COMP4421 Assignment 3

MOHAMAD, Randitya Setyawan
20316273
rsmohamad@ust.hk

November 30, 2018

Exercises

1. **Lossless Compression**

Original Source			**Source Reduction**									
Intensity	**Prob**	**Code**	**1**		**2**		**3**		**4**		**5**	
1	0.25	10	0.25	10	0.25	10	**0.3125**	11	**0.4375**	0	**0.5625**	1
3	0.25	01	0.25	01	0.25	01	0.25	10	0.3125	11	0.4375	0
9	0.1875	00	0.1875	00	0.1875	00	0.25	01	0.25	10		
7	0.125	110	0.125	110	**0.1875**	111	0.1875	00				
2	0.0625	1110	**0.125**	1111	0.125	110						
12	0.0625	11111	0.0625	1110								
15	0.0625	11110										

Intensity	**Frequency**	**Compressed (bits)**	**Original Size**	**Compressed Size**
1	4	2	16	8
3	4	2	16	8
9	3	2	12	6
7	2	3	8	6
2	1	4	4	4
12	1	5	4	5
15	1	5	4	5
Total			64	42

$$\begin{aligned} \textit{Compression ratio} &= \textit{uncompressed}/\textit{compressed} \\ &= 64/42 \\ &= 1.5238 \end{aligned}$$

2. **Adaboost Learning Algorithm**

Finding the weights:

```
T = 1
Sample weights:
[0.11111111 0.11111111 0.11111111 0.11111111 0.11111111 0.11111111
 0.11111111 0.11111111 0.11111111]
Error = 0.444444
Classifier weight = 0.111572

T = 2
Sample weights:
[0.10557281 0.10557281 0.10557281 0.10557281 0.10557281 0.11803399
 0.11803399 0.11803399 0.11803399]
Error = 0.670820
Classifier weight = -0.355949

T = 3
Sample weights:
[0.092548   0.13211548 0.13211548 0.092548   0.092548   0.10347181
 0.10347181 0.10347181 0.1477096 ]
Error = 0.288568
Classifier weight = 0.451175

T = 4
Sample weights:
[0.0794725  0.11344975 0.11344975 0.12478422 0.12478422 0.139513
 0.08885295 0.08885295 0.12684067]
Error = 0.750432
Classifier weight = -0.550458

T = 5
Sample weights:
[0.06716772 0.09588425 0.09588425 0.18287912 0.18287912 0.11791211
 0.07509579 0.07509579 0.10720185]
Error = 0.538358
Classifier weight = -0.076866
```

(a) H3 and H1 have the highest weights. Final classifier:

$$H(\mathbf{x}) = sgn(0.4512 \cdot h_3(\mathbf{x}) + 0.1116 \cdot h_1(\mathbf{x}))$$

(b) Final classifier response:

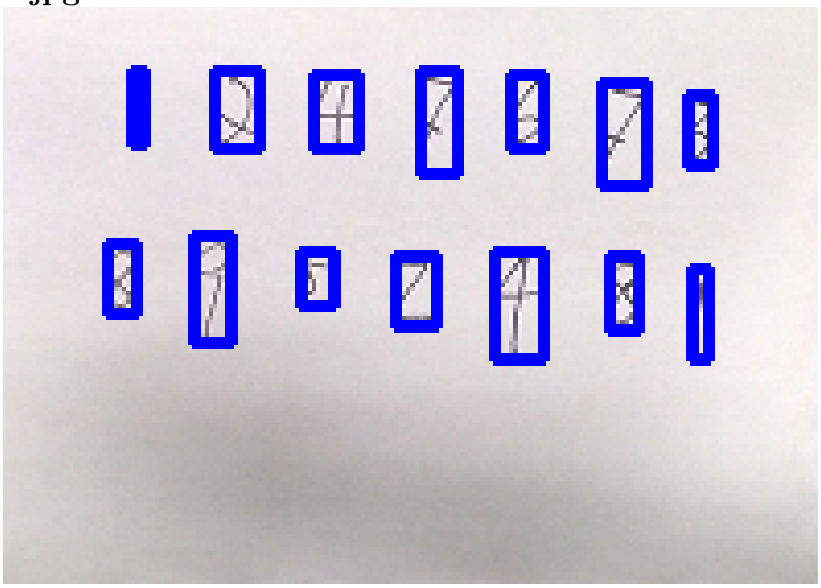
```
[1, 1, -1, 1, -1, -1, -1, 1, -1]
```

Programming

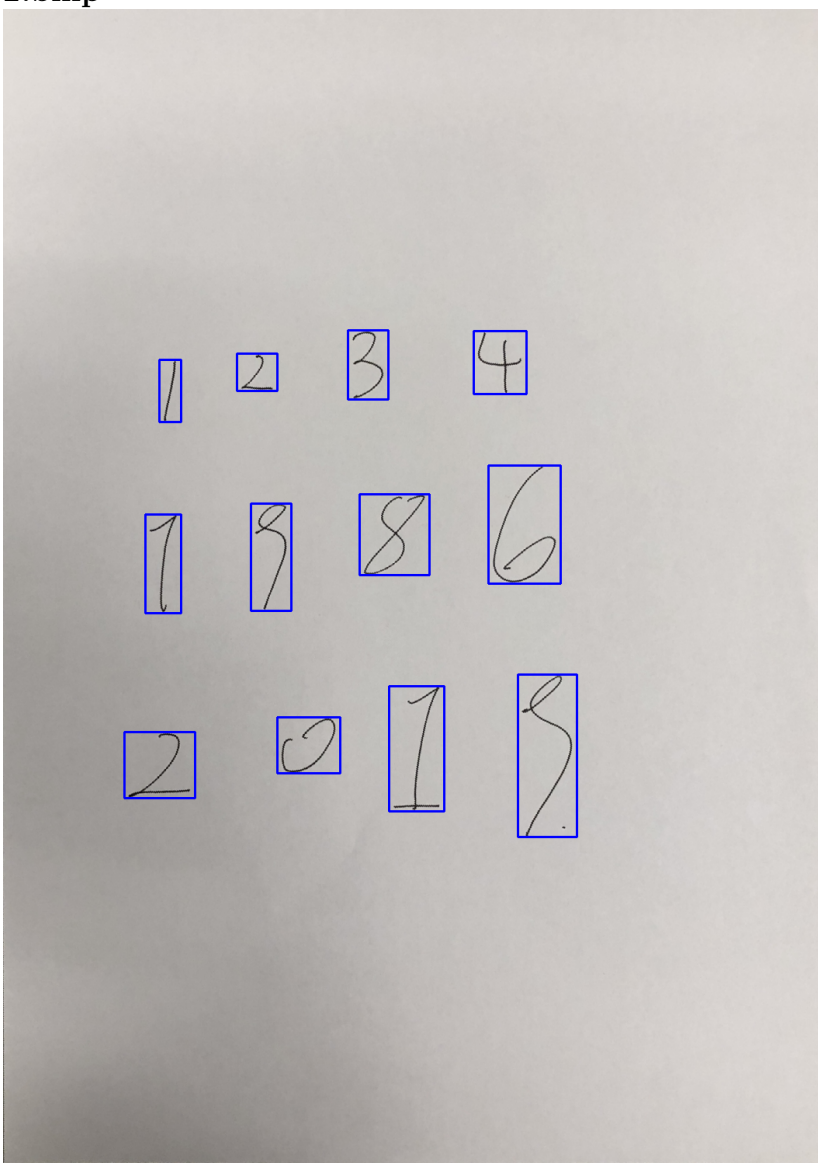
1. **Digit Segmentation**
   I used the given segmentation algoritm (row and column projections) with additional preprocessing and different method to handle digits that are close.

   - **Step 1**: Clean and binarize the image
     - Convert the image to grayscale
     - Obtain the background (image sans digits) by dilating the grayscale image
     - Subtract the background to get a clean image of digits
     - Binarize the digits using OTSU algorithm
   - **Step 2**: Separate the rows using the y-projections on the binarized image
   - **Step 3**: Separate the columns in each row using x-projections
   - **Step 4**: Create ROIs using the obtained rows and columns
   - **Step 5**: Each ROI may contain 2 close digits, separate them by finding connected components

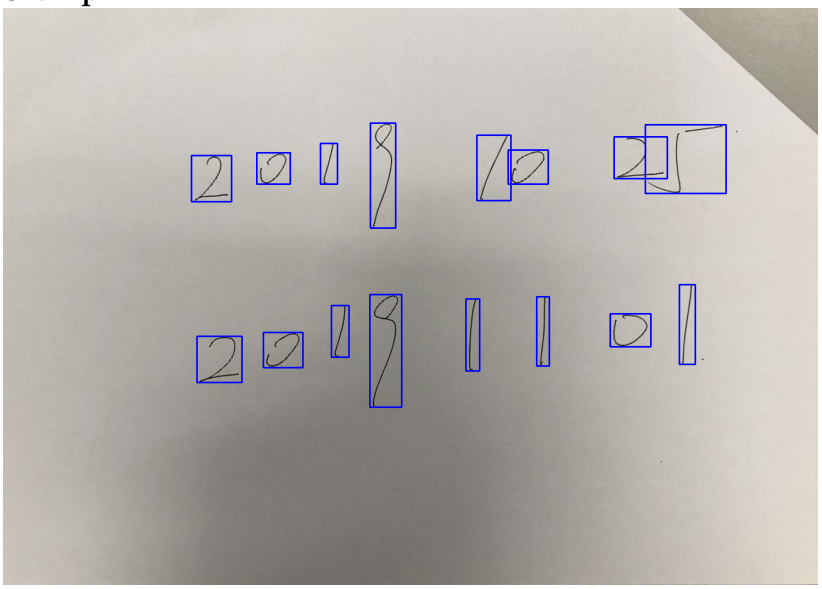
   1.jpg



2.bmp

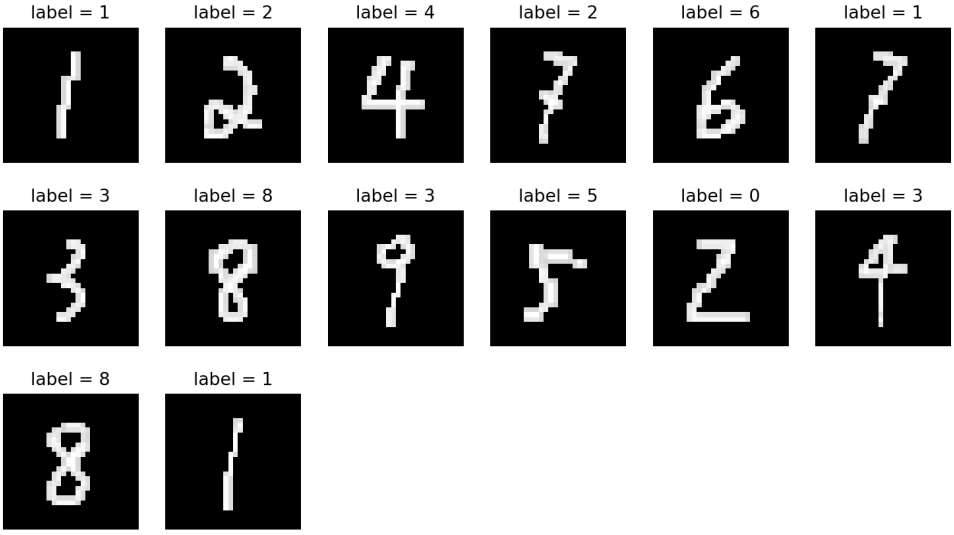


3.bmp

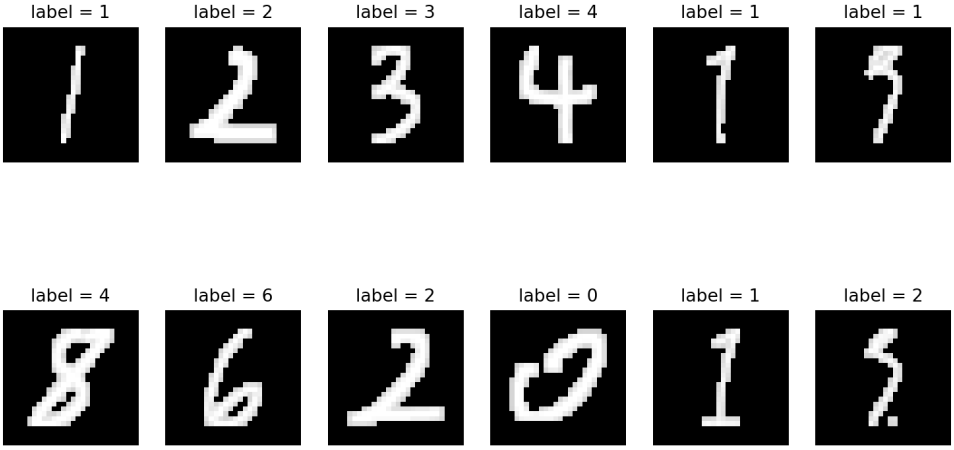


2. **Adaboost Classification**
   I used 5 decision trees for the weak classifiers

   1.jpg - accuracy=0.6429



   2.bmp - accuracy=0.833



3.bmp - accuracy=0.8125