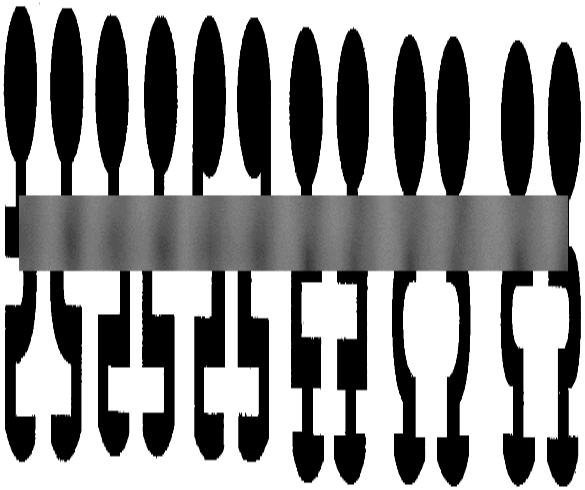
- The program system of real keys' test



4. Template Matching (8)

- Result for the real keys test for one line

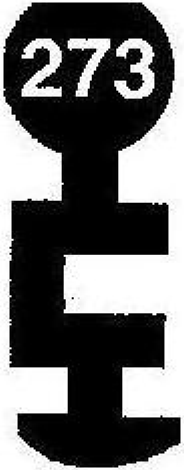
Original



742,00 38,00 91,0 234,00

Position picture

(changed for the presentation not the one used in lab)

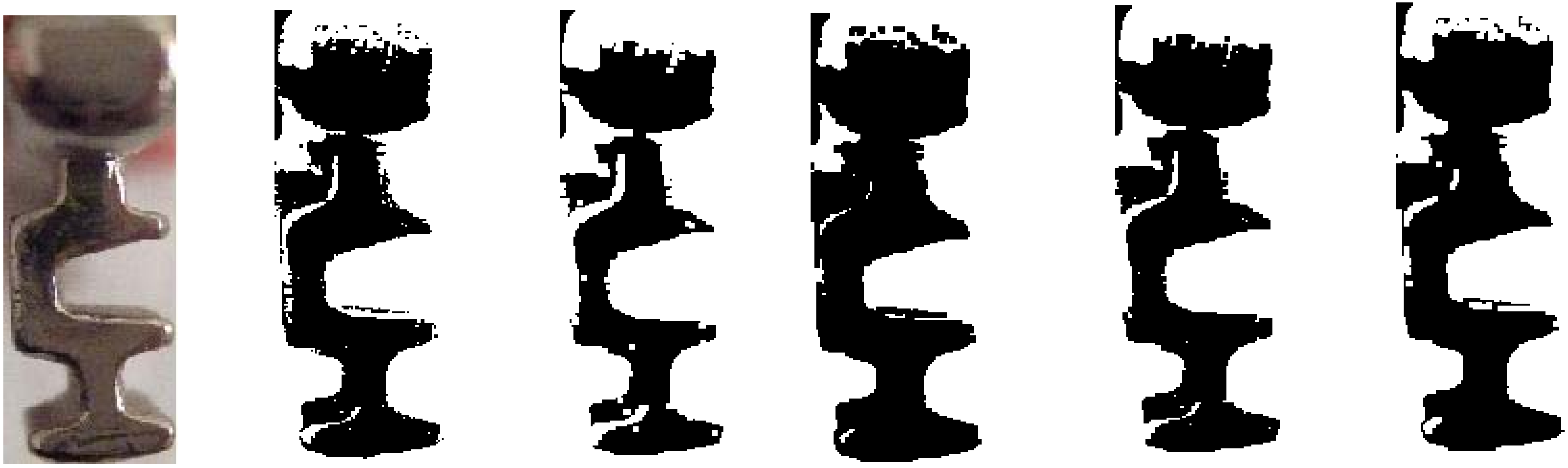


Show result

4. Template Matching (10)

- Pictures Test for key No.273

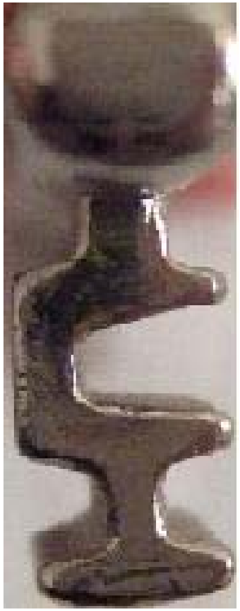
  threshold value 90



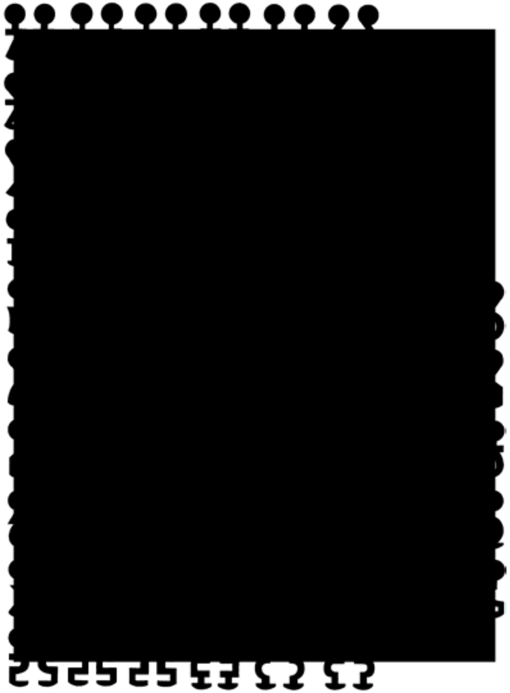
Original Binary Dilation Erosion Closing Opening

4. Template Matching (11)

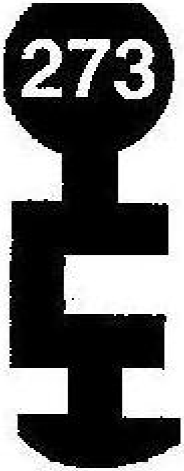
- Result for key 273 (opening)



Original



Position picture



759,0	2564,0
91,0	234,0

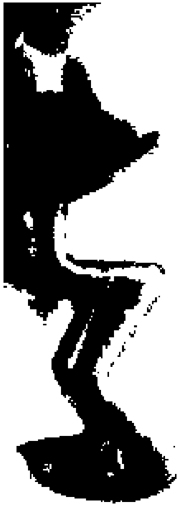
Show result

4. Template Matching (12)

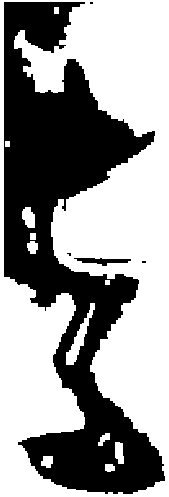
- Pictures Test for the key not include in the template

threshold value 110

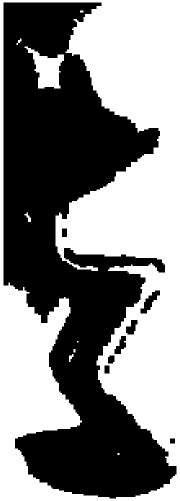
Original



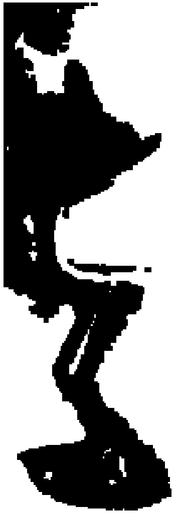
Binary



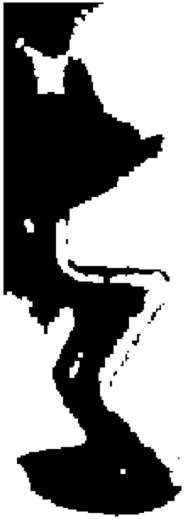
Dilation



Erosion



Closing

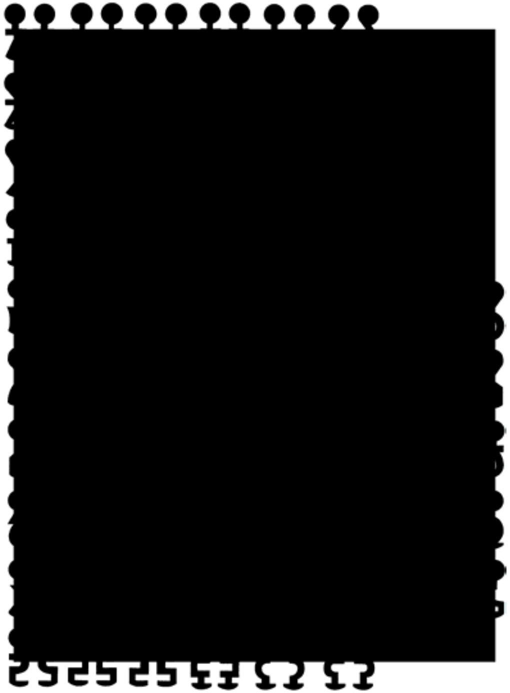


Opening

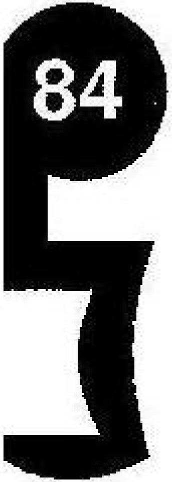
4. Template Matching (13)

- Result for the key not include in the template (opening)

Original



Position picture



668,0	1365,0
114,0	307,0

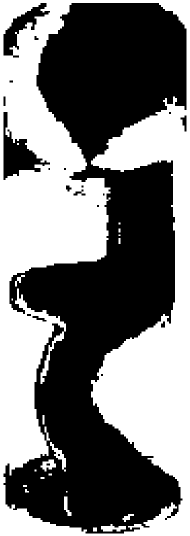
Show result

4. Template Matching (14)

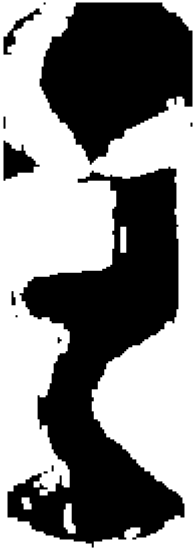
- Pictures Test for key 115

  threshold value 100

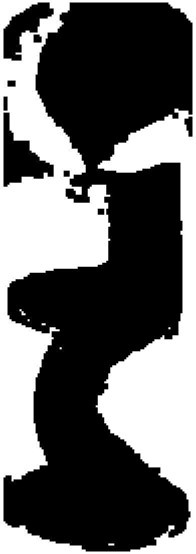
Original



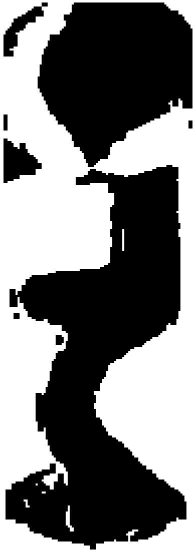
Binary



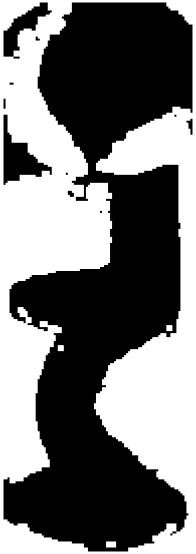
Dilation



Erosion



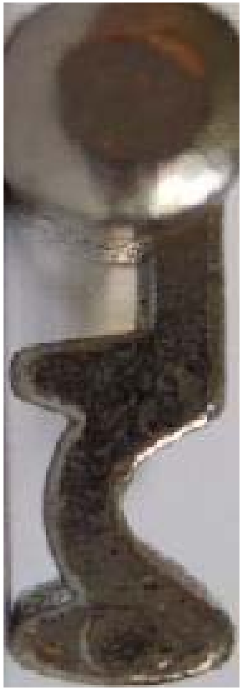
Closing



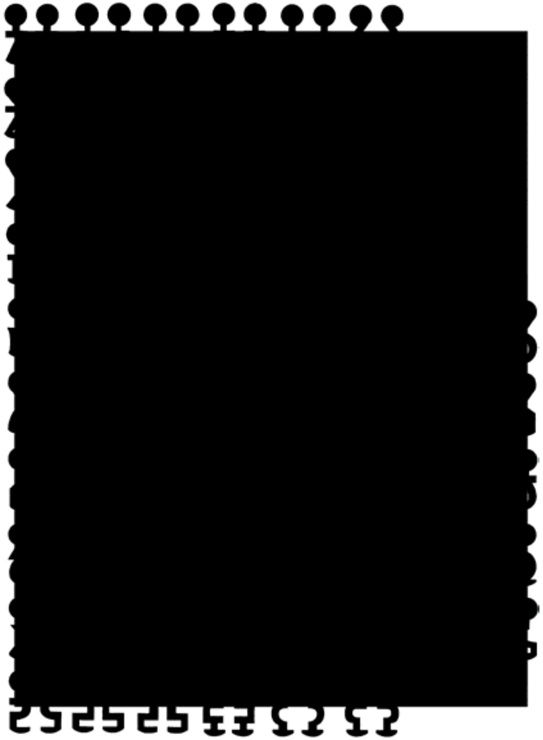
Opening

4. Template Matching (15)

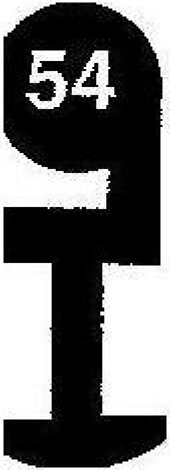
- Result for key 115 (opening)



Original



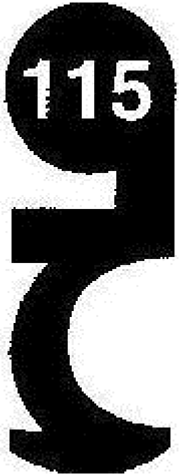
Position picture



668,00 847,00

91,00 234,00

Show result

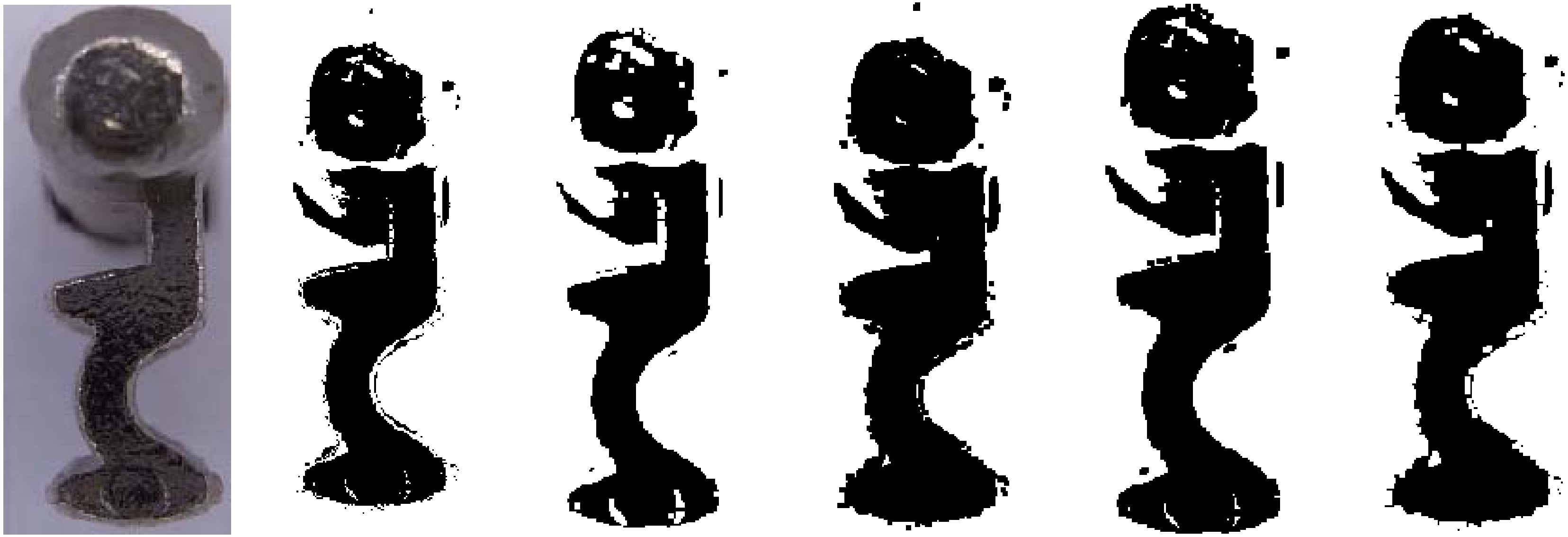


Real result

4. Template Matching (16)

- Pictures Test for key 115

threshold value 80



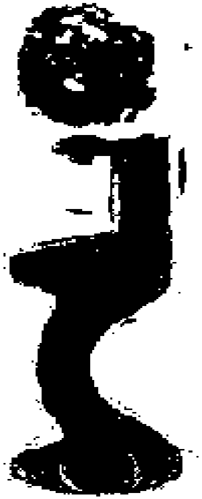
Original Binary Dilation Erosion Closing Opening

4. Template Matching (17)

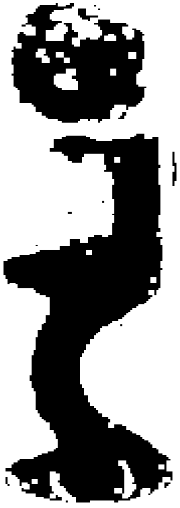
- Pictures Test for key 115

  threshold value 80 with some cut of shadow

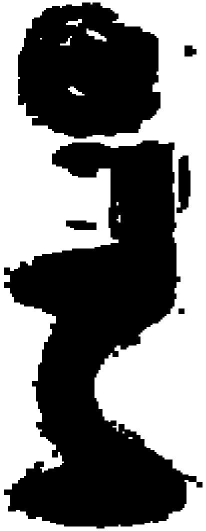
Original



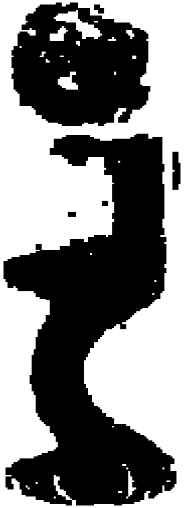
Binary



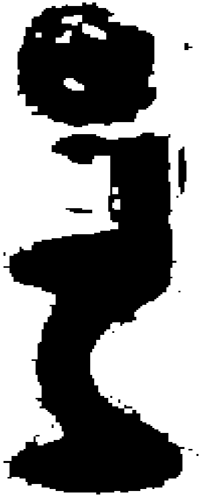
Dilation



Erosion



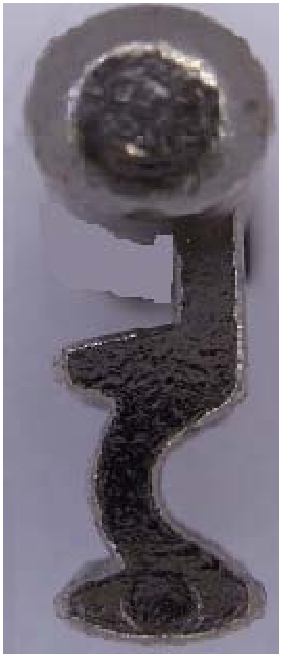
Closing



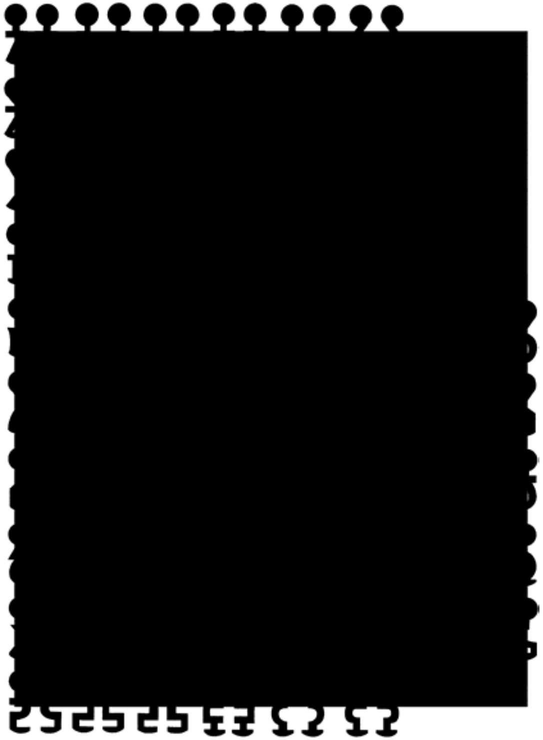
Opening

4. Template Matching (18)

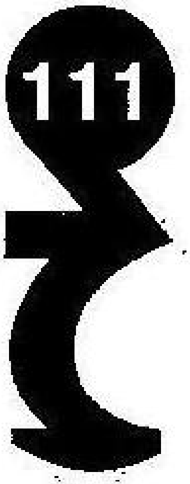
- Result for key 115 without shadow (clothing)



Original



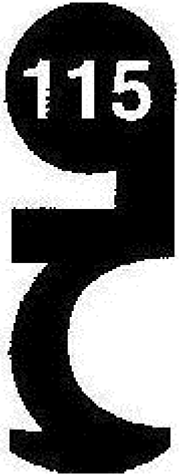
Position picture



1278,0 1644,00

132,00 312,00

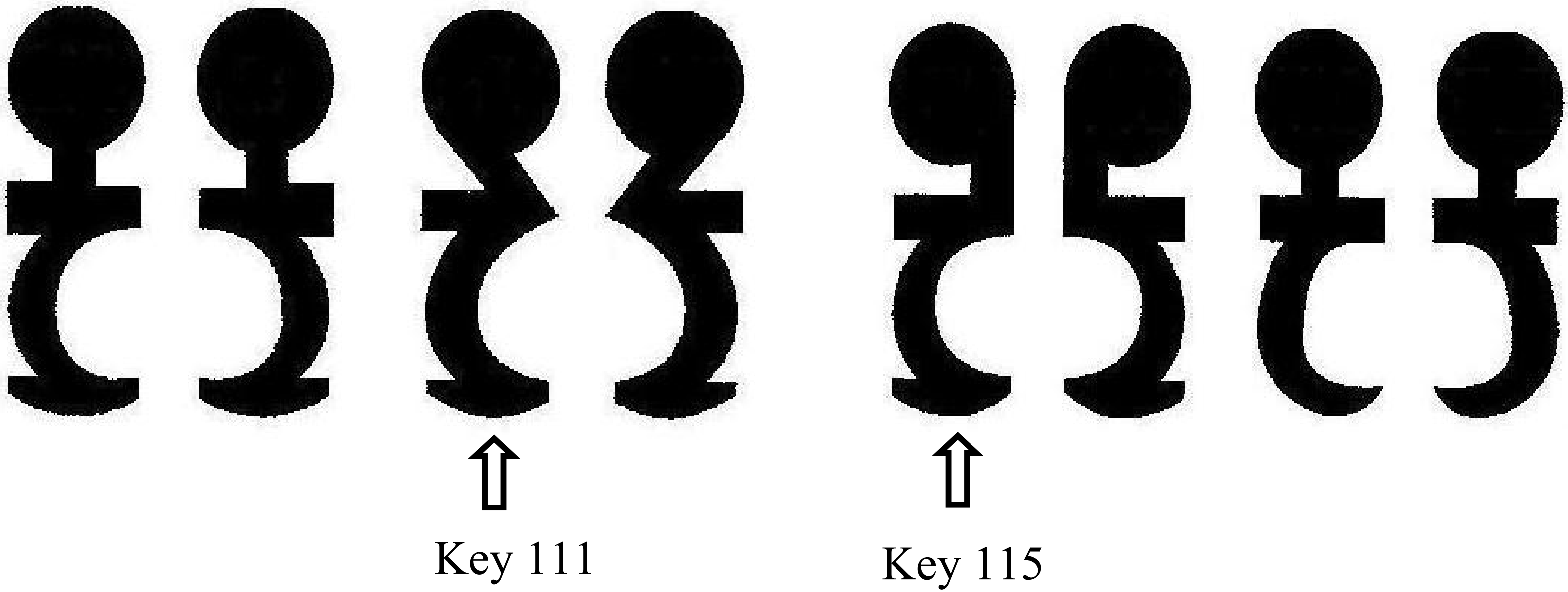
Show result



Real result

4. Template Matching (19)

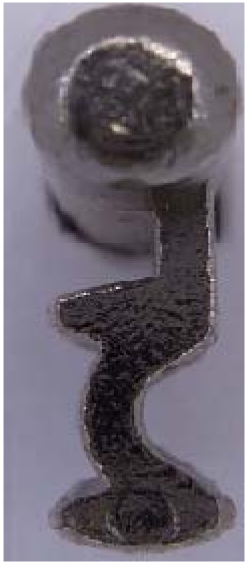
- Key 111 looks quite similar as key 115, in order to find out the reason which causes this misindentify I cut out the two small blocks from the template and do some further tests.



4. Template Matching (20)

- Result comparison of sub templates (1)

Test picture

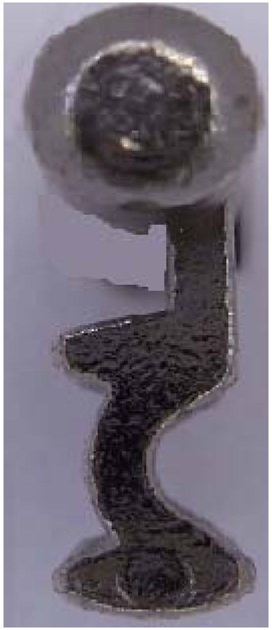


	Result of 111 block		Result of 115 block	
Error	102,10		112,06	
Position	273,00	12,00	5,00	4,00
	120,00	279,00	120,00	279,00

4. Template Matching (21)

- Result comparison of sub templates (2)

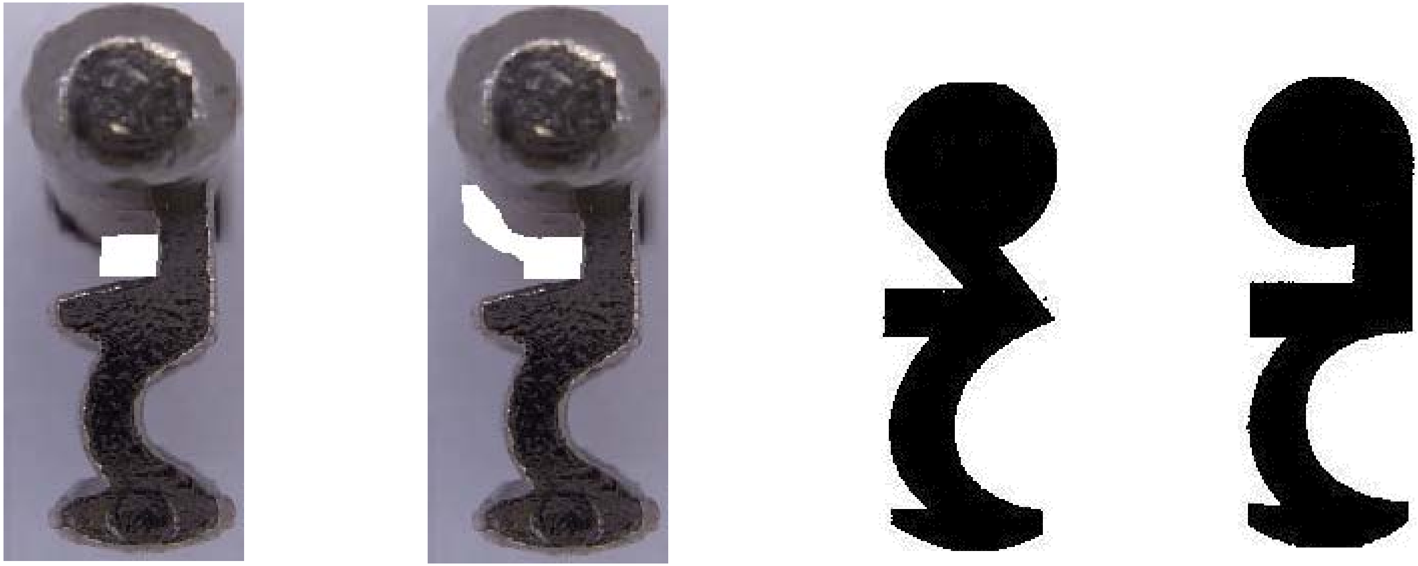
Test picture



	Result of 111 block		Result of 115 block	
Error	177,53		191,95	
Position	266,00	0,00	4,00	0,00
	132,00	312,00	132,00	312,00

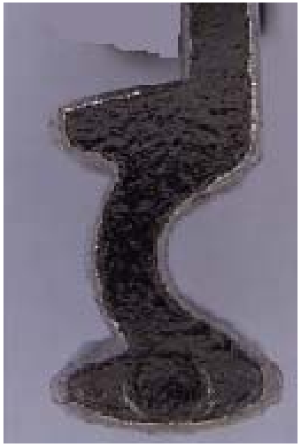
4. Template Matching (22)

- I also change the picture by hand to make it seems more like the template one. But they two also can not work. The error of key 111 always smaller than key 115.



4. Template Matching (23)

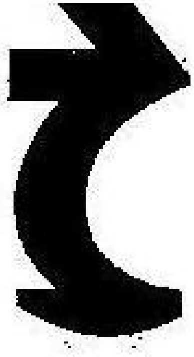
- I think maybe the round head is the key point for the misidentify because it is so big and may happen to be judged as the part of the interim hilt. So I cut the head out only use the body to do the template matching again.



Used image

Position information

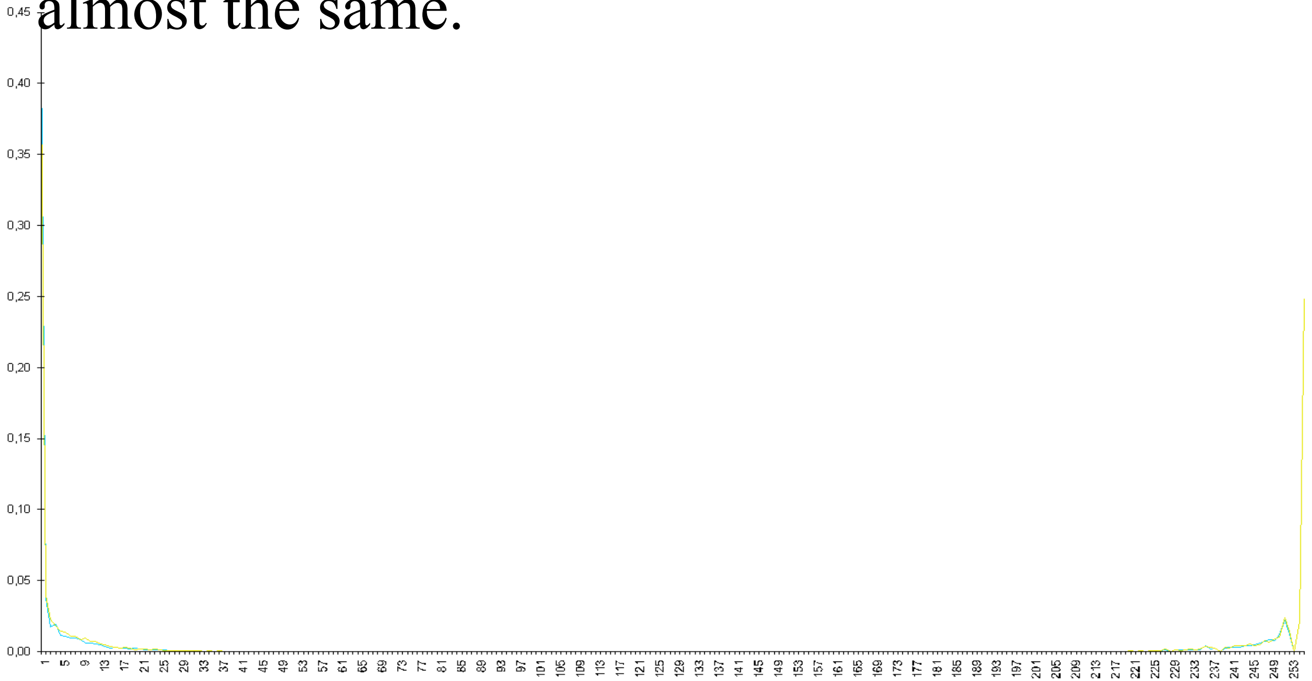
1281,0	1782,0
120,0	160,0



Show result

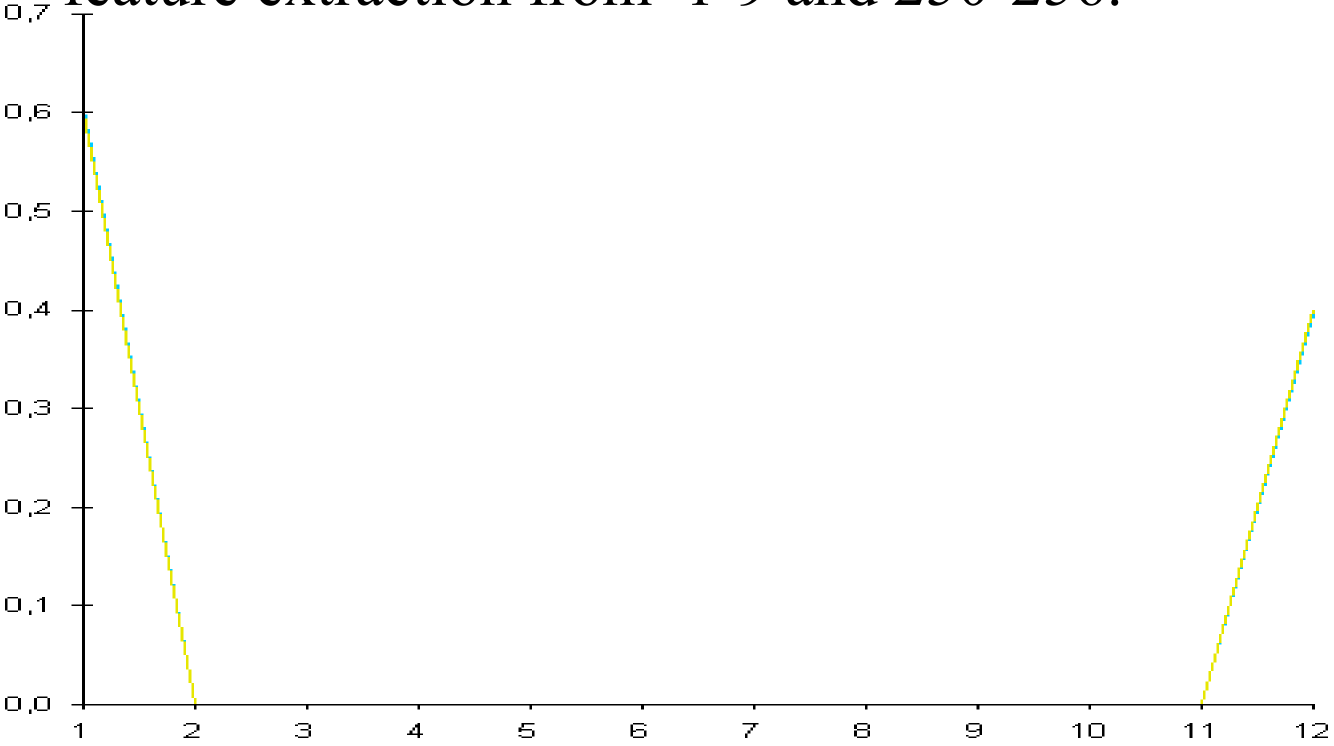
5.Histogram (1)

- For the exist problem, I use histogram to analyze the keys No.111 and No.115. The following is the results. They almost the same.



5.Histogram (2)

- The histogram of keys No.111 and No.115 after feature extraction from 1-9 and 250-256.

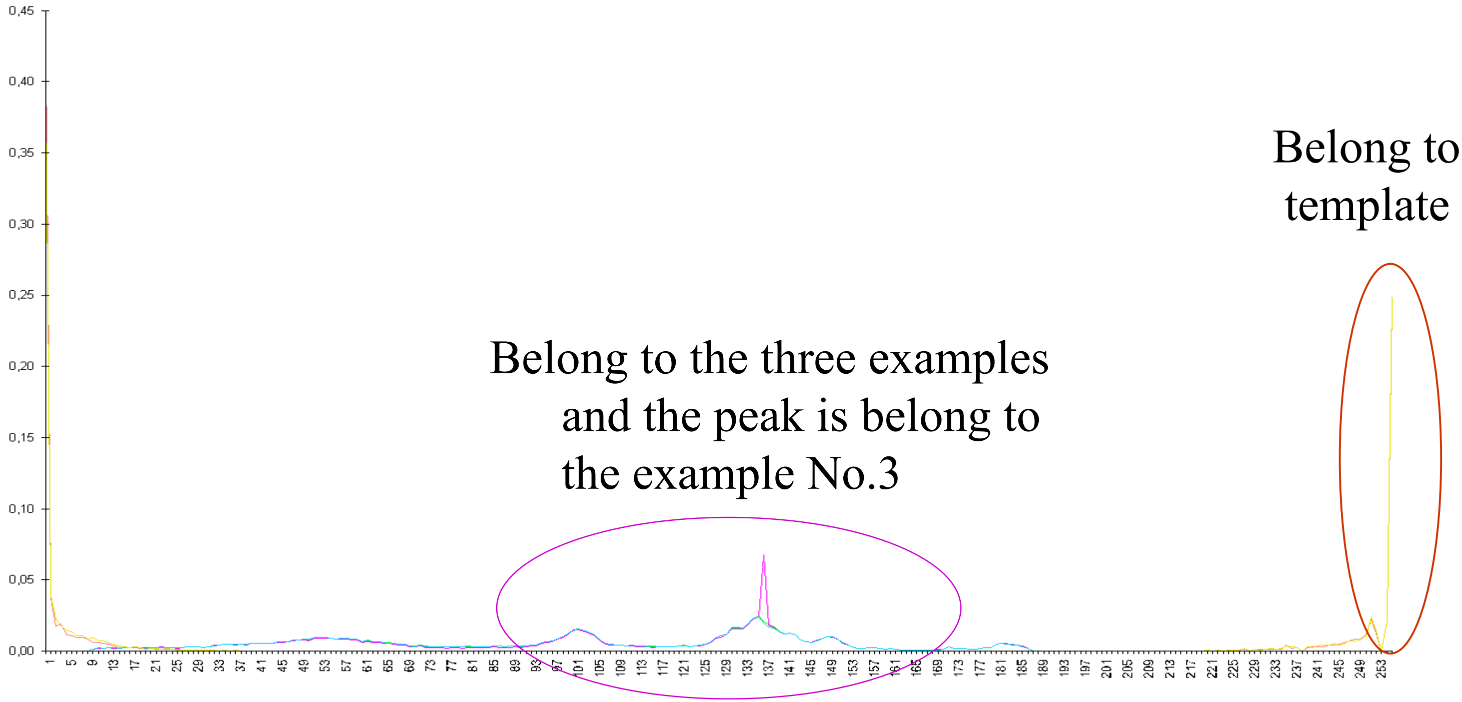


5.Histogram (5)

- These are the pictures I used in the test.

5.Histogram (6)

- The following is the histogram for these three examples, key No.111 and No.115. I also use the feather extraction to



Belong to the template keys

Belong to the three examples and the peak is belong to the example No.3

Conclusion

- After the work of this project, man can easily recognize a key by matching its picture with the brochure. But the requirement of this picture is strict. it should be frontal and without any rotation.
- A subsequent work is making the system a little more robust. Which means the pictures with rotation can also be recognized. This may need a large database with rotation training data.
- A little suggestion is that maybe we can combine the camera, processing procedure and the result showing part together in one program. An idea is using MFC to build a customer operation interface.

Thank You for your attention!