L3-EMNIST

September 28, 2020

[1]:
```
from google.colab import drive
drive.mount('/content/gdrive',force_remount=True)
```

```
Mounted at /content/gdrive
```

[9]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

import data

[6]:
```
train = pd.read_csv('/content/gdrive/My Drive/Dacon/ComputerVision/' + 'train.
 ↪csv')
train_digit = train['digit'].values
train_letter = train['letter'].values
```

[15]:
```
train
```

[15]:

	id	digit	letter	0	1	2	3	4	...	776	777	778	779	780	781	782	783
0	1	5	L	1	1	1	4	3	...	0	1	2	4	4	4	3	4
1	2	0	B	0	4	0	0	4	...	0	1	4	1	4	2	1	2
2	3	4	L	1	1	2	2	1	...	3	0	2	0	3	0	2	2
3	4	9	D	1	2	0	2	0	...	2	0	1	4	0	0	1	1
4	5	6	A	3	0	2	4	0	...	3	2	1	3	4	3	1	2
...	...	...	...	..	..	..	..	..	...	...	...	...	...	...	...	...	...
2043	2044	6	V	2	4	3	4	2	...	2	0	0	1	3	1	4	0
2044	2045	1	L	3	2	2	1	1	...	4	2	1	2	3	4	1	1
2045	2046	9	A	4	0	4	0	2	...	1	1	3	4	2	2	0	0
2046	2047	0	Z	2	3	3	0	3	...	1	1	0	4	1	4	3	

```
1
2047  2048      5      Z  4  2  2  1  3  ...    4    0    4    3    2    4    3
4

[2048 rows x 787 columns]
```

to look at the unique values

[7]:
```
print('digit  : ', np.unique(train_digit))
print('letter : ', np.unique(train_letter))
```

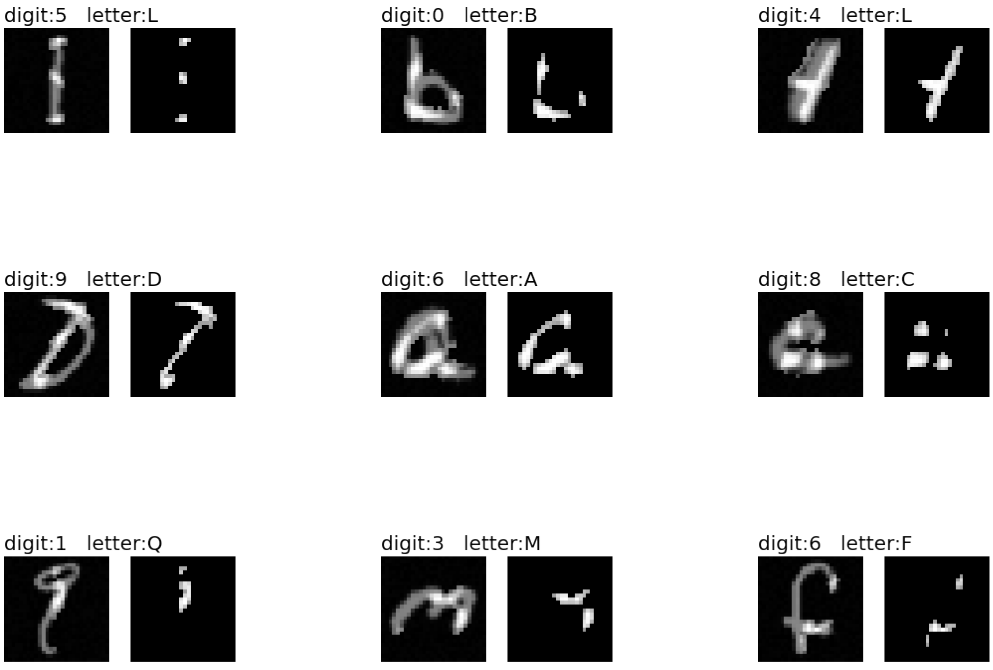
```
digit  :  [0 1 2 3 4 5 6 7 8 9]
letter :  ['A' 'B' 'C' 'D' 'E' 'F' 'G' 'H' 'I' 'J' 'K' 'L' 'M' 'N' 'O' 'P' 'Q'
'R'
 'S' 'T' 'U' 'V' 'W' 'X' 'Y' 'Z']
```

see how numbers are hidden

[11]:
```
for idx in range(0, 100, 3):
    plt.figure(figsize=(20,30))
    plt.subplot(1,9,1)
    plt.imshow(train_img[idx].reshape(28,28),cmap='gray')
    plt.axis('off')

    plt.title('digit:{}   letter:{}'.format(train_digit[idx],␣
 ↪train_letter[idx]), loc='left', fontsize=20)

    plt.subplot(1,9,2)
    data = np.where(train_img>=150, train_img, 0)
    plt.imshow(data[idx].reshape(28,28),cmap='gray')
    plt.axis('off')

    plt.subplot(1,9,3)
    plt.imshow(np.zeros((28,28,3))+1,cmap='gray')
    plt.axis('off')


    plt.subplot(1,9,4)
    plt.imshow(train_img[idx+1].reshape(28,28),cmap='gray')
    plt.axis('off')

    plt.title('digit:{}   letter:{}'.format(train_digit[idx+1],␣
 ↪train_letter[idx+1]), loc='left', fontsize=20)

    plt.subplot(1,9,5)
    data = np.where(train_img>=150, train_img, 0)
    plt.imshow(data[idx+1].reshape(28,28),cmap='gray')
    plt.axis('off')

    plt.subplot(1,9,6)
```

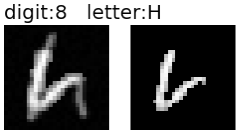
```
    plt.imshow(np.zeros((28,28,3))+1,cmap='gray')
    plt.axis('off')


    plt.subplot(1,9,7)
    plt.imshow(train_img[idx+2].reshape(28,28),cmap='gray')
    plt.axis('off')

    plt.title('digit:{}   letter:{}'.format(train_digit[idx+2],␣
 ↪train_letter[idx+2]), loc='left', fontsize=20)

    plt.subplot(1,9,8)
    data = np.where(train_img>=150, train_img, 0)
    plt.imshow(data[idx+2].reshape(28,28),cmap='gray')
    plt.axis('off')

    plt.subplot(1,9,9)
    plt.imshow(np.zeros((28,28,3))+1,cmap='gray')
    plt.axis('off')

plt.show()
```

```
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:2: RuntimeWarning:
More than 20 figures have been opened. Figures created through the pyplot
interface (`matplotlib.pyplot.figure`) are retained until explicitly closed and
may consume too much memory. (To control this warning, see the rcParam
`figure.max_open_warning`).
```

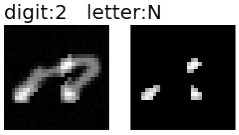


digit:8 letter:J

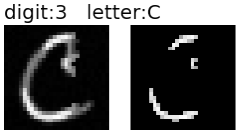
digit:8 letter:H



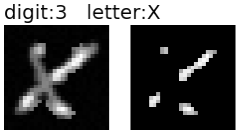
digit:2 letter:N



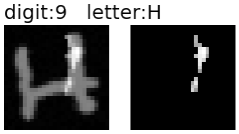
digit:3 letter:C



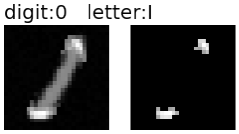
digit:3 letter:X



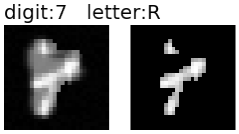
digit:9 letter:H



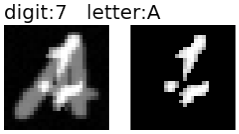
digit:0 letter:I



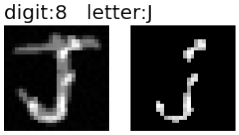
digit:7 letter:R



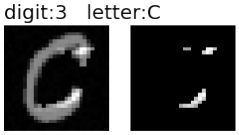
digit:7 letter:A



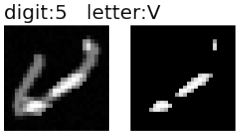
digit:8 letter:J



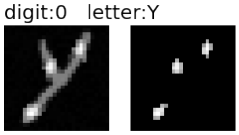
digit:3 letter:C



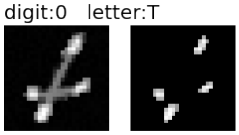
digit:5 letter:V



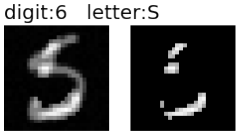
digit:0 letter:Y



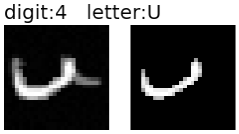
digit:0 letter:T



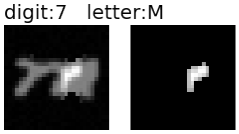
digit:6 letter:S



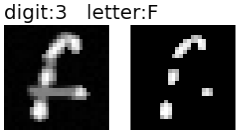
digit:4 letter:U



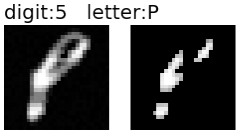
digit:7 letter:M



digit:3 letter:F

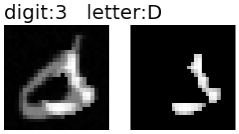


digit:5 letter:P

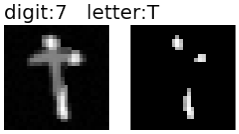


digit:0 letter:H

digit:3 letter:D

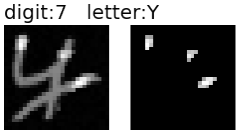


digit:7 letter:T

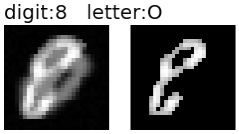


digit:2 letter:K

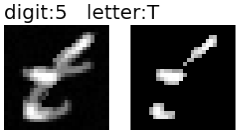
digit:7 letter:Y



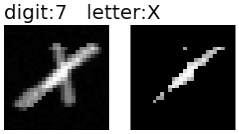
digit:8 letter:O



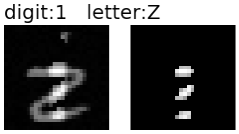
digit:5 letter:T



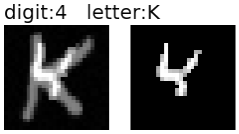
digit:7 letter:X



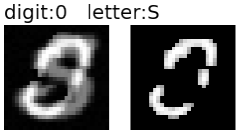
digit:1 letter:Z



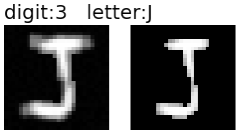
digit:4 letter:K



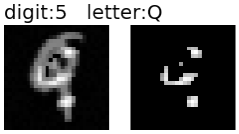
digit:0 letter:S



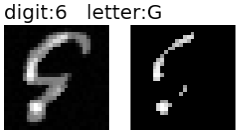
digit:3 letter:J



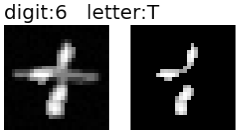
digit:5 letter:Q



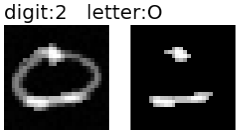
digit:6 letter:G



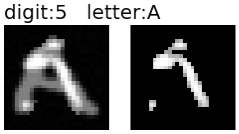
digit:6 letter:T



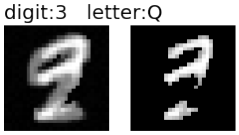
digit:2 letter:O



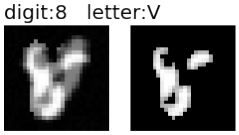
digit:5 letter:A



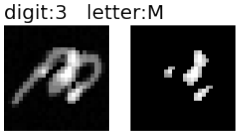
digit:3 letter:Q



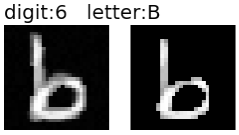
digit:8 letter:V



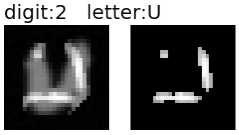
digit:3 letter:M



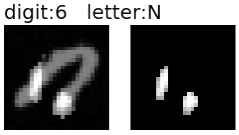
digit:6 letter:B



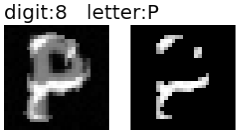
digit:2 letter:U



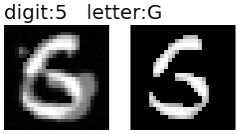
digit:6 letter:N



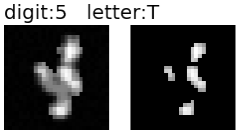
digit:8 letter:P



digit:5 letter:G

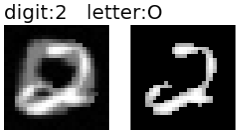


digit:5 letter:T

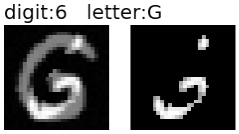


digit:7 letter:F

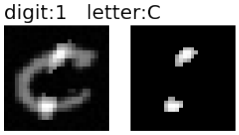
digit:2 letter:O



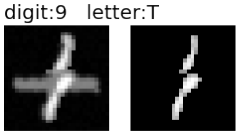
digit:6 letter:G



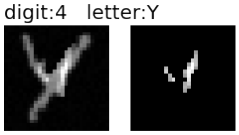
digit:1 letter:C



digit:9 letter:T

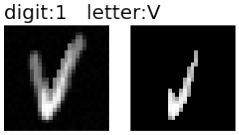


digit:4 letter:Y

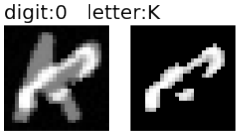


digit:8 letter:B

digit:1 letter:V

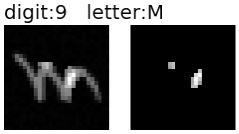


digit:0 letter:K

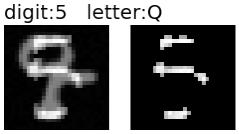


digit:7 letter:G

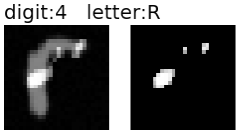
digit:9 letter:M



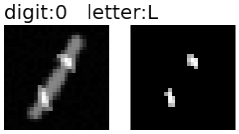
digit:5 letter:Q



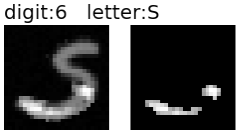
digit:4 letter:R



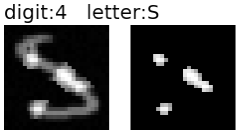
digit:0 letter:L



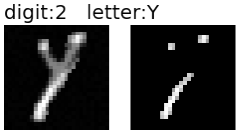
digit:6 letter:S



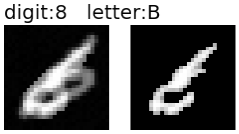
digit:4 letter:S



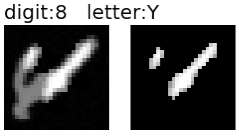
digit:2 letter:Y



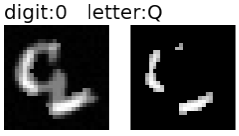
digit:8 letter:B



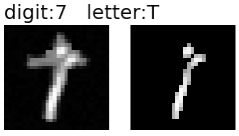
digit:8 letter:Y



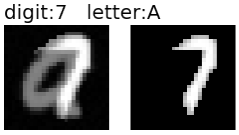
digit:0 letter:Q



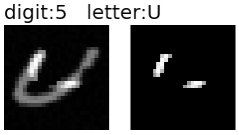
digit:7 letter:T



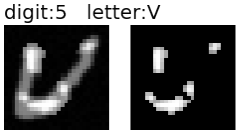
digit:7 letter:A



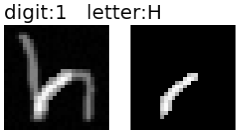
digit:5 letter:U



digit:5 letter:V

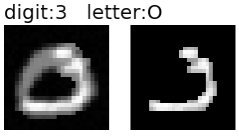


digit:1 letter:H

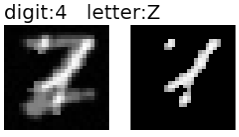


digit:6 letter:G

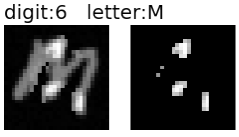
digit:3 letter:O



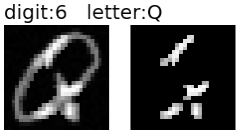
digit:4 letter:Z



digit:6 letter:M



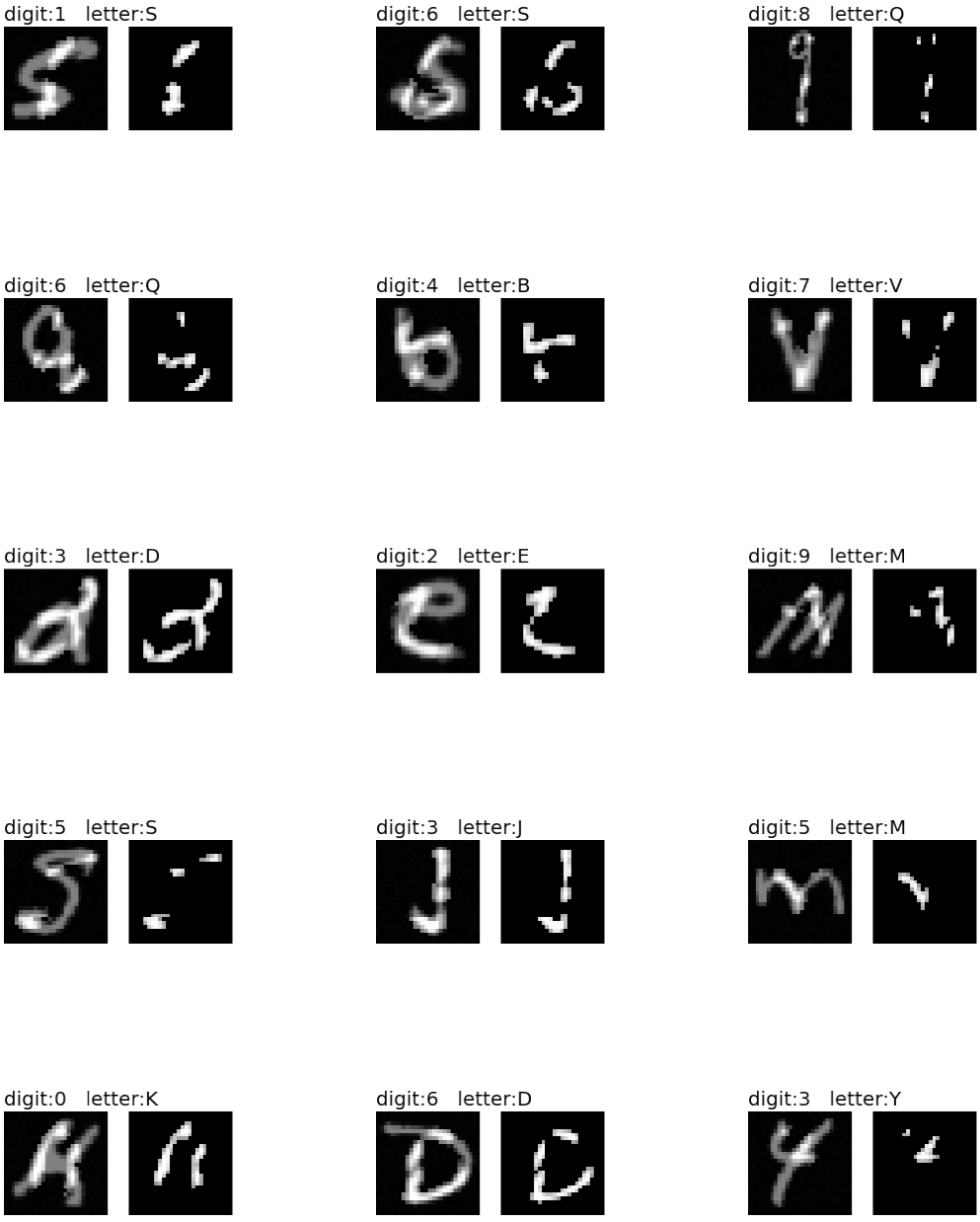
digit:6 letter:Q



digit:5 letter:H

digit:8 letter:P

digit:8 letter:Y

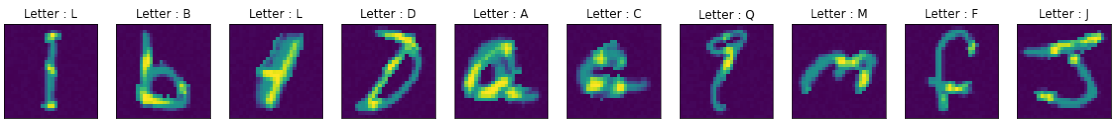


[22]:
```
from tensorflow.keras.utils import to_categorical
```

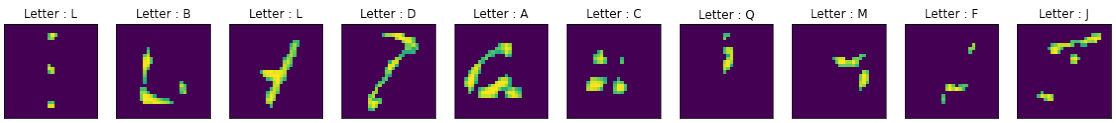
[42]:
```
def show_10_imgs(imgs, labels = [False]):
    plt.figure(figsize=(20, 5))
    for i in range(10):
        ax = plt.subplot(2, 10, i + 1)
        plt.imshow(imgs[i])
        if labels[0]:
```

```
            letter = labels[i]
            ax.set_title(f'Letter : {letter}')
        ax.get_xaxis().set_visible(False)
        ax.get_yaxis().set_visible(False)

    plt.show()
```

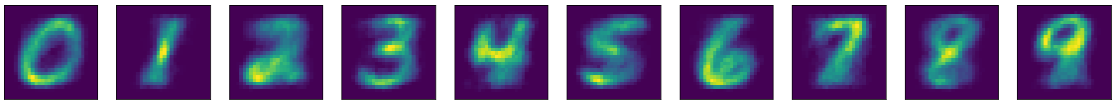
[43]:
```
X = train.iloc[:,3:].values.reshape((-1,28,28))
show_10_imgs(X,train_letter)  #
```



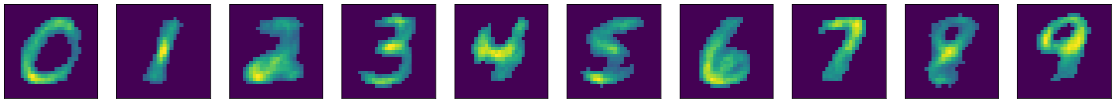
[44]:
```
X = X / 255.
threshold = 0.6
X[X < threshold] = 0
show_10_imgs(X,train_letter)
```



[45]:
```
digit_ref = np.zeros((10, 28, 28))
for (idx, digit) in enumerate(train_digit):
    digit = np.argmax(to_categorical(digit,10))
    digit_ref[digit] += X[idx]

digit_ref = digit_ref / np.max(digit_ref)
show_10_imgs(digit_ref)
```



[46]:
```
digit_ref[digit_ref < 0.25] = 0
show_10_imgs(digit_ref)
```



[47]:
```
digit_ref[digit_ref != 0] = 1
show_10_imgs(digit_ref)
```