Team member: Zhengqi Fang(netID: zf4),**Lingkai Kong**(netID: lingkai2)**, Dongxuan Zhang**(netID: dz3)

3 credit

CS 440 Machine Problem 3

Part 1:Digit classification

1.1:

Implementation:

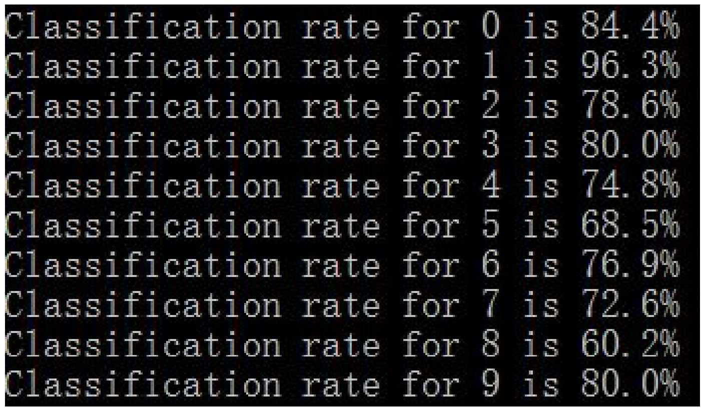
First read in both train data and train label, and in order to store the training result, I create a numpy array of size (2,10,28,28), let us call it 'train_result'. Then I go through each pixel in each image, if such pixel is background (in other words, it's " " in the train data), then I increment train-result [0][corresponding label][x][y] (x,y is the corresponding position for the pixel in the image). Otherwise, I increment the train-result [1][corresponding label][x][y].

Then for each image in the test sample, I just use the maximum a posteriori (MAP) classification. The testing process is much of the same as training process. First check if such pixel is background or not, then add the corresponding probability. After going through every pixel in the image, I will find the digit that has the largest probability and check whether it is the same as its own label.

Smoothing:

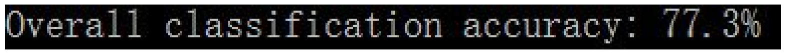
I choose 0.1 as the smoothing constant. Because as I increase the smoothing constant, the classification accuracy decreases.

Below image shows the classification rate for each digit:



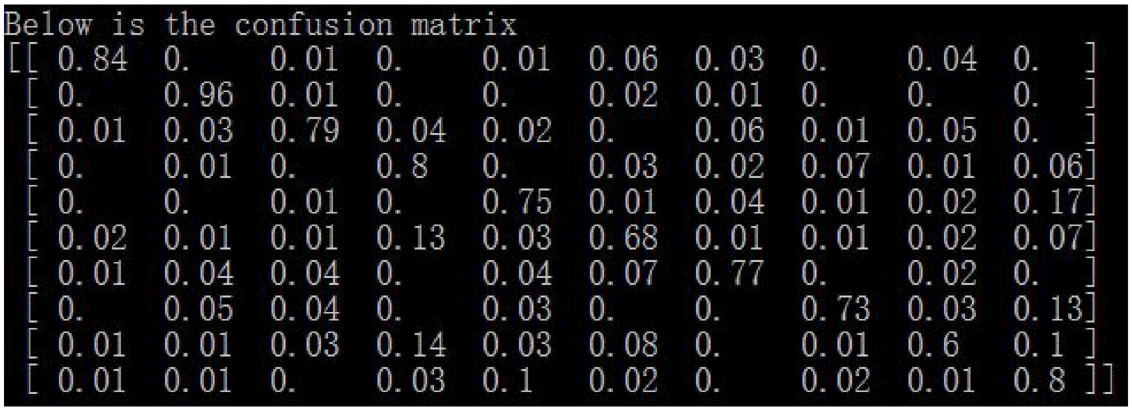
```
Classification rate for 0 is 84.4%
Classification rate for 1 is 96.3%
Classification rate for 2 is 78.6%
Classification rate for 3 is 80.0%
Classification rate for 4 is 74.8%
Classification rate for 5 is 68.5%
Classification rate for 6 is 76.9%
Classification rate for 7 is 72.6%
Classification rate for 8 is 60.2%
Classification rate for 9 is 80.0%
```

Below is the overall accuracy: 77.3%



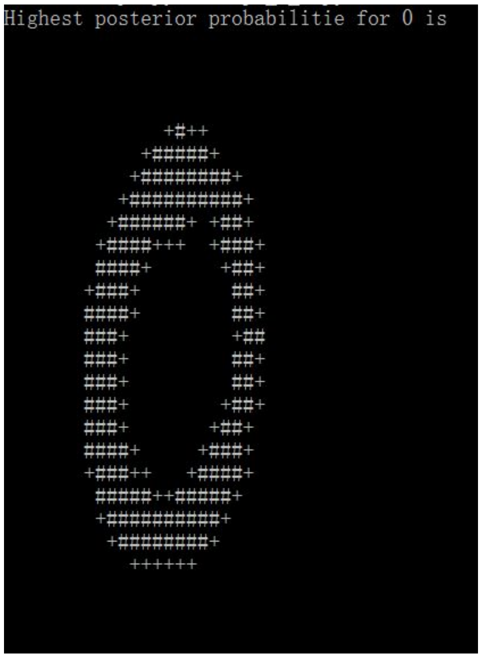
```
Overall classification accuracy: 77.3%
```

Below is the confusion matrix:

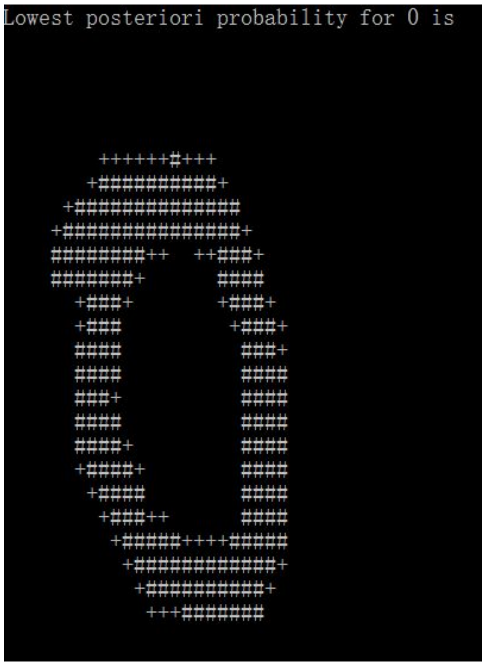


```
Below is the confusion matrix
[[ 0.84  0.    0.01  0.    0.01  0.06  0.03  0.    0.04  0.  ]
 [ 0.    0.96  0.01  0.    0.    0.02  0.01  0.    0.    0.  ]
 [ 0.01  0.03  0.79  0.04  0.02  0.    0.06  0.01  0.05  0.  ]
 [ 0.    0.01  0.    0.8   0.    0.03  0.02  0.07  0.01  0.06]
 [ 0.    0.    0.01  0.    0.75  0.01  0.04  0.01  0.02  0.17]
 [ 0.02  0.01  0.01  0.13  0.03  0.68  0.01  0.01  0.02  0.07]
 [ 0.01  0.04  0.04  0.    0.04  0.07  0.77  0.    0.02  0.  ]
 [ 0.    0.05  0.04  0.    0.03  0.    0.    0.73  0.03  0.13]
 [ 0.01  0.01  0.03  0.14  0.03  0.08  0.    0.01  0.6   0.1 ]
 [ 0.01  0.01  0.    0.03  0.1   0.02  0.    0.02  0.01  0.8 ]]
```

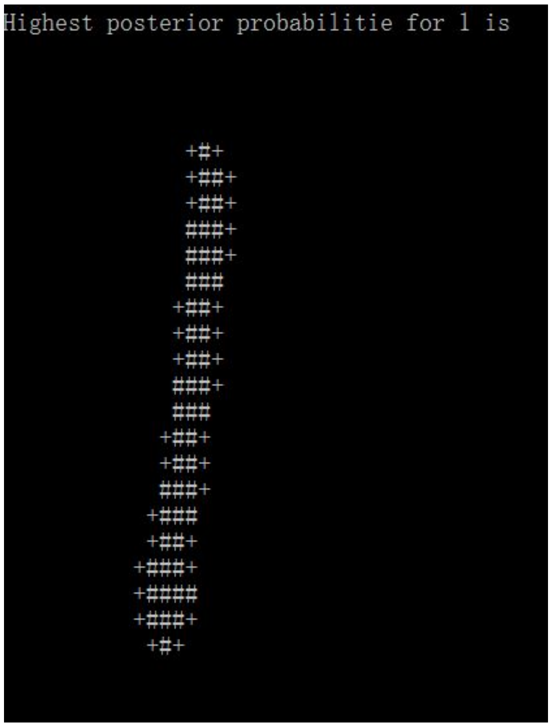
Below are the test examples that have the highest and lowest posterior probabilities for each digit



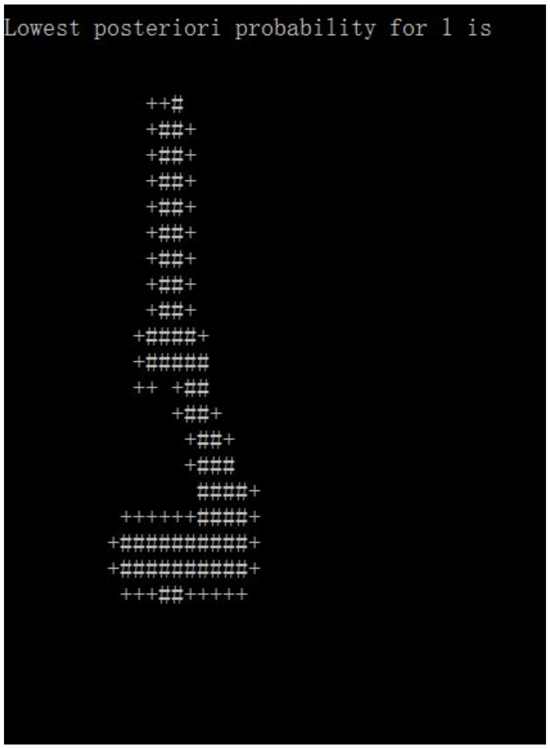
```
Highest posterior probabilitie for 0 is




       +#++
     +#####+
    +########+
   +##########+
  +######+ +##+
 +####+++  +###+
 ####+      +##+
+###+        ##+
####+        ##+
###+         +##
###+         ##+
###+         ##+
###+        +##+
###+       +##+
####+     +###+
+###++   +####+
 #####++#####+
 +##########+
  +########+
    ++++++
```



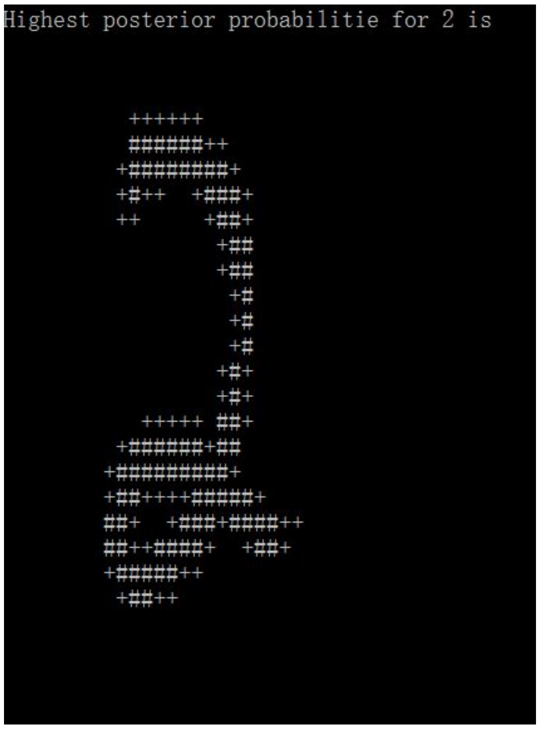
```
Lowest posteriori probability for 0 is




    ++++++#+++
   +##########+
 +##############
+###############+
########++  ++###+
#######+      ####
  +###+       +###+
  +###         +###+
  ####          ###+
  ####          ####
  ###+          ####
  ####          ####
  ####+         ####
  +####+        ####
   +####        ####
    +###++      ####
     +#####++++#####
      +############+
       +##########+
        +++#######
```



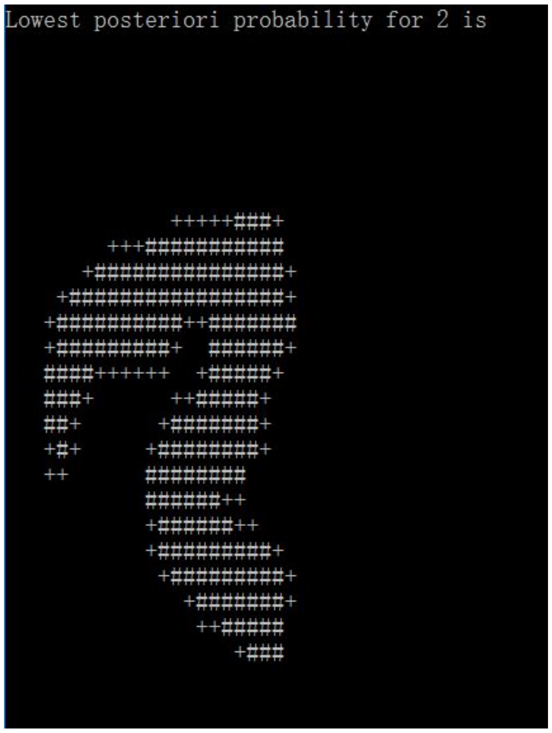
```
Highest posterior probabilitie for 1 is




              +#+
              +##+
              +##+
              ###+
              ###+
              ###
             +##+
             +##+
             +##+
             ###+
             ###
            +##+
            +##+
            ###+
           +###
           +##+
          +###+
          +####
          +###+
           +#+
```



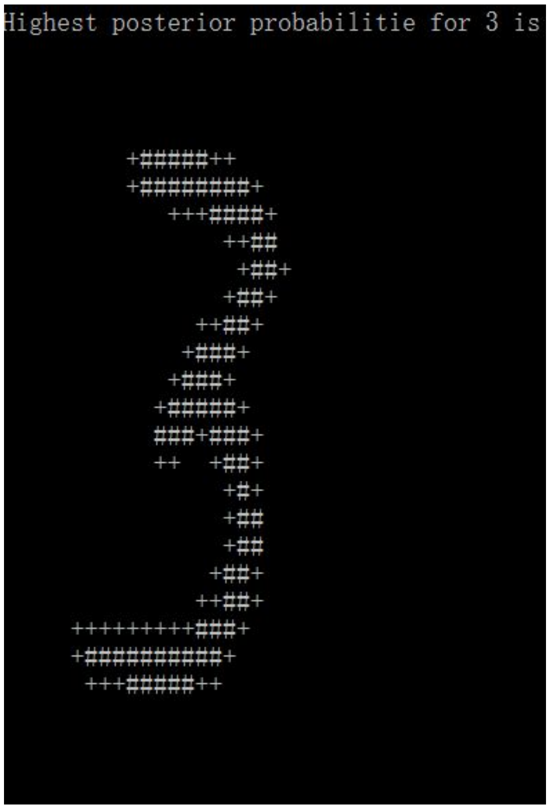
```
Lowest posteriori probability for 1 is


           ++#
           +##+
           +##+
           +##+
           +##+
           +##+
           +##+
           +##+
           +##+
          +####+
          +#####
          ++ +##
             +##+
              +##+
              +###
               ####+
         ++++++####+
        +##########+
        +##########+
         +++##+++++
```



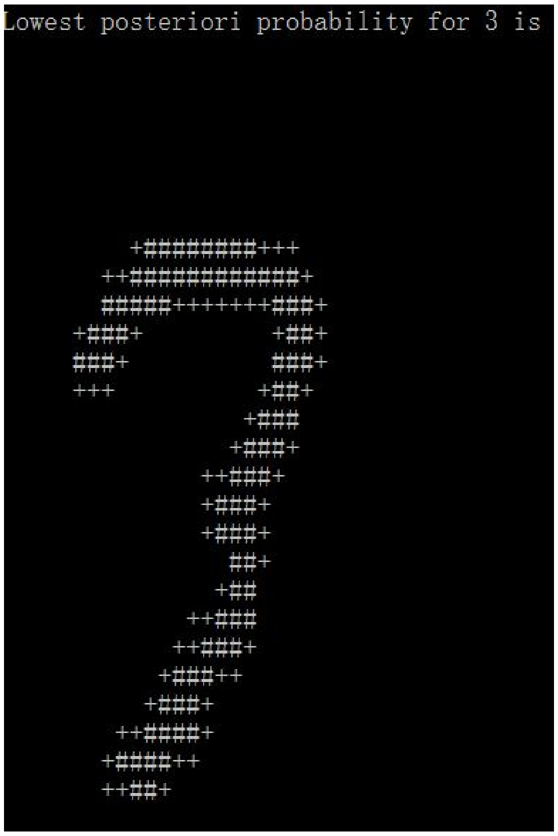
```
Highest posterior probabilitie for 2 is



          ++++++
          ######++
         +########+
         +#++  +###+
         ++     +##+
                 +##
                 +##
                  +#
                  +#
                  +#
                 +#+
                 +#+
           +++++ ##+
         +######+##
        +#########+
        +##++++#####+
        ##+  +###+####++
        ##++####+  +##+
        +#####++
         +##++
```



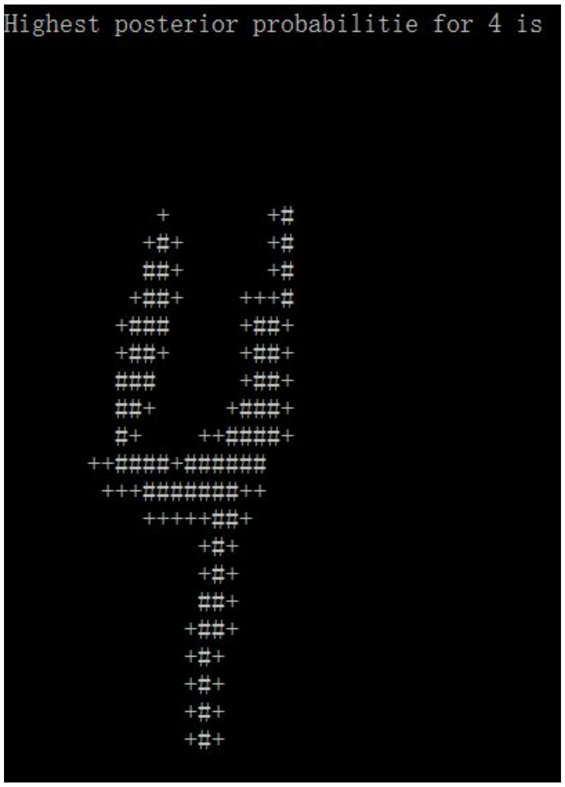
```
Lowest posteriori probability for 2 is







             +++++###+
        +++###########
      +###############+
    +#################+
   +##########++#######
   +#########+  ######+
   ####++++++  +#####+
   ###+      ++#####+
   ##+      +#######+
   +#+     +########+
   ++      ########
           ######++
           +######++
           +#########+
            +#########+
              +#######+
               ++#####
                  +###
```



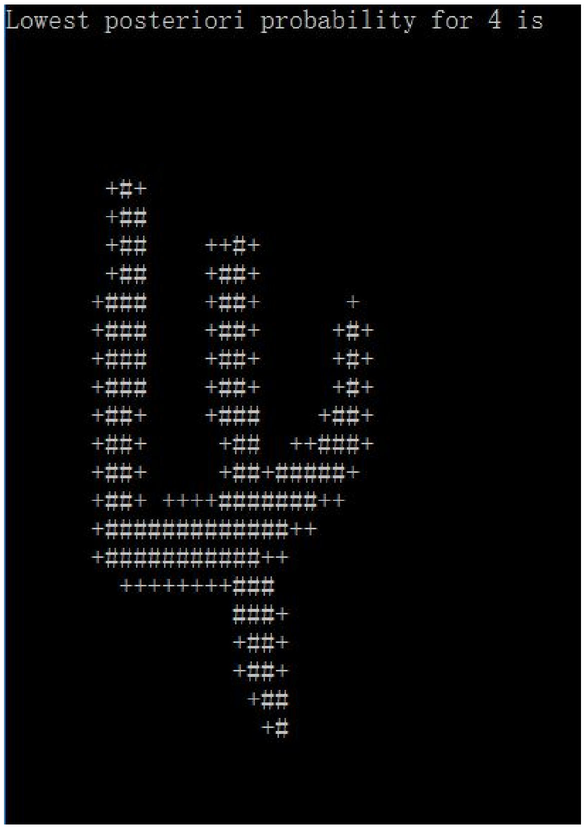
```
Highest posterior probabilitie for 3 is




         +#####++
         +#########+
            +++####+
                ++##
                 +##+
                +##+
              ++##+
             +###+
            +###+
           +#####+
           ###+###+
           ++  +##+
                +#+
                +##
                +##
               +##+
              ++##+
     +++++++++###+
     +##########+
      +++#####++
```



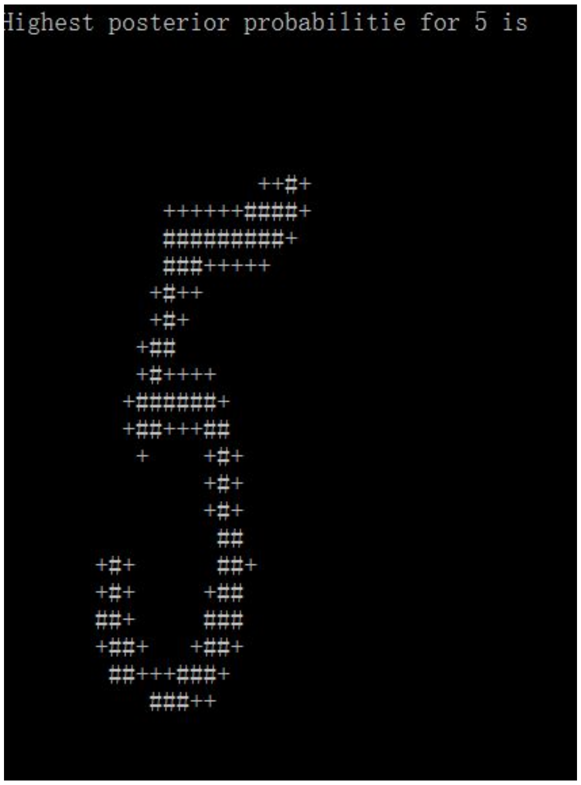
```
Lowest posteriori probability for 3 is







         +########+++
       ++############+
       #####+++++++###+
     +###+         +##+
     ###+          ###+
     +++          +##+
                 +###
                +###+
              ++###+
              +###+
              +###+
                ##+
               +##
             ++###
            ++###+
           +###++
          +###+
        ++####+
       +####++
       ++##+
```



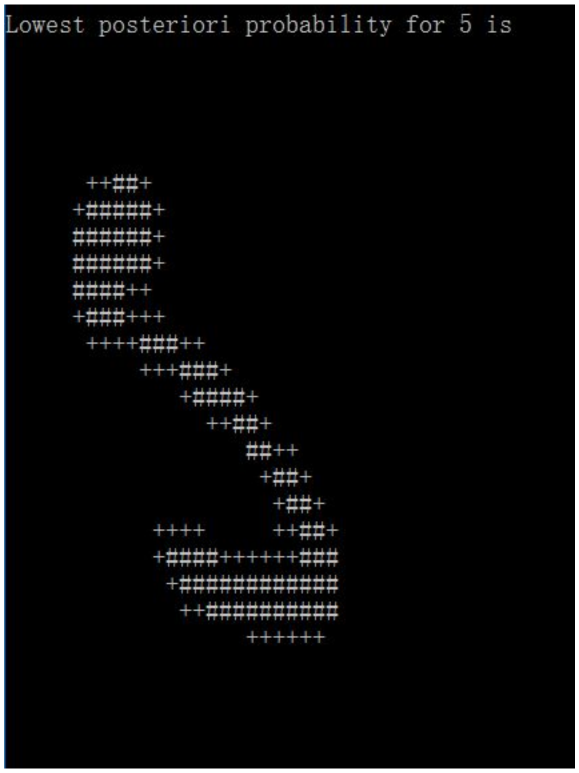
```
Highest posterior probabilitie for 4 is







           +       +#
          +#+      +#
          ##+      +#
         +##+    +++#
        +###     +##+
        +##+     +##+
        ###      +##+
        ##+     +###+
        #+    ++####+
      ++####+######
       +++#######++
          +++++##+
              +#+
              +#+
              ##+
             +##+
             +#+
             +#+
             +#+
             +#+
```



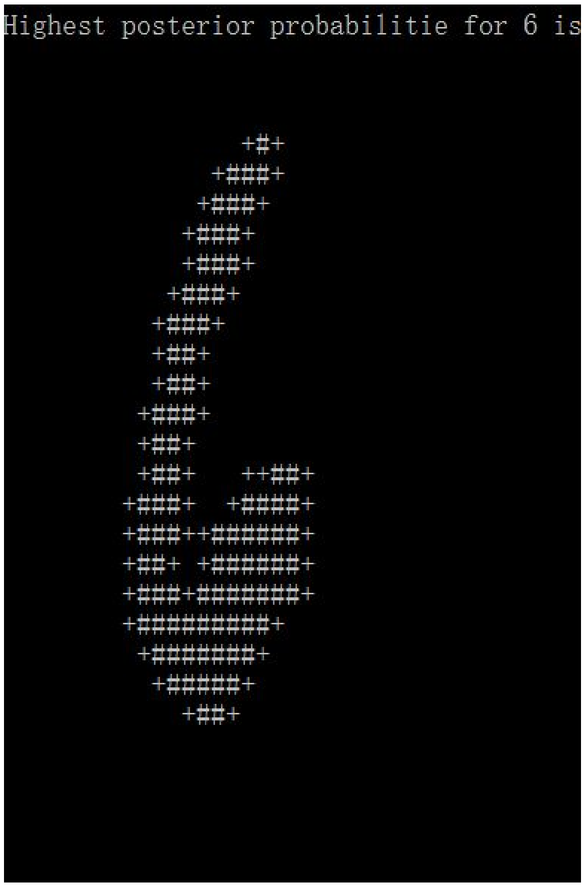
```
Lowest posteriori probability for 4 is





       +#+
       +##
       +##    ++#+
       +##    +##+
      +###    +##+      +
      +###    +##+     +#+
      +###    +##+     +#+
      +###    +##+     +#+
      +##+    +###    +##+
      +##+     +##  ++###+
      +##+     +##+#####+
      +##+ ++++#######++
      +#############++
      +###########++
        ++++++++###
                ###+
                +##+
                +##+
                 +##
                  +#
```



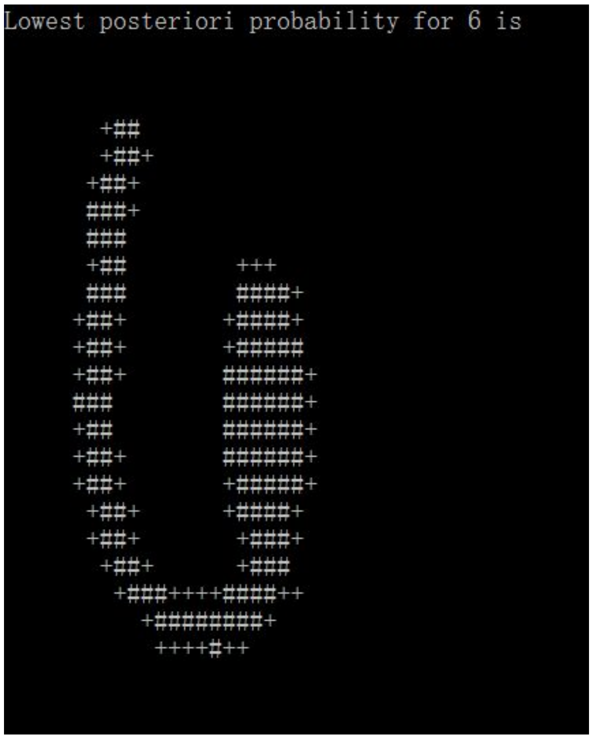
```
Highest posterior probabilitie for 5 is




                  ++#+
           ++++++####+
           #########+
           ###+++++
          +#++
          +#+
         +##
         +#++++
        +######+
        +##+++##
         +    +#+
              +#+
              +#+
               ##
      +#+      ##+
      +#+     +##
      ##+     ###
      +##+   +##+
       ##+++###+
          ###++
```



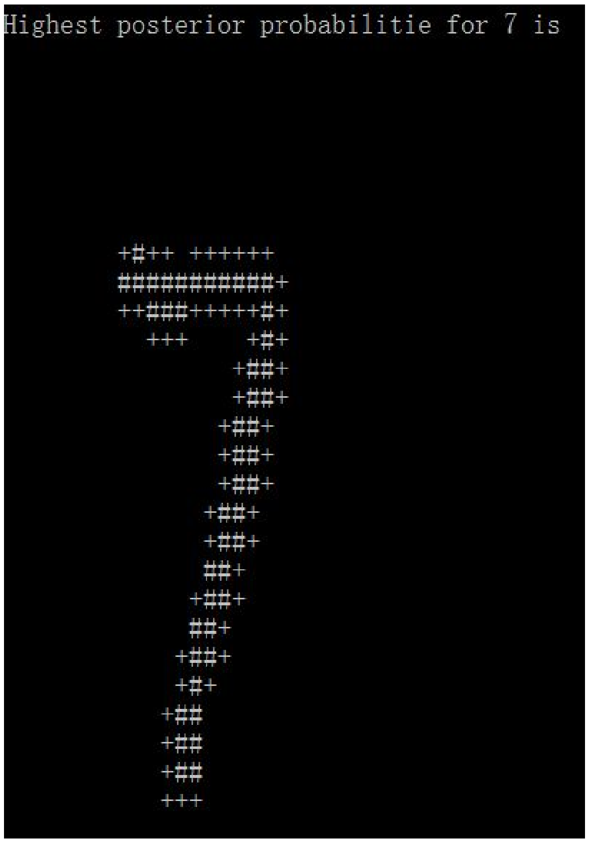
```
Lowest posteriori probability for 5 is




      ++##+
     +#####+
     ######+
     ######+
     ####++
     +###+++
      ++++###++
          +++###+
             +####+
               ++##+
                  ##++
                   +##+
                    +##+
           ++++     ++##+
           +####++++++###
            +############
             ++##########
                  ++++++
```



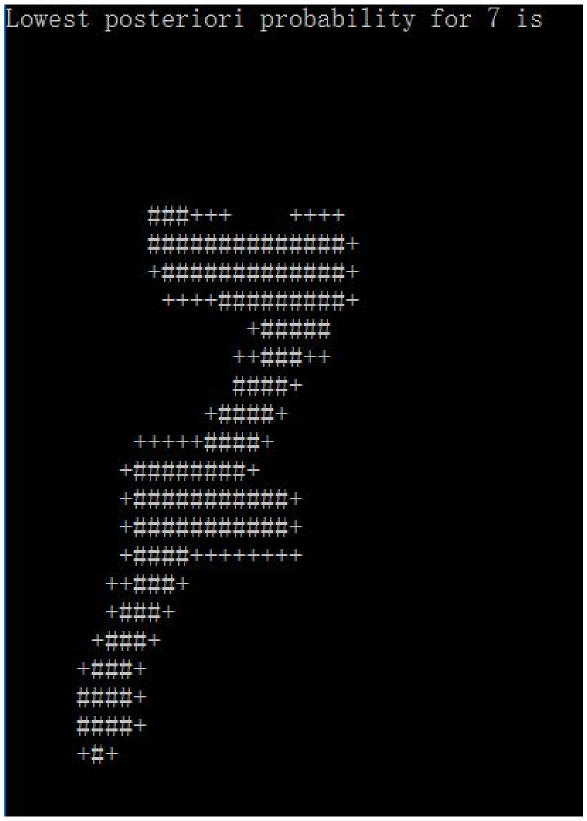
```
Highest posterior probabilitie for 6 is



                +#+
              +###+
             +###+
            +###+
            +###+
           +###+
          +###+
          +##+
          +##+
         +###+
         +##+
         +##+   ++##+
        +###+  +####+
        +###++######+
        +##+ +######+
        +###+#######+
        +#########+
         +#######+
          +#####+
            +##+
```



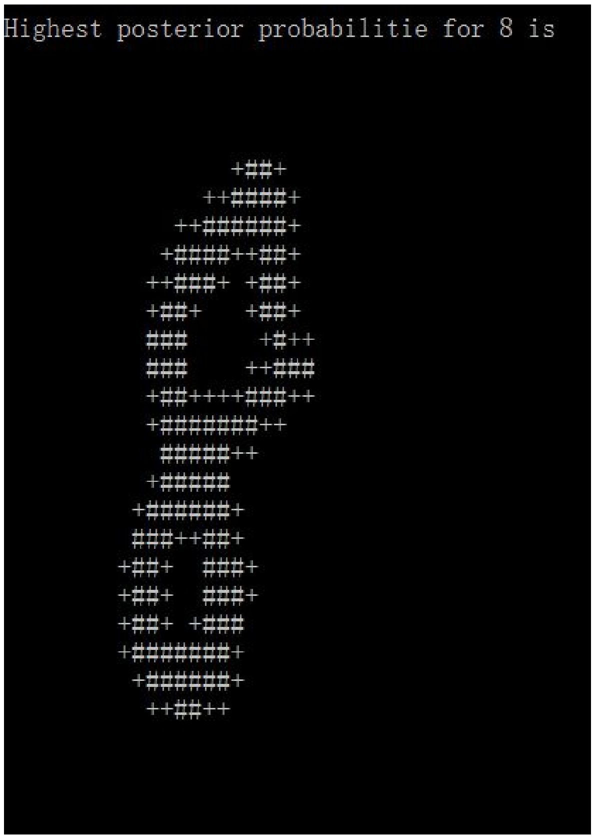
```
Lowest posteriori probability for 6 is



       +##
       +##+
      +##+
      ###+
      ###
      +##          +++
      ###          ####+
     +##+         +####+
     +##+         +#####
     +##+         ######+
     ###          ######+
     +##          ######+
     +##+         ######+
     +##+         +#####+
      +##+        +####+
      +##+         +###+
       +##+        +###
        +###++++####++
          +########+
           ++++#++
```



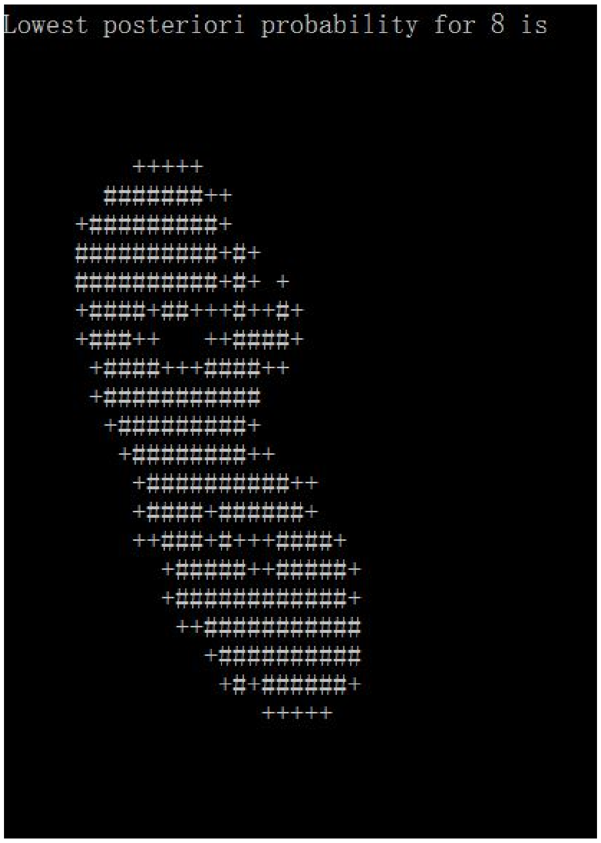
```
Highest posterior probabilitie for 7 is







        +#++ ++++++
        ###########+
        ++###+++++#+
          +++    +#+
                +##+
                +##+
               +##+
               +##+
               +##+
              +##+
              +##+
              ##+
             +##+
             ##+
            +##+
            +#+
           +##
           +##
           +##
           +++
```



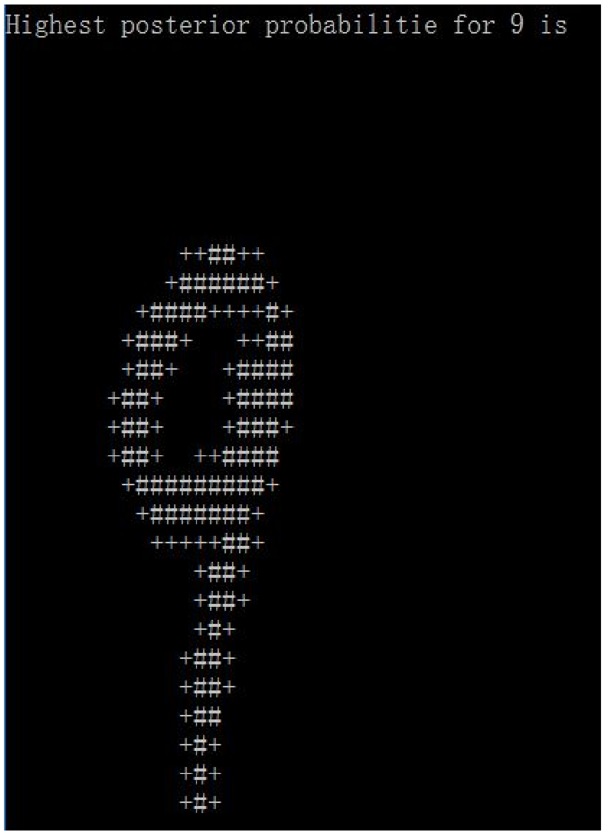
```
Lowest posteriori probability for 7 is







          ###+++    ++++
          ##############+
          +#############+
           ++++#########+
                 +#####
                ++###++
                ####+
              +####+
         +++++####+
        +#########+
        +############+
        +############+
        +####+++++++++
       ++###+
       +###+
      +###+
     +###+
     ####+
     ####+
     +#+
```



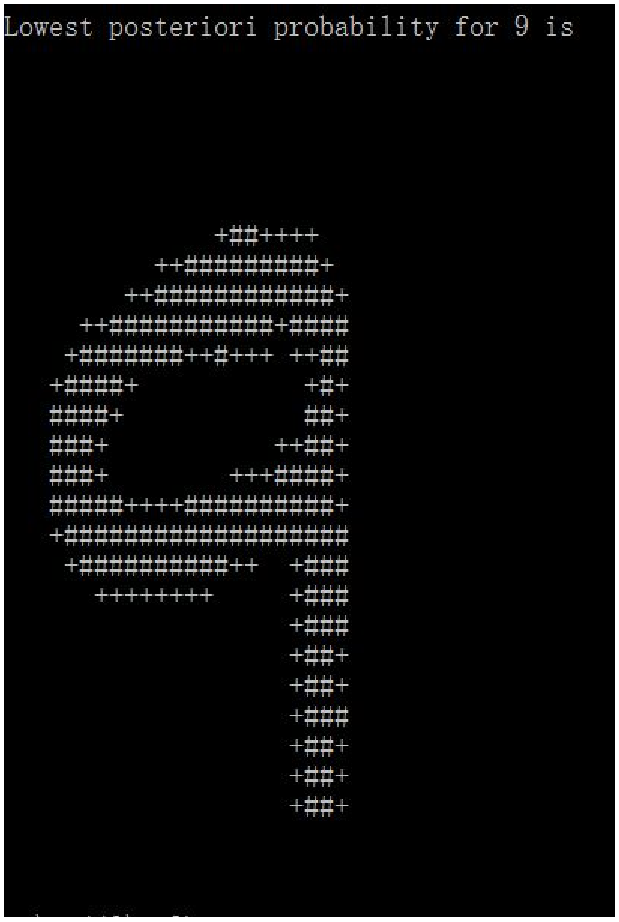
```
Highest posterior probabilitie for 8 is




                +##+
              ++####+
            ++######+
           +####++##+
          ++###+ +##+
          +##+   +##+
          ###     +#++
          ###    ++###
          +##++++###++
          +#######++
           ######++
          +#####
         +######+
         ###++##+
        +##+  ###+
        +##+  ###+
        +##+ +###
        +#######+
         +######+
          ++##++
```



```
Lowest posteriori probability for 8 is




         +++++
       #######++
     +#########+
     ##########+#+
     ##########+#+ +
     +####+##+++#++#+
     +###++   ++####+
      +####+++####++
      +###########
       +#########+
        +########++
         +##########++
         +####+######+
         ++###+#+++####+
           +#####++#####+
           +############+
            ++###########
              +##########
               +#+######+
                  +++++
```



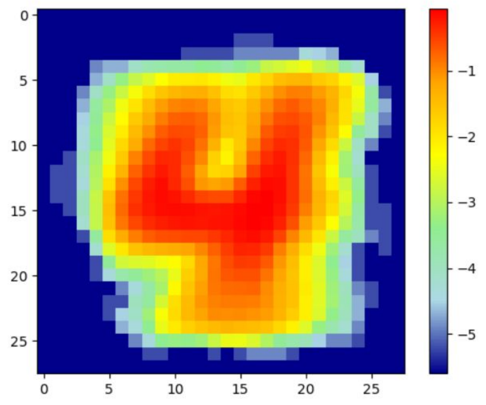
```
Highest posterior probabilitie for 9 is







            ++##++
           +######+
         +####++++#+
        +###+   ++##
        +##+   +####
       +##+    +####
       +##+    +###+
       +##+  ++####
        +#########+
         +#######+
          +++++##+
             +##+
             +##+
             +#+
            +##+
            +##+
            +##
            +#+
            +#+
            +#+
```

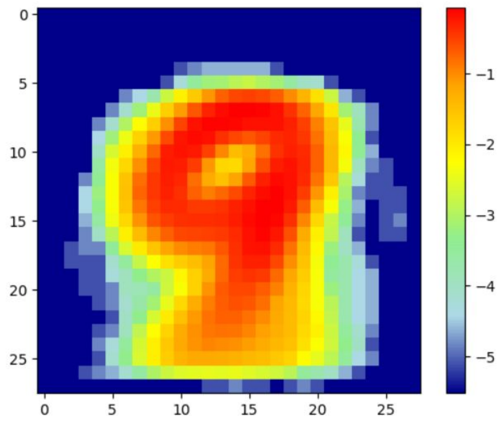


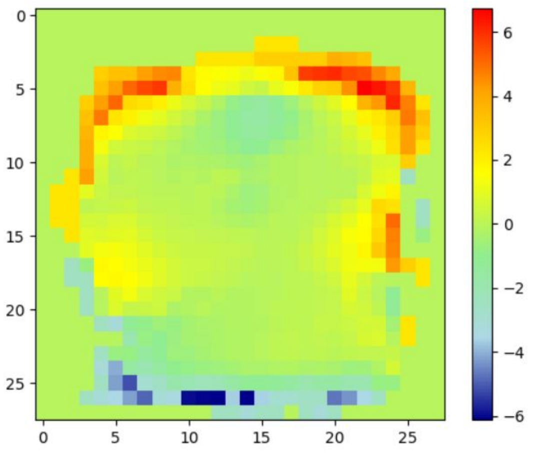
```
Lowest posteriori probability for 9 is






              +##++++
          ++##########+
        ++############+
     ++###########+####
    +#######++#+++ ++##
   +####+           +#+
   ####+            ##+
   ###+           ++##+
   ###+        +++####+
   #####++++##########+
   +###################
    +##########++  +###
      ++++++++     +###
                   +###
                   +##+
                   +##+
                   +###
                   +##+
                   +##+
                   +##+
```

four pairs of digits that have the highest confusion rates are (4,9) (7,9) (8,4) (5,3)
Below are their likelihood and their odds ratio

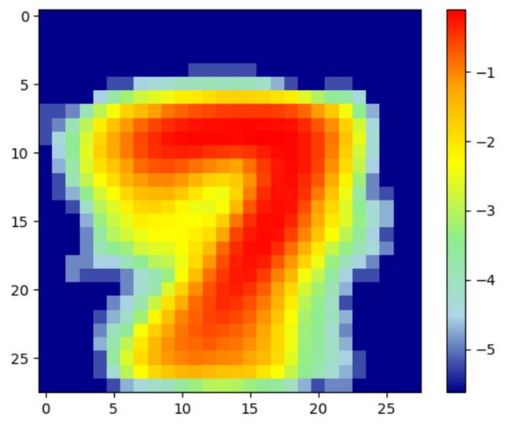
(4,9)

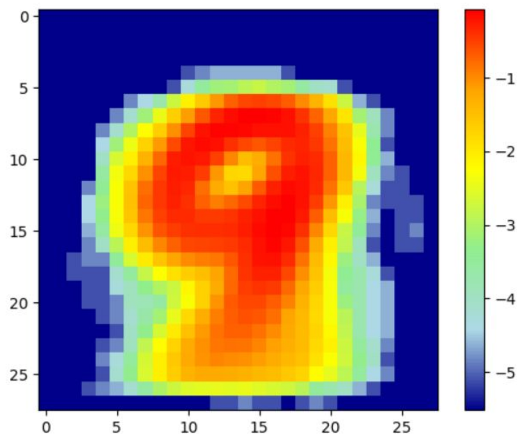


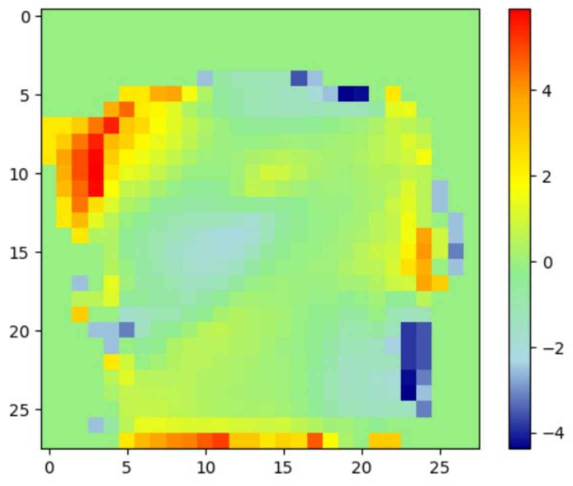




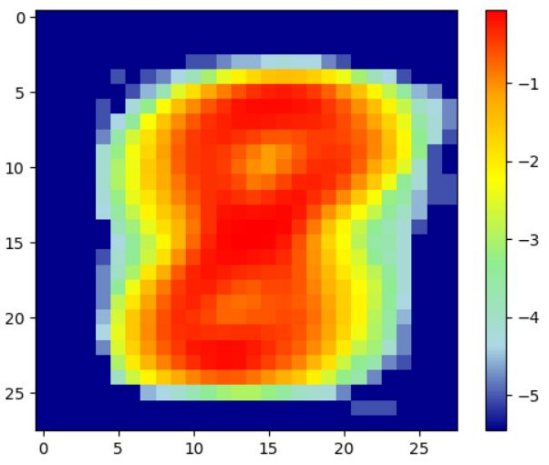
(7,9)

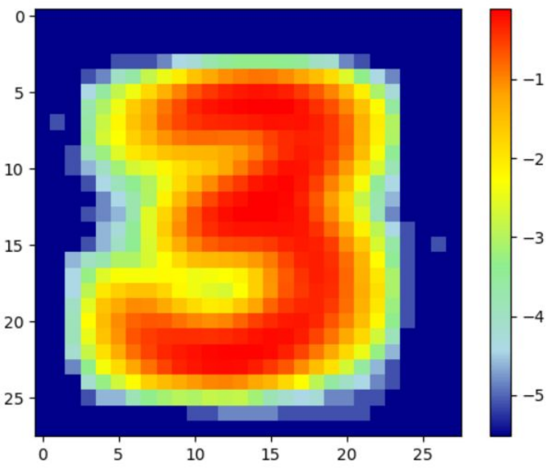


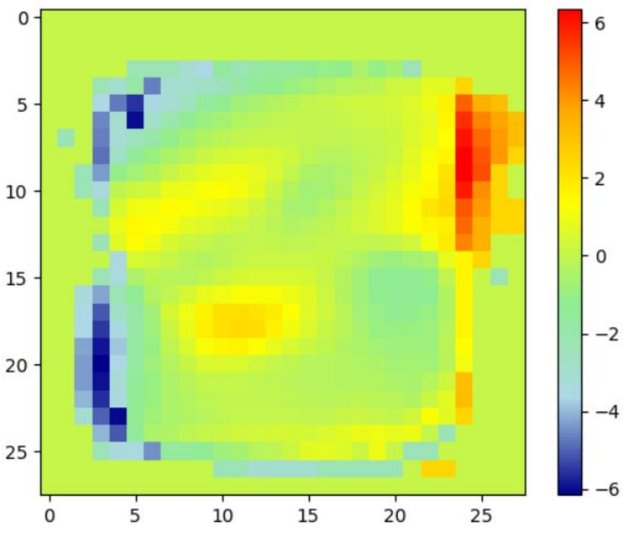




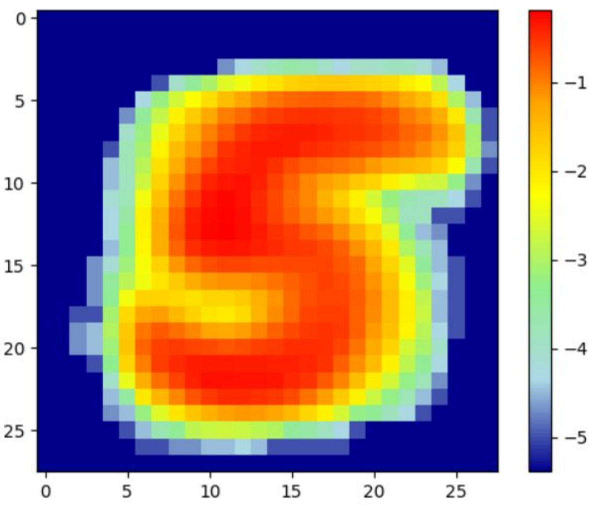
(8,3)



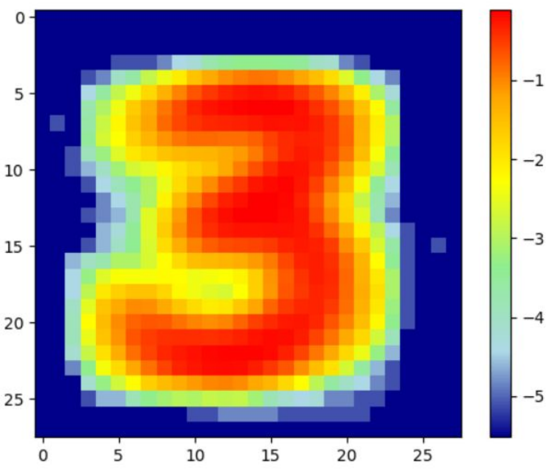




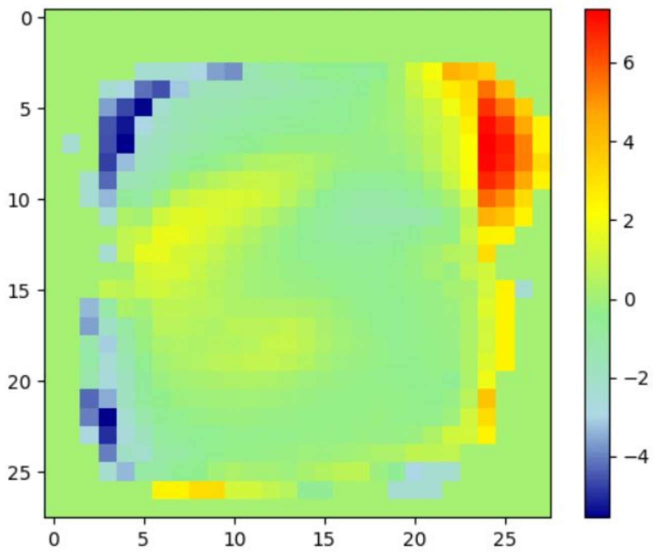
(5,3)



0
5
10
15
20
25
0
5
10
15
20
25
−1
−2
−3
−4
−5



0
5
10
15
20
25
0
5
10
15
20
25
−1
−2
−3
−4
−5



0
5
10
15
20
25
0
5
10
15
20
25
6
4
2
0
−2
−4

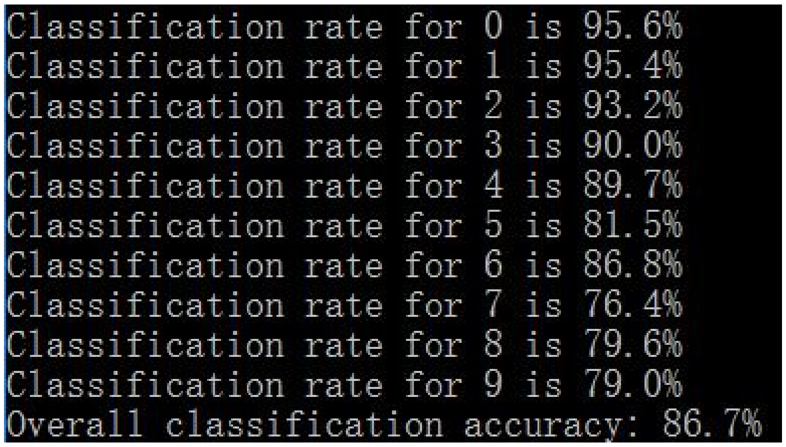
1.2:

For all disjoint patches and overlap patches, I use 0.1 as the smoothing constant.

Disjoint patches:

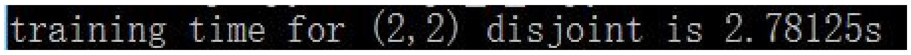
Size 2*2:

Test accuracy:



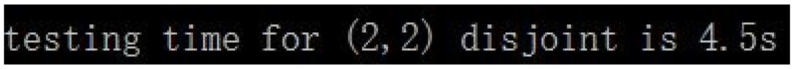
```
Classification rate for 0 is 95.6%
Classification rate for 1 is 95.4%
Classification rate for 2 is 93.2%
Classification rate for 3 is 90.0%
Classification rate for 4 is 89.7%
Classification rate for 5 is 81.5%
Classification rate for 6 is 86.8%
Classification rate for 7 is 76.4%
Classification rate for 8 is 79.6%
Classification rate for 9 is 79.0%
Overall classification accuracy: 86.7%
```

Training time:



```
training time for (2,2) disjoint is 2.78125s
```

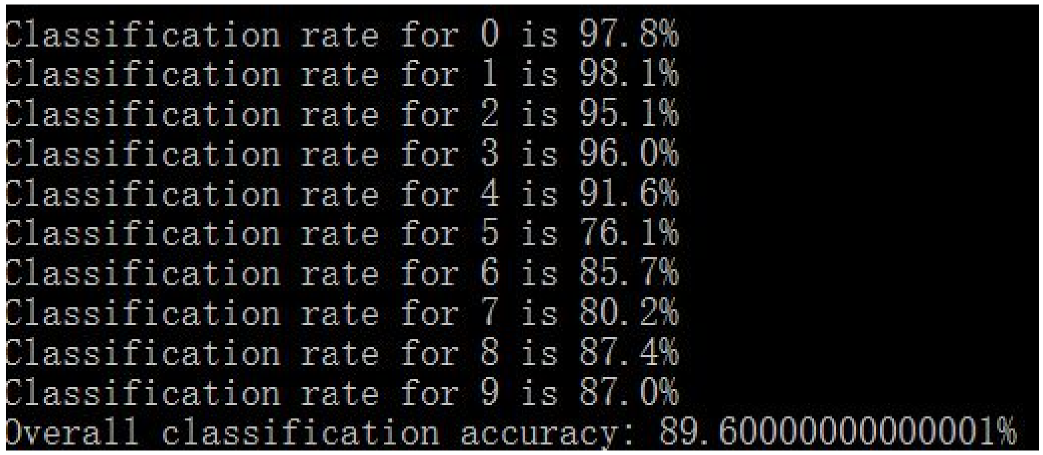
Testing time:



```
testing time for (2,2) disjoint is 4.5s
```

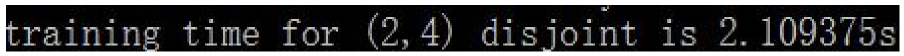
2*4:

Test accuracy:



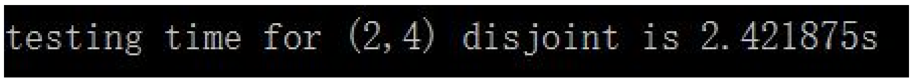
```
Classification rate for 0 is 97.8%
Classification rate for 1 is 98.1%
Classification rate for 2 is 95.1%
Classification rate for 3 is 96.0%
Classification rate for 4 is 91.6%
Classification rate for 5 is 76.1%
Classification rate for 6 is 85.7%
Classification rate for 7 is 80.2%
Classification rate for 8 is 87.4%
Classification rate for 9 is 87.0%
Overall classification accuracy: 89.60000000000001%
```

Training time:



```
training time for (2,4) disjoint is 2.109375s
```

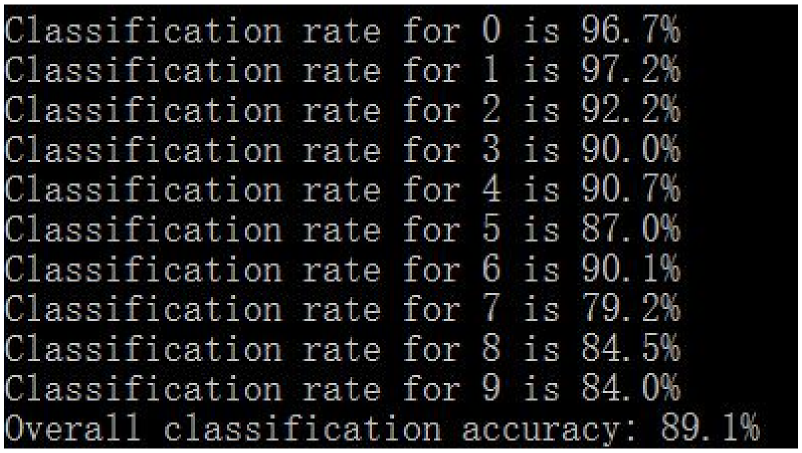
Testing time:



```
testing time for (2,4) disjoint is 2.421875s
```

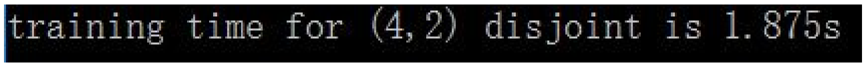
4*2:

Test accuracy:



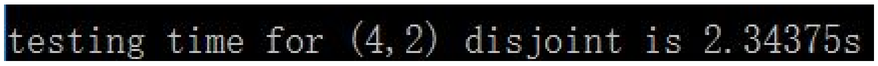
```
Classification rate for 0 is 96.7%
Classification rate for 1 is 97.2%
Classification rate for 2 is 92.2%
Classification rate for 3 is 90.0%
Classification rate for 4 is 90.7%
Classification rate for 5 is 87.0%
Classification rate for 6 is 90.1%
Classification rate for 7 is 79.2%
Classification rate for 8 is 84.5%
Classification rate for 9 is 84.0%
Overall classification accuracy: 89.1%
```

Training time:



```
training time for (4,2) disjoint is 1.875s
```

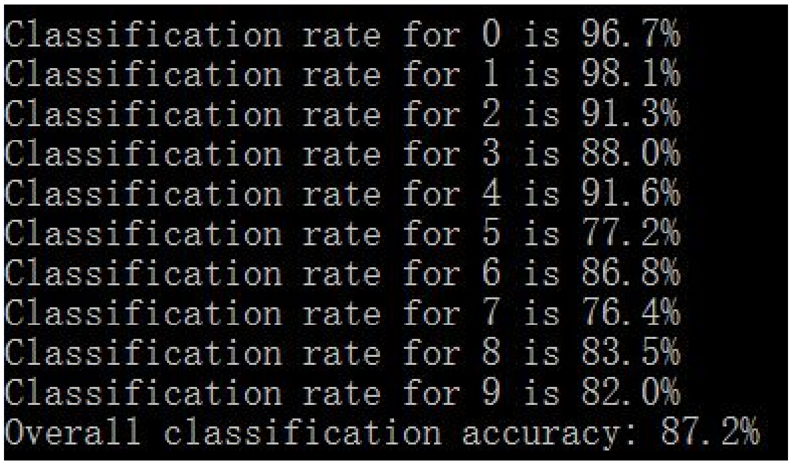
Testing time:



```
testing time for (4,2) disjoint is 2.34375s
```

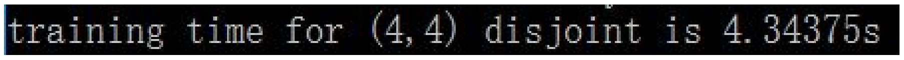
4*4:

Test accuracy:



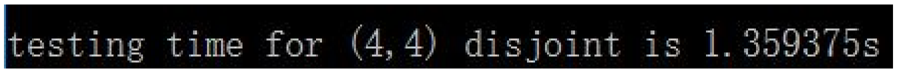
```
Classification rate for 0 is 96.7%
Classification rate for 1 is 98.1%
Classification rate for 2 is 91.3%
Classification rate for 3 is 88.0%
Classification rate for 4 is 91.6%
Classification rate for 5 is 77.2%
Classification rate for 6 is 86.8%
Classification rate for 7 is 76.4%
Classification rate for 8 is 83.5%
Classification rate for 9 is 82.0%
Overall classification accuracy: 87.2%
```

Training time:



```
training time for (4,4) disjoint is 4.34375s
```

Testing time:

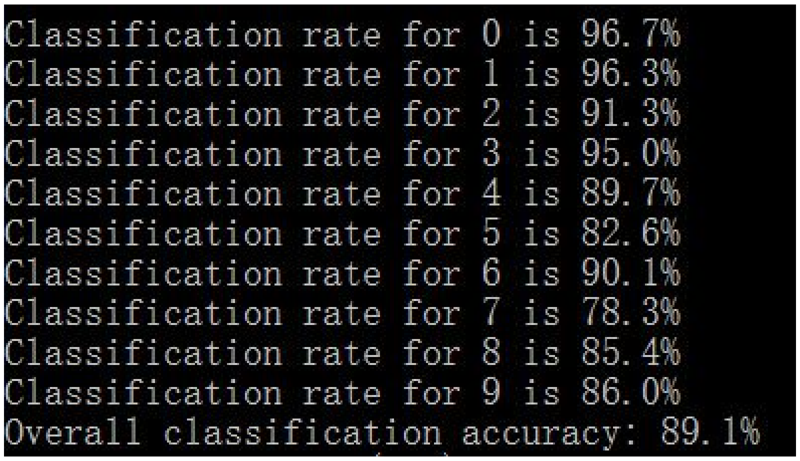


```
testing time for (4,4) disjoint is 1.359375s
```

overlapping patches:

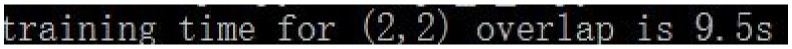
2*2

Test accuracy:



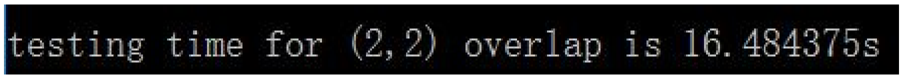
```
Classification rate for 0 is 96.7%
Classification rate for 1 is 96.3%
Classification rate for 2 is 91.3%
Classification rate for 3 is 95.0%
Classification rate for 4 is 89.7%
Classification rate for 5 is 82.6%
Classification rate for 6 is 90.1%
Classification rate for 7 is 78.3%
Classification rate for 8 is 85.4%
Classification rate for 9 is 86.0%
Overall classification accuracy: 89.1%
```

Training time:



```
training time for (2,2) overlap is 9.5s
```

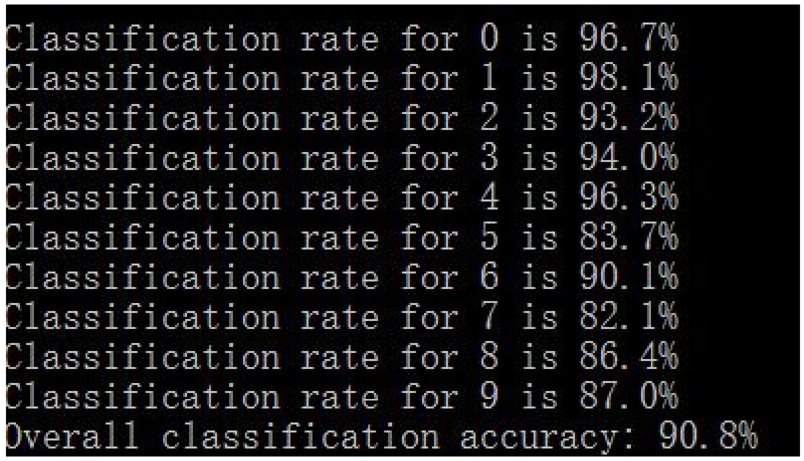
Testing time:



```
testing time for (2,2) overlap is 16.484375s
```

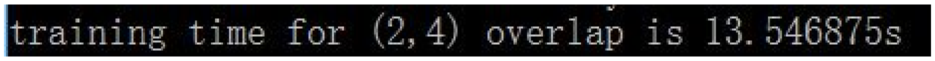
2*4

Test accuracy:



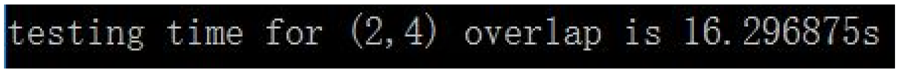
```
Classification rate for 0 is 96.7%
Classification rate for 1 is 98.1%
Classification rate for 2 is 93.2%
Classification rate for 3 is 94.0%
Classification rate for 4 is 96.3%
Classification rate for 5 is 83.7%
Classification rate for 6 is 90.1%
Classification rate for 7 is 82.1%
Classification rate for 8 is 86.4%
Classification rate for 9 is 87.0%
Overall classification accuracy: 90.8%
```

Training time:



```
training time for (2,4) overlap is 13.546875s
```

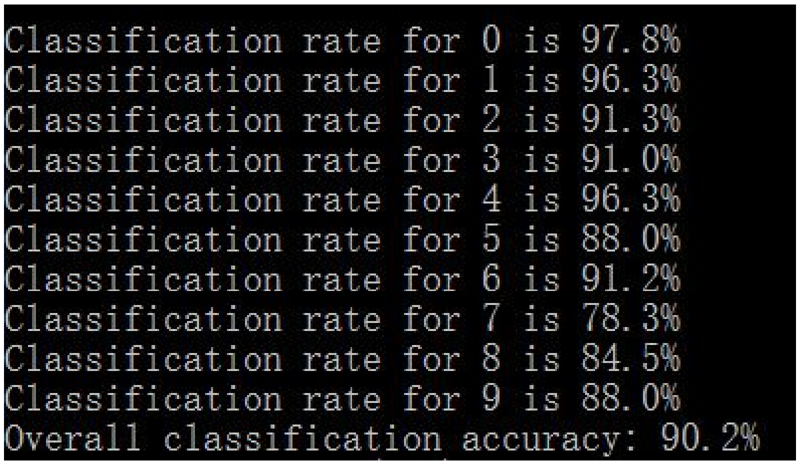
Testing time:



```
testing time for (2,4) overlap is 16.296875s
```

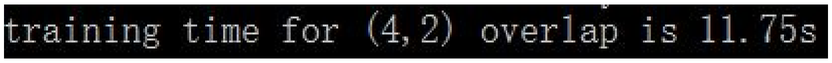
4*2

Test accuracy:



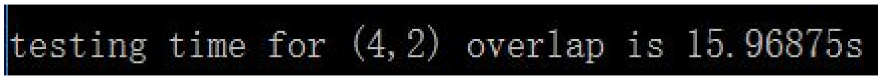
```
Classification rate for 0 is 97.8%
Classification rate for 1 is 96.3%
Classification rate for 2 is 91.3%
Classification rate for 3 is 91.0%
Classification rate for 4 is 96.3%
Classification rate for 5 is 88.0%
Classification rate for 6 is 91.2%
Classification rate for 7 is 78.3%
Classification rate for 8 is 84.5%
Classification rate for 9 is 88.0%
Overall classification accuracy: 90.2%
```

Training time:



```
training time for (4,2) overlap is 11.75s
```

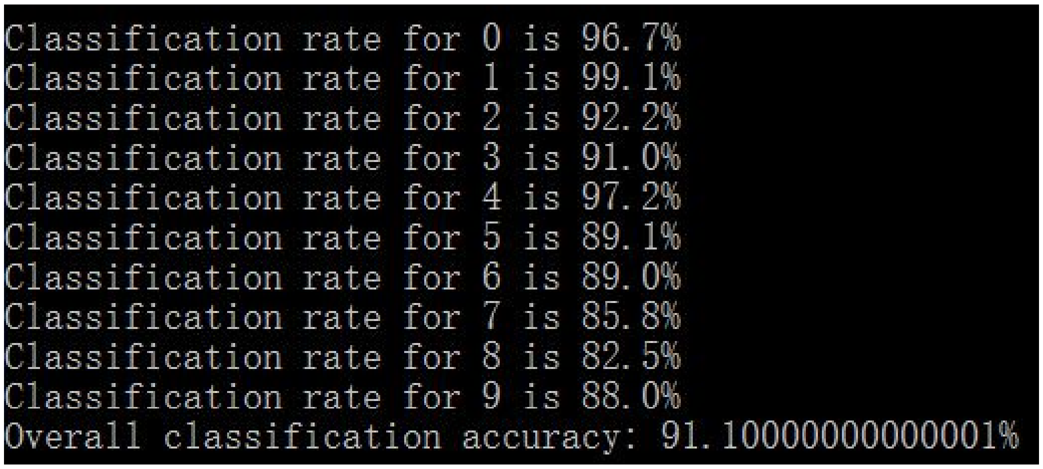
Testing time:



```
testing time for (4,2) overlap is 15.96875s
```

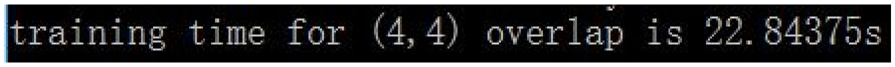
4*4

Test accuracy:



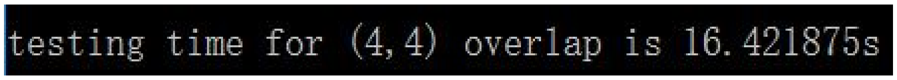
```
Classification rate for 0 is 96.7%
Classification rate for 1 is 99.1%
Classification rate for 2 is 92.2%
Classification rate for 3 is 91.0%
Classification rate for 4 is 97.2%
Classification rate for 5 is 89.1%
Classification rate for 6 is 89.0%
Classification rate for 7 is 85.8%
Classification rate for 8 is 82.5%
Classification rate for 9 is 88.0%
Overall classification accuracy: 91.10000000000001%
```

Training time:



```
training time for (4,4) overlap is 22.84375s
```

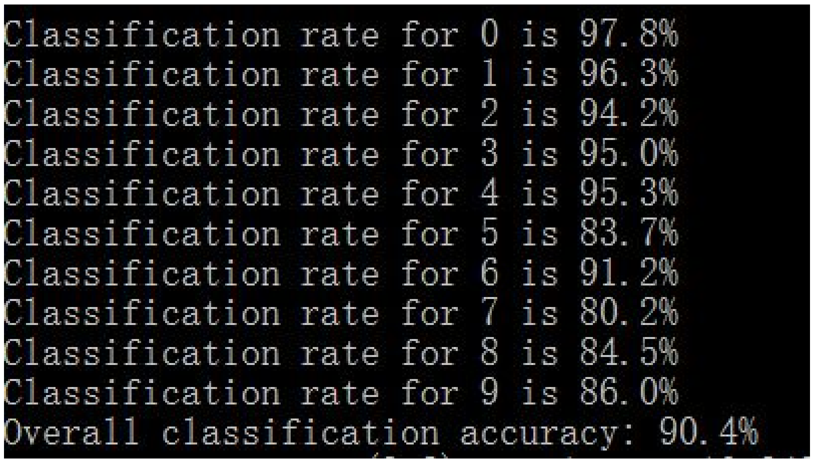
Testing time:



```
testing time for (4,4) overlap is 16.421875s
```

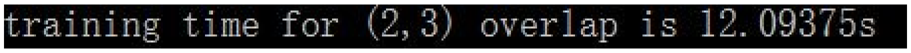
2*3

Test accuracy:



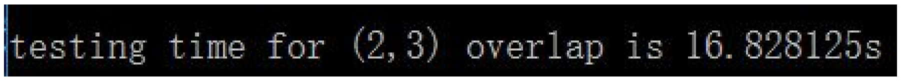
```
Classification rate for 0 is 97.8%
Classification rate for 1 is 96.3%
Classification rate for 2 is 94.2%
Classification rate for 3 is 95.0%
Classification rate for 4 is 95.3%
Classification rate for 5 is 83.7%
Classification rate for 6 is 91.2%
Classification rate for 7 is 80.2%
Classification rate for 8 is 84.5%
Classification rate for 9 is 86.0%
Overall classification accuracy: 90.4%
```

Training time:



```
training time for (2,3) overlap is 12.09375s
```

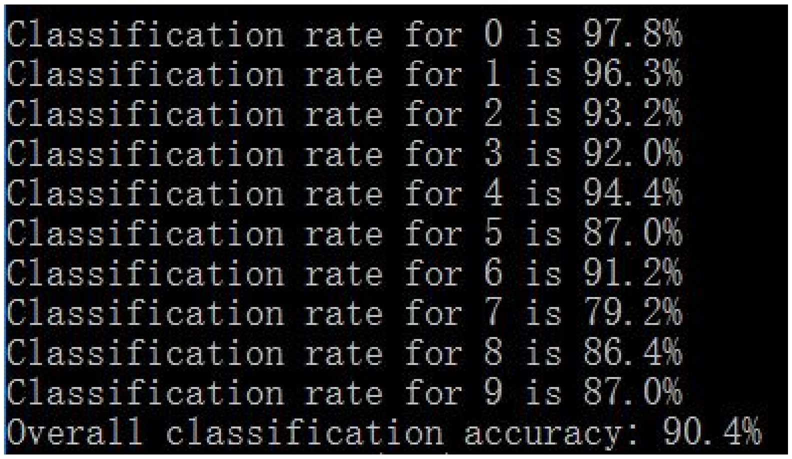
Testing time:



```
testing time for (2,3) overlap is 16.828125s
```

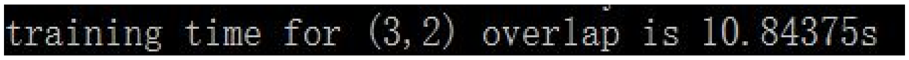
3*2

Test accuracy:



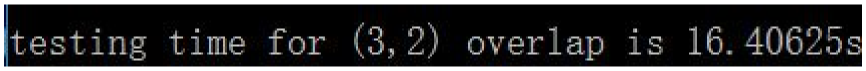
```
Classification rate for 0 is 97.8%
Classification rate for 1 is 96.3%
Classification rate for 2 is 93.2%
Classification rate for 3 is 92.0%
Classification rate for 4 is 94.4%
Classification rate for 5 is 87.0%
Classification rate for 6 is 91.2%
Classification rate for 7 is 79.2%
Classification rate for 8 is 86.4%
Classification rate for 9 is 87.0%
Overall classification accuracy: 90.4%
```

Training time:



```
training time for (3,2) overlap is 10.84375s
```

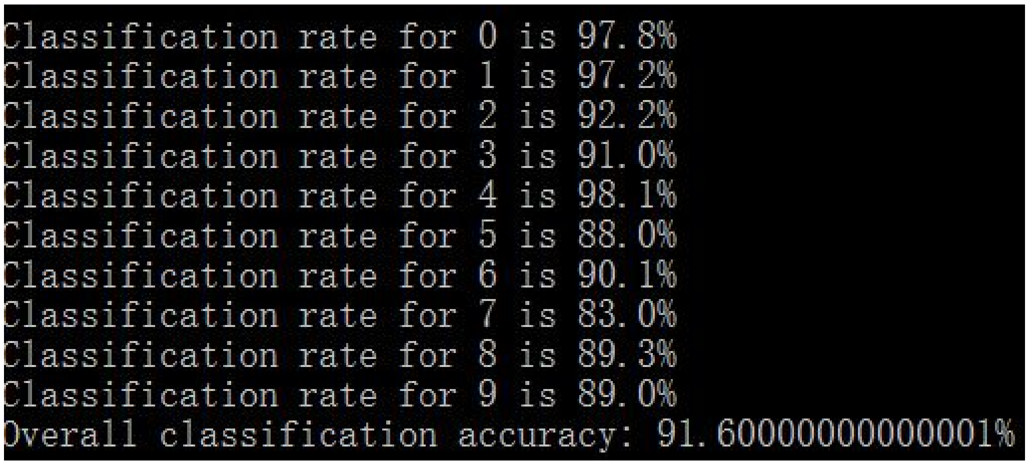
Testing time:



```
testing time for (3,2) overlap is 16.40625s
```

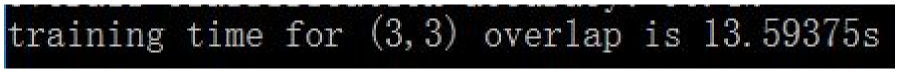
3*3

Test accuracy:



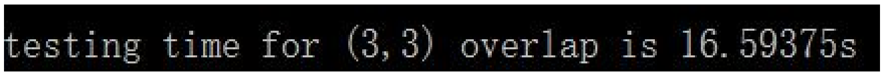
```
Classification rate for 0 is 97.8%
Classification rate for 1 is 97.2%
Classification rate for 2 is 92.2%
Classification rate for 3 is 91.0%
Classification rate for 4 is 98.1%
Classification rate for 5 is 88.0%
Classification rate for 6 is 90.1%
Classification rate for 7 is 83.0%
Classification rate for 8 is 89.3%
Classification rate for 9 is 89.0%
Overall classification accuracy: 91.60000000000001%
```

Training time:



```
training time for (3,3) overlap is 13.59375s
```

Testing time:



```
testing time for (3,3) overlap is 16.59375s
```

Accuracy Trends:

Both the disjoint patches and overlapping patches behave better than signal pixel.

For disjoint patches, as we increase the size of patch, the overall accuracy increases. But if we increase the size too much, like 4*4, the overall accuracy will drop down. For patches that have the same size , the overall accuracy is slightly different, but within a reasonable range.

For overlapping patches, as we increase the size of patch, the overall accuracy increases. For patch that has the same size , the overall accuracy is almost the same.

Overall , the overlapping patches behaves the best among the three. The highest accuracy for overlapping patches is 91.6, which is the largest. I think the reason why such patches behave the best is that it take the most thing into consider. For the same size like 2*2, disjoint patch consider 14*14 features, but overlapping consider 27*27, such increase in consideration will increase the overall accuracy.

Time trend:

For disjoint patches, as we increase the size of patch, the running time and testing time will decrease. But if we increase the size too much, like 4*4, the running time and testing time will increase. For patches that have the same size, the time values are only slightly different.

For overlapping patches, as we increase the size of the patch, the running time and testing time increase. For patches that have the same size , the time values are only slightly different.

Overall, for training time and testing time, disjoint patches are faster. Because as I said above, overlapping patches take more into consideration. It has to do more computation, therefore it is much slower than disjoint patches.

Part 2 Audio Classification

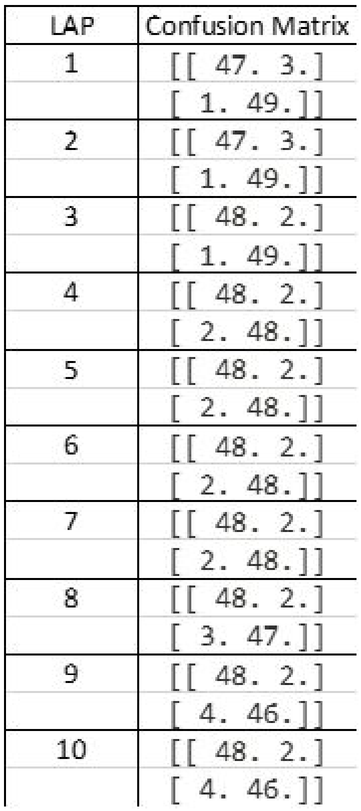
2.1

Implementation

The implementation of this part is similar to part 1. We use two arrays to store P(F_{ij} = f | class = yes) and P(F_{ij} = f | class = no), where F_{ij} represents the i, j-th block of the spectrogram and it has value of 0 or 1. When testing the data, the program will calculate the probability of yes and no respectively and compare the result. It will make the prediction according to the one with higher probability.

Laplacian smoothing constant & Confusion matrix

We tried several different laplacian smoothing constants (from 1 to 10) and the results (represented as confusion matrices) are listed below: (Note that the first column/row is "No" and the second column/row is "Yes")



LAP	Confusion Matrix
1	[[47. 3.]
	[1. 49.]]
2	[[47. 3.]
	[1. 49.]]
3	[[48. 2.]
	[1. 49.]]
4	[[48. 2.]
	[2. 48.]]
5	[[48. 2.]
	[2. 48.]]
6	[[48. 2.]
	[2. 48.]]
7	[[48. 2.]
	[2. 48.]]
8	[[48. 2.]
	[3. 47.]]
9	[[48. 2.]
	[4. 46.]]
10	[[48. 2.]
	[4. 46.]]

From the table above, we can see that when choosing a laplacian smoothing constant (LAP) value of 3, the correctness of this classifier on test data reaches max (97%). When decreasing the LAP value, the classifier generates more error on dealing with "No" data; when increasing the LAP value, the classifier generates more error on dealing with "Yes" data.

2.2

Implementation

Not much change is needed compare to part 2.1. First gather features from the training set, store the features in 5 different feature arrays and, when testing, add up the probability of each label, the label with maximum probability is the predicted result.

Best Accuracy Result

Laplace : 2.5

Confusion matrix:

\Predicted Actual \	1	2	3	4	5
1	7	0	0	0	1
2	0	7	0	1	0
3	0	0	8	0	0
4	0	2	0	6	0
5	0	0	0	0	8

Accuracy: 90%

Tried Laplace Factor and their respective accuracy

Laplace Smoothing Factor	Accuracy
1	87.5%
2	87.5%
3	87.5%
4	82.5%
5	80%
10	80%
50	87.5%
60	90%
100	20%

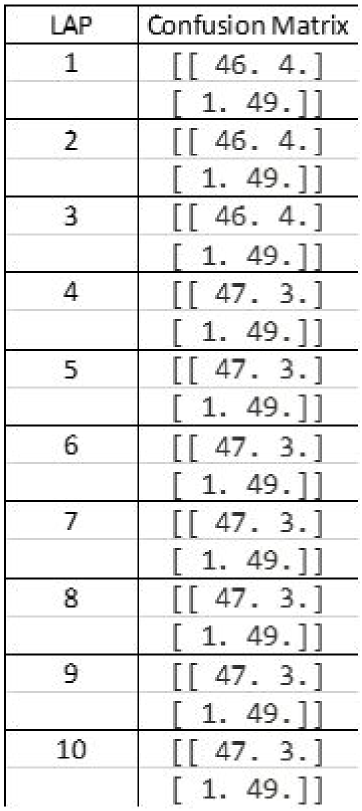
Discussion

Overall accuracy would be around 80% - 90%. Given the limited amount of training data, we think the accuracy is pretty reasonable. We tried some small laplace value at first and thought that the value for highest accuracy would be somewhere around 2, after some trials the value 2.5 stood out to beat others with a 90%. Then we gradually increase the value and found a noticeable decrease in accuracy (values 4 - 10). However, when we tried some big numbers such as 50 and 60, the accuracy is surprisingly good (around 90% mark again), then it rapidly drops to 20% for any value close or greater than 100 (all predicted value becomes “1”, the classifier is no longer functional). The classifier’s behavior under different Laplace values is not easy to explain, the results may be more reliable if given bigger training set and testing set.

Extra Credit

Classification using average column

The laplacian smoothing constants and corresponding confusion matrices are listed below:



LAP	Confusion Matrix
1	[[46. 4.]
	[1. 49.]]
2	[[46. 4.]
	[1. 49.]]
3	[[46. 4.]
	[1. 49.]]
4	[[47. 3.]
	[1. 49.]]
5	[[47. 3.]
	[1. 49.]]
6	[[47. 3.]
	[1. 49.]]
7	[[47. 3.]
	[1. 49.]]
8	[[47. 3.]
	[1. 49.]]
9	[[47. 3.]
	[1. 49.]]
10	[[47. 3.]
	[1. 49.]]

From the table above we can see that when using average column to reduce the dimension of the feature, the laplacian smoothing constant (LAP) has less impact on the result compared to that in 2.1. That is, when decreasing or increasing the LAP, the confusion matrix doesn’t change too much. Moreover, it seems that the classifier reaches max correctness (96%) when LAP is greater than 4.

Disclaimer:

Zhengqi Fang (netID: zf4) is responsible for part2.1

Lingkai Kong(netID: lingkai2) is responsible for part1

Dongxuan Zhang(netID: dz3) is responsible for part2.2

Each member is responsible for the report on the part he works on.