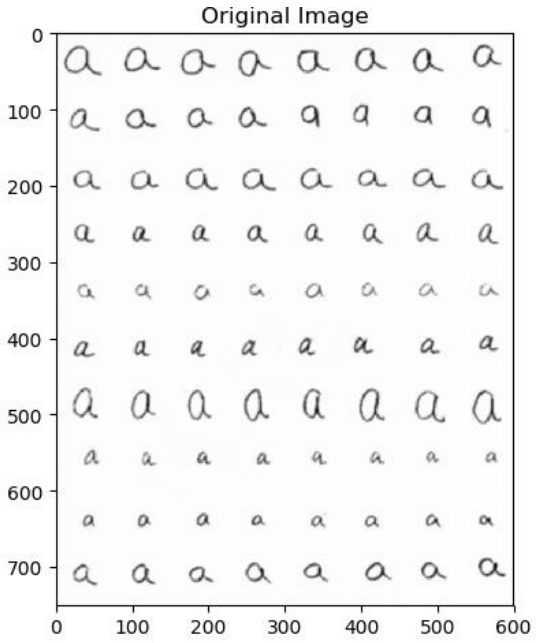
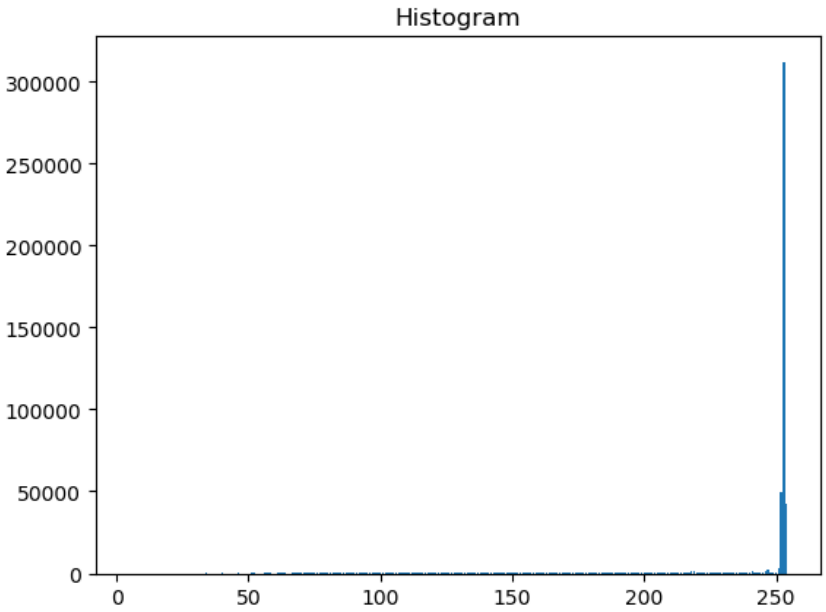
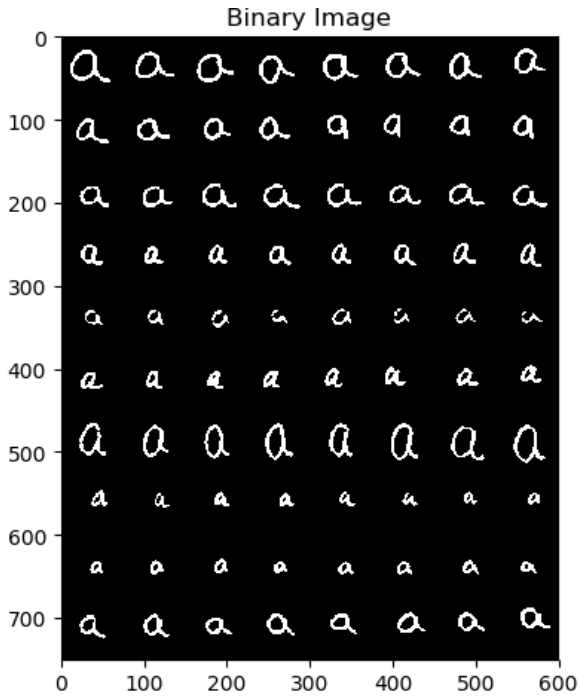
# 1 Train

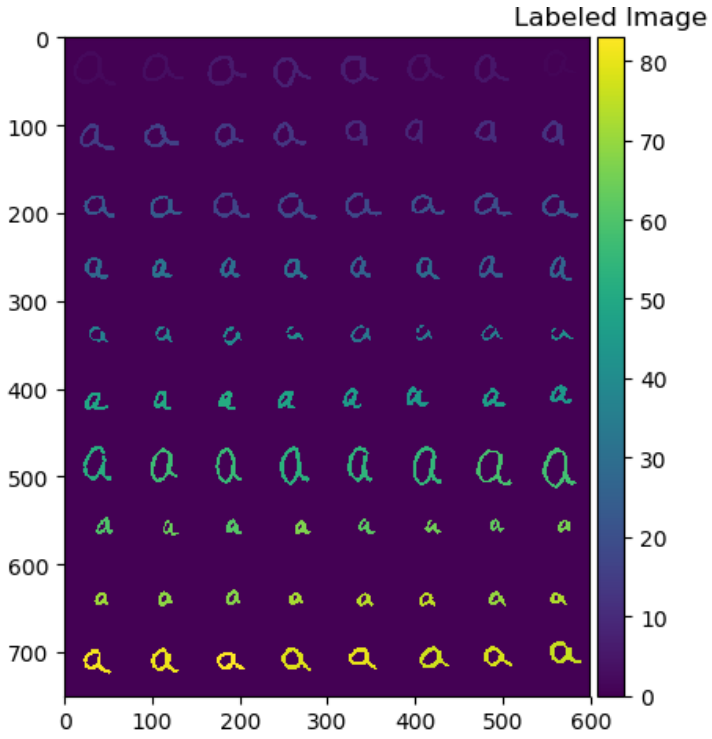
In the phase of generating training data set.

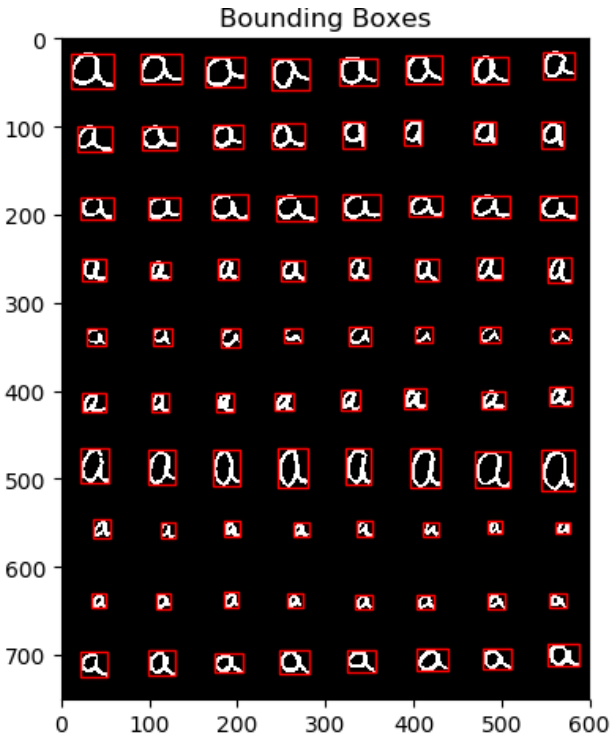
## 1.1 a



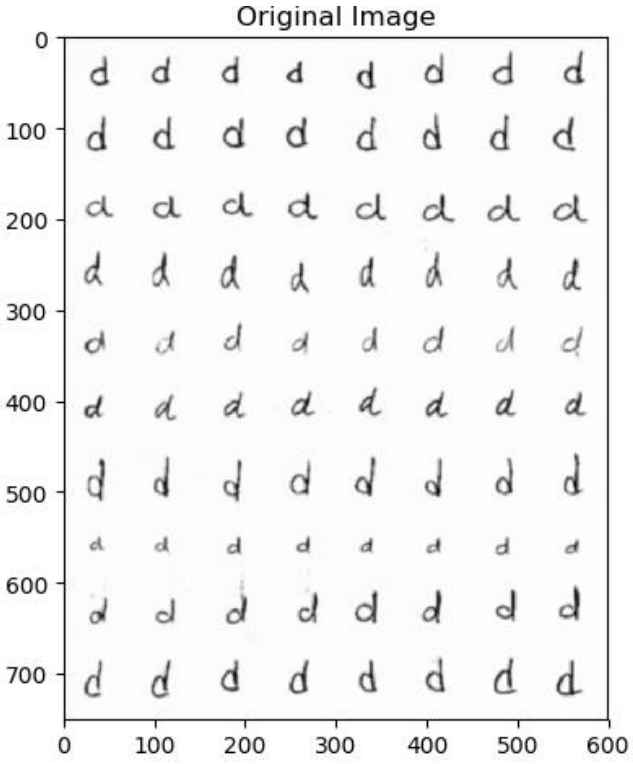



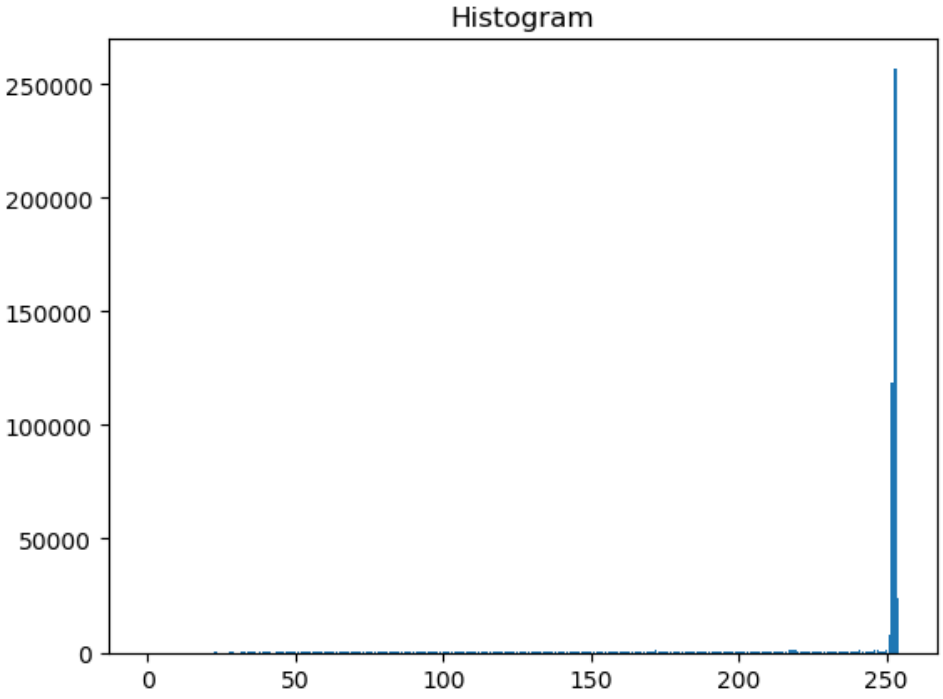

Binary Image
0
100
200
300
400
500
600
700
0
100
200
300
400
500
600

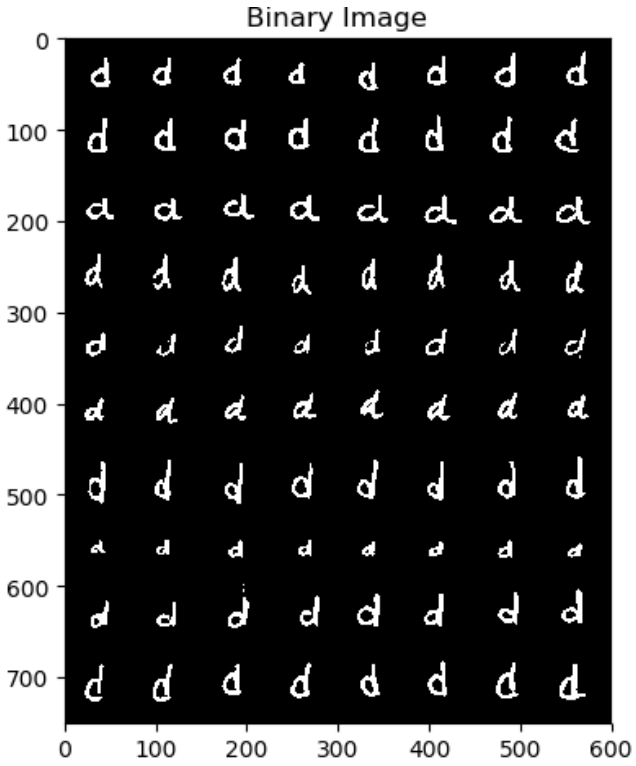

Labeled Image
0
100
200
300
400
500
600
700
0
100
200
300
400
500
600
80
70
60
50
40
30
20
10
0

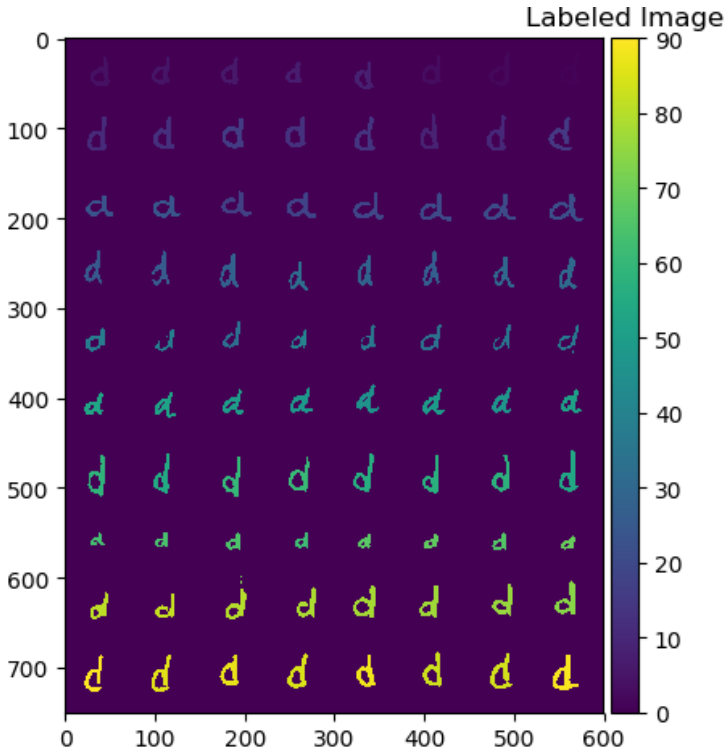

Bounding Boxes
0
100
200
300
400
500
600
700
0
100
200
300
400
500
600

1.2 d


Original Image
0
100
200
300
400
500
600
700
0
100
200
300
400
500
600

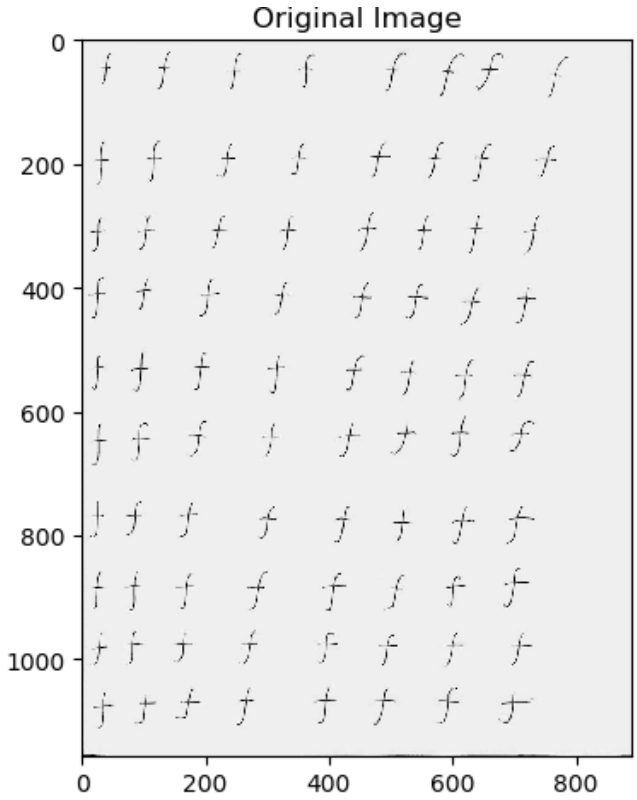

Histogram
250000
200000
150000
100000
50000
0
0
50
100
150
200
250

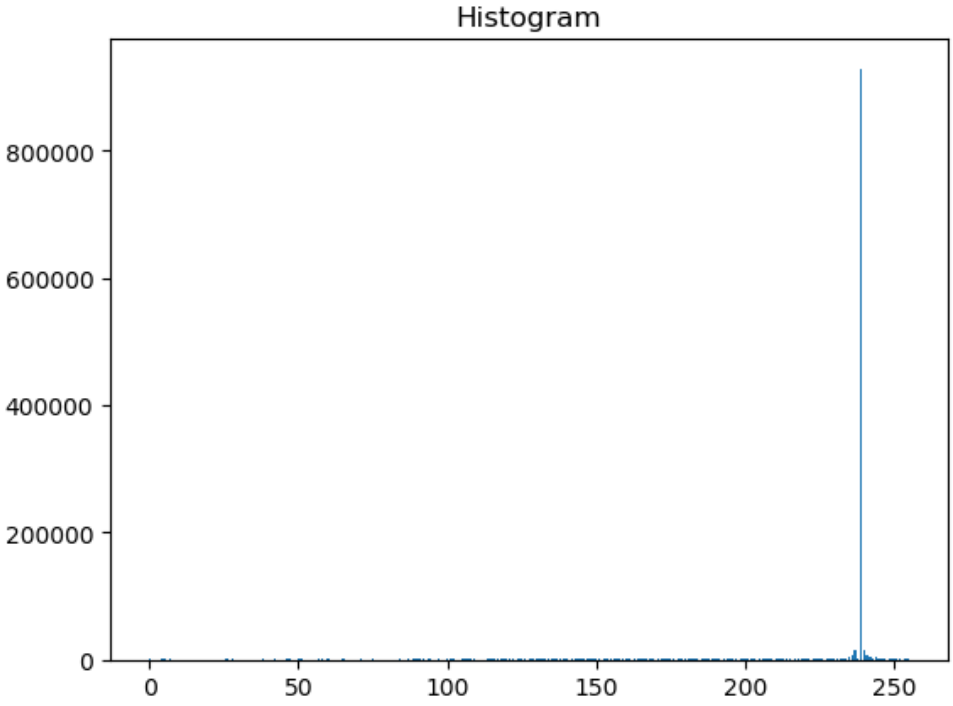

Binary Image
0
100
200
300
400
500
600
700
0
100
200
300
400
500
600

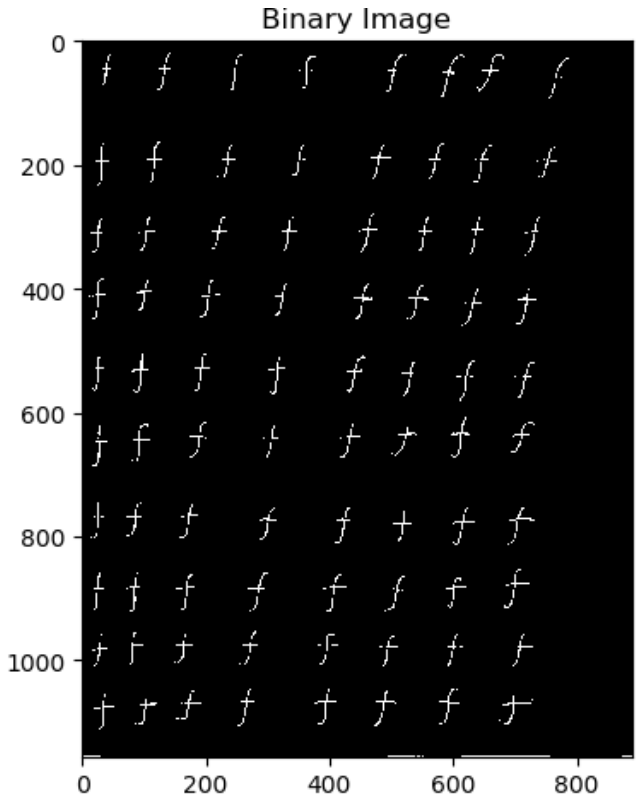

Labeled Image
0
100
200
300
400
500
600
700
0
100
200
300
400
500
600
90
80
70
60
50
40
30
20
10
0

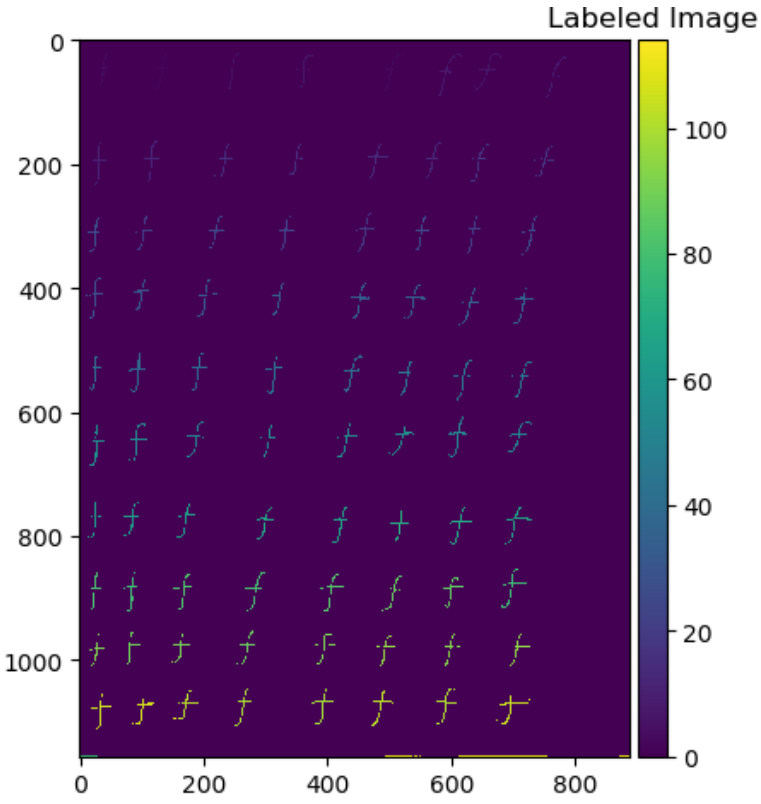

Bounding Boxes
0
100
200
300
400
500
600
700
0
100
200
300
400
500
600

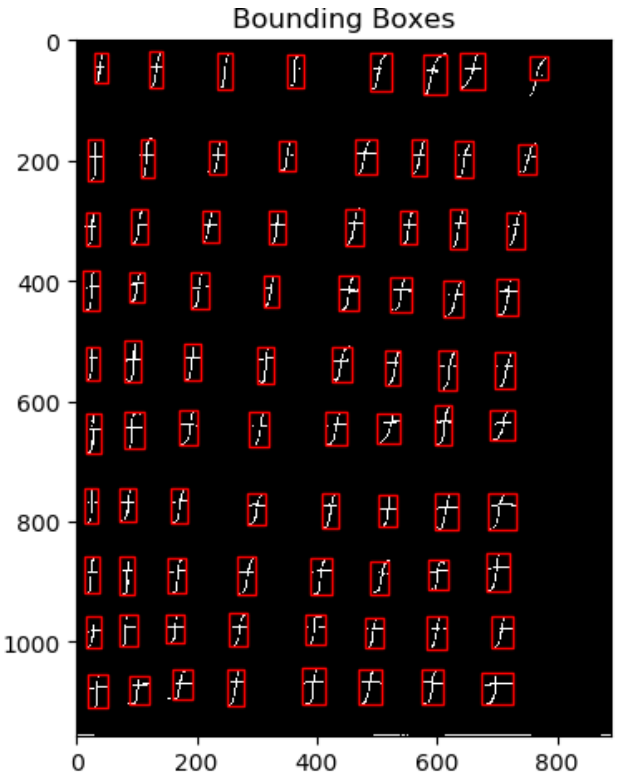
1.3 f


Original Image
0
200
400
600
800
1000
0
200
400
600
800

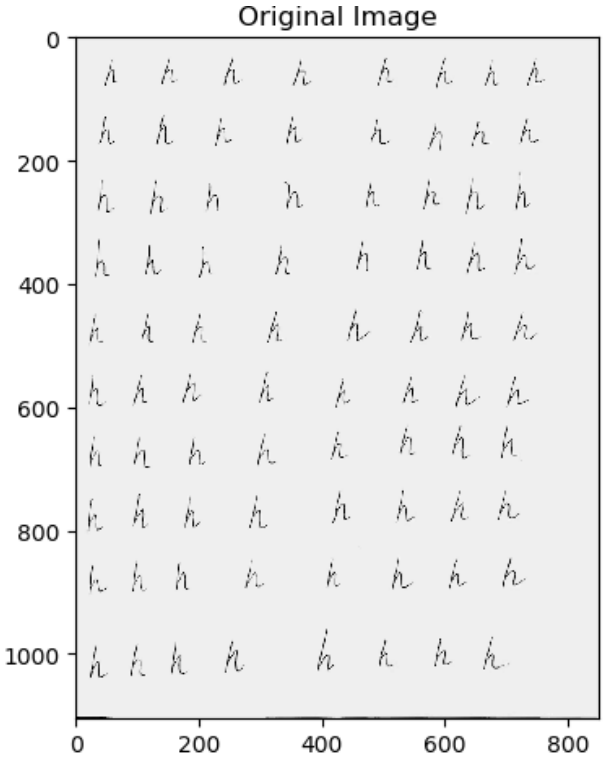

Histogram
800000
600000
400000
200000
0
0
50
100
150
200
250

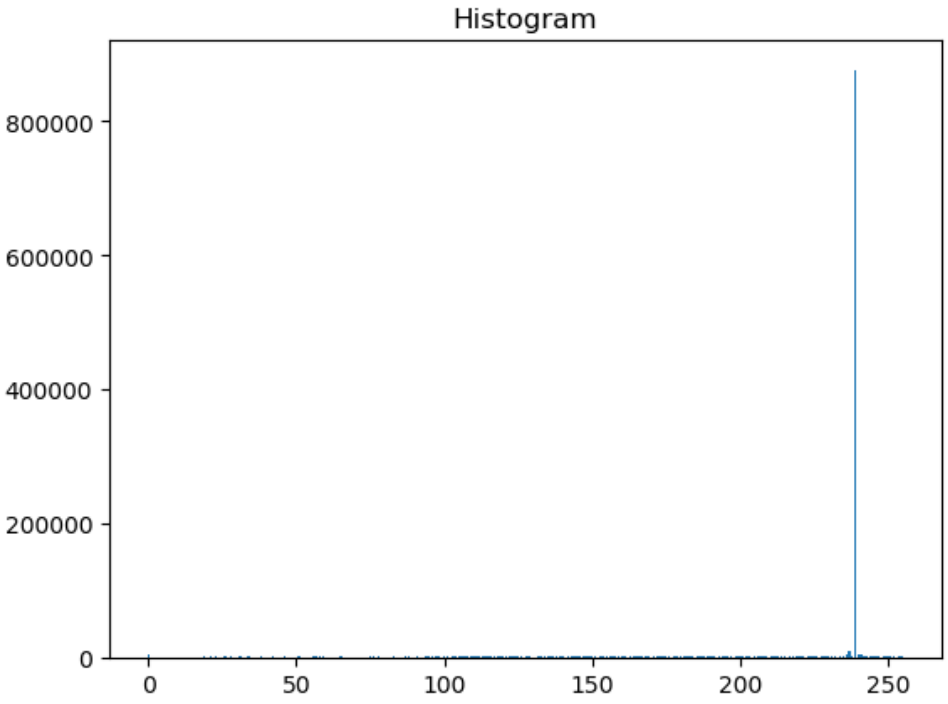

Binary Image
0
200
400
600
800
1000
0
200
400
600
800

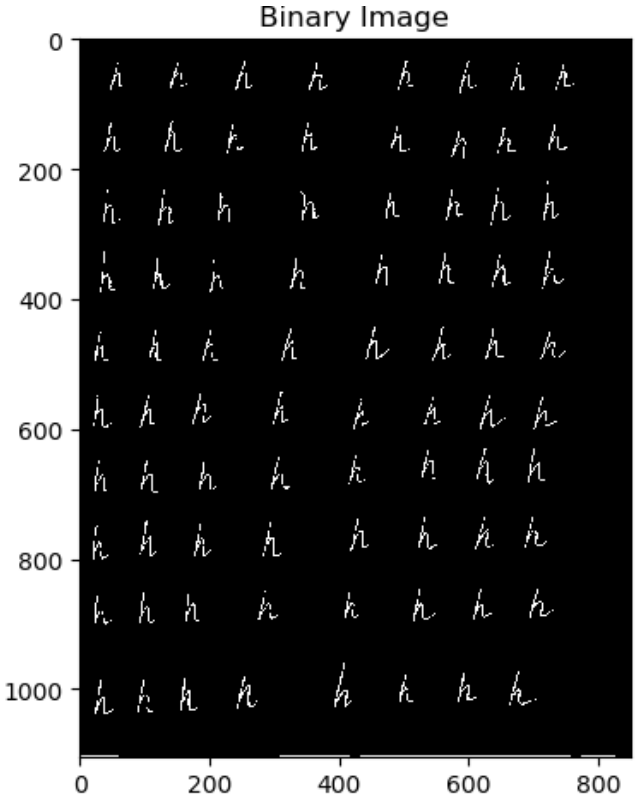

Labeled Image
0
200
400
600
800
1000
0
200
400
600
800
100
80
60
40
20
0

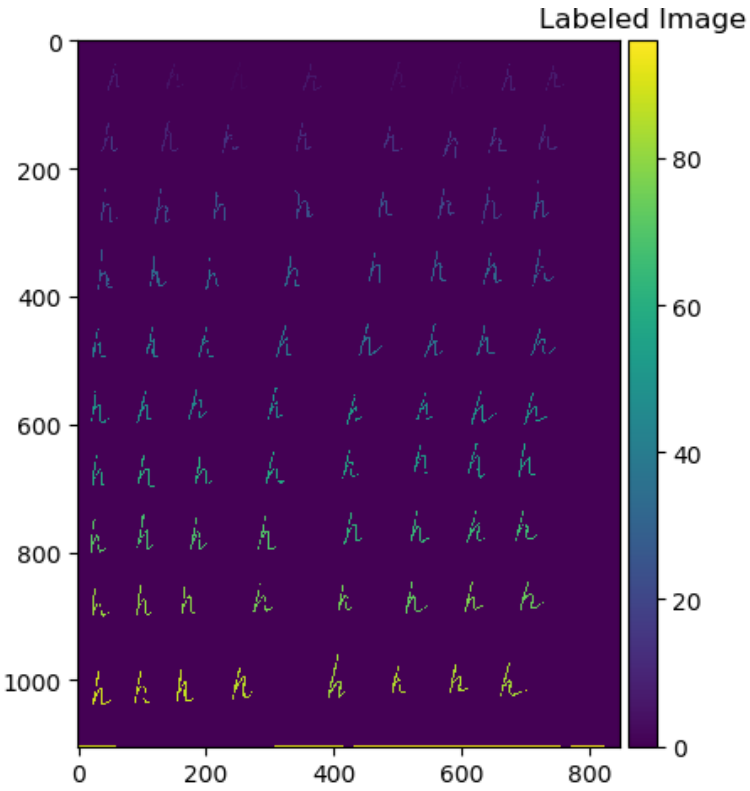

Bounding Boxes
0
200
400
600
800
1000
0
200
400
600
800

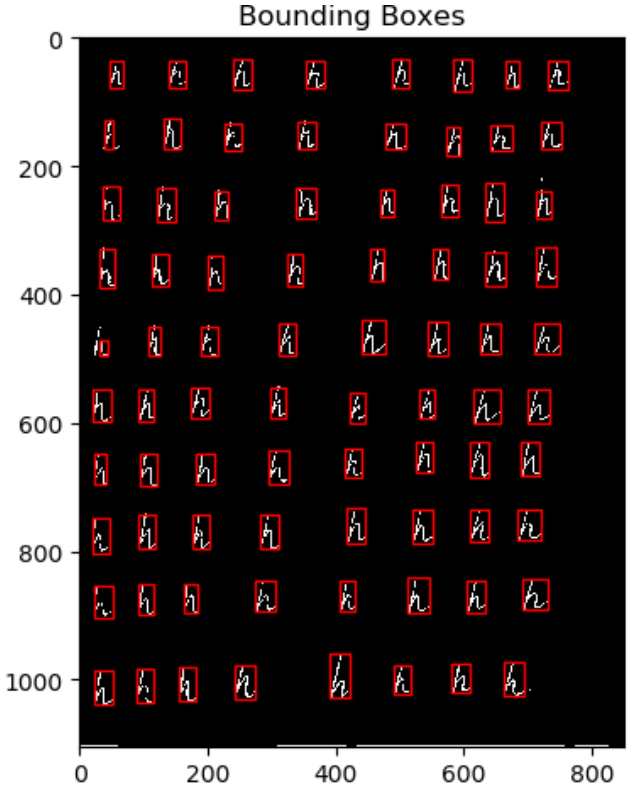
1.4 h


Original Image
0
200
400
600
800
1000
0
200
400
600
800

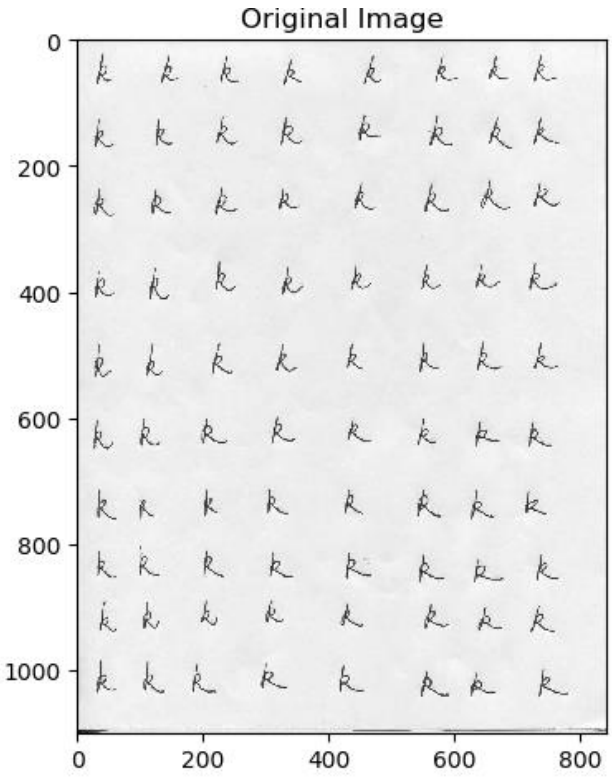

Histogram
800000
600000
400000
200000
0
0
50
100
150
200
250

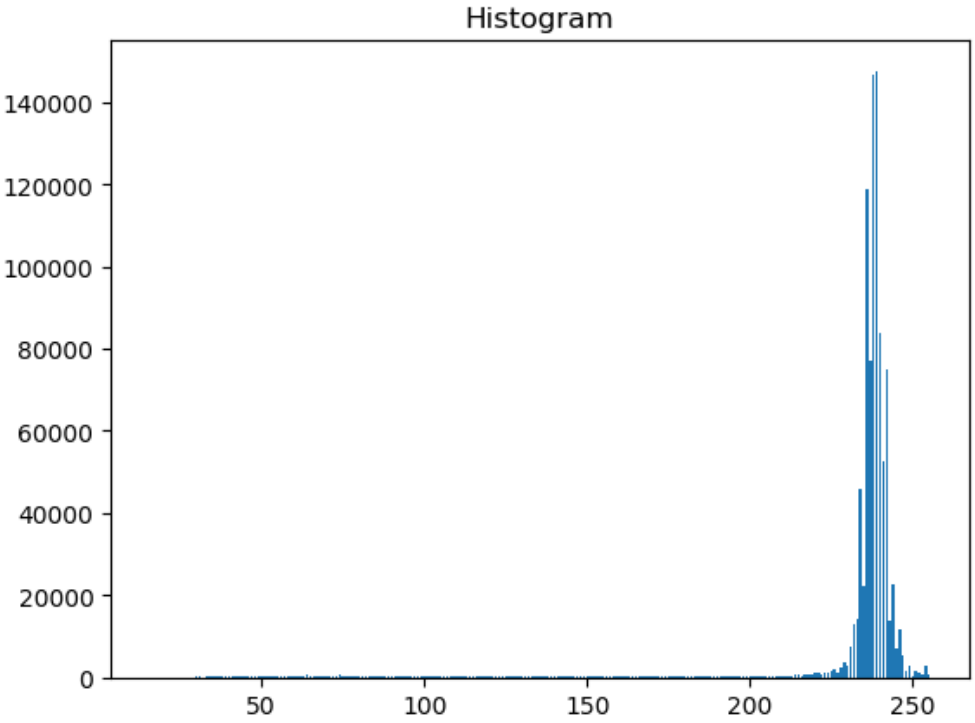

Binary Image
0
200
400
600
800
1000
0
200
400
600
800

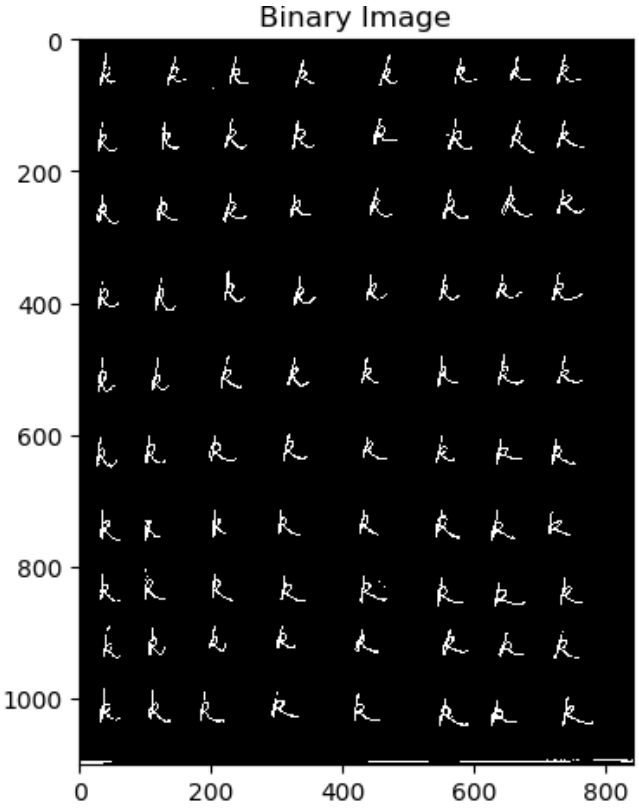

Labeled Image
0
200
400
600
800
1000
0
200
400
600
800
0
20
40
60
80

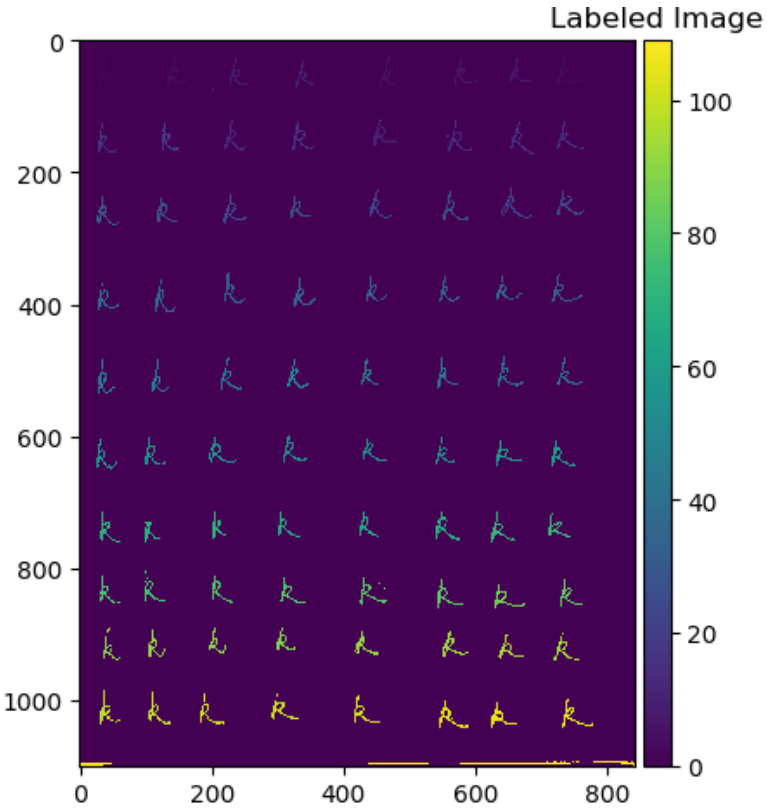

Bounding Boxes
0
200
400
600
800
1000
0
200
400
600
800

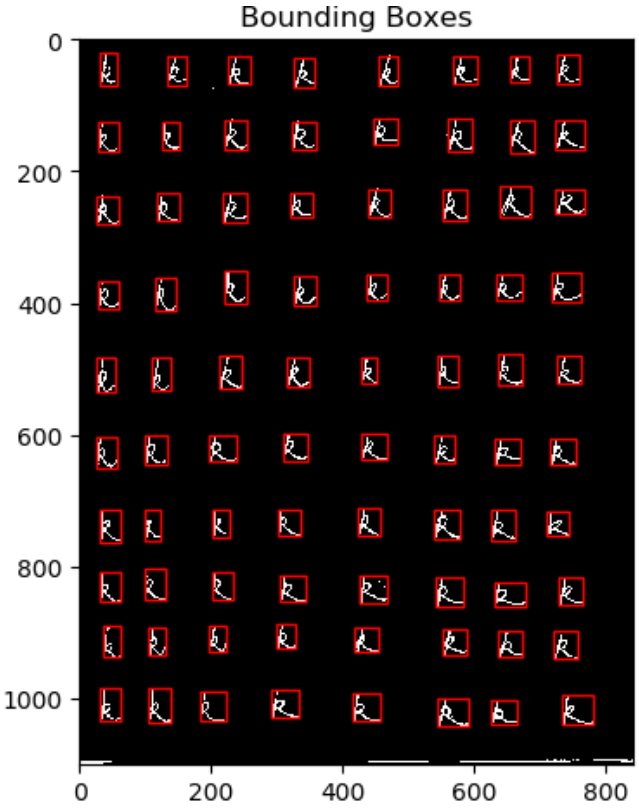
1.5 k


Original Image
0
200
400
600
800
1000
0
200
400
600
800

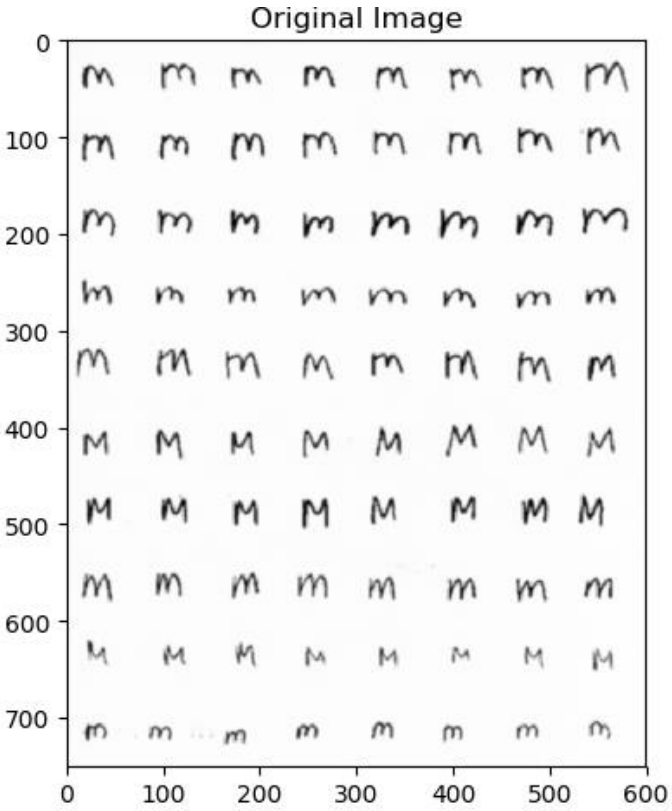

Histogram
140000
120000
100000
80000
60000
40000
20000
0
50
100
150
200
250

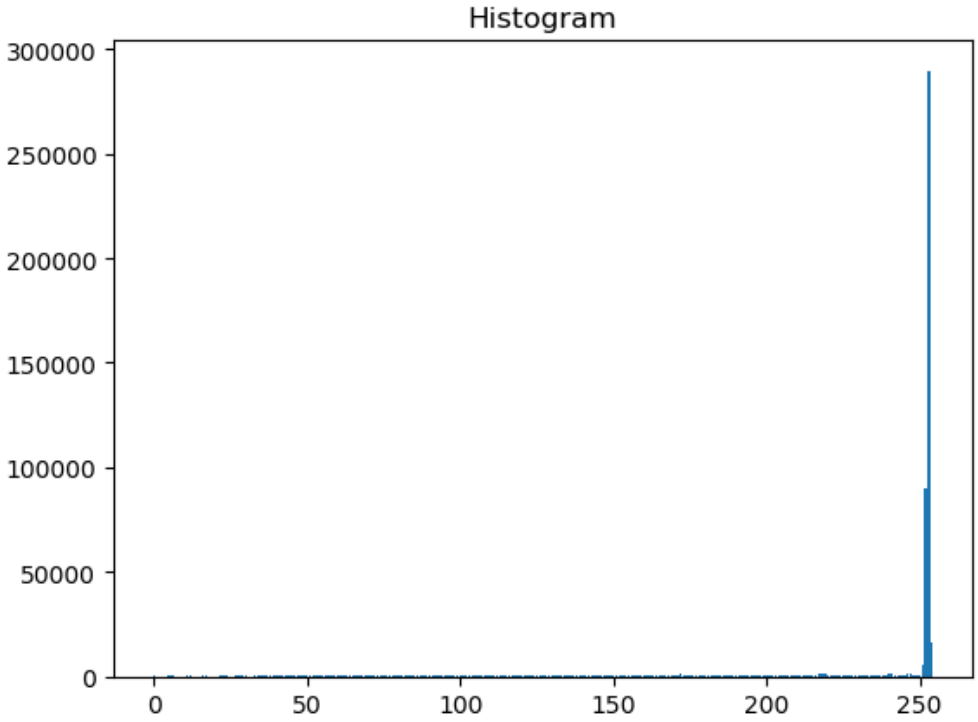

Binary Image
0
200
400
600
800
1000
0
200
400
600
800

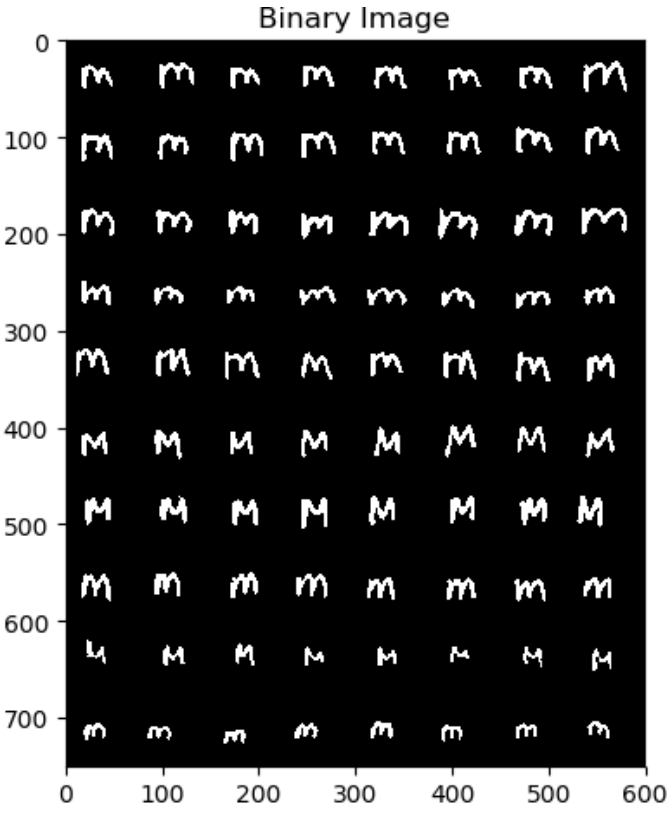

Labeled Image
0
200
400
600
800
1000
0
200
400
600
800
100
80
60
40
20
0

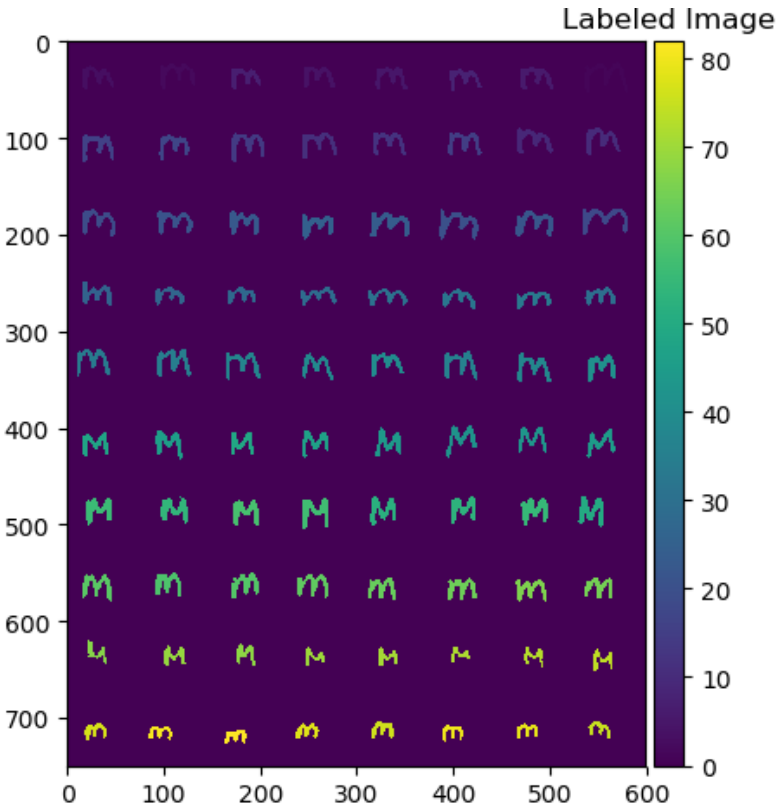

Bounding Boxes
0
200
400
600
800
1000
0
200
400
600
800

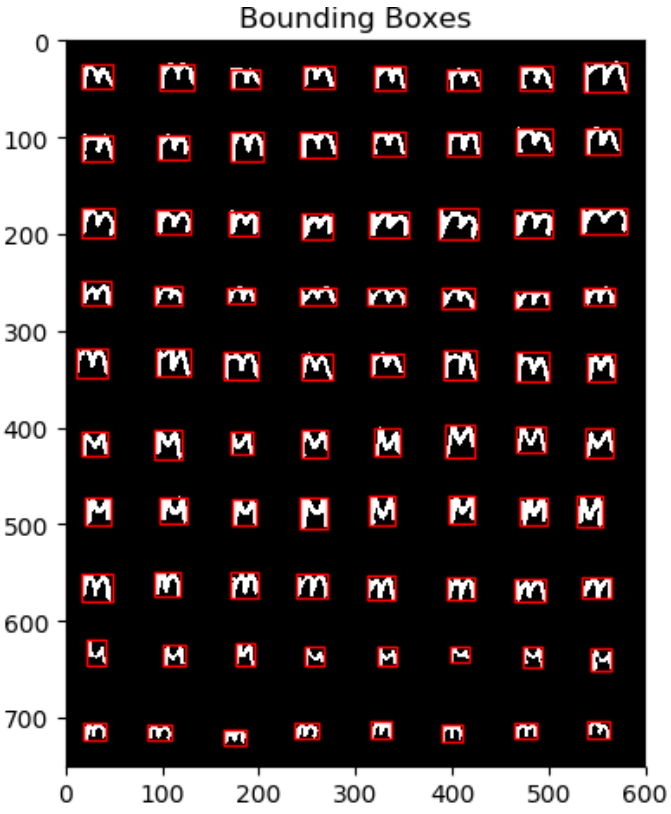
1.6 m


Original Image
0
100
200
300
400
500
600
700
0
100
200
300
400
500
600

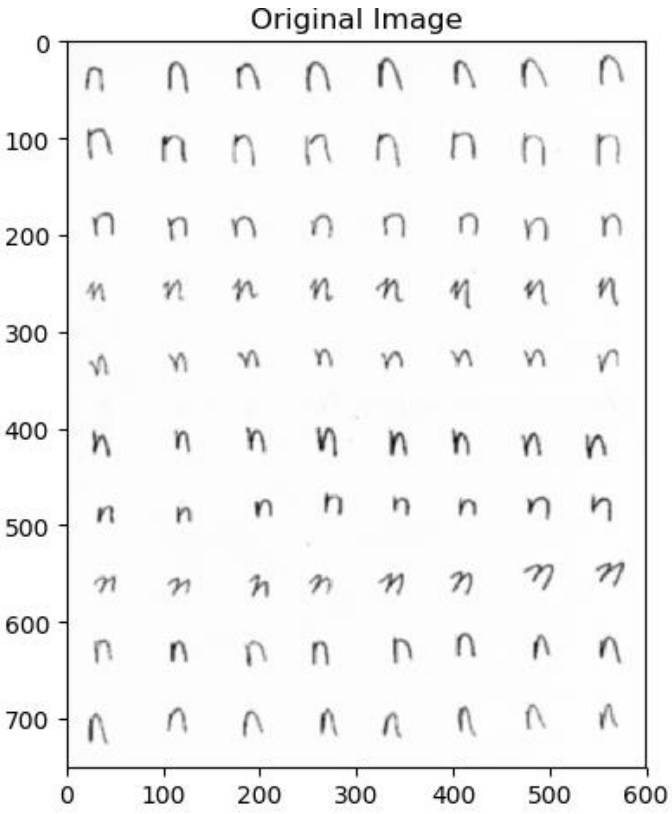

Histogram
300000
250000
200000
150000
100000
50000
0
0
50
100
150
200
250

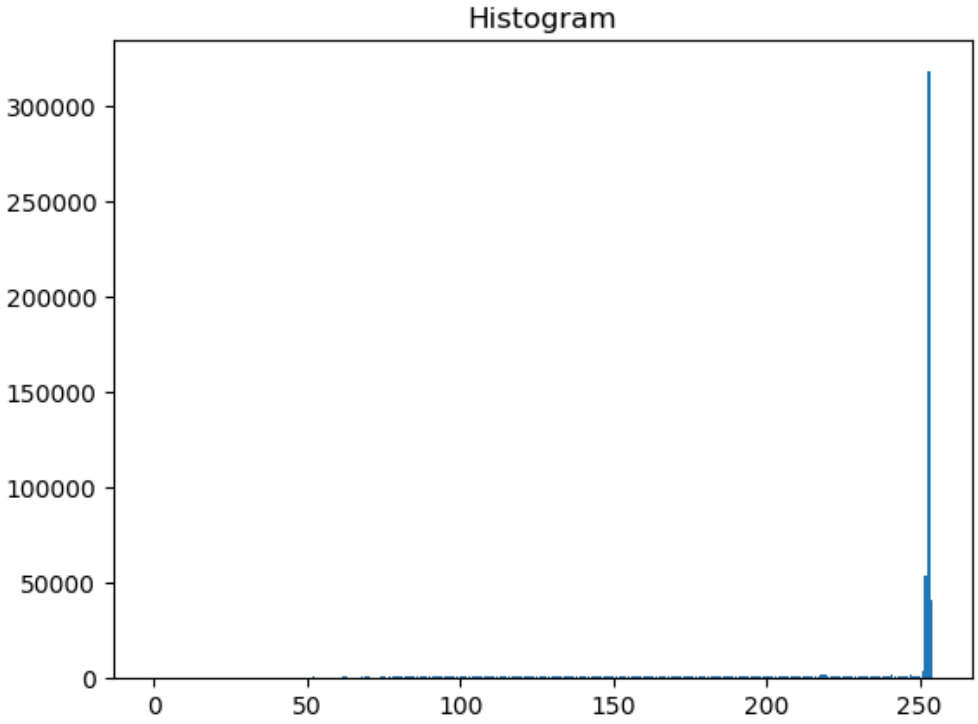

Binary Image
0
100
200
300
400
500
600
700
0
100
200
300
400
500
600

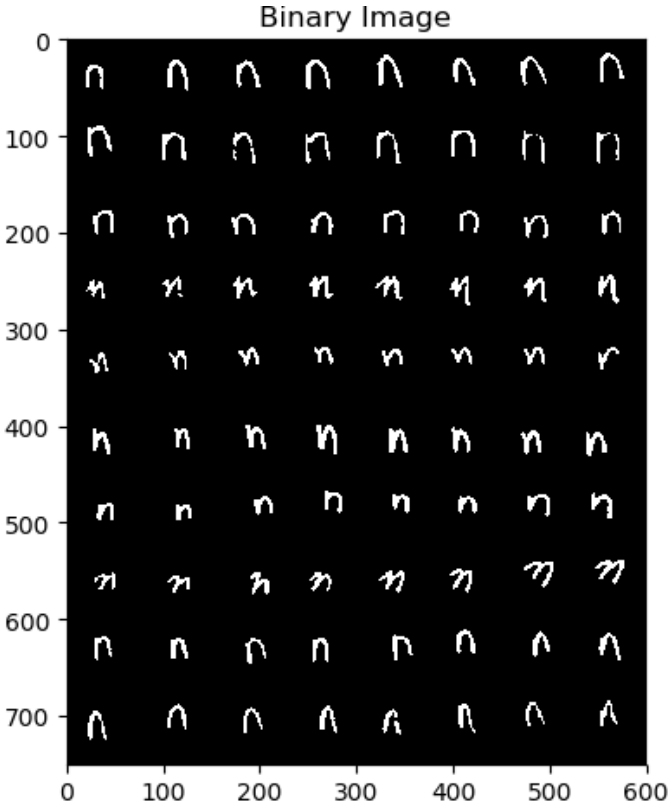

Labeled Image
0
100
200
300
400
500
600
700
0
100
200
300
400
500
600
0
10
20
30
40
50
60
70
80

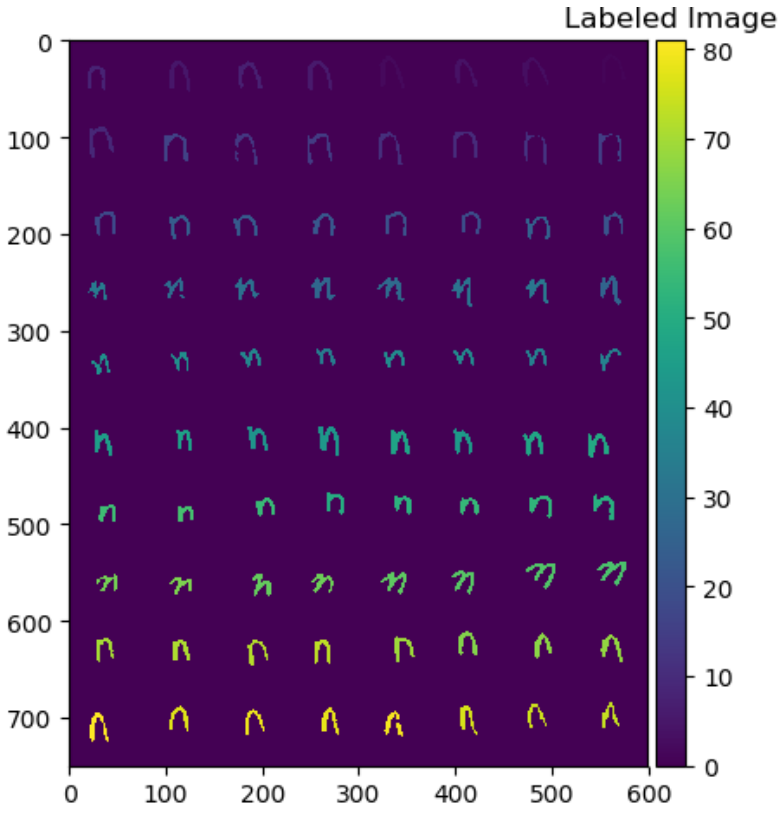

Bounding Boxes
0
100
200
300
400
500
600
700
0
100
200
300
400
500
600

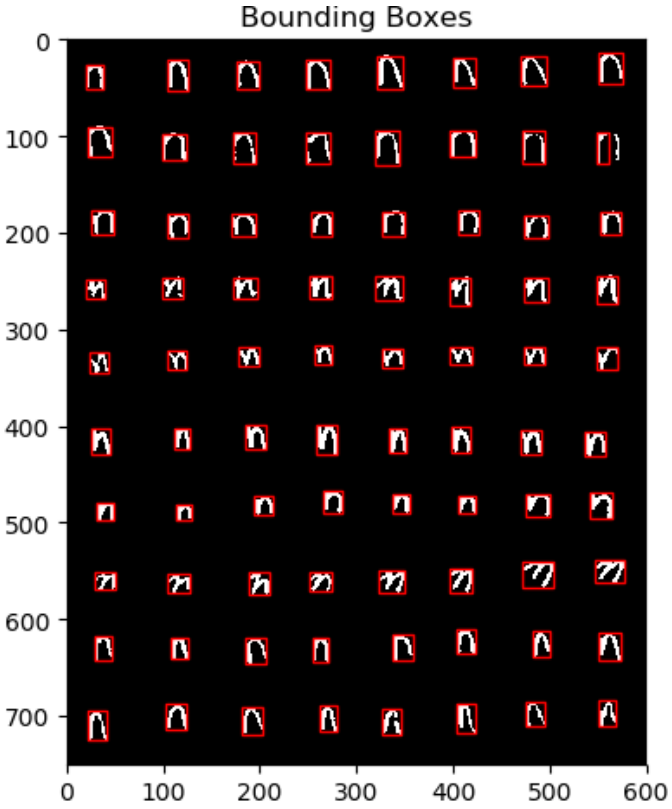
1.7 n


Original Image
0
100
200
300
400
500
600
700
0
100
200
300
400
500
600

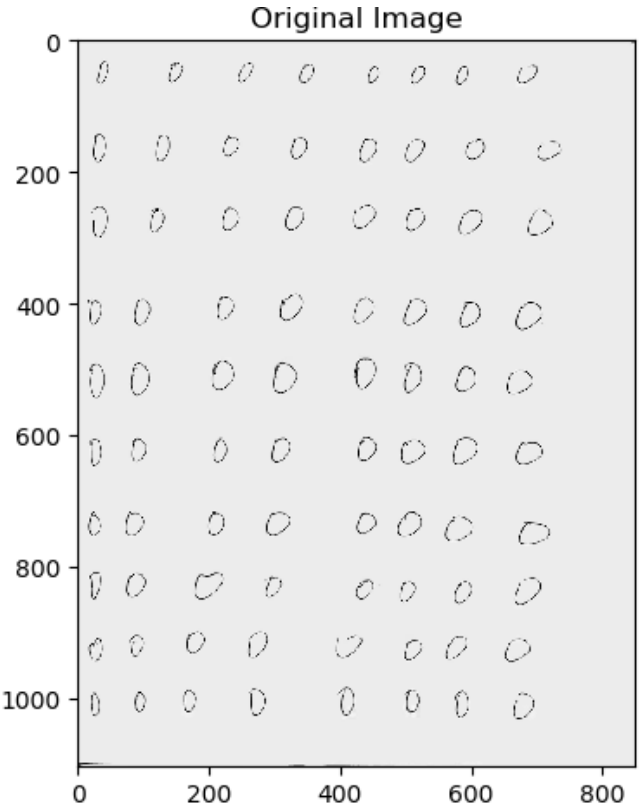

Histogram
300000
250000
200000
150000
100000
50000
0
0
50
100
150
200
250

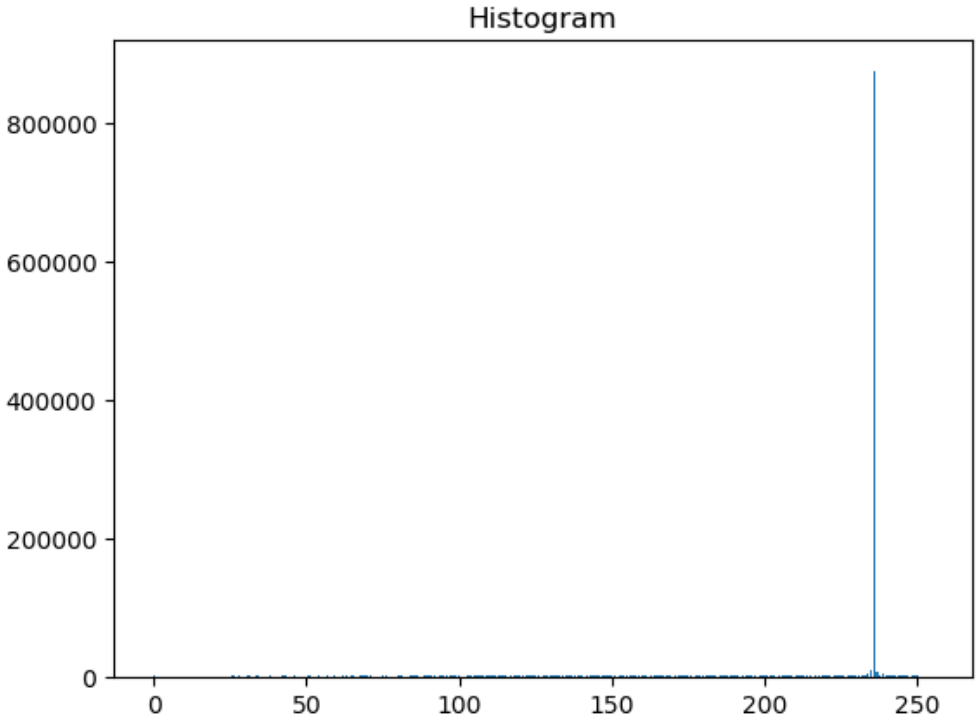

Binary Image
0
100
200
300
400
500
600
700
0
100
200
300
400
500
600

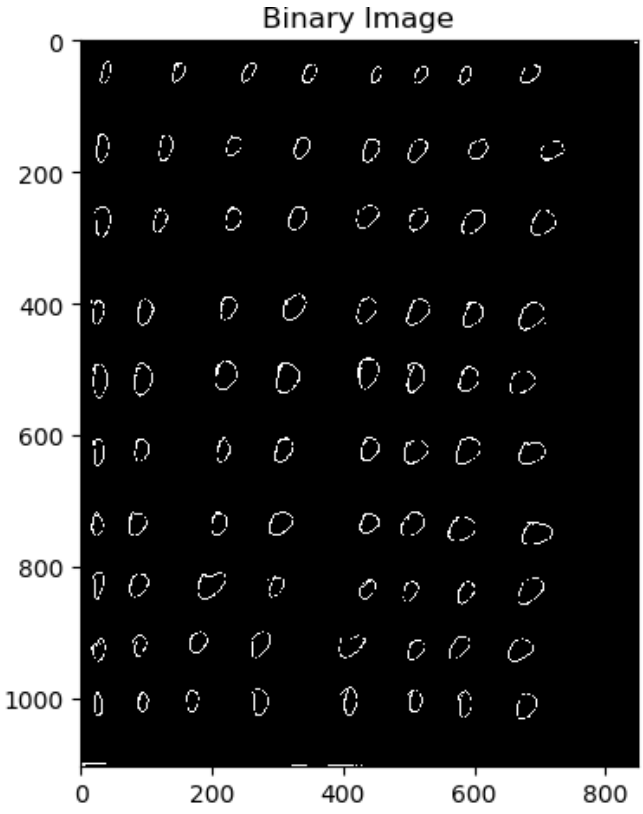

Labeled Image
0
100
200
300
400
500
600
700
0
100
200
300
400
500
600
80
70
60
50
40
30
20
10
0

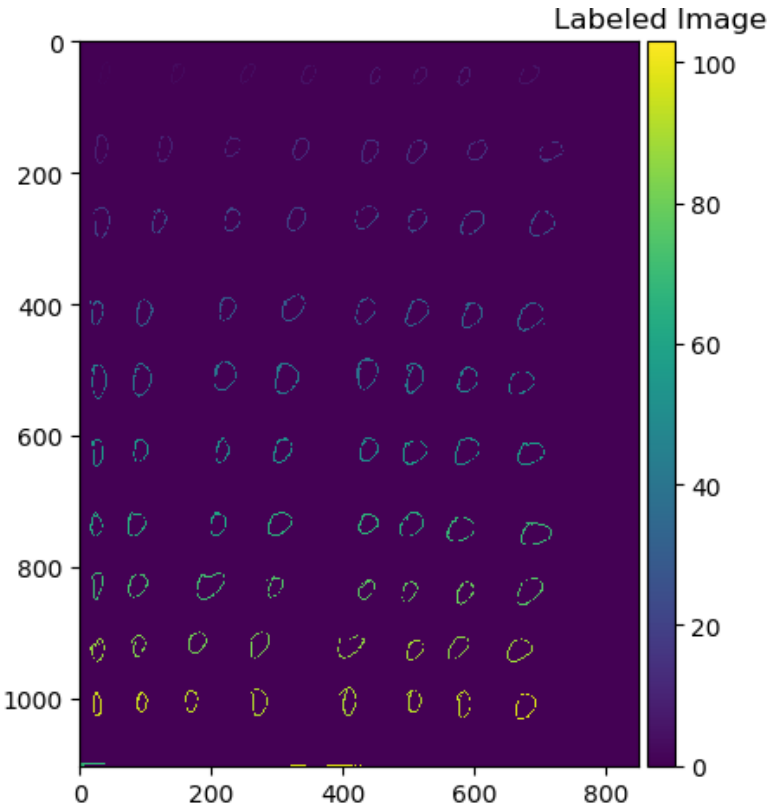

Bounding Boxes
0
100
200
300
400
500
600
700
0
100
200
300
400
500
600

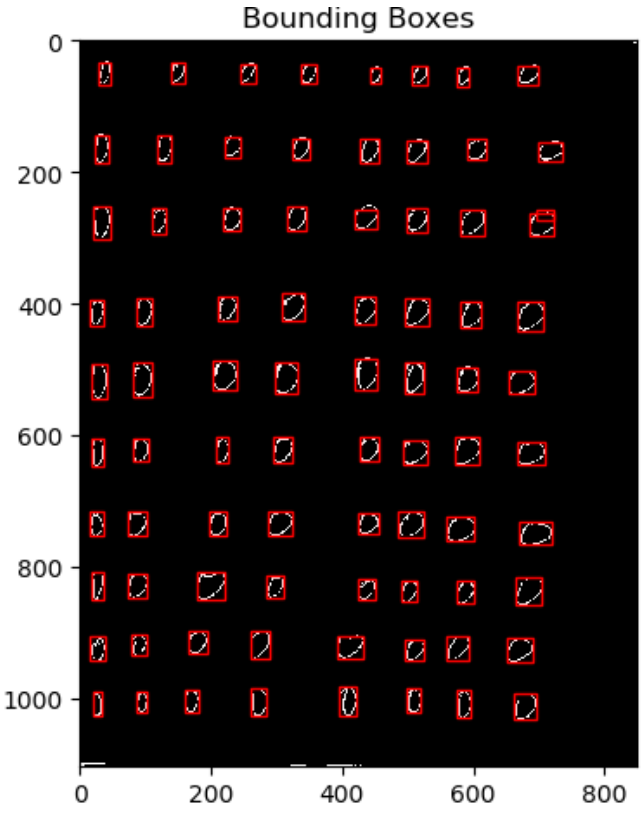
1.8 o


Original Image
0
200
400
600
800
1000
0
200
400
600
800

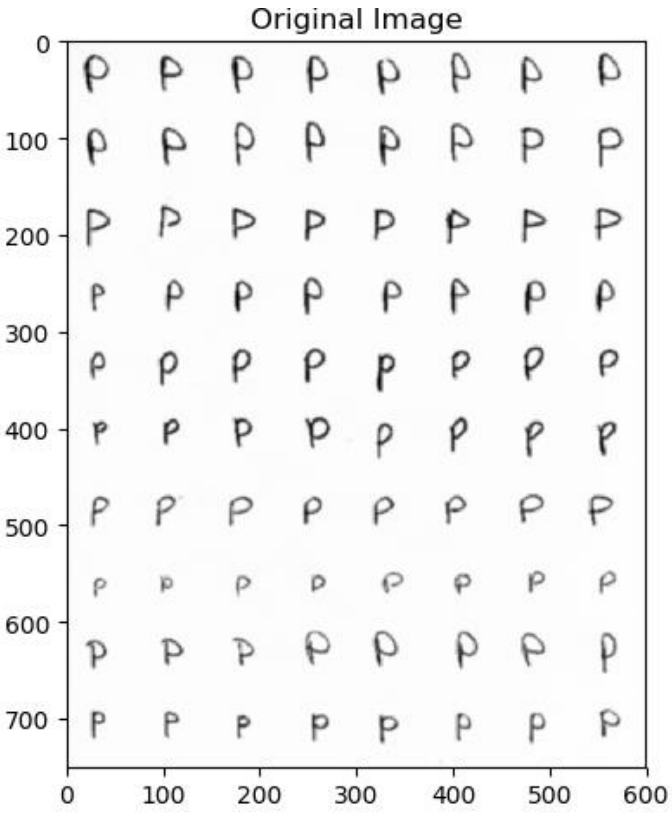

Histogram
800000
600000
400000
200000
0
0
50
100
150
200
250

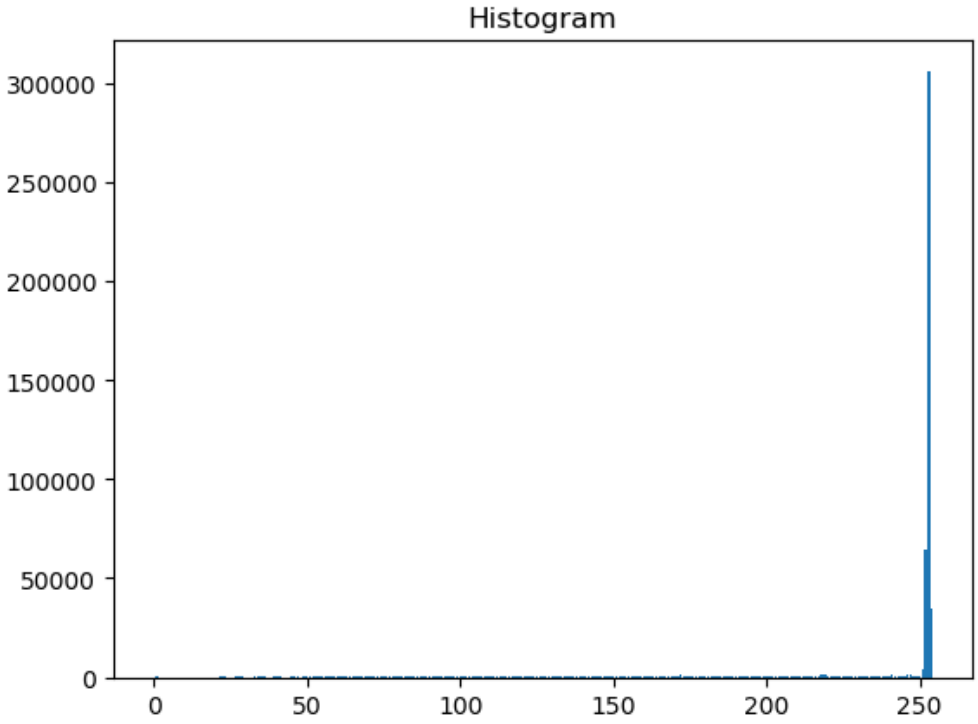

Binary Image
0
200
400
600
800
1000
0
200
400
600
800

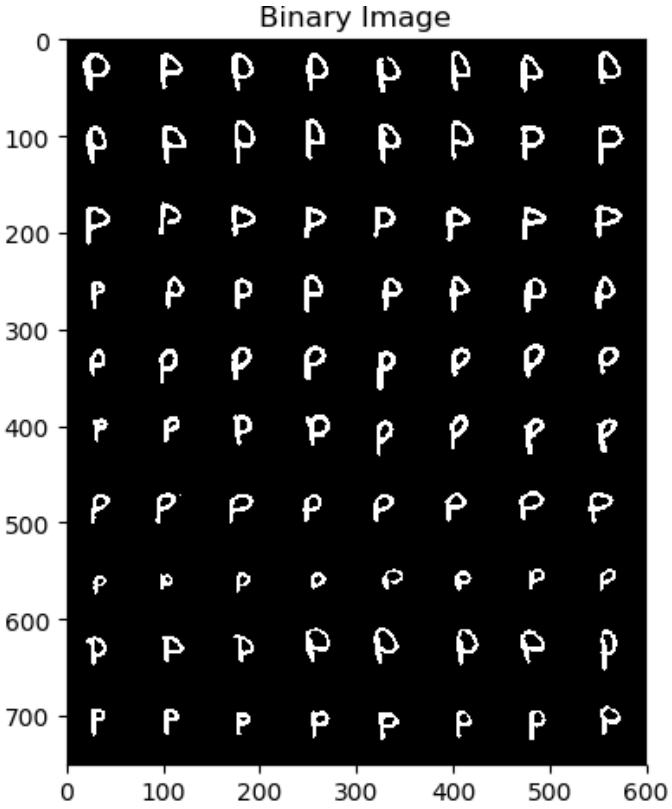

Labeled Image
0
200
400
600
800
1000
0
200
400
600
800
100
80
60
40
20
0

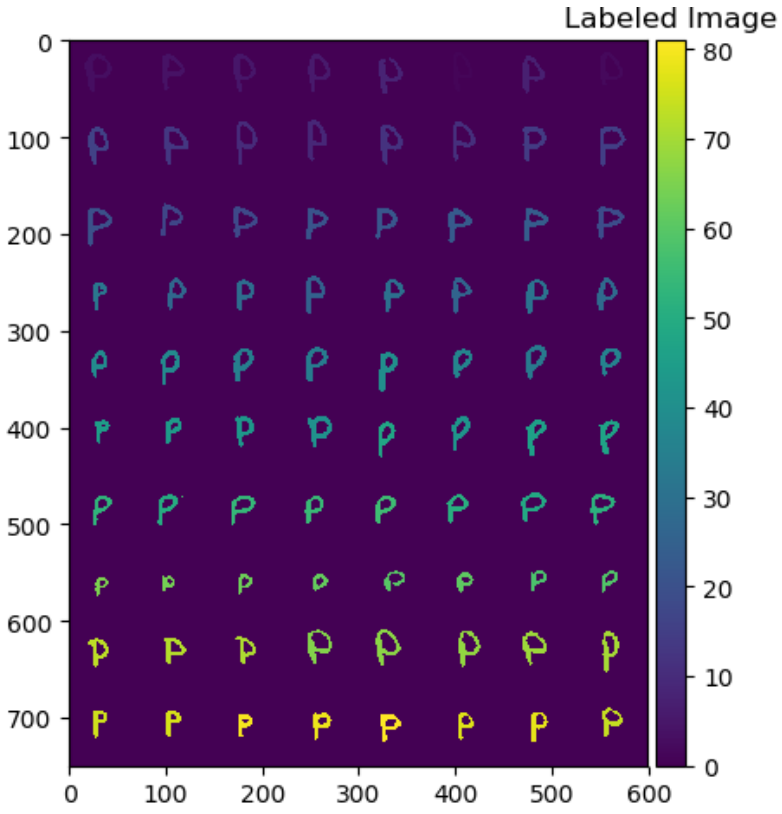

Bounding Boxes
0
200
400
600
800
1000
0
200
400
600
800

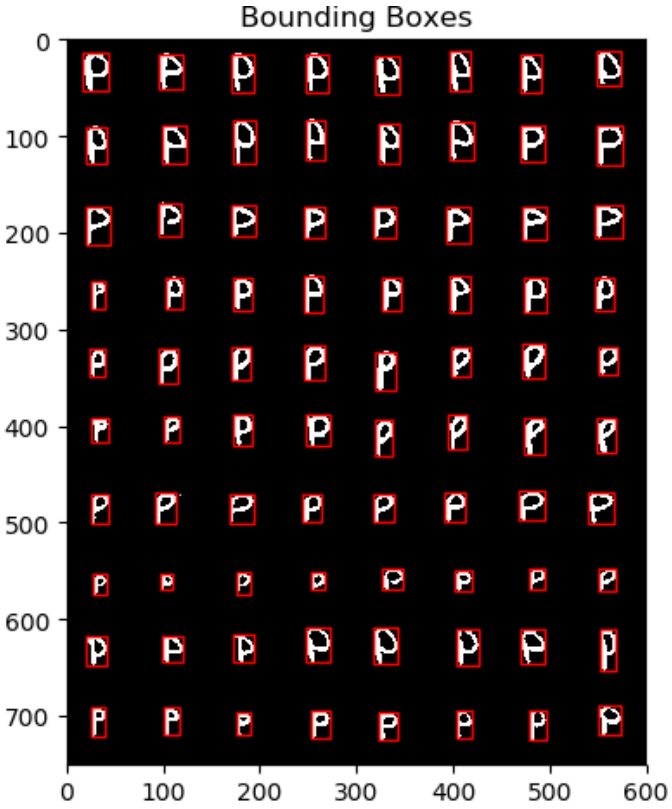
1.9 p


Original Image
0
100
200
300
400
500
600
700
0
100
200
300
400
500
600

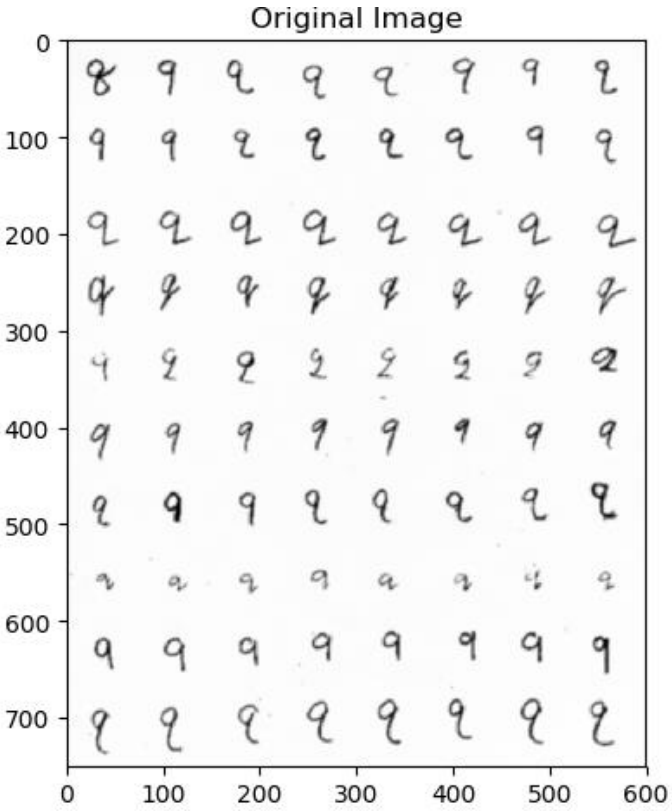

Histogram
300000
250000
200000
150000
100000
50000
0
0
50
100
150
200
250

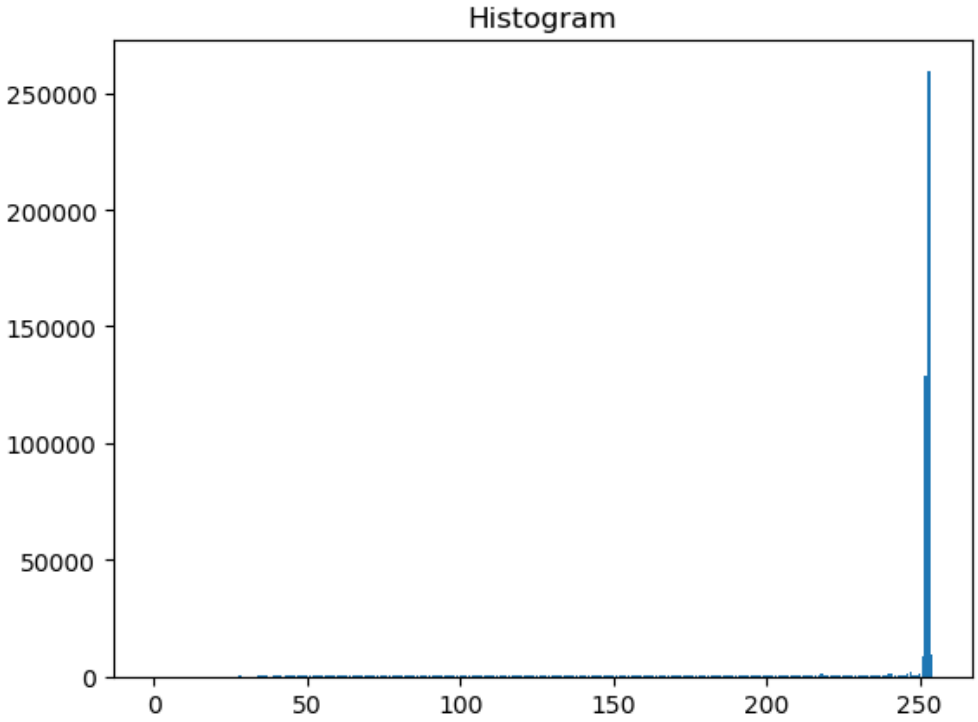

Binary Image
0
100
200
300
400
500
600
700
0
100
200
300
400
500
600

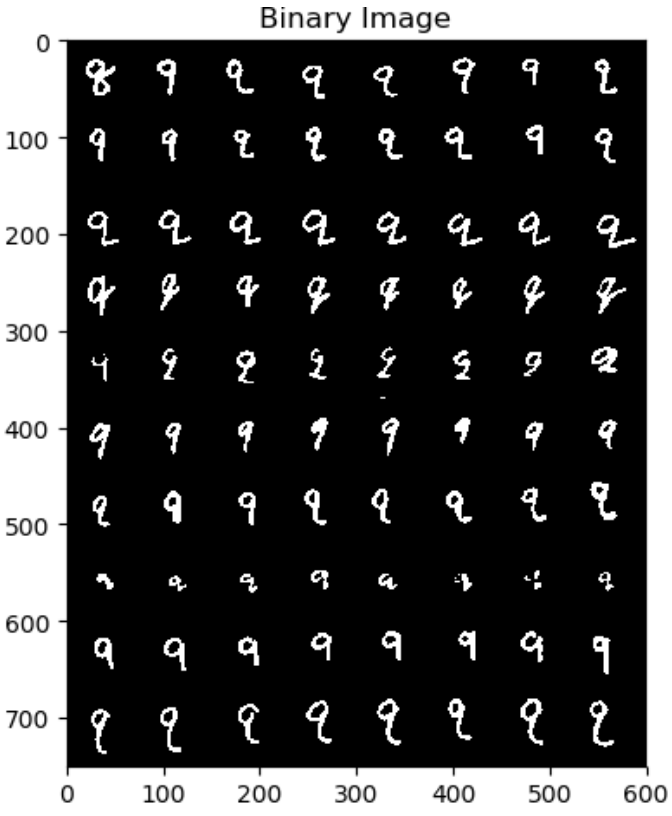

Labeled Image
0
100
200
300
400
500
600
700
0
100
200
300
400
500
600
80
70
60
50
40
30
20
10
0

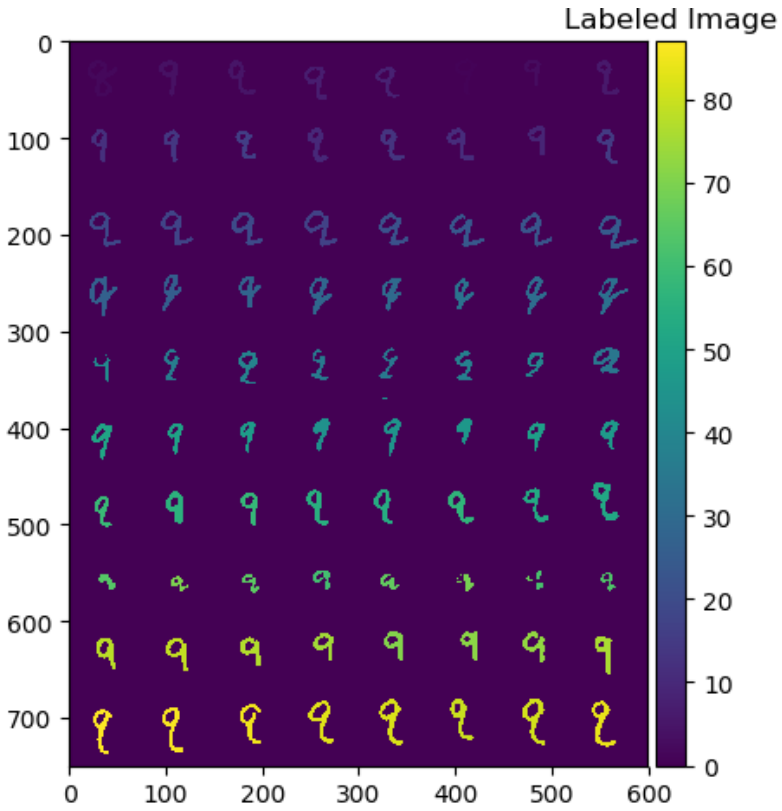

Bounding Boxes
0
100
200
300
400
500
600
700
0
100
200
300
400
500
600

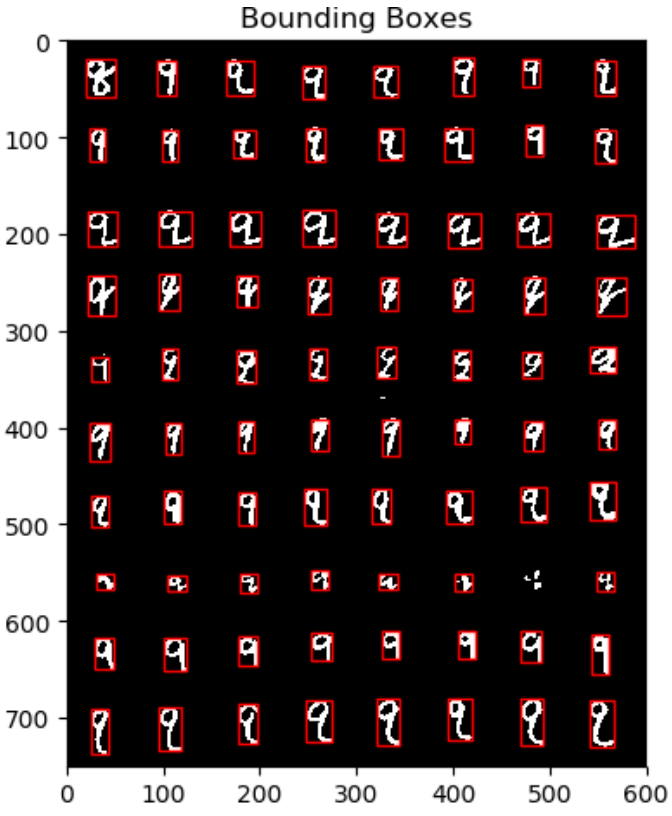
1.10 q


Original Image
0
100
200
300
400
500
600
700
0
100
200
300
400
500
600

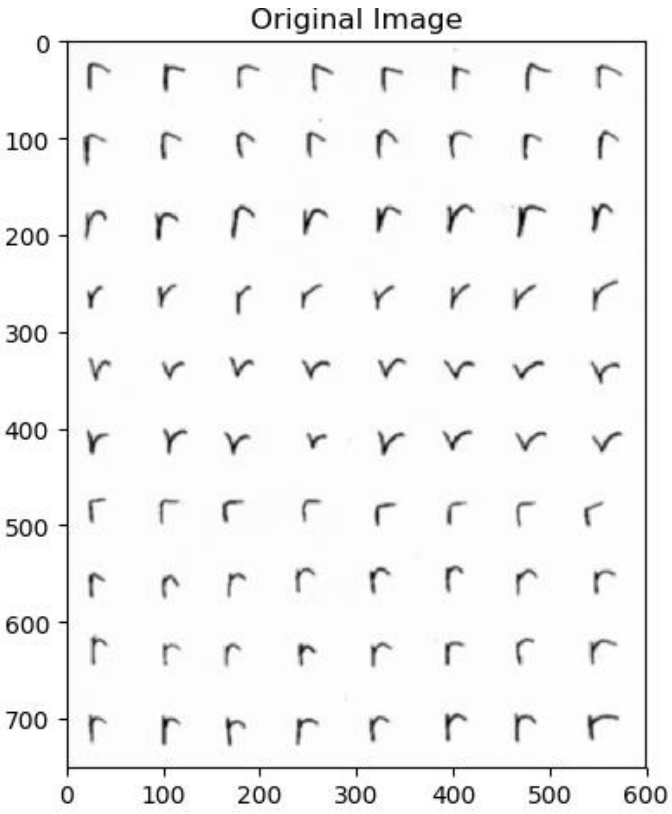

Histogram
250000
200000
150000
100000
50000
0
0
50
100
150
200
250

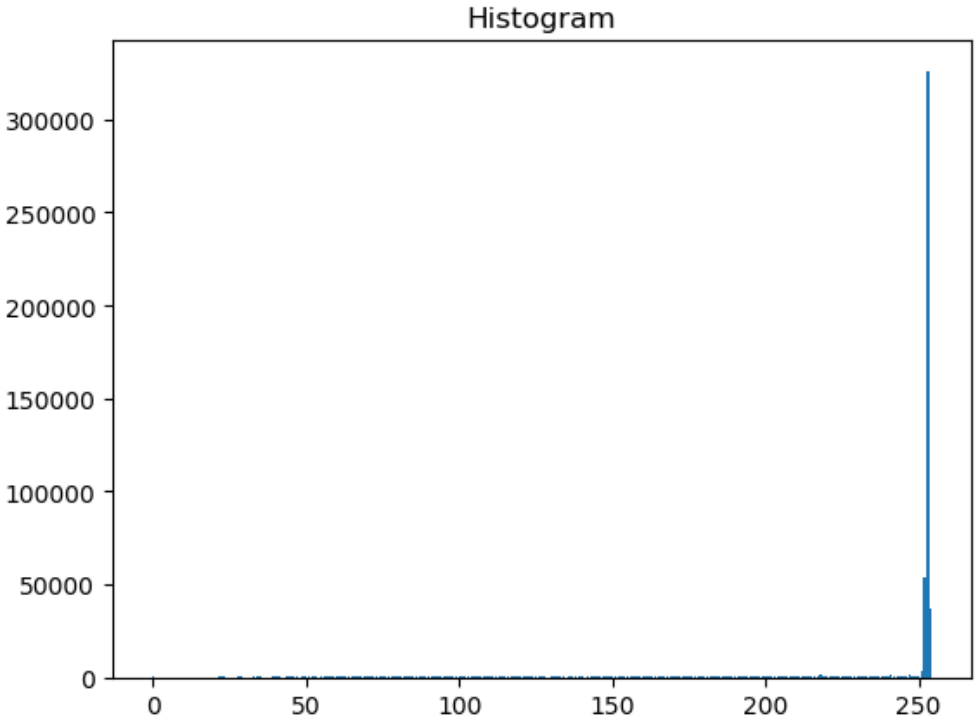

Binary Image
0
100
200
300
400
500
600
700
0
100
200
300
400
500
600

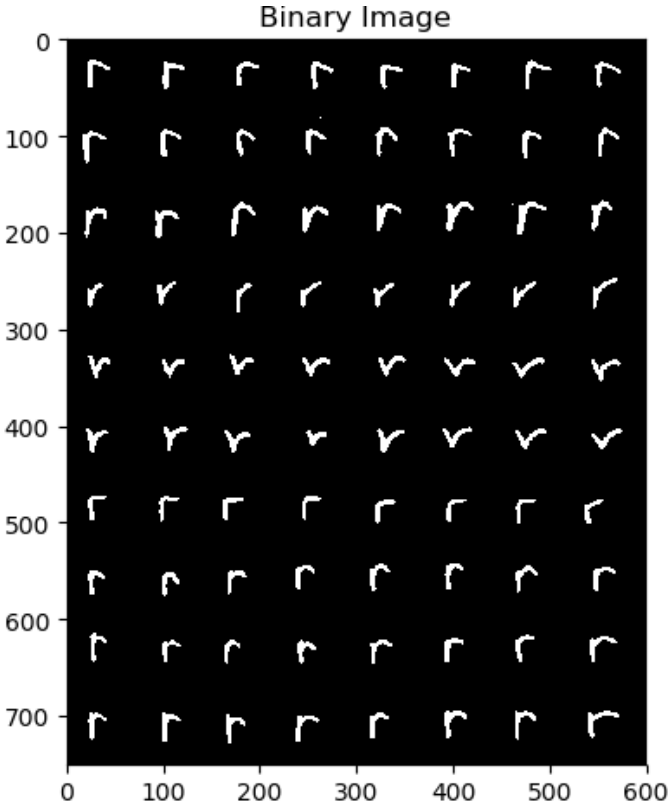

Labeled Image
0
100
200
300
400
500
600
700
0
100
200
300
400
500
600
0
10
20
30
40
50
60
70
80

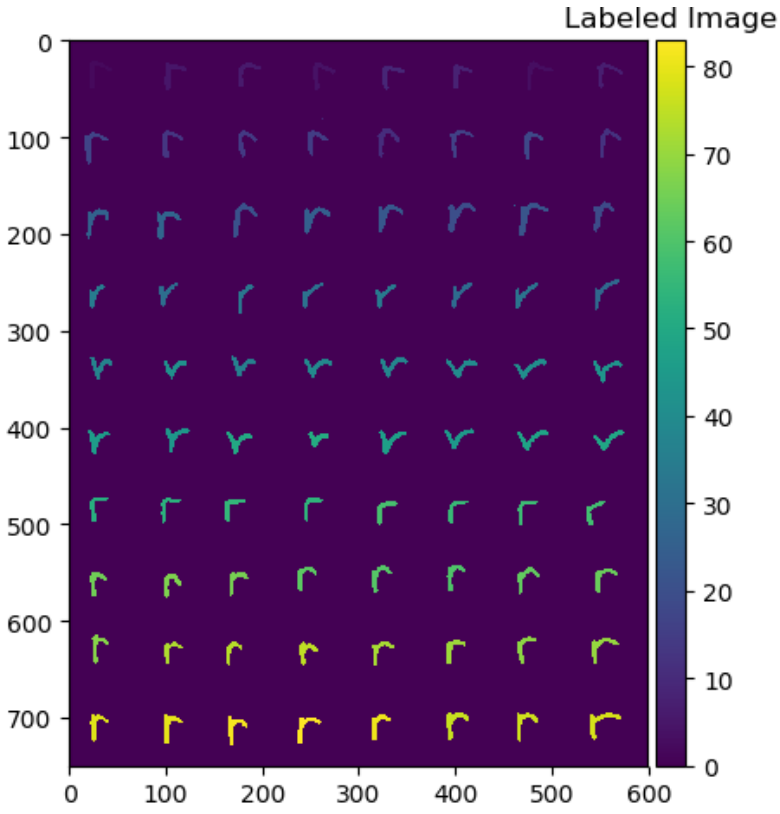

Bounding Boxes
0
100
200
300
400
500
600
700
0
100
200
300
400
500
600

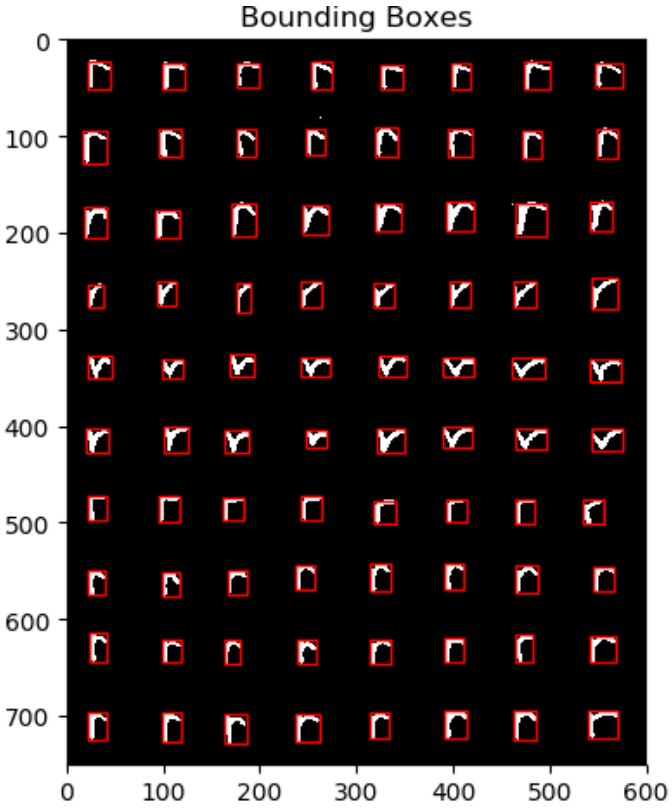
1.11 r


Original Image
0
100
200
300
400
500
600
700
0
100
200
300
400
500
600

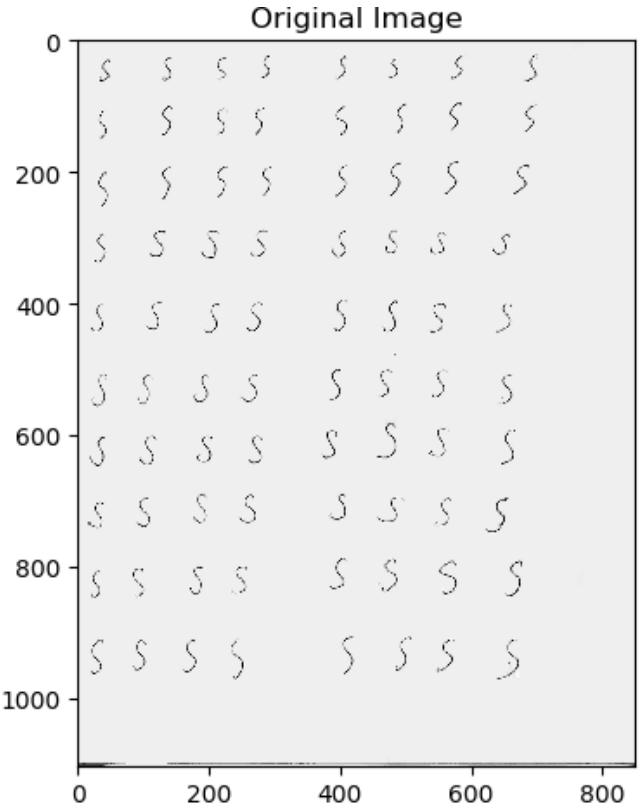

Histogram
300000
250000
200000
150000
100000
50000
0
0
50
100
150
200
250

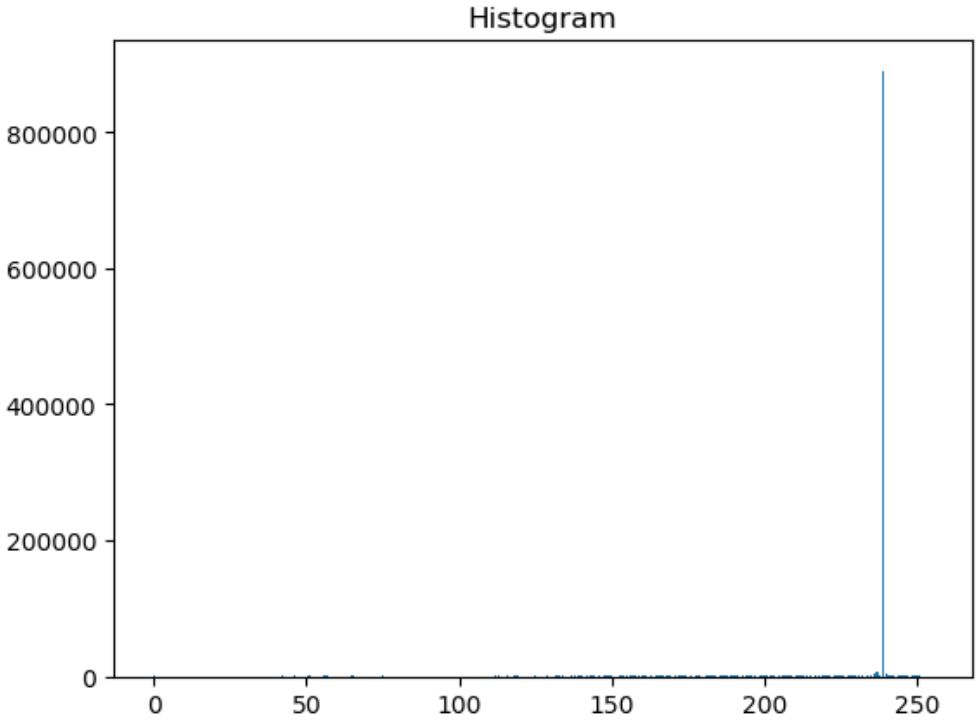

Binary Image
0
100
200
300
400
500
600
700
0
100
200
300
400
500
600

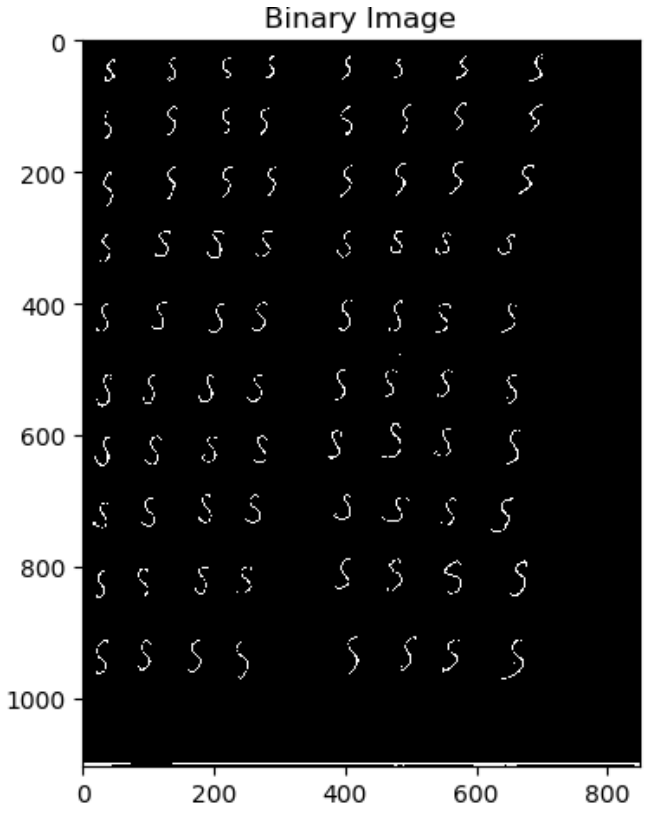

Labeled Image
0
100
200
300
400
500
600
700
0
100
200
300
400
500
600
80
70
60
50
40
30
20
10
0

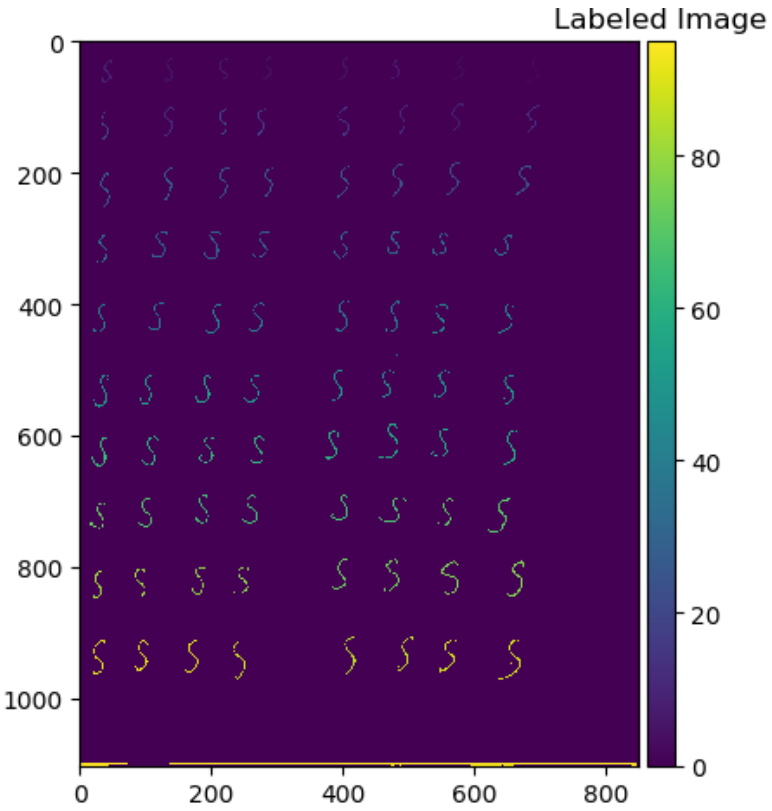

Bounding Boxes
0
100
200
300
400
500
600
700
0
100
200
300
400
500
600

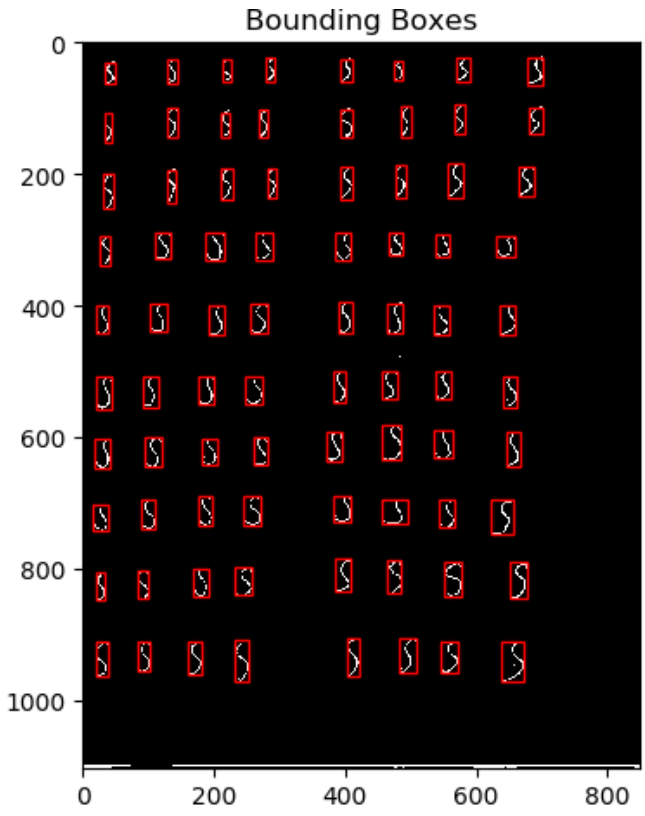
1.12 s


Original Image
0
200
400
600
800
1000
0
200
400
600
800

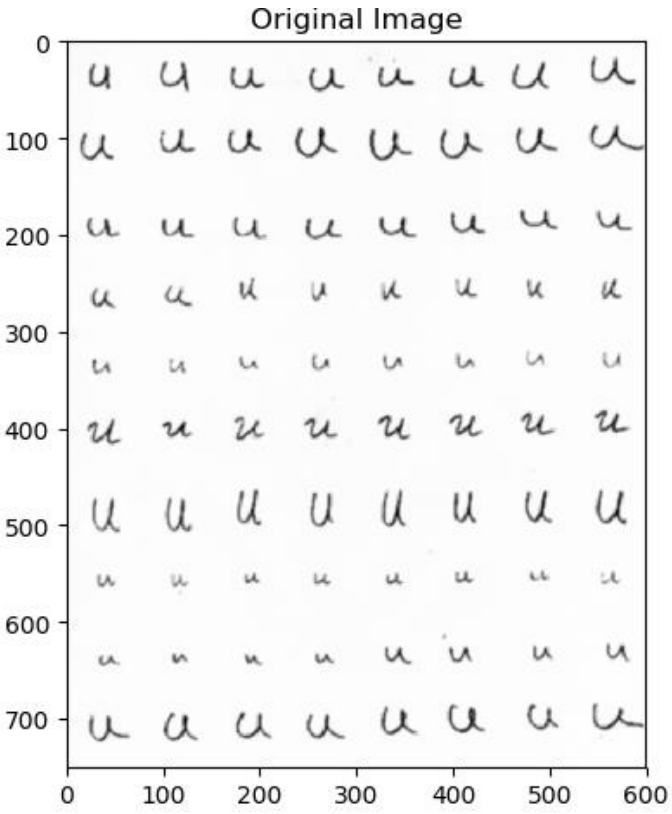

Histogram
800000
600000
400000
200000
0
0
50
100
150
200
250

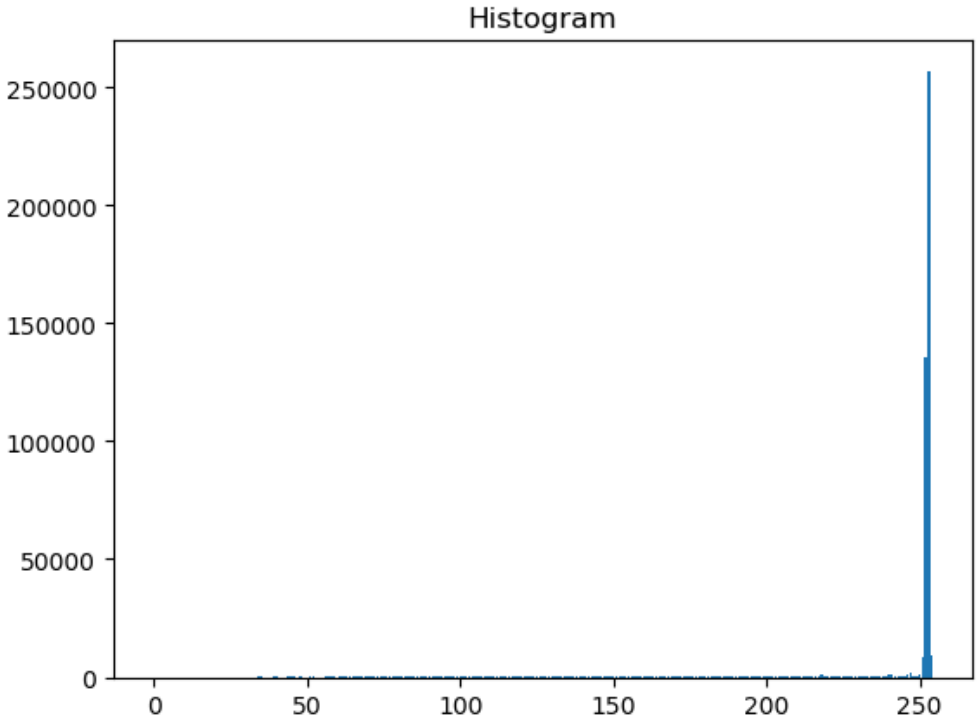

Binary Image
0
200
400
600
800
1000
0
200
400
600
800

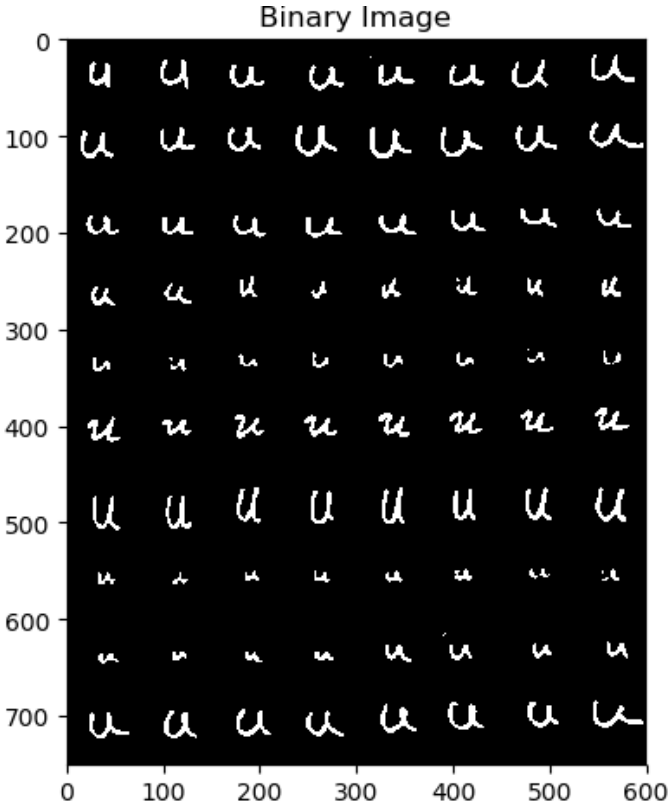

Labeled Image
0
200
400
600
800
1000
0
200
400
600
800
0
20
40
60
80

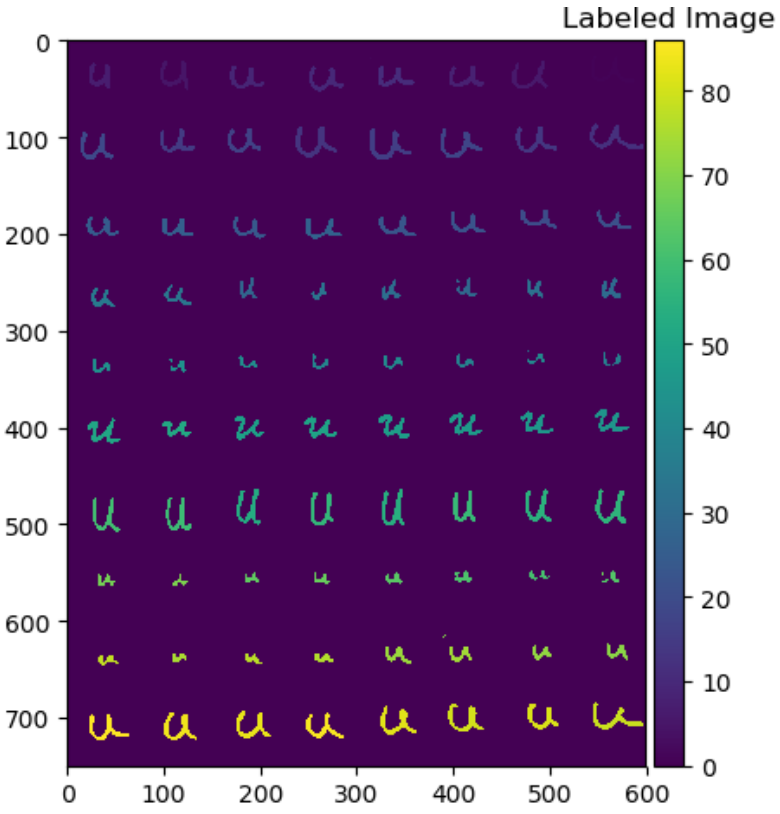

Bounding Boxes
0
200
400
600
800
1000
0
200
400
600
800

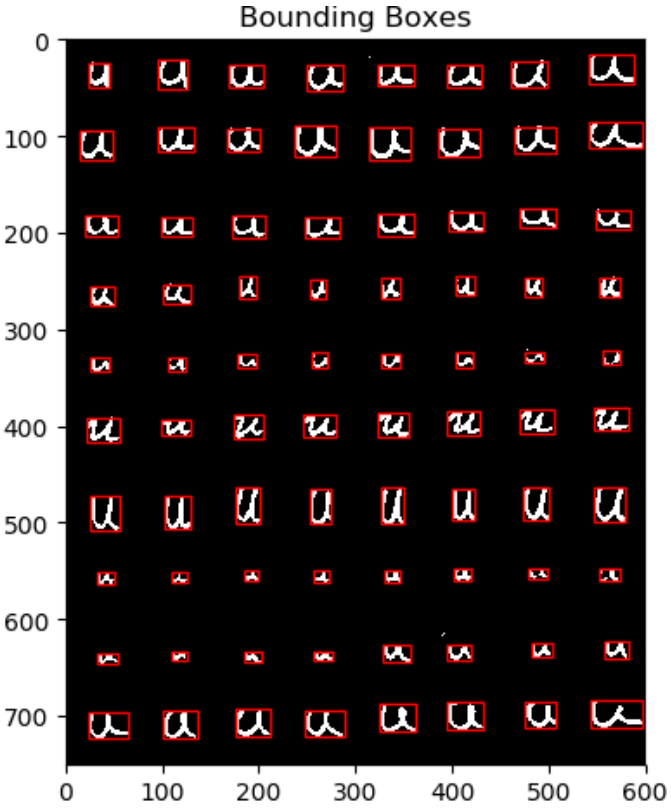
1.13 u


Original Image
0
100
200
300
400
500
600
700
0
100
200
300
400
500
600

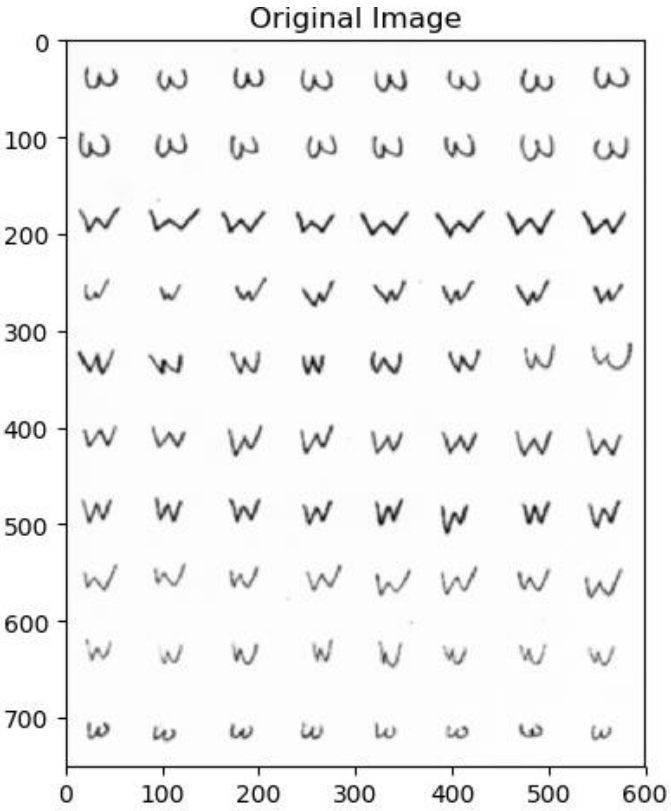

Histogram
250000
200000
150000
100000
50000
0
0
50
100
150
200
250

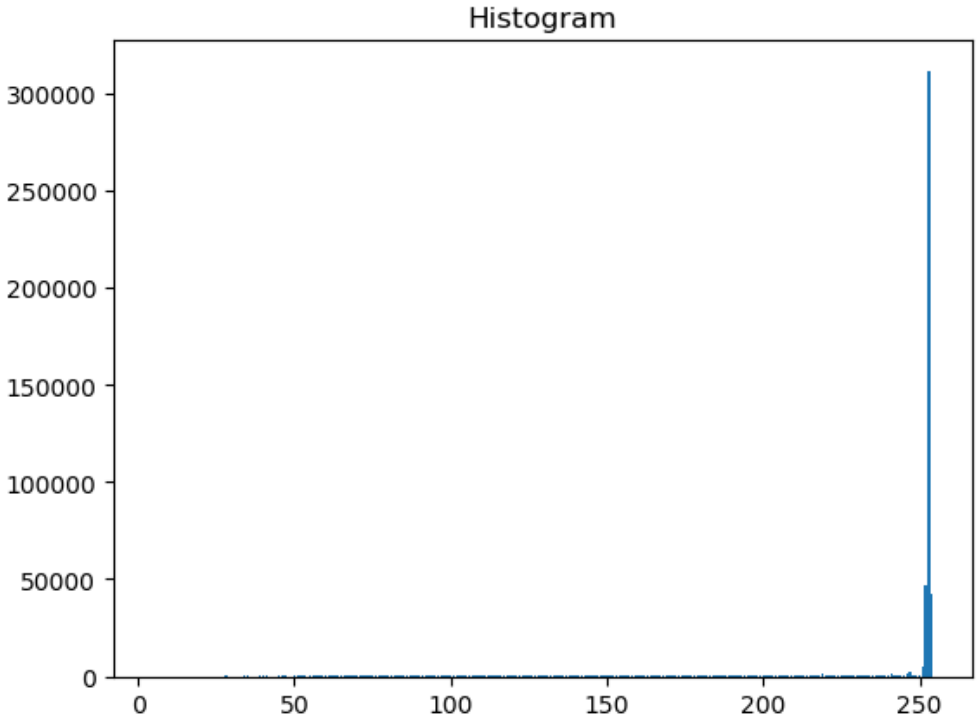

Binary Image
0
100
200
300
400
500
600
700
0
100
200
300
400
500
600

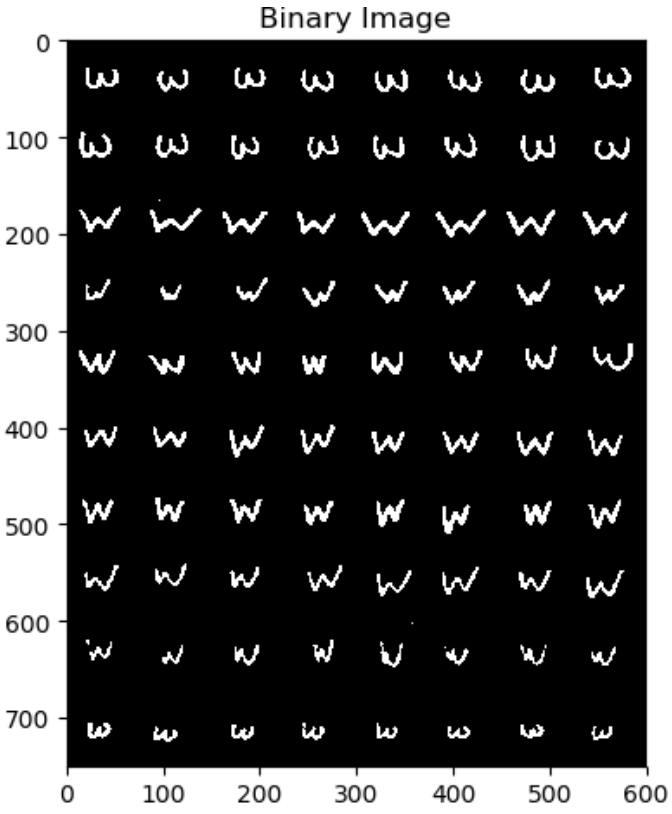

Labeled Image
0
100
200
300
400
500
600
700
0
100
200
300
400
500
600
80
70
60
50
40
30
20
10
0

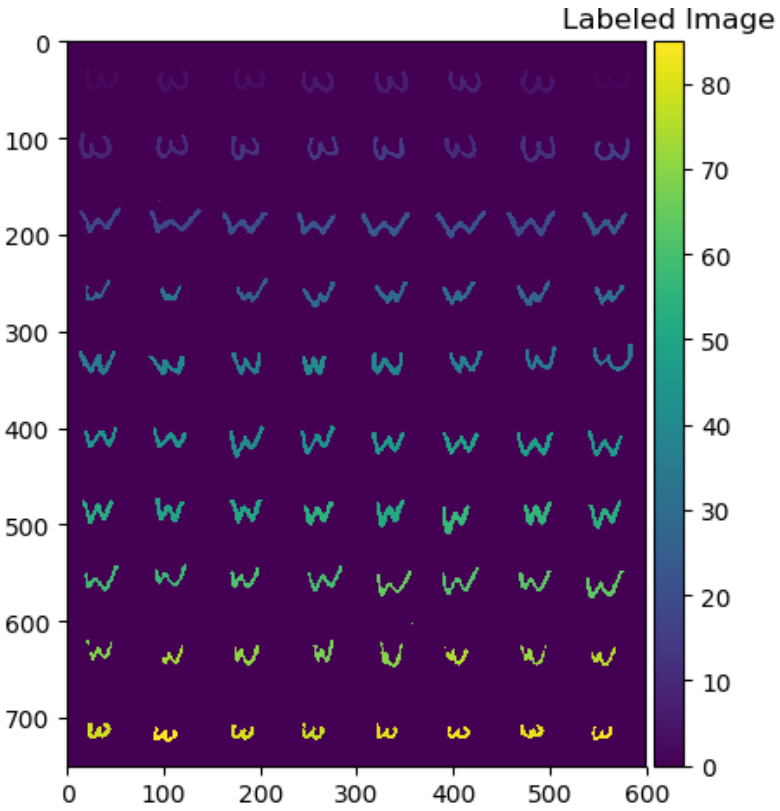

Bounding Boxes
0
100
200
300
400
500
600
700
0
100
200
300
400
500
600

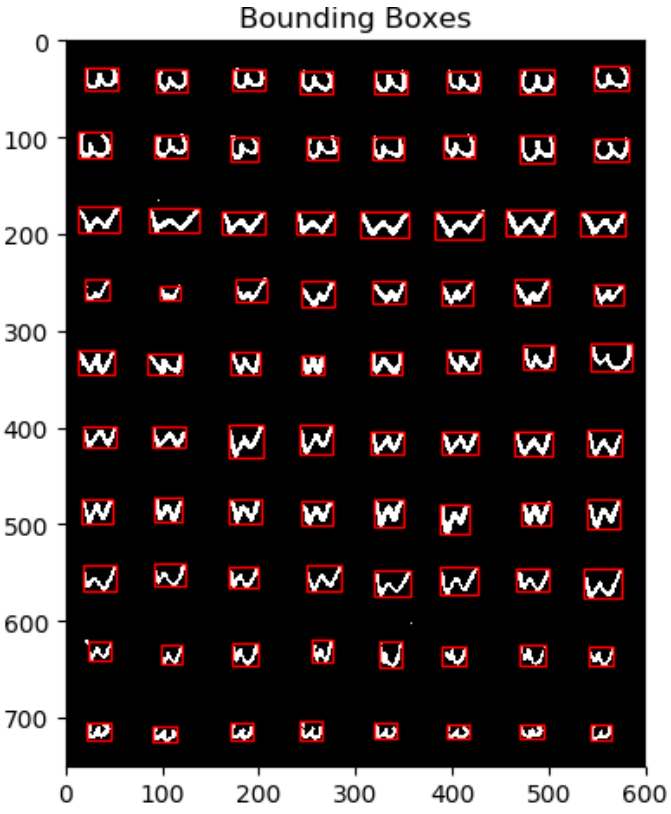
1.14 w


Original Image
0
100
200
300
400
500
600
700
0
100
200
300
400
500
600

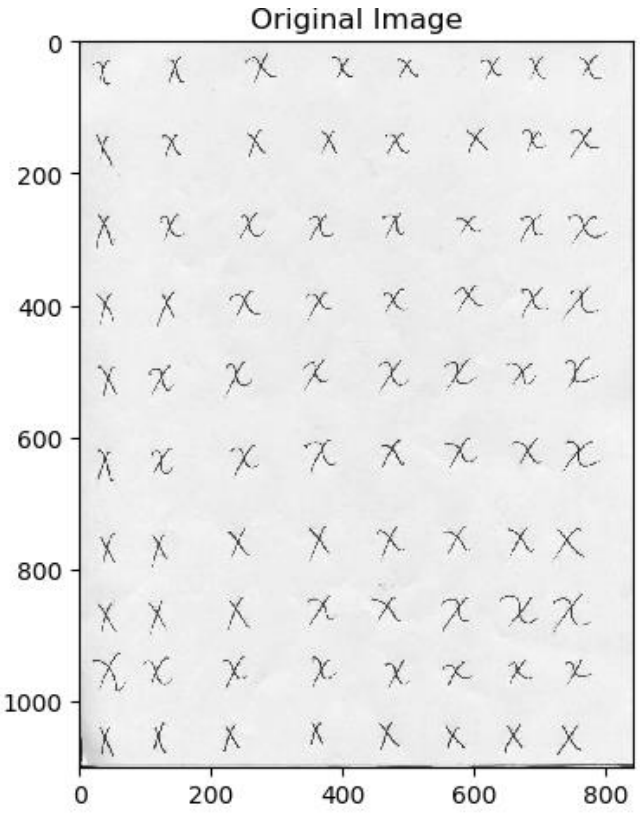

Histogram
300000
250000
200000
150000
100000
50000
0
0
50
100
150
200
250

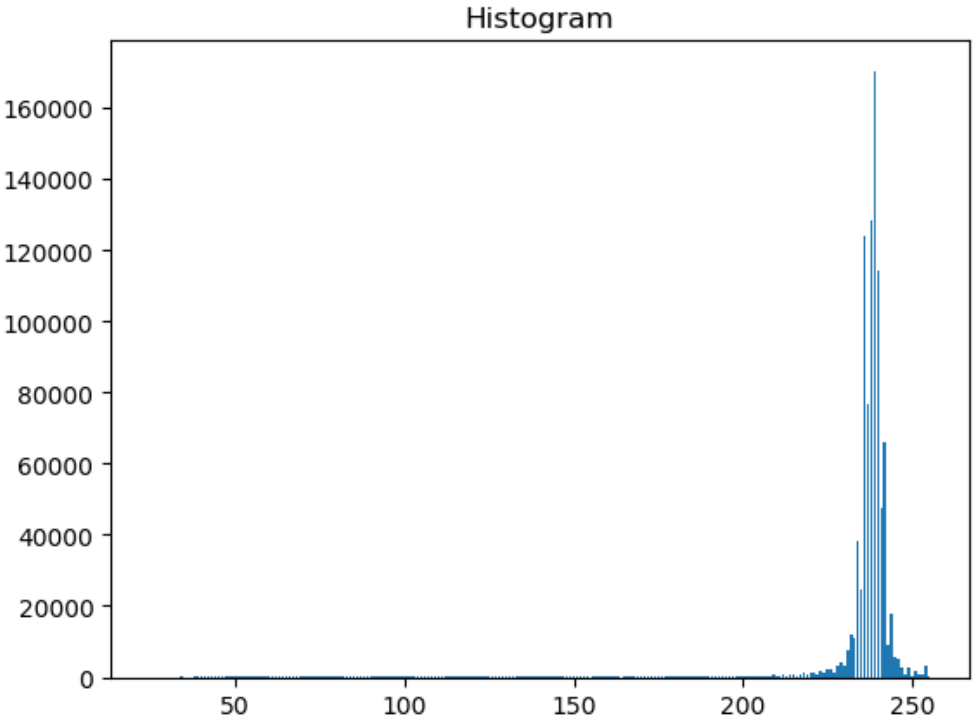

Binary Image
0
100
200
300
400
500
600
700
0
100
200
300
400
500
600

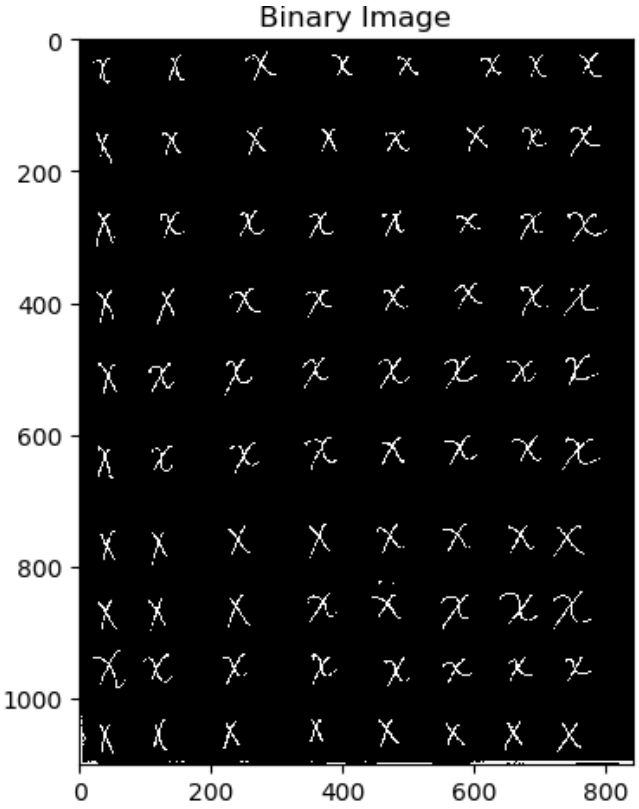

Labeled Image
0
100
200
300
400
500
600
700
0
100
200
300
400
500
600
0
10
20
30
40
50
60
70
80

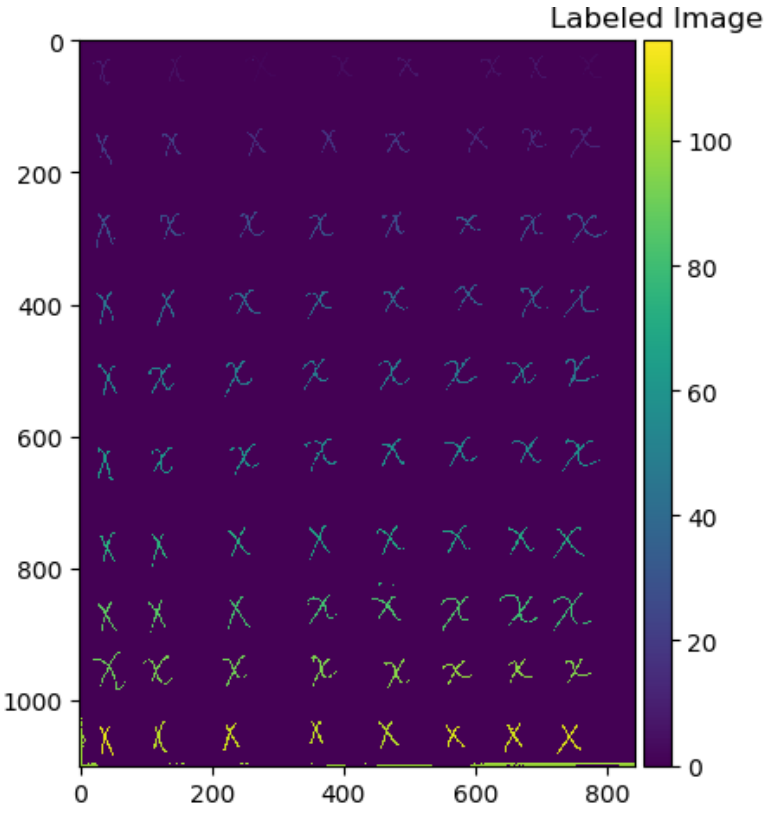

Bounding Boxes
0
100
200
300
400
500
600
700
0
100
200
300
400
500
600

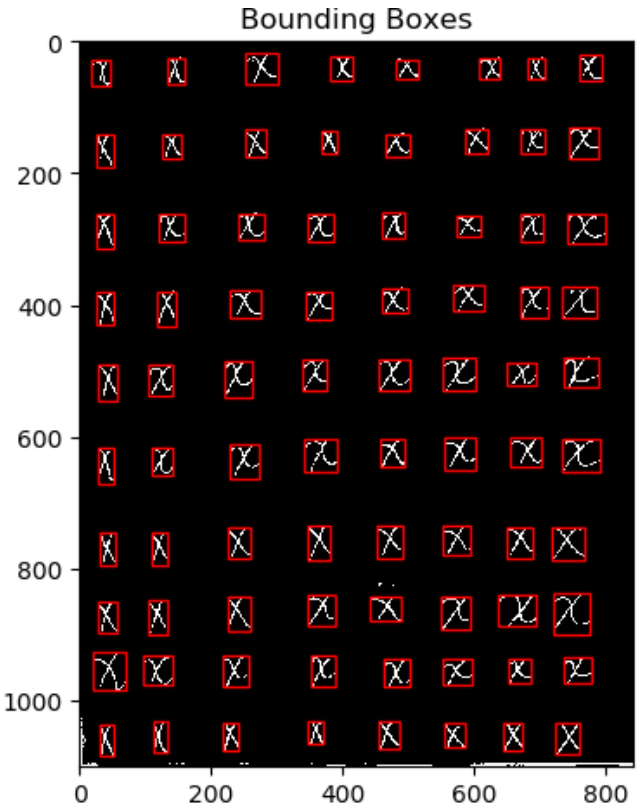
1.15 x


Original Image
0
200
400
600
800
1000
0
200
400
600
800

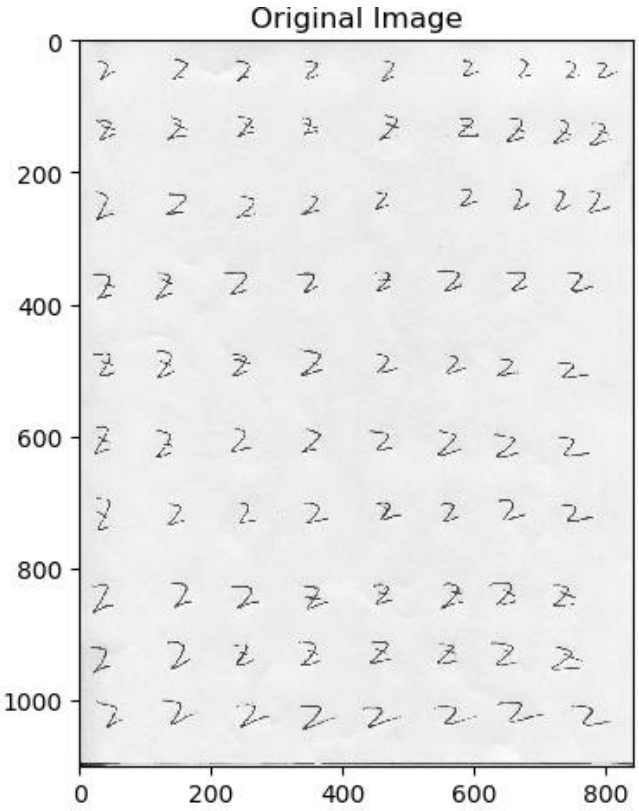

Histogram
160000
140000
120000
100000
80000
60000
40000
20000
0
50
100
150
200
250

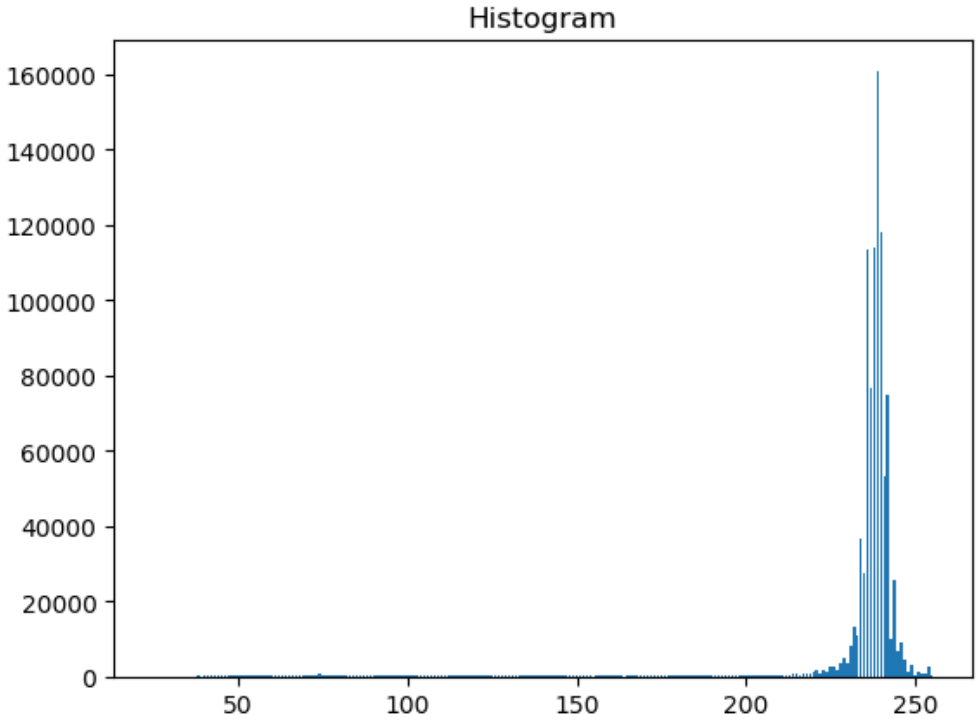

Binary Image
0
200
400
600
800
1000
0
200
400
600
800

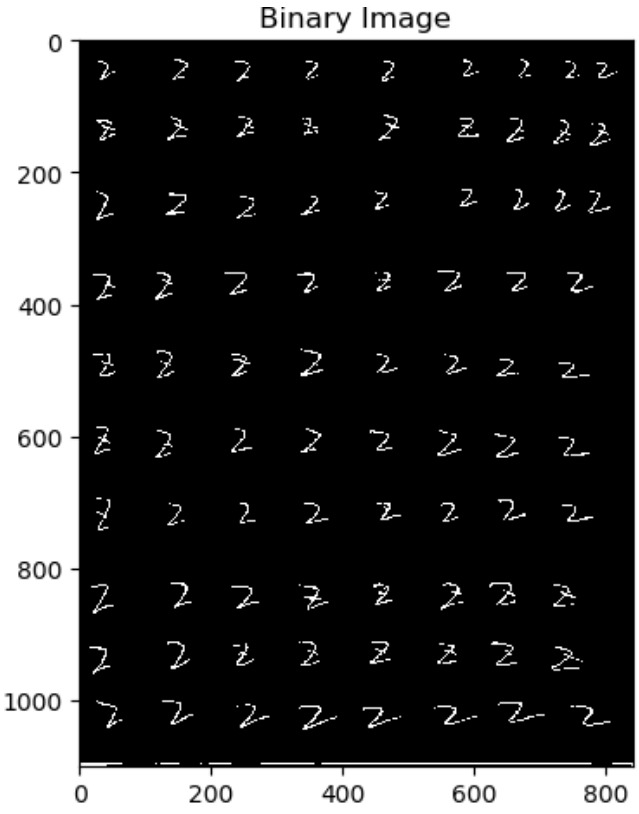

Labeled Image
0
200
400
600
800
1000
0
200
400
600
800
100
80
60
40
20
0

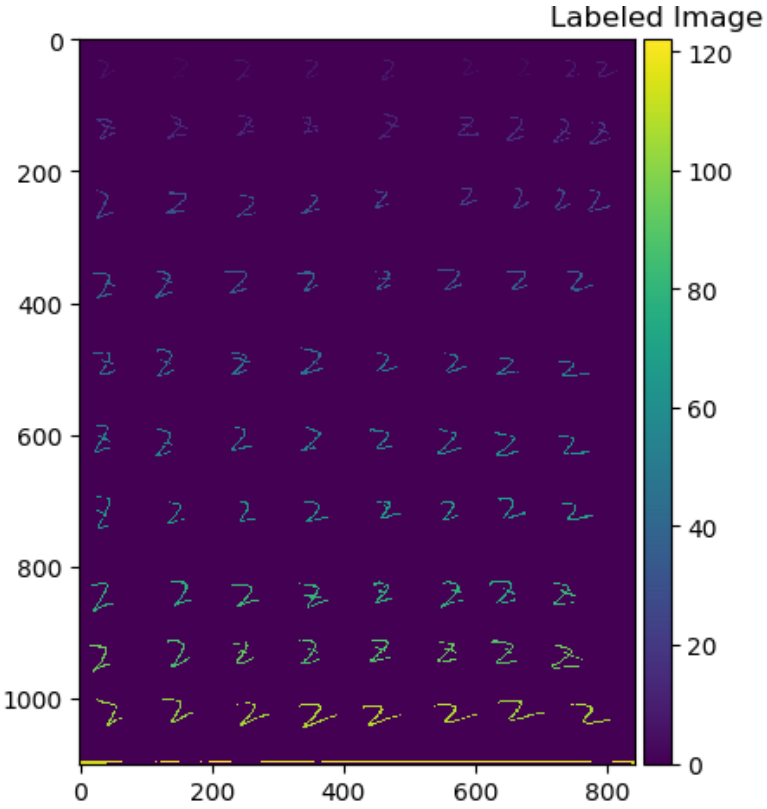

Bounding Boxes
0
200
400
600
800
1000
0
200
400
600
800

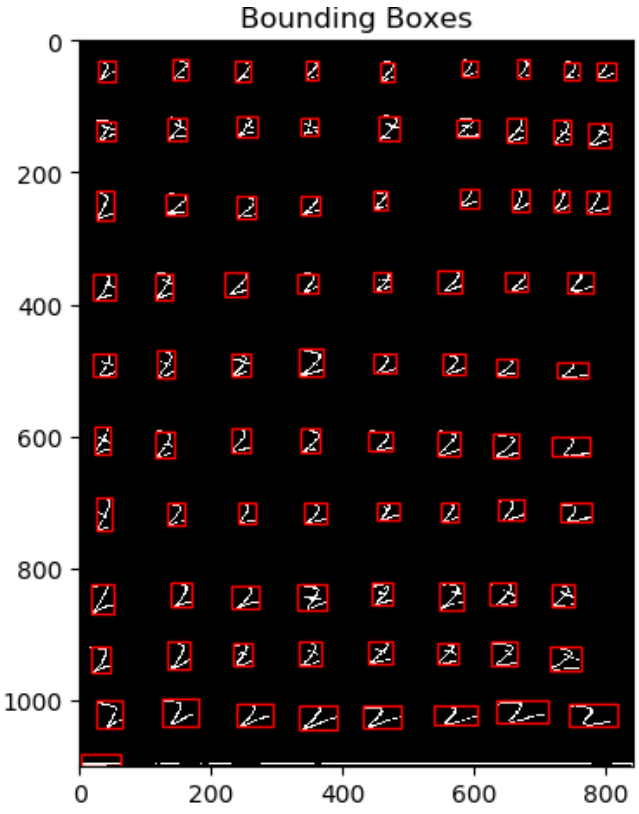
1.16 z


Original Image
0
200
400
600
800
1000
0
200
400
600
800

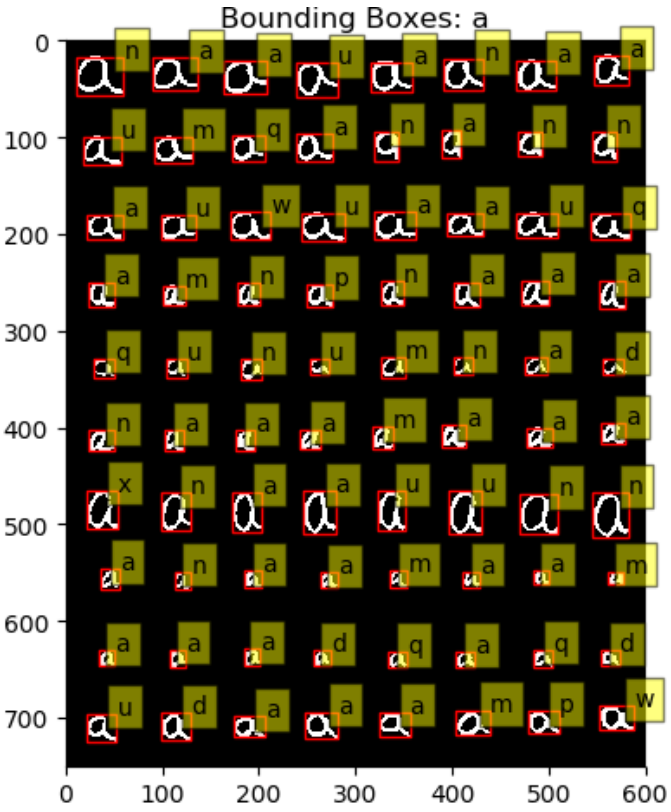

Histogram
160000
140000
120000
100000
80000
60000
40000
20000
0
50
100
150
200
250

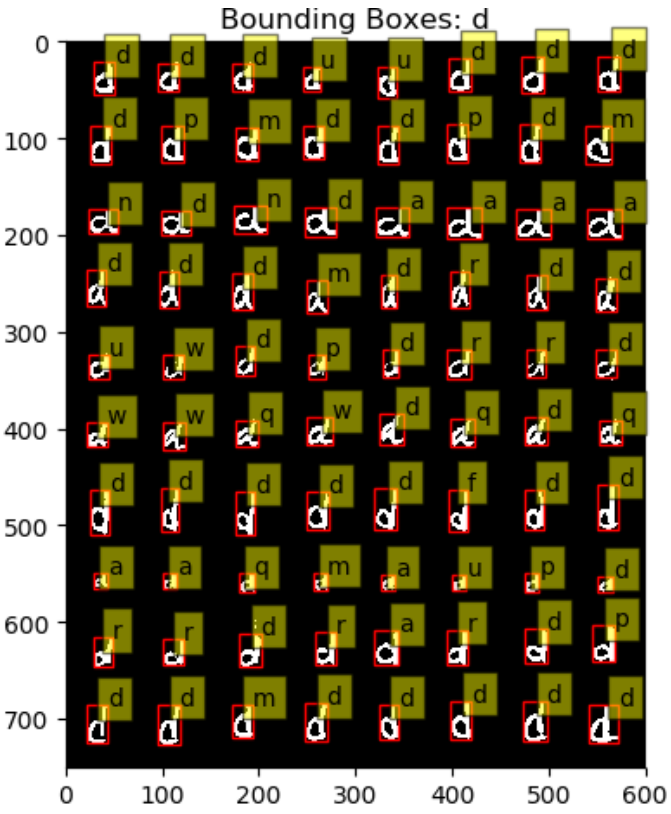

Binary Image
0
200
400
600
800
1000
0
200
400
600
800

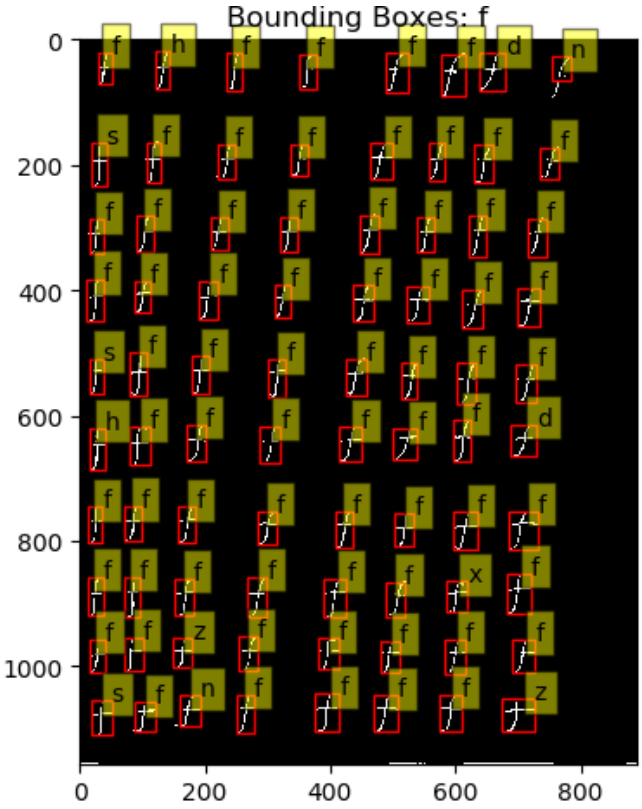

Labeled Image
0
200
400
600
800
1000
0
200
400
600
800
120
100
80
60
40
20
0

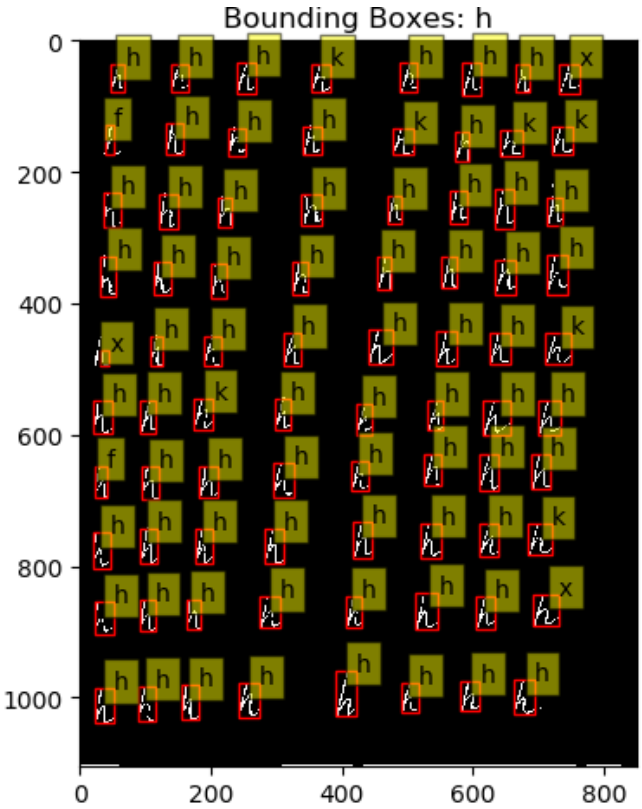

Bounding Boxes
0
200
400
600
800
1000
0
200
400
600
800

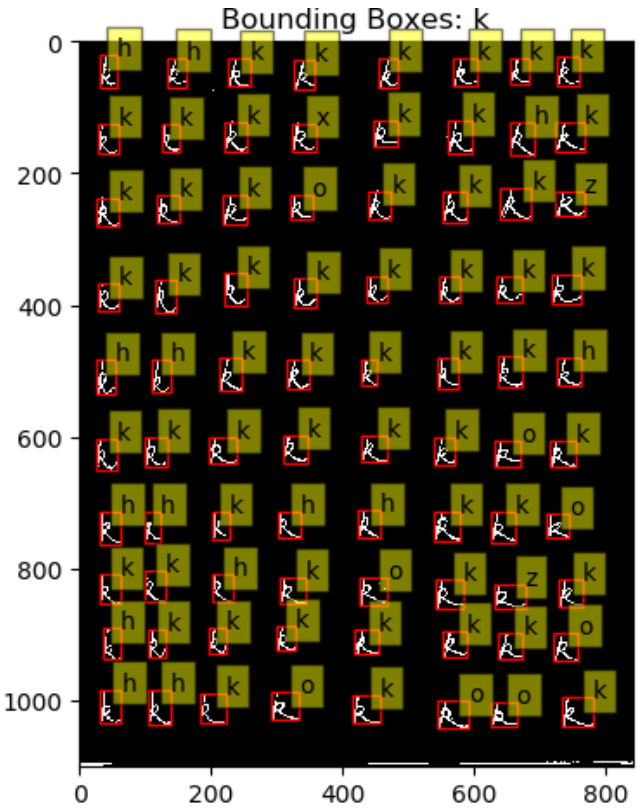
1.17 Connected component image with bounding boxes and recognition results

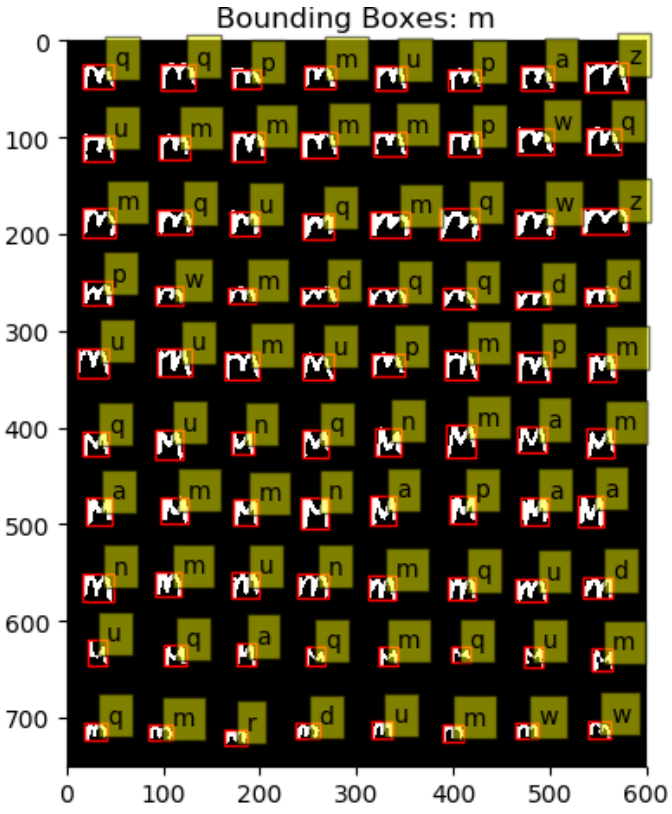


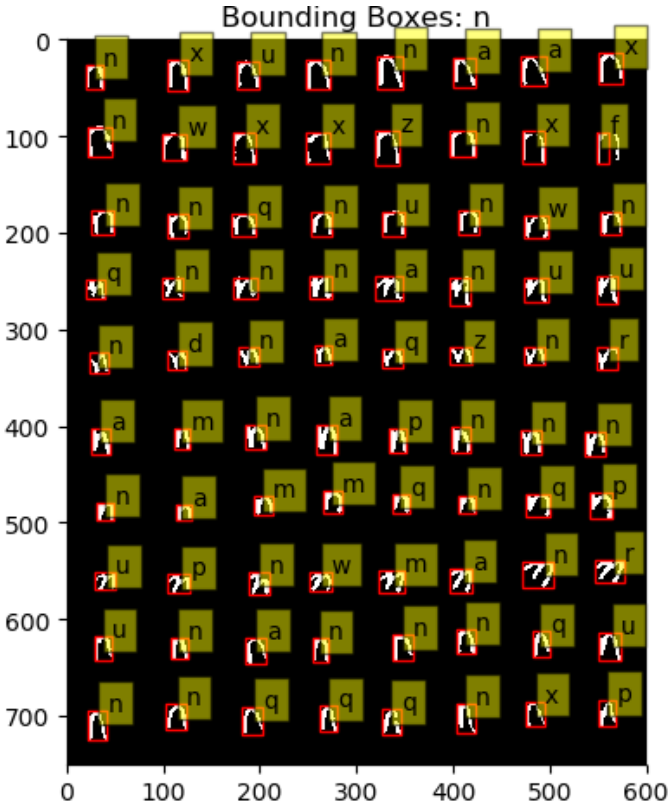



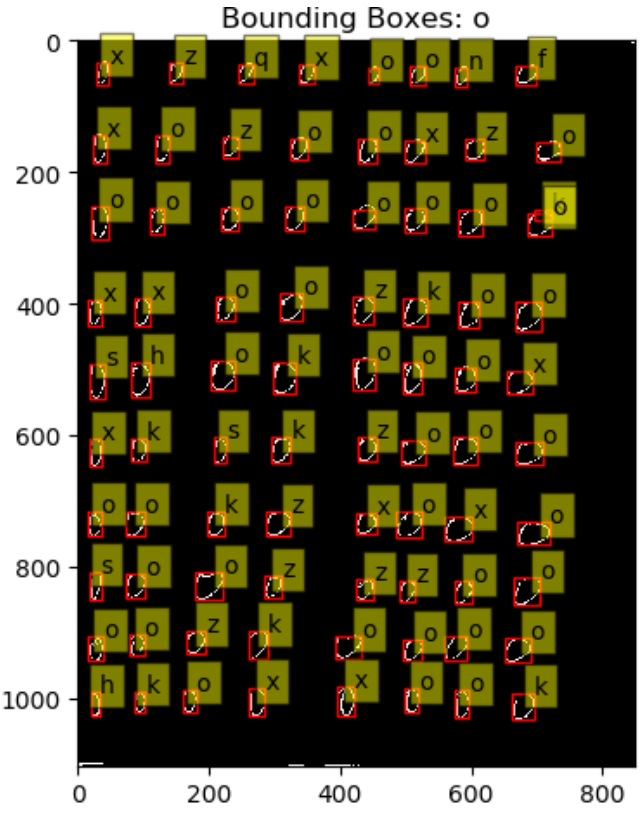

Bounding Boxes: f
0
200
400
600
800
1000
0
200
400
600
800

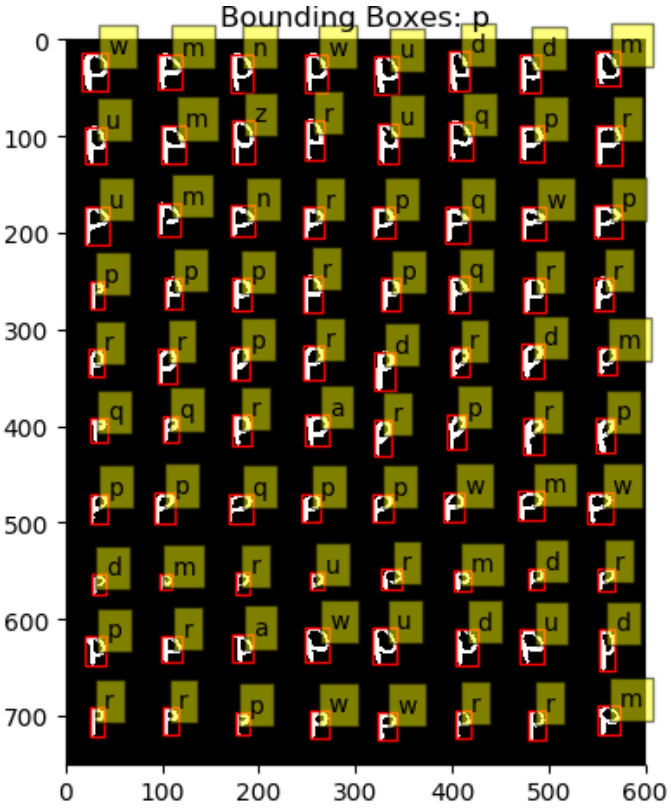

Bounding Boxes: h
0
200
400
600
800
1000
0
200
400
600
800

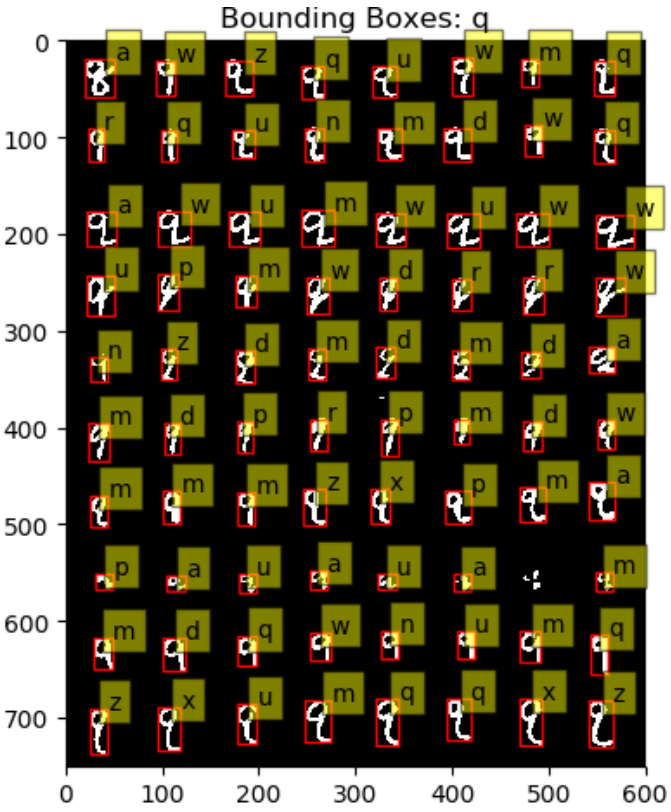

Bounding Boxes: k
0
200
400
600
800
1000
0
200
400
600
800

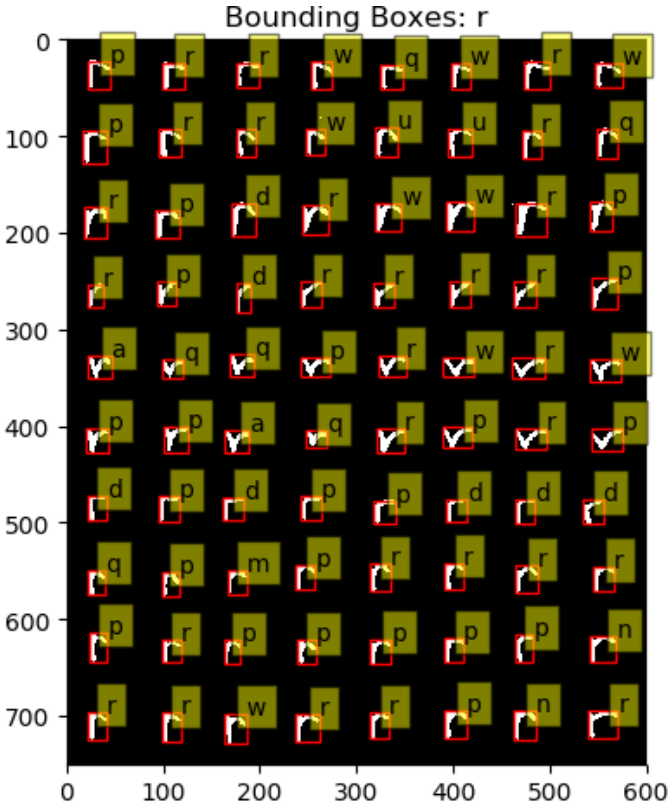

Bounding Boxes: m
0
100
200
300
400
500
600
700
0
100
200
300
400
500
600

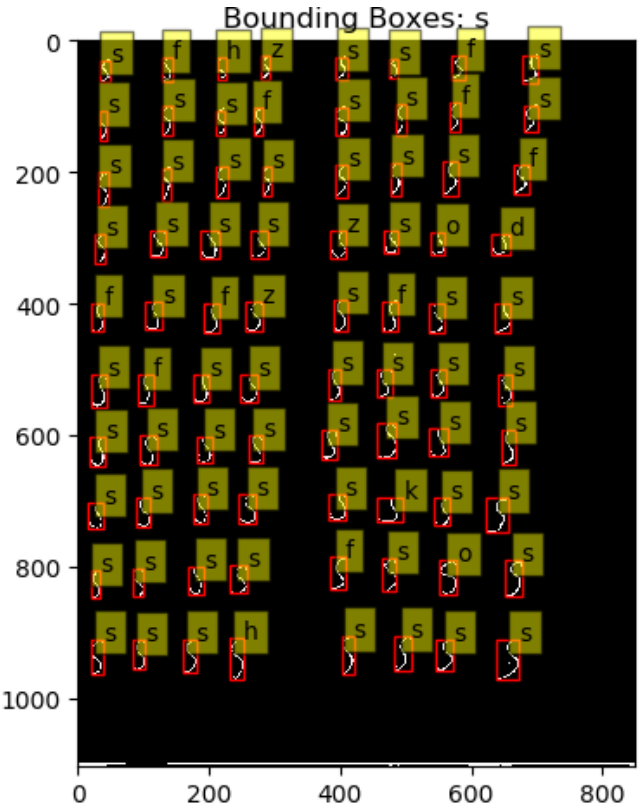

Bounding Boxes: n
0
100
200
300
400
500
600
700
0
100
200
300
400
500
600

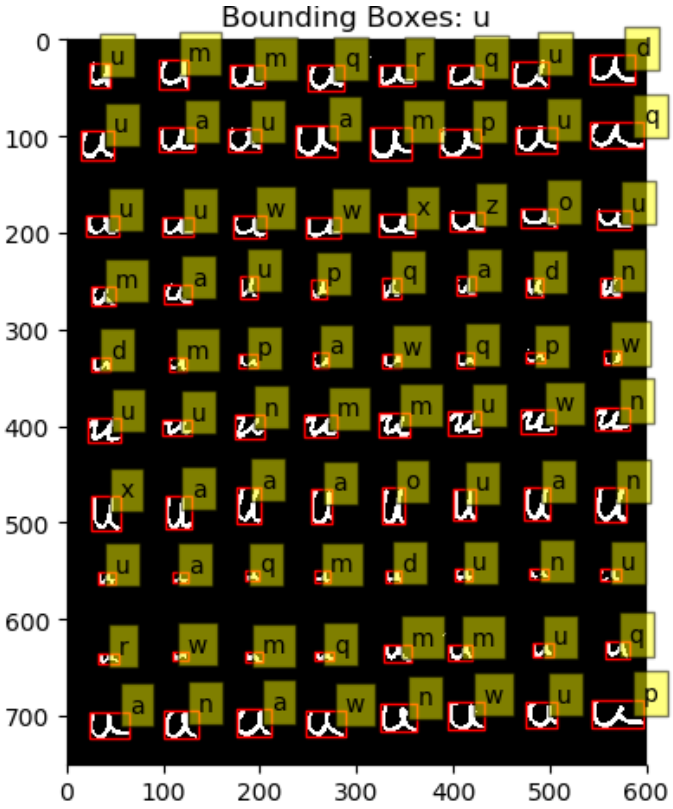

Bounding Boxes: o
0
200
400
600
800
1000
0
200
400
600
800

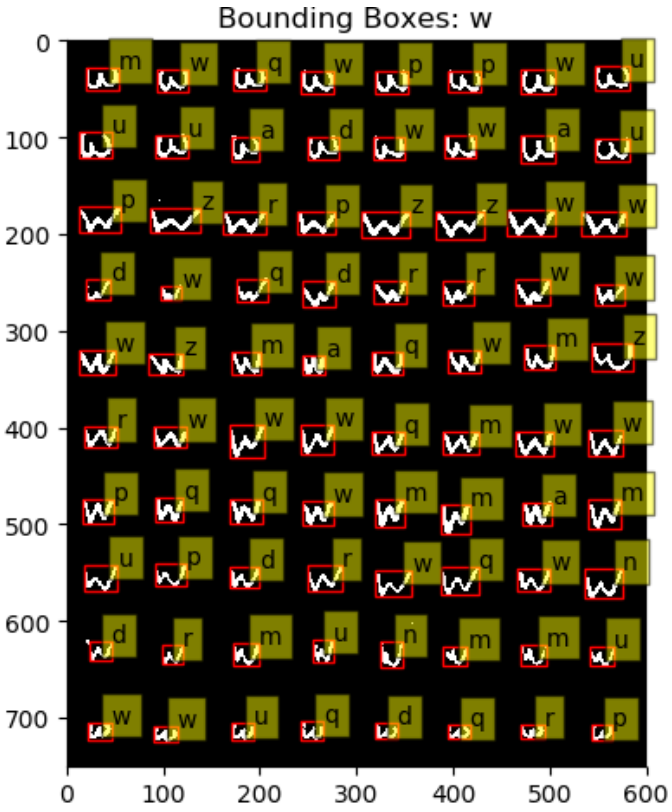

Bounding Boxes: p
0
100
200
300
400
500
600
700
0
100
200
300
400
500
600

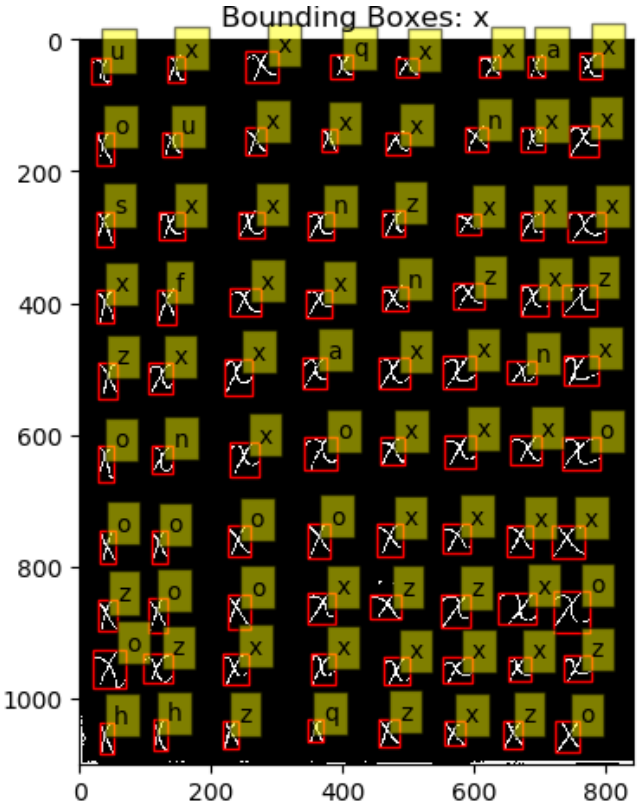

Bounding Boxes: q
0
100
200
300
400
500
600
700
0
100
200
300
400
500
600

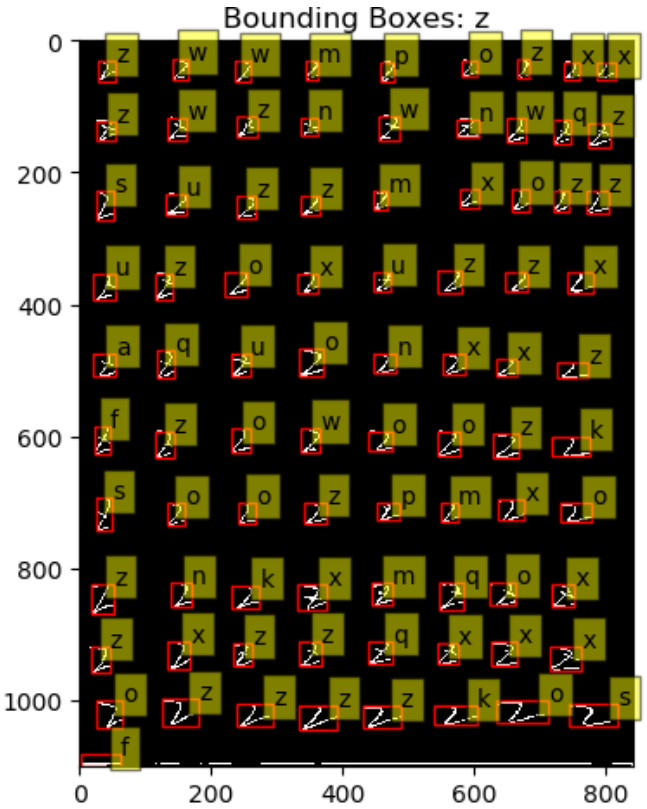

Bounding Boxes: r
0
100
200
300
400
500
600
700
0
100
200
300
400
500
600


Bounding Boxes: s
0
200
400
600
800
1000
0
200
400
600
800


Bounding Boxes: u
0
100
200
300
400
500
600
700
0
100
200
300
400
500
600


Bounding Boxes: w
0
100
200
300
400
500
600
700
0
100
200
300
400
500
600


Bounding Boxes: x
0
200
400
600
800
1000
0
200
400
600
800

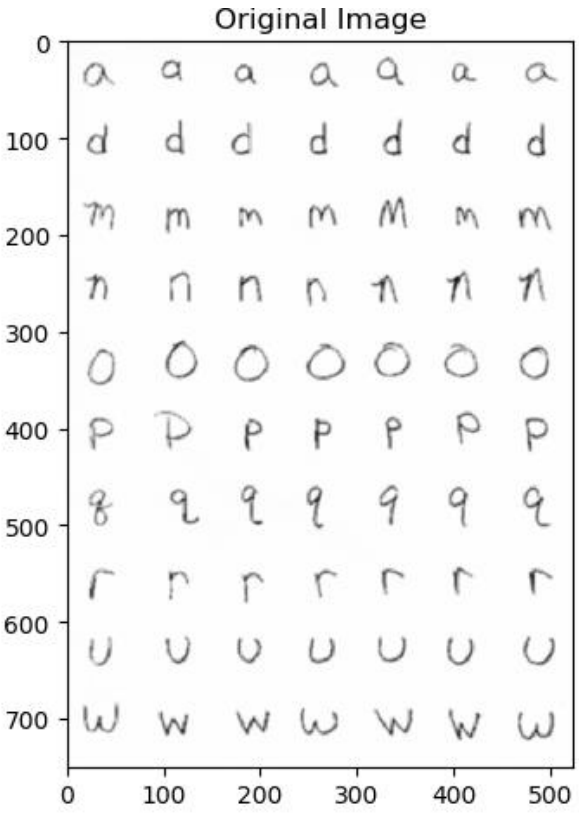

Bounding Boxes: z
0
200
400
600
800
1000
0
200
400
600
800

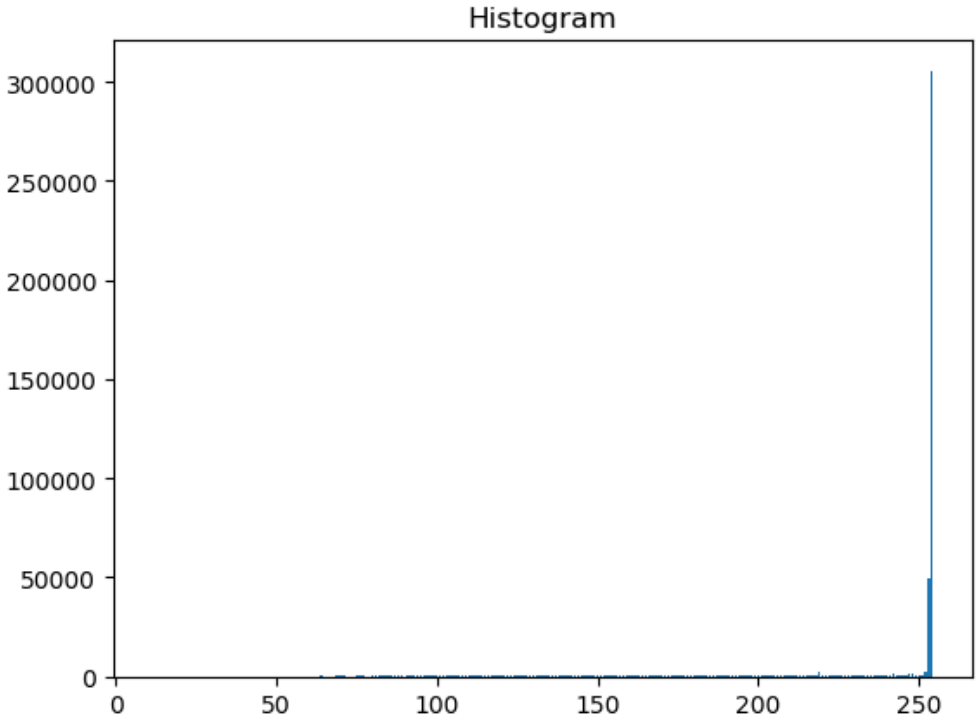
### 1.18 Accuracy of train

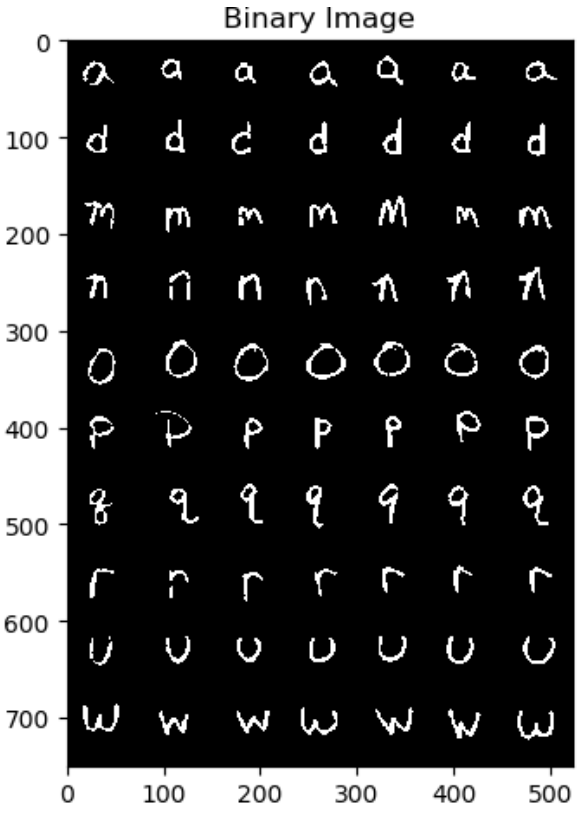
Accuracy = 0.45015576324

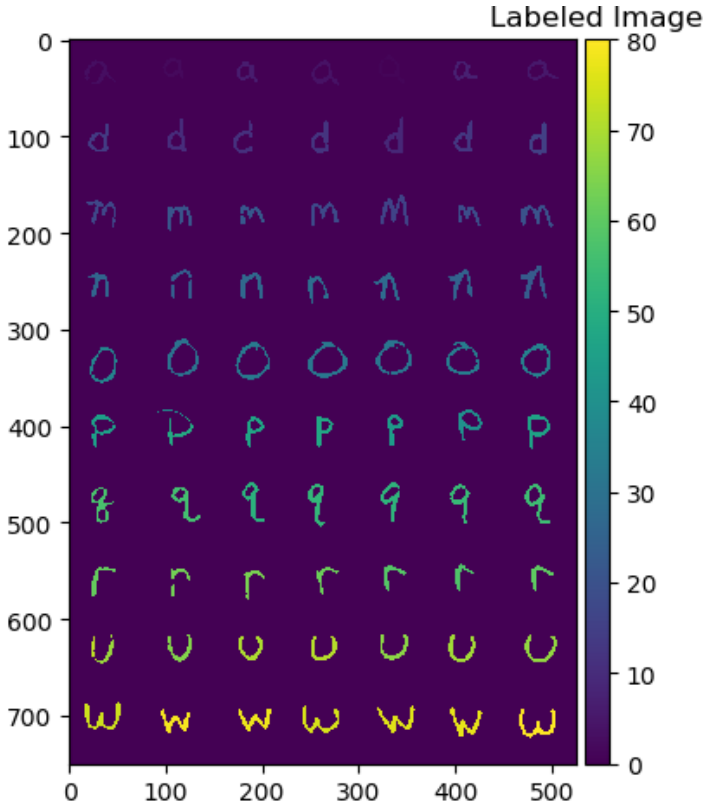
# 2 Test

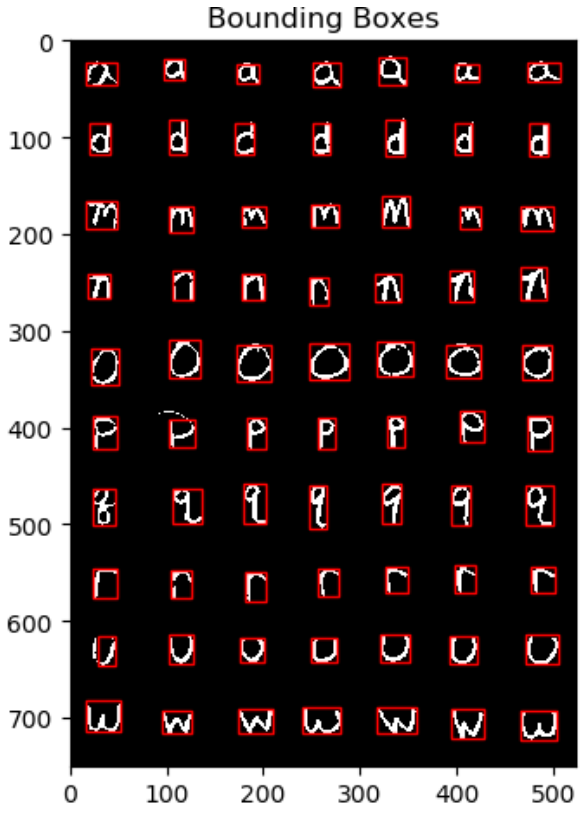
## 2.1 Test1



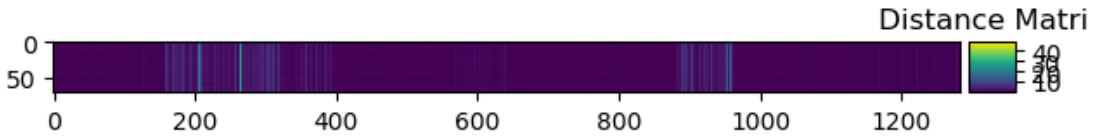

Histogram
300000
250000
200000
150000
100000
50000
0
0
50
100
150
200
250

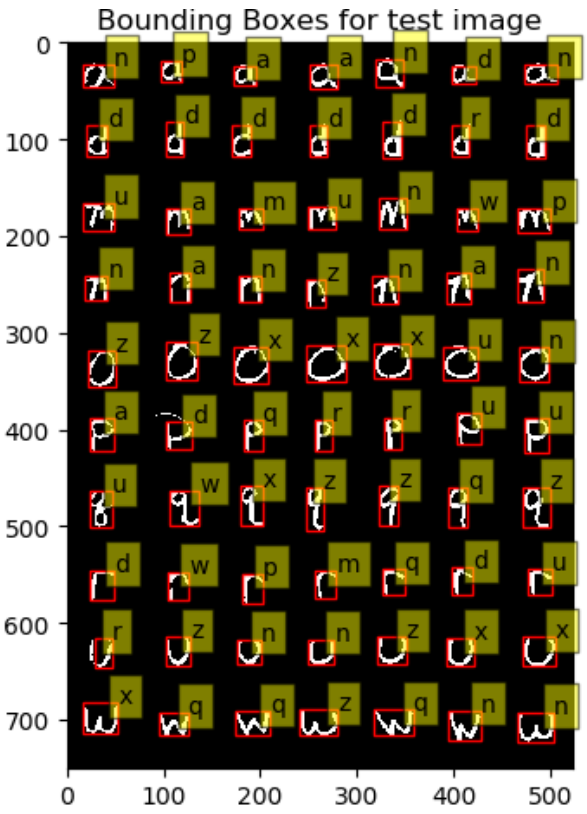

Binary Image
0
100
200
300
400
500
600
700
0
100
200
300
400
500


Labeled Image
0
100
200
300
400
500
600
700
0
100
200
300
400
500
0
10
20
30
40
50
60
70
80


Bounding Boxes
0
100
200
300
400
500
600
700
0
100
200
300
400
500

Distance Matrix for test data



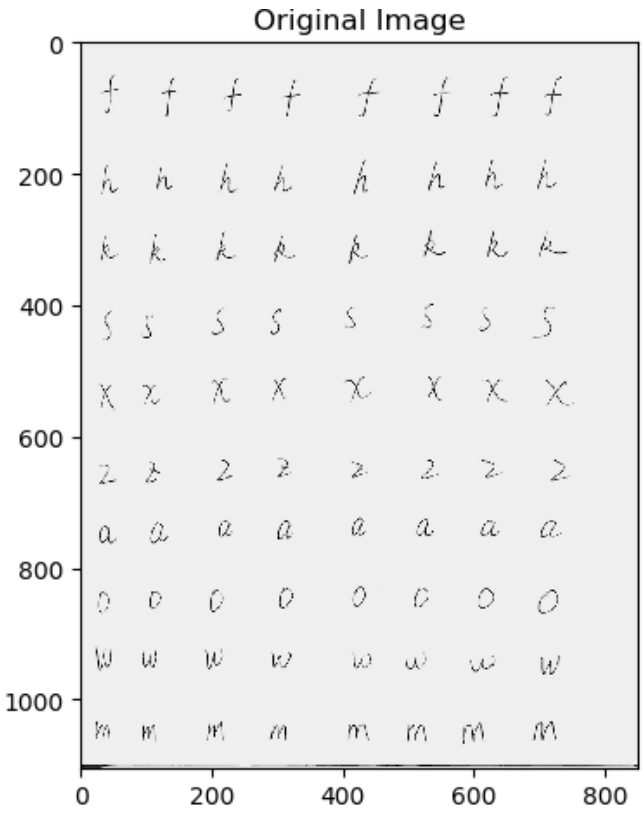



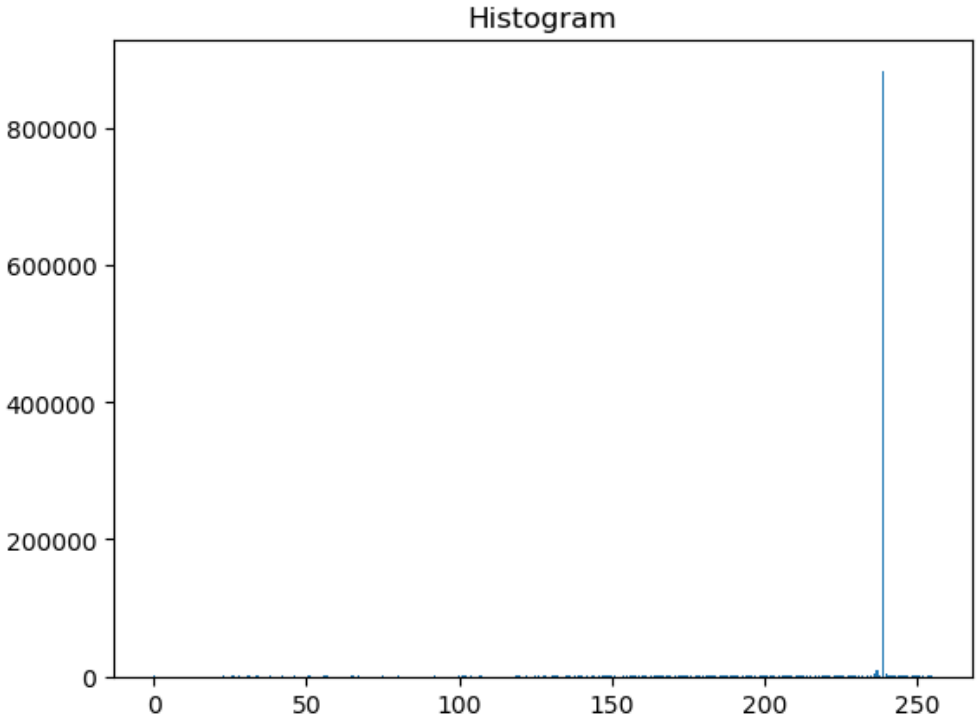

Because the Distance matrix is too larger, so I just paste part of it. And the Distance matrix for

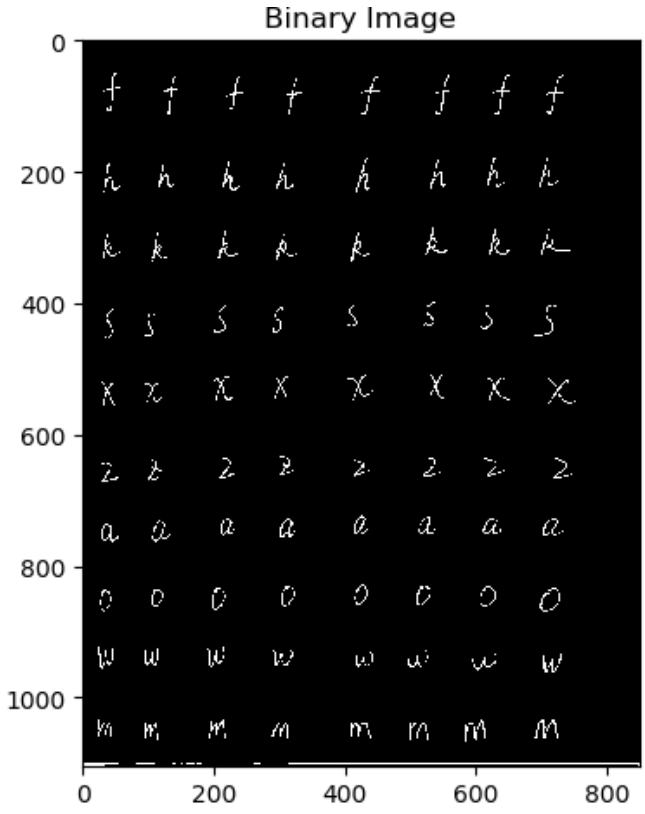
test data will be stored in the file ***result.txt***

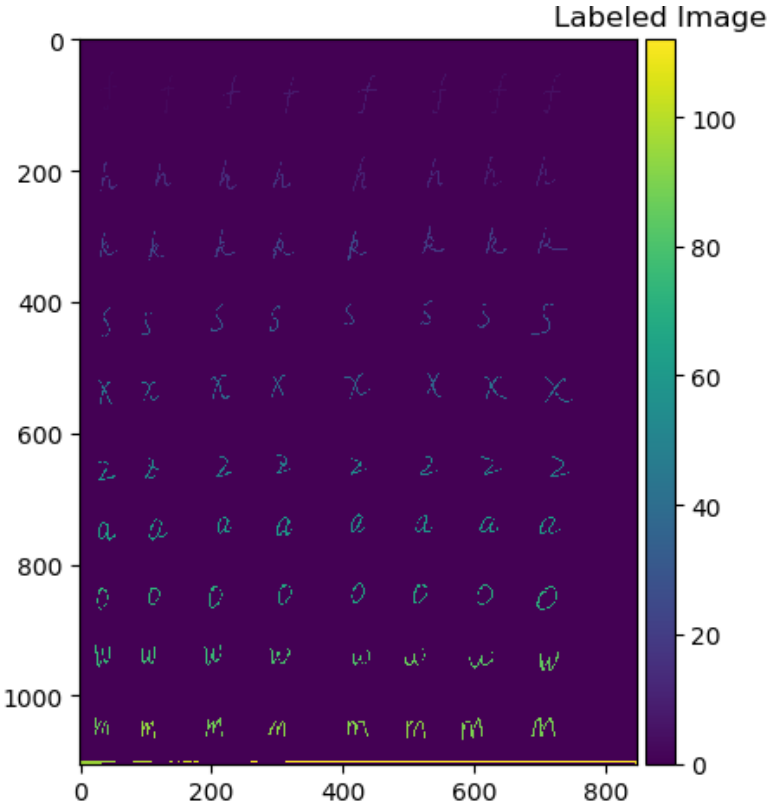
```
array([[0.02172063, 0.1210053 , 0.0612633 , ..., 1.25253535, 0.59022696,
        0.95748361],
       [0.05698179, 0.19949207, 0.13975971, ..., 1.32987715, 0.66856265,
        1.03051269],
       [0.05027196, 0.09227331, 0.03262388, ..., 1.22465227, 0.56161287,
        0.93211971],
       ...,
       [0.21609619, 0.07579045, 0.13378008, ..., 1.05965941, 0.39716629,
        0.77408997],
       [0.0106205 , 0.13498218, 0.07528268, ..., 1.26562754, 0.60398878,
        0.96883791],
       [0.20067429, 0.05919077, 0.11801316, ..., 1.07596215, 0.4121265 ,
        0.79172351]])
```

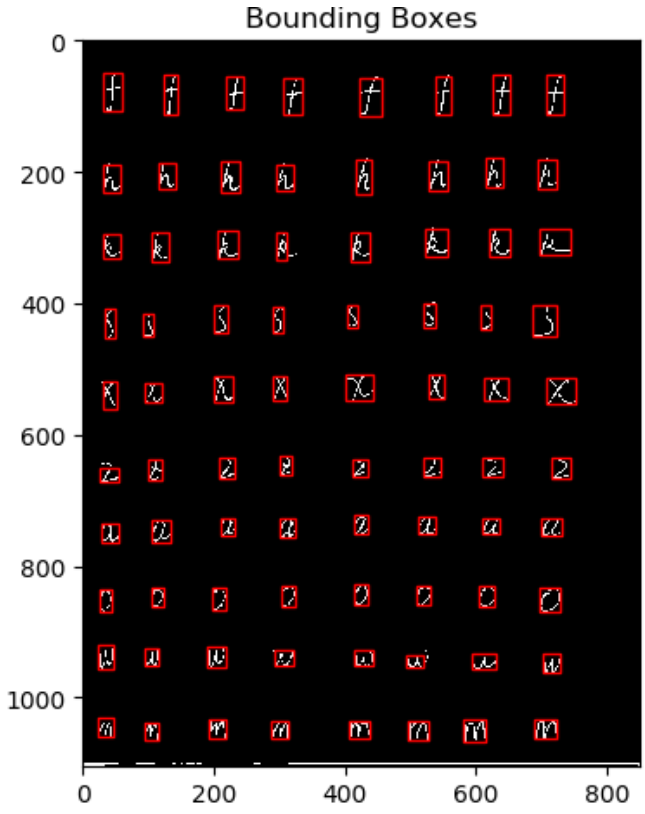
## 2.2 Test2

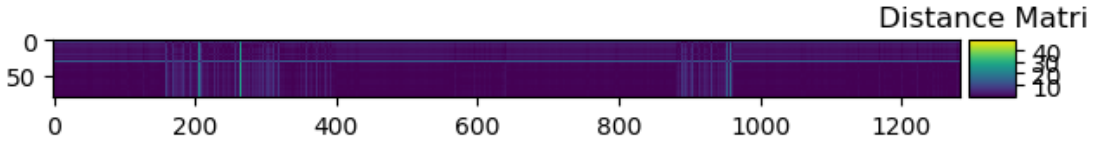


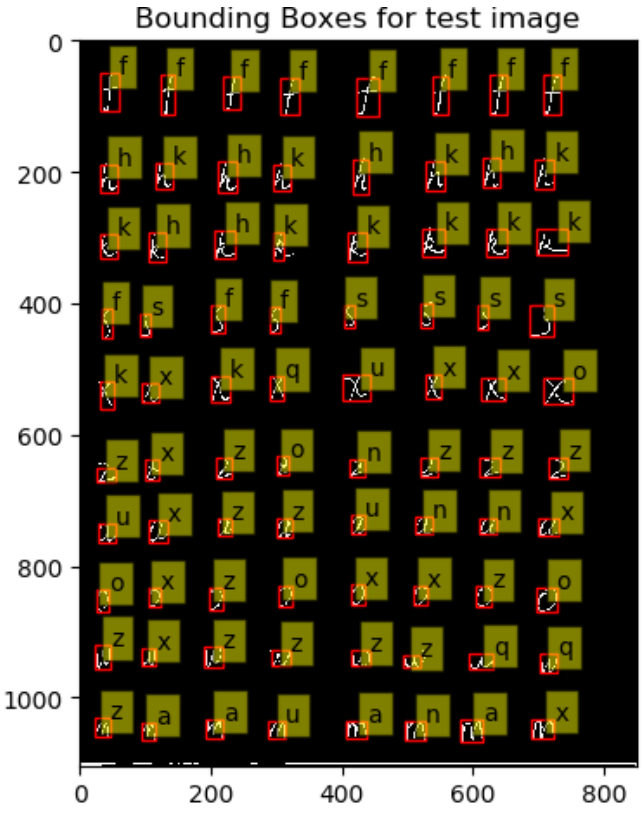

Histogram
800000
600000
400000
200000
0
0
50
100
150
200
250


Binary Image
0
200
400
600
800
1000
0
200
400
600
800


Labeled Image
0
200
400
600
800
1000
0
200
400
600
800
0
20
40
60
80
100


Bounding Boxes
0
200
400
600
800
1000
0
200
400
600
800




Because the Distance matrix is too larger, so I just paste part of it. And the Distance matrix for test data will be stored in the file ***result.txt***

array([[4.04692318e+00, 3.95569768e+00, 3.99272412e+00, ...,

```
  3.19100764e+00, 3.65618794e+00, 3.22495900e+00],
 [8.12322397e+00, 8.06047347e+00, 8.08567660e+00, ...,
  7.46769752e+00, 7.84383656e+00, 7.45405157e+00],
 [5.45060408e+00, 5.37441276e+00, 5.40514258e+00, ...,
  4.70424372e+00, 5.11971456e+00, 4.70336293e+00],
 ...,
 [1.20176062e-02, 1.30712160e-01, 7.09764204e-02, ...,
  1.26215787e+00, 5.99929000e-01, 9.66719302e-01],
 [9.50359461e-02, 4.75343347e-02, 1.23228489e-02, ...,
  1.18057213e+00, 5.16982425e-01, 8.90910593e-01],
 [6.63631060e-03, 1.36003605e-01, 7.63180281e-02, ...,
  1.26758787e+00, 6.05194269e-01, 9.72299642e-01]])
```

# 3 Recognition Rates and Other Values

## 3.1 Threshold value you have picked

For original train and test phase, the threshold is between the value below.

```
threshold = {'height':[10.0, 80.0], 'width':[12.0, 85.0]}
```

## 3.2 Number of components you obtained for test image and Recognition rate for the test image

| | Test1 | Test2 |
|---|---|---|
| Number of components you obtained | 70 | 80 |
| Recognition rate | 0.2 (14/70) | 0.425 (34/80) |
| | | |

# 4 Enhancements

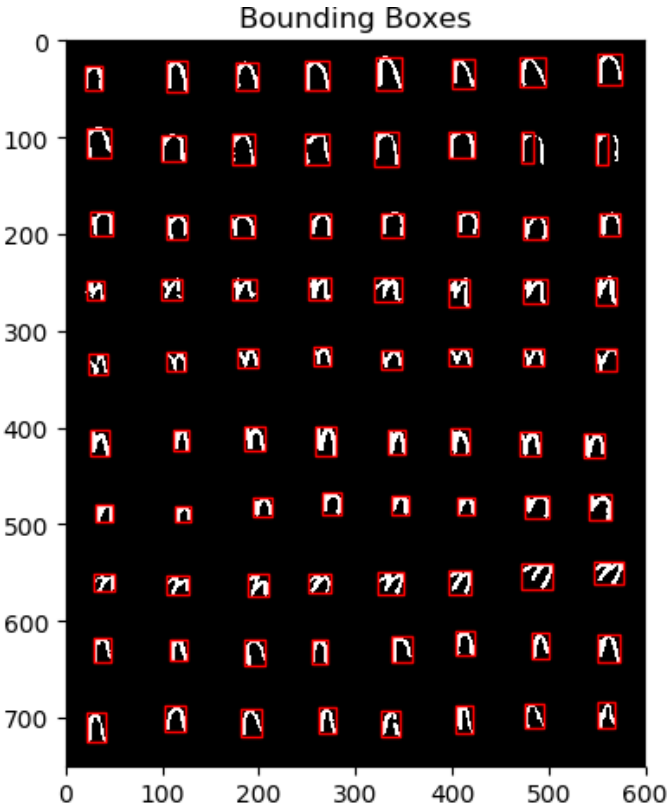
In this part we try 8 methods separately, (1) is the method to change the Connectivity from default 8-connectivity to 4-connectivity, (2, 3, 4) are methods automatically get the threshold used to get the binary image (default is 200), (5, 6, 7, 8) are methods use Mathematical Morphology to get the image.

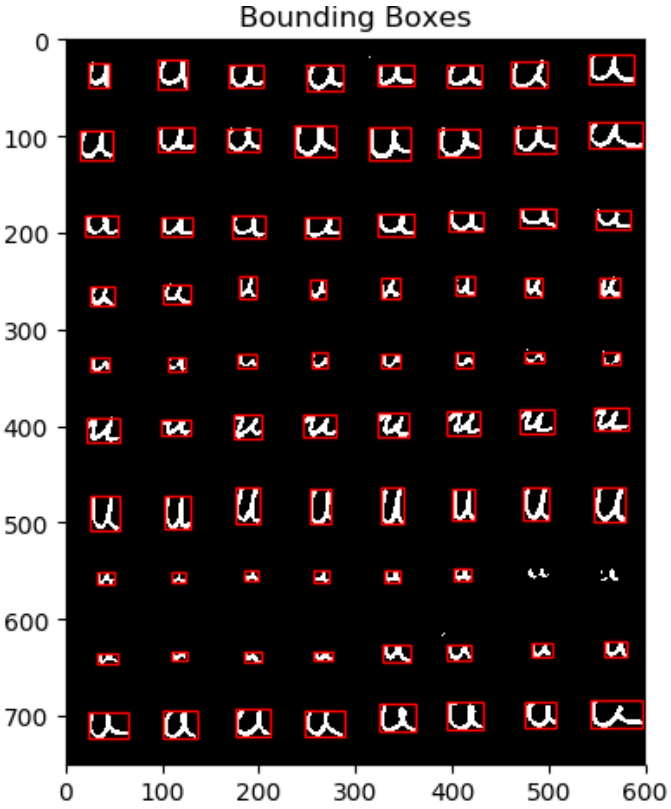
I think if the method's train error is bigger than the origin train error, the method will have a bigger test error than the origin too. So I just choose one in the (2, 3, 4) which is the best and the best of (5, 6, 7, 8), and with (1) if they are better than the origin. So, for combining all the enhancements, there will be maximum 3 methods. If the method's train error is smaller than the original, I will test it at further step or it will just end at this phase and be abandoned.
**The original train accuracy = 0.45015576324**

## 4.1 For enhance_1_train.py

I think change the Connectivity from default 8-connectivity to 4-connectivity will remove some small noise hooks onto the characters so I think it will have good result. In label phase of character **O**, there is a o separate to two parts and regarded as two connected component, and I think this method will solve this problem.
**Train accuracy = 0.458786936236**, so in train phase it is a good enhance.

Compare to the origin, the right part of n of row 2 column 7 has been ignored which is bad. And the O is unchanged.






From these examples, we can know that this method ignore some characters that is less obvious which could cause bad outcome, but for the obvious character, it will strength it.

In the test phase,

| | Test1 | Test2 |
|---|---|---|
| Number of components you obtained | 70 | 80 |
| Recognition rate | 0.2 (14/70) | 0.45 (36/80) |
| Origin Recognition rate | 0.2 (14/70) | 0.425 (34/80) |

It did work.

## 4.2 For enhance_2_train, enhance_3_train, enhance_4_train

| | enhance_2_train | enhance_3_train | enhance_4_train |
|---|---|---|---|
| Method of generate threshold | Peak and Valley method | P-tile method | Triangle algorithm |
| Train accuracy | 0.317335243553 | 0.31027857829 | 0.401815181518 |

For enhance_2_train, histograms are not smooth and it is not easy to find two peaks for our problems, so it is not surprised that the error is big. So not move on.

For enhance_3_train, firstly we don't know the p%, so we just know for th = 200 is approximately p = 0.05, and test the 0.45 to 0.65, we can find p=0.05 is good enough. We

get
threhold = 197
threhold = 191
threhold = 238
threhold = 241
threhold = 230
threhold = 174
threhold = 220
threhold = 238
threhold = 186
threhold = 192
threhold = 225
threhold = 241
threhold = 210
threhold = 196
threhold = 230
threhold = 230
Counter({'z': 84, 'f': 81, 'k': 81, 'x': 81, 'a': 80, 'n': 80, 'p': 80, 'r': 80, 'w': 80, 'd': 79, 'm': 79, 'u': 79, 'q': 77})
But for this method, we have a problem because we don't have the best p%, so the error is big so we just move on.

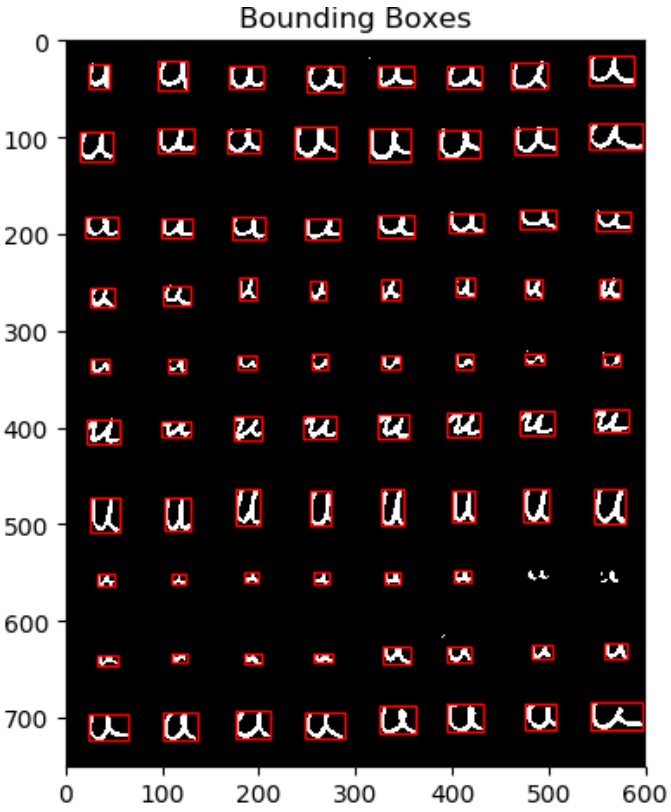
For enhance_4_train, use equations in the xy-coordinate to calculate the distance between P and the line. So we get
threshold = 250
threshold = 250
threshold = 238
threshold = 238
threshold = 230
threshold = 250
threshold = 250
threshold = 253
threshold = 250
threshold = 250
threshold = 250
threshold = 238
threshold = 250
threshold = 250
threshold = 230
threshold = 230
Counter({'z': 84, 'a': 81, 'd': 81, 'f': 81, 'k': 81, 'p': 81, 's': 81, 'r': 81, 'w': 81, 'x': 81, 'h': 80, 'm': 80, 'n': 80, 'q': 80, 'u': 79})
This technique is particularly effective when the object pixels produce a weak peak in the histogram. But in our problems, the peak is especially strong. So we move on.

So the methods in this part will be abandoned.

## 4.3 For enhance_5_train, enhance_6_train, enhance_7_train, enhance_8_train

| | enhance_5_train | enhance_6_train | enhance_7_train | enhance_8_train |
|---|---|---|---|---|
| Method of generate threshold | dilation | erosion | opening | closing |
| Train accuracy | 0.453696498054 | 0.354122621564 | 0.342710997442 | 0.450934579439 |

For 5 and 6, because the characters written by people may have some disconnected handwriting in the paper, and dilation can expand the edge and small hole, so it will better find the character, and the erosion will expand the background which will enhance the disconnected part, so it will have bad effects. So we can easily choose dilation.
For train in dilation, actually it ignore more not obvious character like example below,




But in test, it actually have this result

| | Test1 | Test2 |
|---|---|---|
| Number of components you obtained | 70 | 80 |
| Recognition rate | 0.2 (14/70) | 0.45 (36/80) |
| Origin Recognition rate | 0.2 (14/70) | 0.425 (34/80) |
| | | |

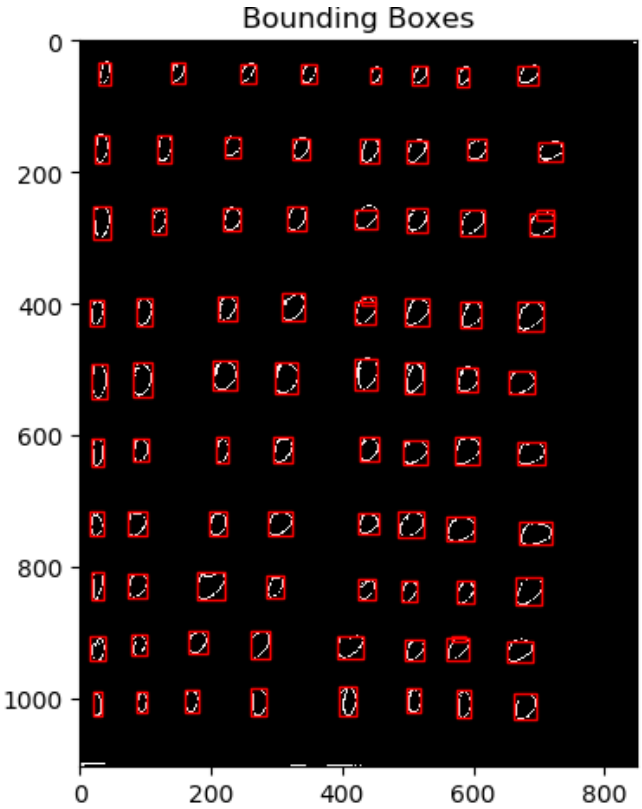
So it is good.

For 7, it is the same reason as 6 because it does erosion then dilation, so it will also remove the small spot. So we just move on.

For 8, we already dilation works, and closing is first dilation then erosion and we can easily find that we choose enhance_5 is good enough.

## 4.4 For combined enhance

Actually I can't see any different from the enhance_1 and enhance_5 in the labeled image, below is one of them.



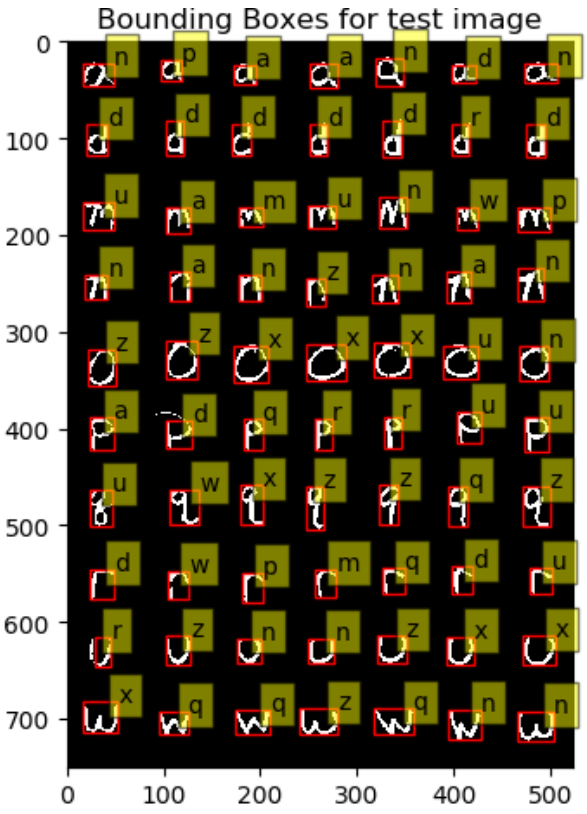
The result is also the same.

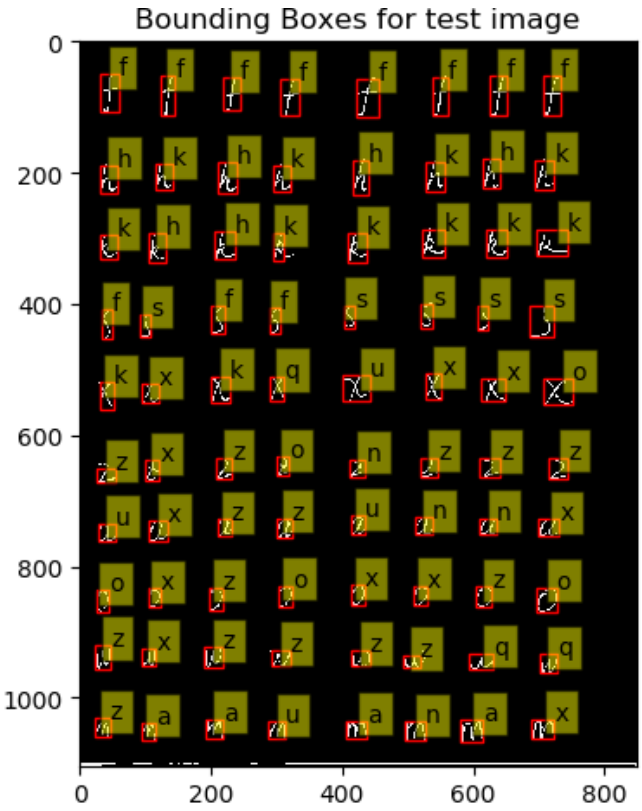
| | Test1 | Test2 |
|---|---|---|
| Number of components you obtained | 70 | 80 |
| Recognition rate | 0.2 (14/70) | 0.425 (34/80) |
| Origin Recognition rate | 0.2 (14/70) | 0.425 (34/80) |
| | | |

# 5 Recognition Rates and Other Values

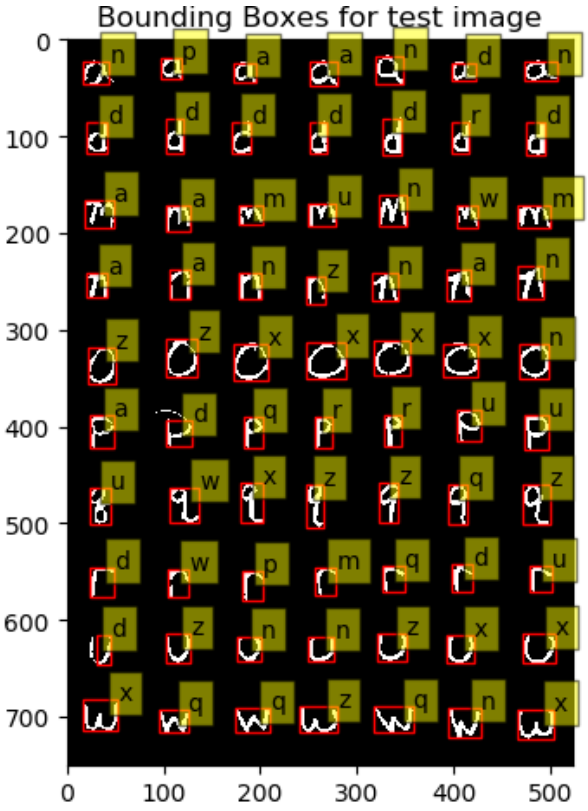
| | Test1 | Test2 |
|---|---|---|
| Number of components you obtained | 70 | 80 |
| Origin Recognition rate | 0.2 (14/70) | 0.425 (34/80) |
| Enhance by connectivity | 0.2 (14/70) | 0.45 (36/80) |
| Enhance by dilation | 0.2 (14/70) | 0.45 (36/80) |
| Enhance by dilation and connectivity | 0.2 (14/70) | 0.45 (36/80) |

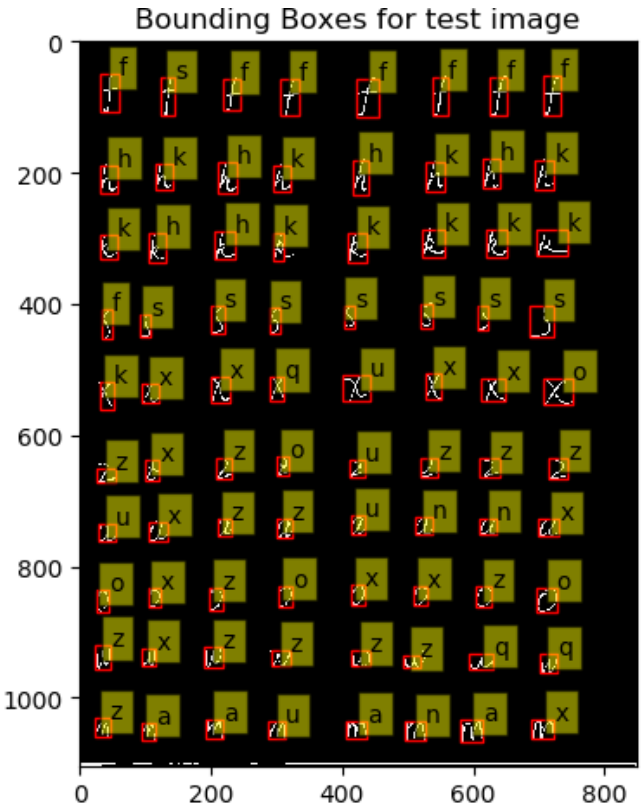
**Origin:**



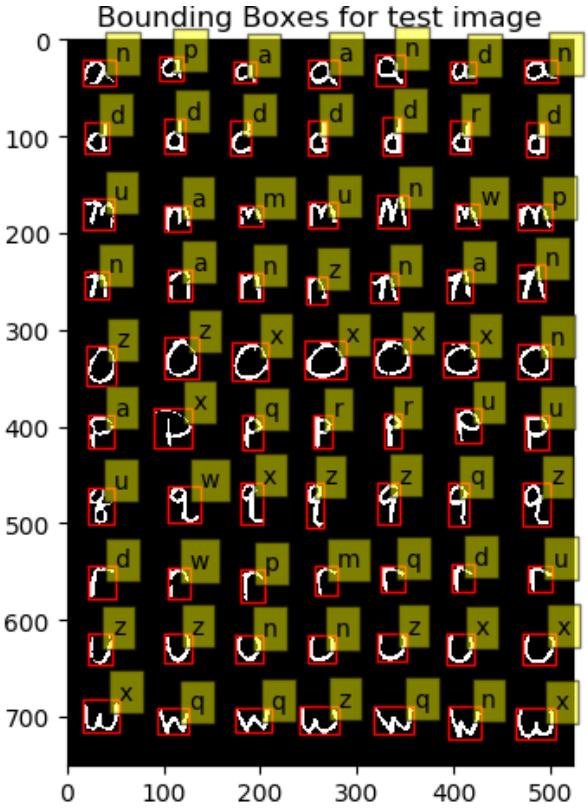


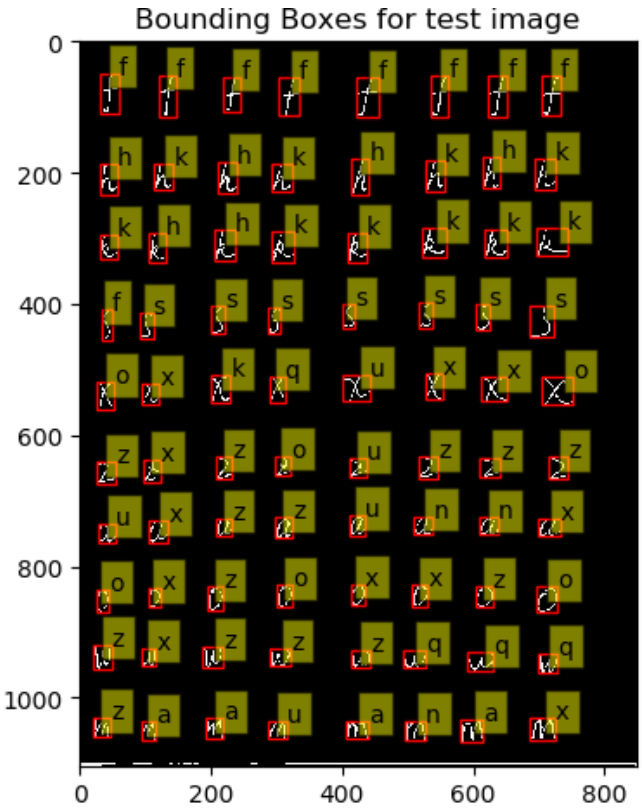

By connectivity:



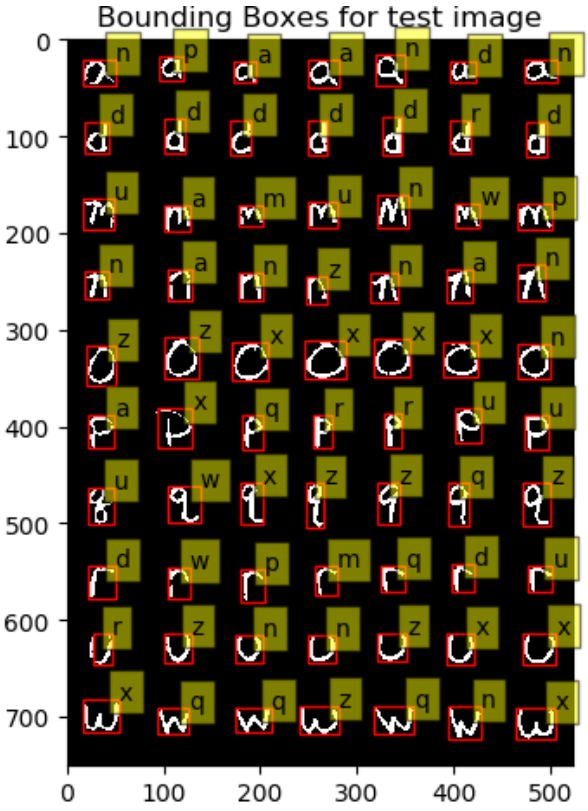

Bounding Boxes for test image
0
200
400
600
800
1000
0
200
400
600
800

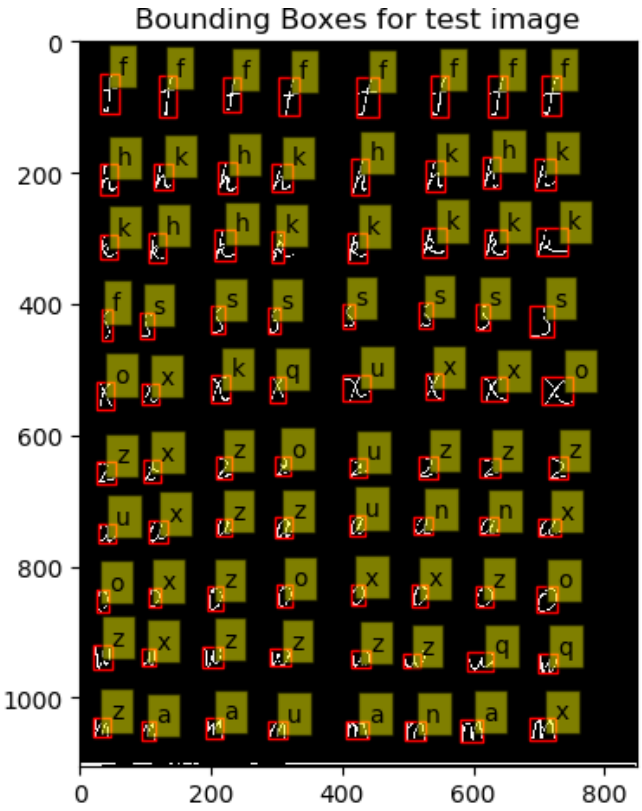
By dilation:


Bounding Boxes for test image
0
100
200
300
400
500
600
700
0
100
200
300
400
500


Bounding Boxes for test image
0
200
400
600
800
1000
0
200
400
600
800

**Combined:**


Bounding Boxes for test image
0
100
200
300
400
500
600
700
0
100
200
300
400
500


Bounding Boxes for test image
0
200
400
600
800
1000
0
200
400
600
800