Assignment-2: Cs215

Saikiran-200050023, Kamal-200050142

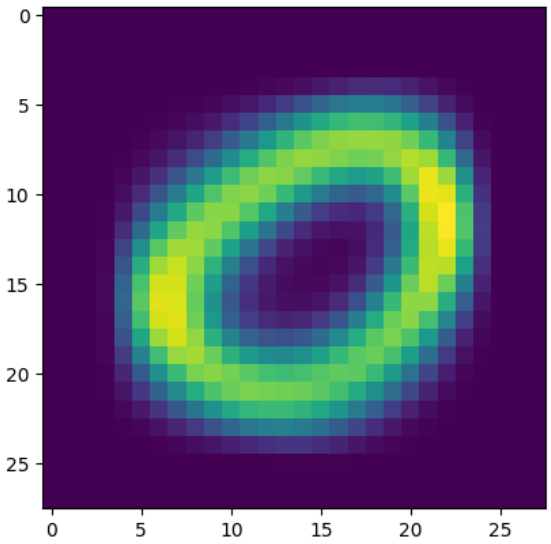
Q4

Instructions to run the code are given in the end

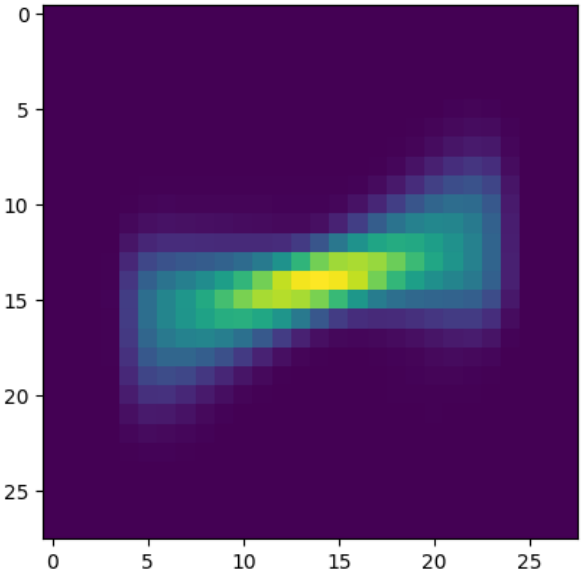
Mean

For extracting the training data from the file “minst.mat” a python package named **h5py** is used (pip install h5py will do the installation). After extracting the data, the images are stored in the matrices digits_train and corresponding labels to the labels_train matrix, 5 new NumPy matrices are defined for storing the mean, covariance matrix, eigen values and eigen vectors for each digit. One can access the data for each digit by matrix[i]. the for loop in the line 18 is initiated to calculate the sum of all matrices and a*transpose(a) for each digit and the next for loop calculates the mean. On running the python script q4.py 10 images names in the format mean_label{i}.png corresponding to each digit is created and stored in the results directory and given below

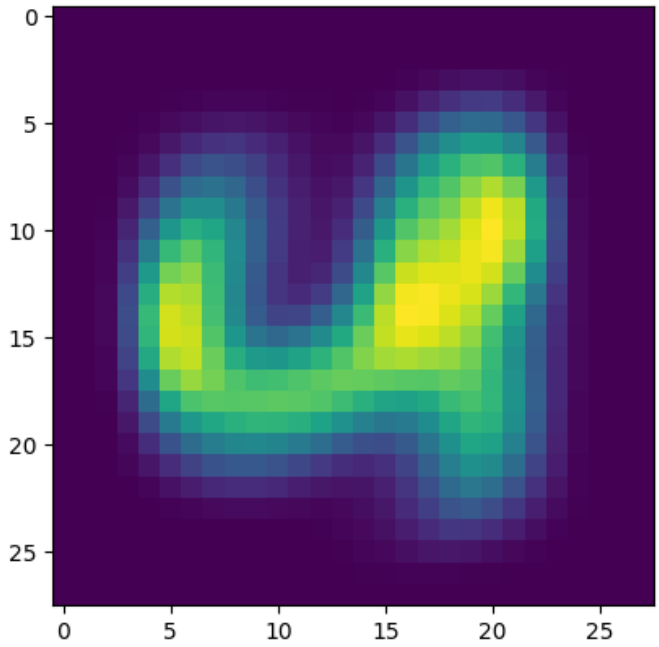
Label-0



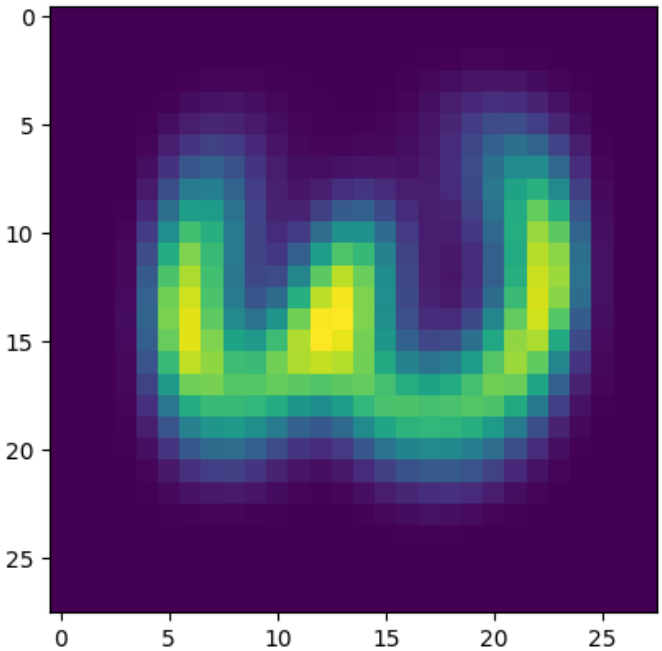
Label-1



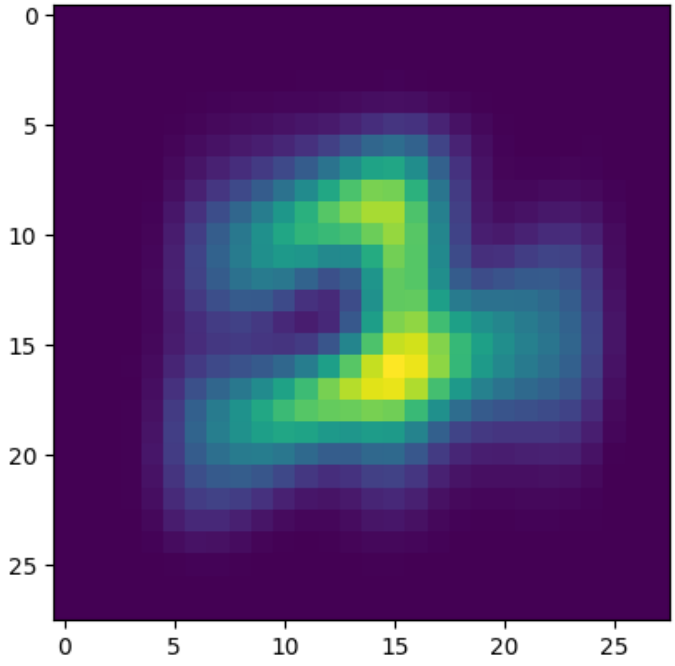
Label-2



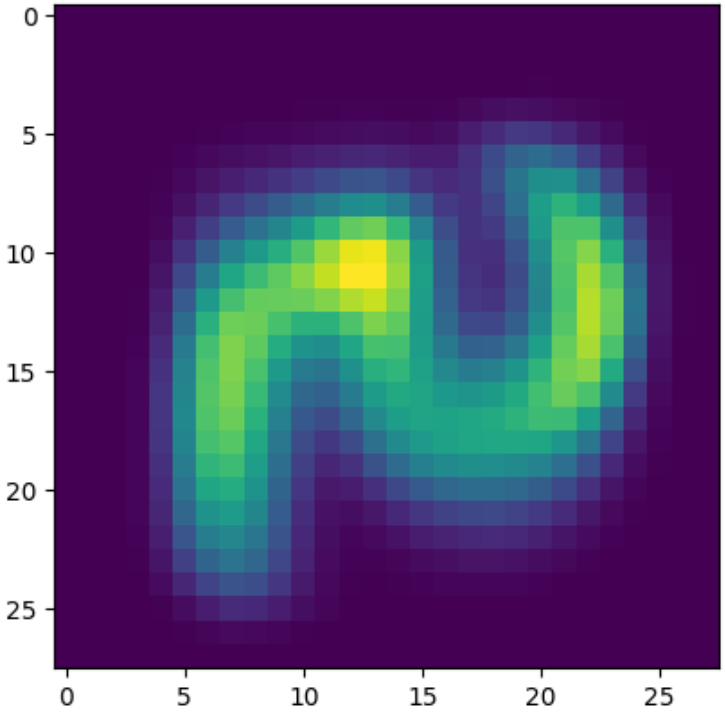
Label-3



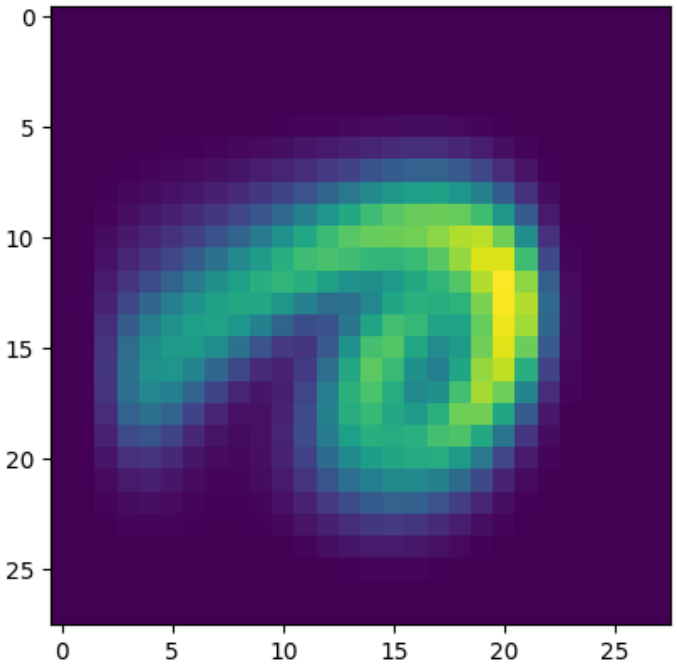
Label-4



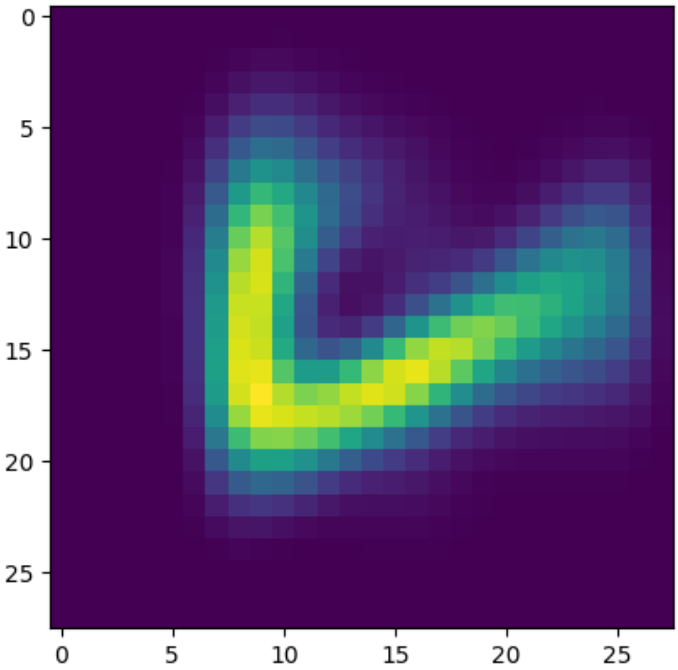
Label-5



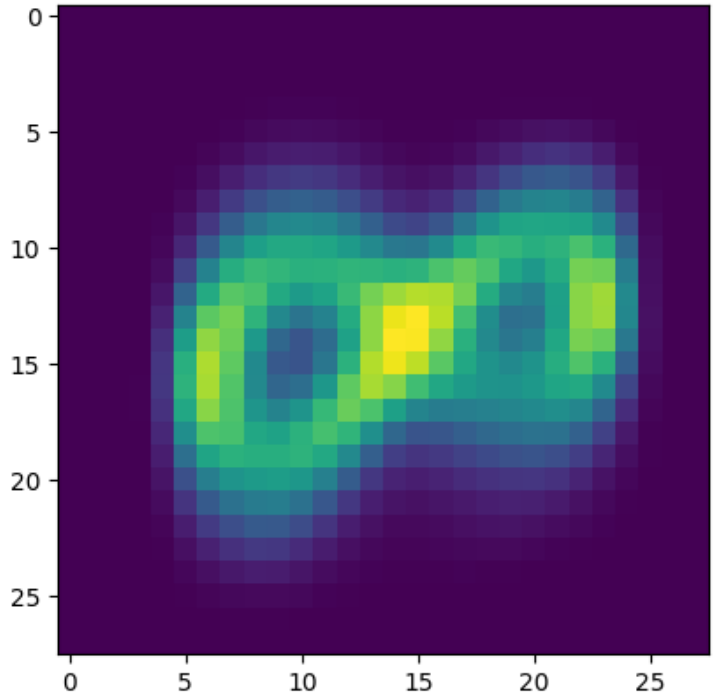
Label-6



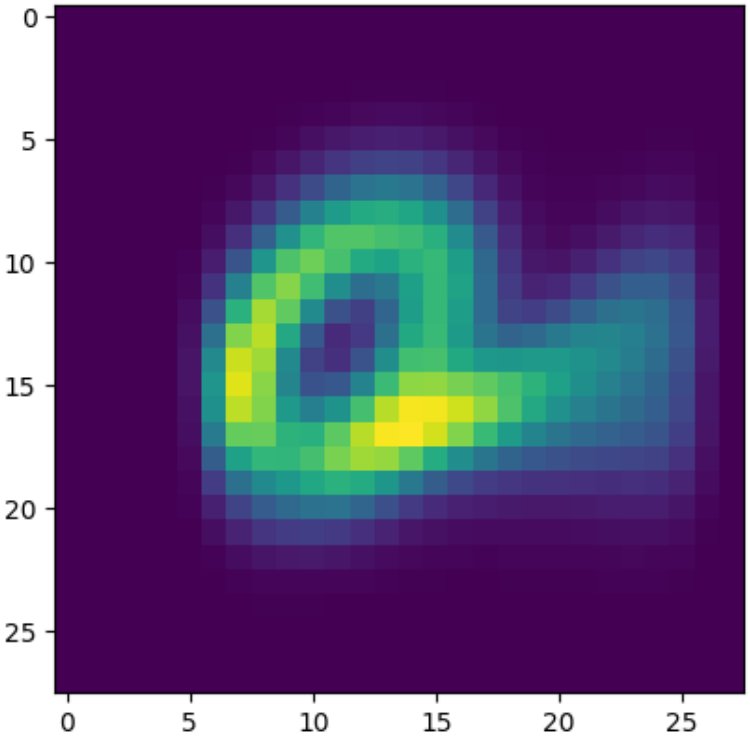
Label-7



Label-8



Label-9



Covariance

As said in the section mean, the covariance matrix for each digit Is stored in the NumPy matrix cov_matrix and one access it by cov_matrix[i] for each digit i, for printing the covariance on the screen please uncomment the lines 43,44,45.

Eigen values and Eigen vectors

The function linalg.eigh from the SciPy library is used for calculating the eigen values and eigen vectors for each digit, the matrices eigen_values[i] and eigen_vectors[i] store the eigen values and eigen vectors respectively. The principal mode variation determined by the eigen value lamda1 is printed on the screen for each digit as.

```
printing the lamda1 of label 0 567161.0850465144
printing the lamda1 of label 1 511989.5256693684
printing the lamda1 of label 2 396868.51455854
printing the lamda1 of label 3 364424.8967841626
printing the lamda1 of label 4 317142.6201851795
printing the lamda1 of label 5 517409.1325334063
printing the lamda1 of label 6 485385.85708529485
printing the lamda1 of label 7 391852.9003784168
printing the lamda1 of label 8 366328.2645461494
printing the lamda1 of label 9 403292.0400330625
$
```

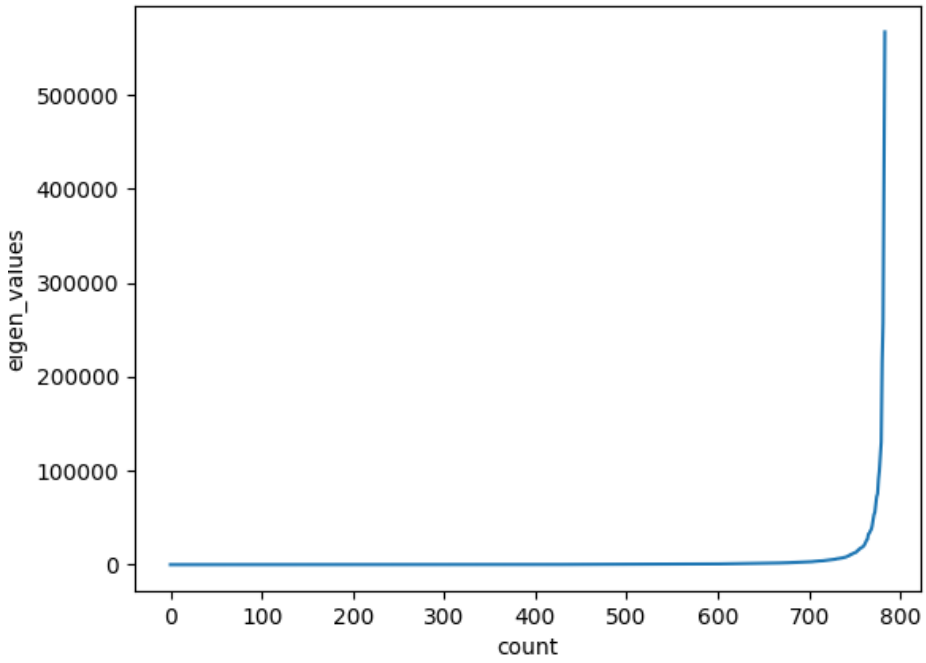
For printing the eigen vectors please uncomment the lines 54 and 55, those are the lines that print the eigen vectors(this lines are commented because they occupy large space in the terminal and is hard to read other data)

Principle and significant modes

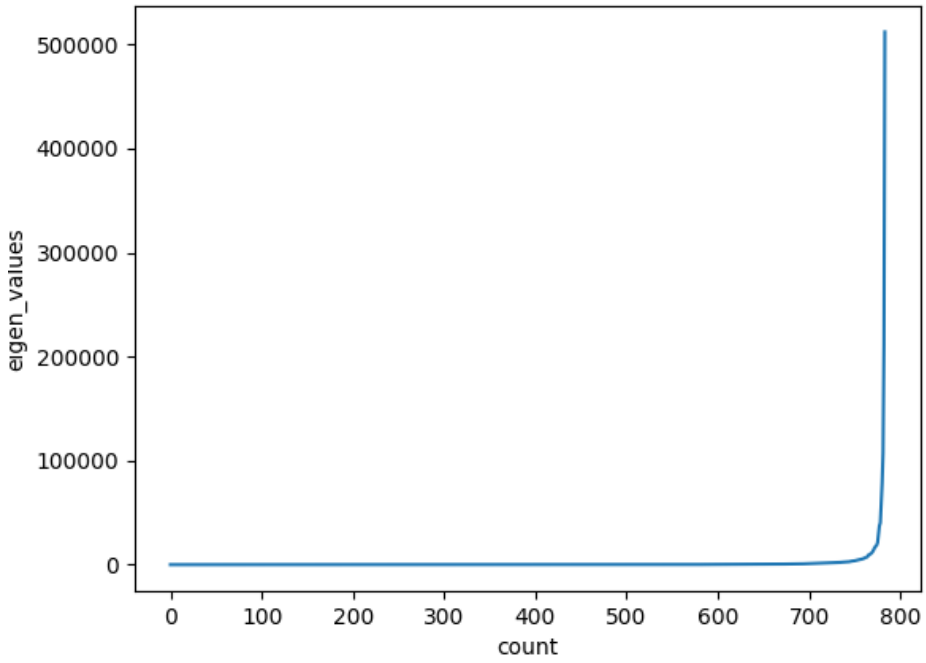
On running the python file q4.py, plots of eigen values for each digit is created and saved in the results directory with the name format “eigenvalues_label{i}.png” , the graphs are also given below for completeness

From the graph we can conclude that only few (2 or 3) in the 784 eigen values are significant/ principal mode variation. Which is far less compared to the 784 eigen values. This behaviour of the graph is excepted, because shape of the of the digits does not change much compared to the orientation of the digits. Thus there will be few where perpendicular distance is minimum (or spread on the eigen vector is maximum), this behaviour is also reflected in the below graphs.

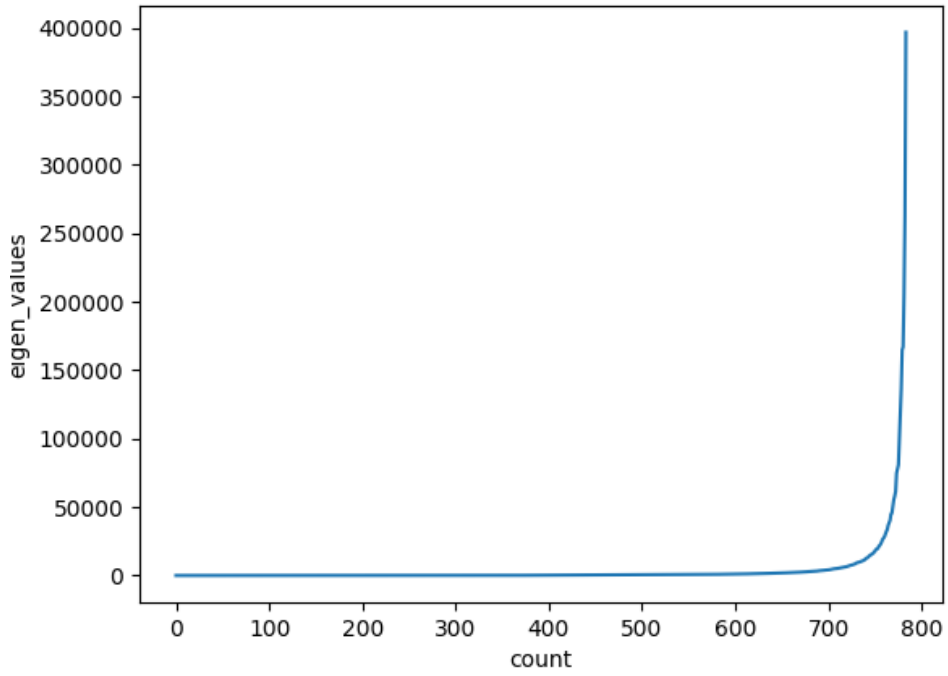
Label-0



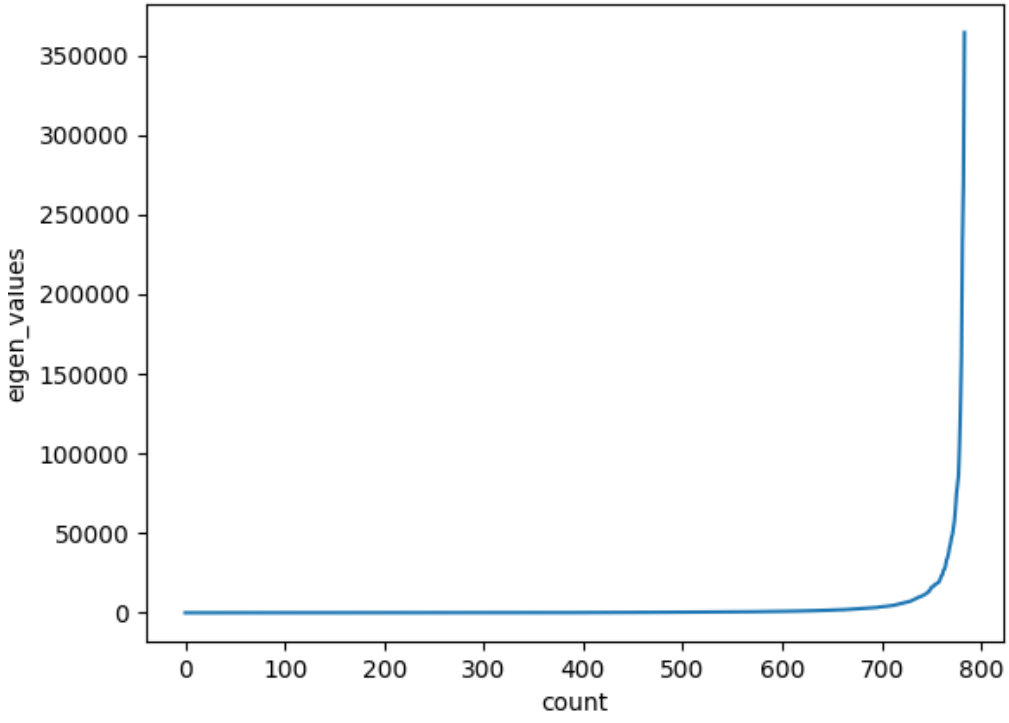
Label-1



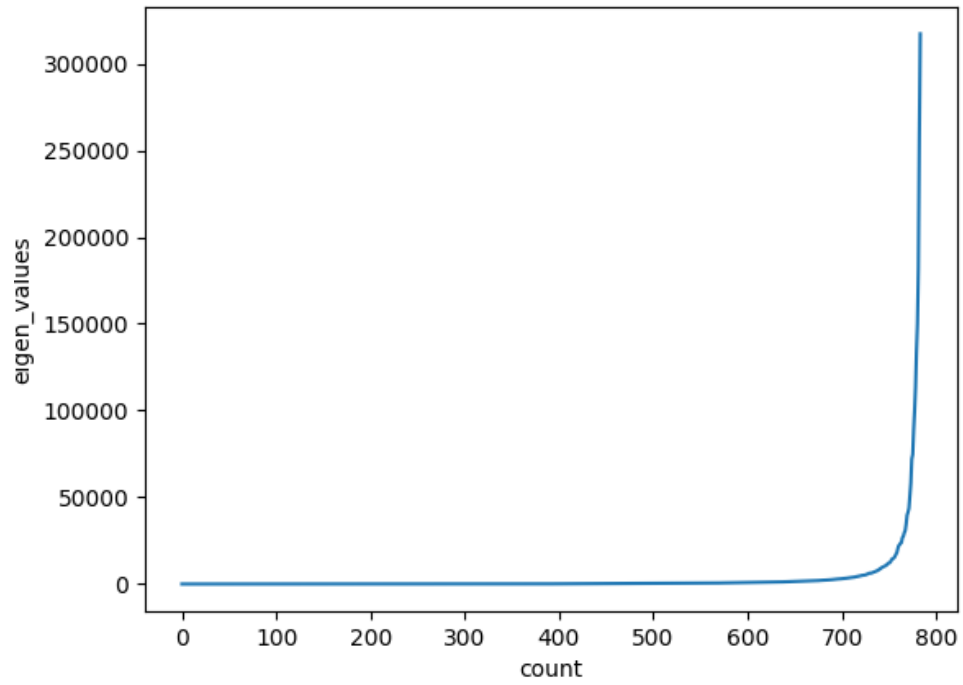
Label-2



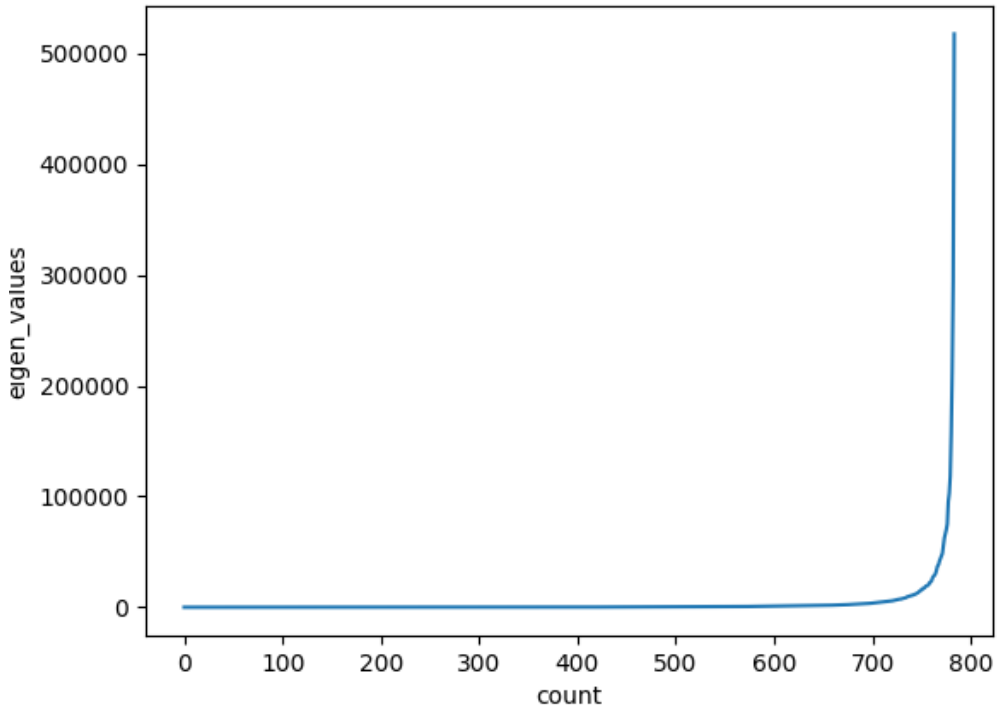
Label-3



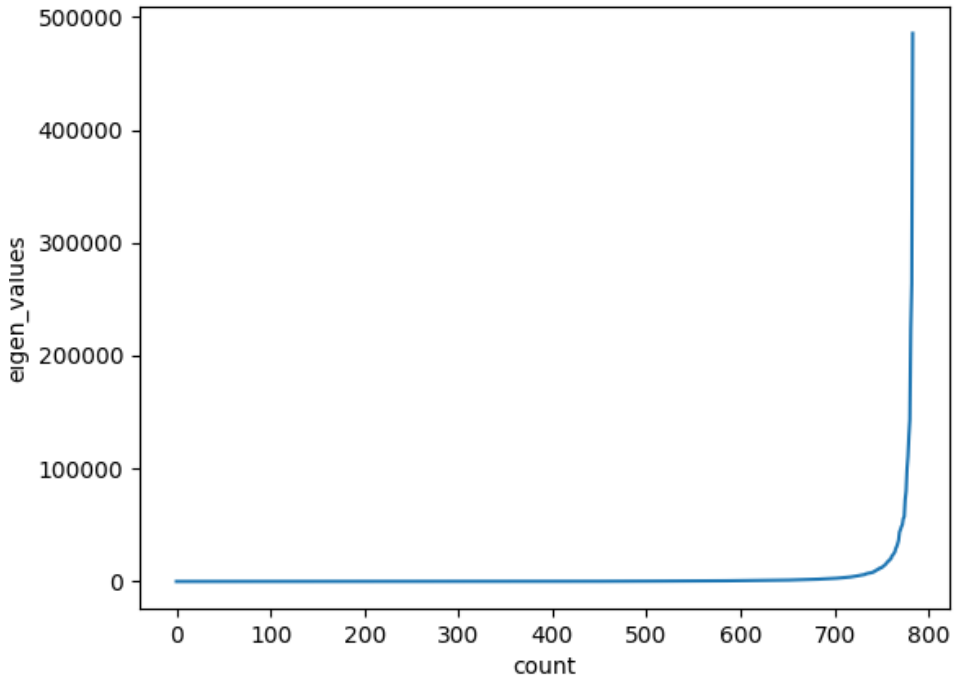
Label-4



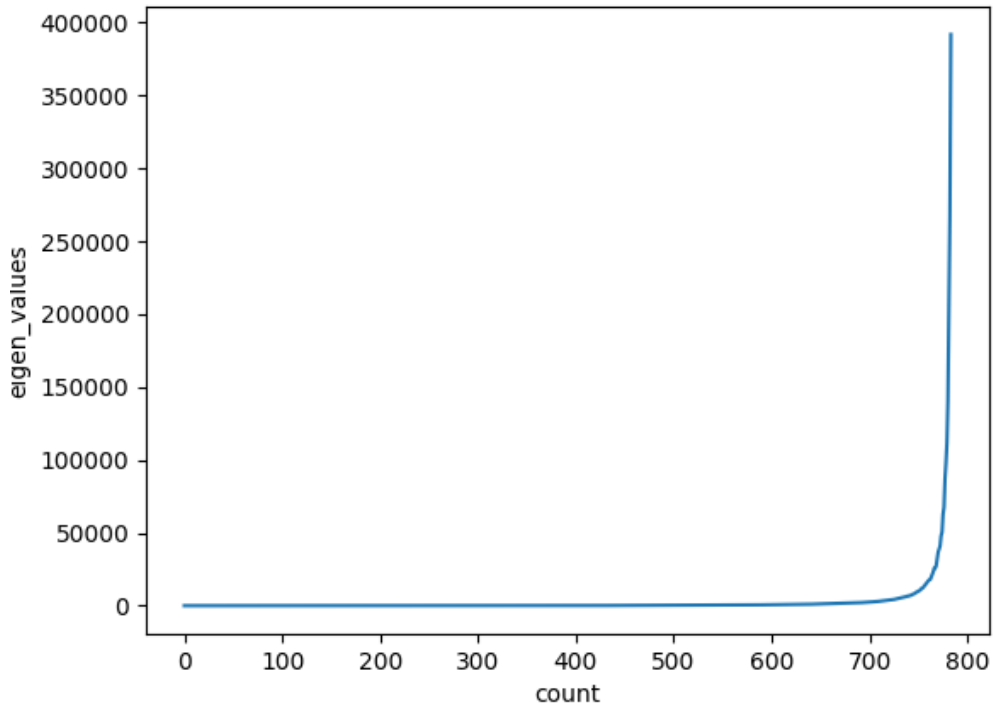
Label-5



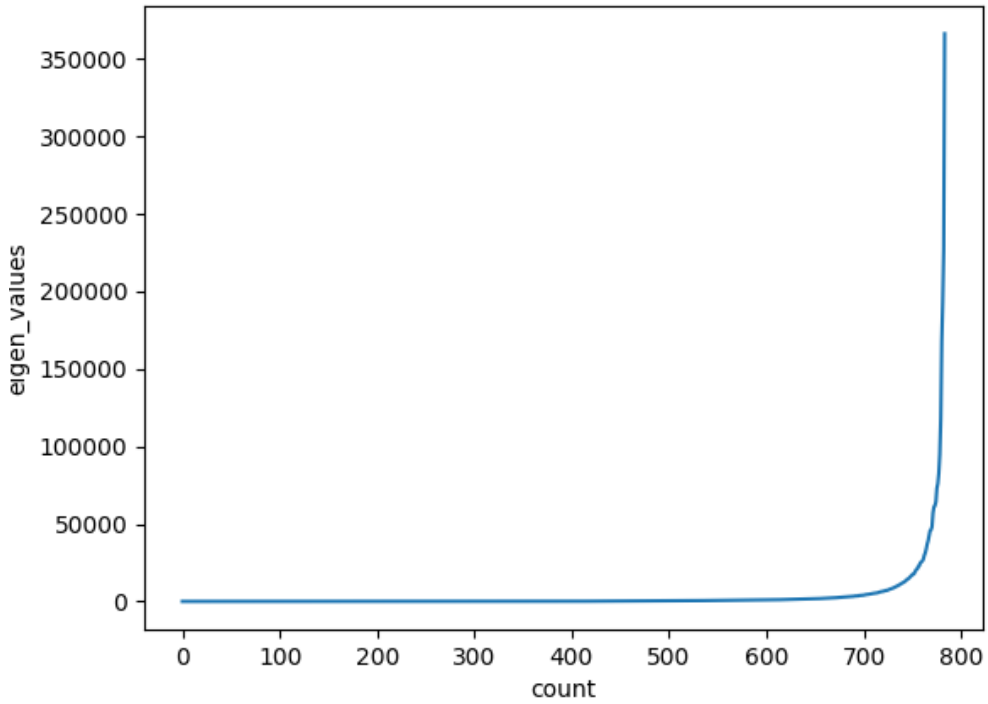
Label-6



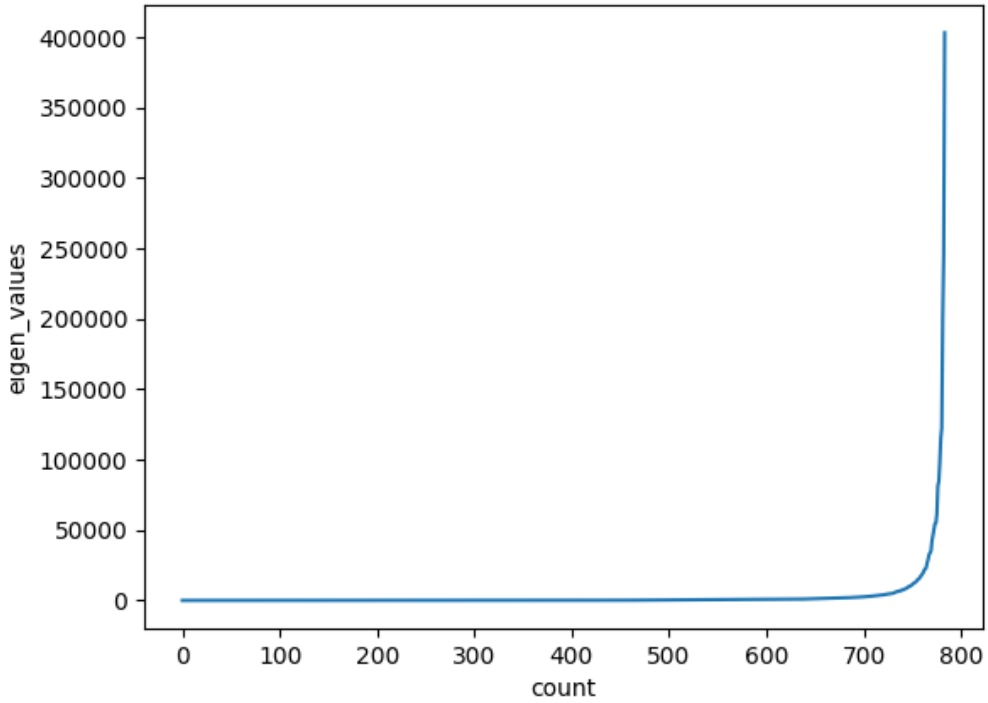
Label-7



Label-8



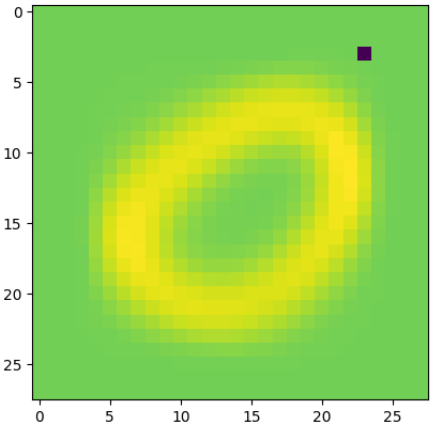
Label-9



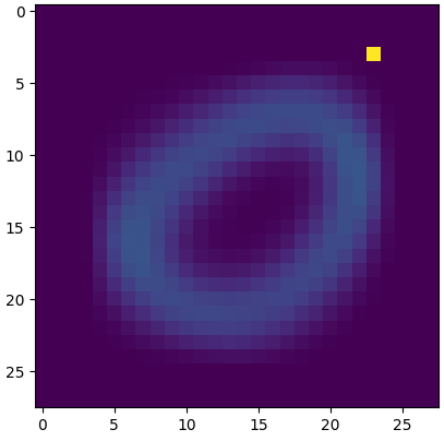
Three images

On running the q4.py python script , two separate files for each digit in the name format “minus_lamda1_label{i}.png” and “plus_lamda1_label{i}.png” are created and saved to the results directory,

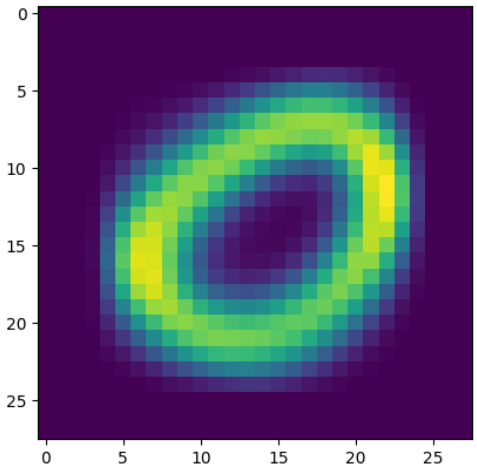
Label-0



Minus_lamda1



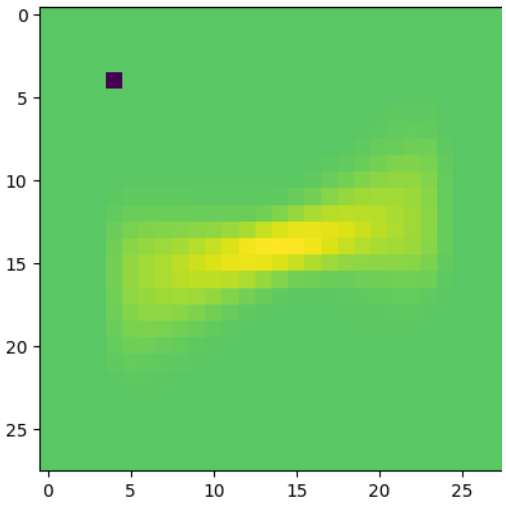
plus_lamda1



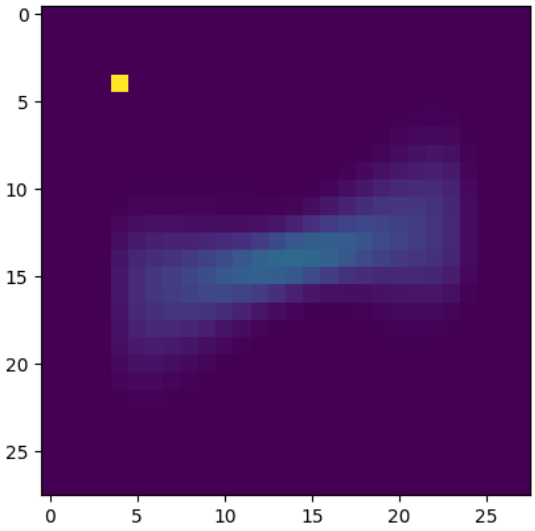
Actual mean

the colours have changed for + and -, orientation of the images changes a bit for + and -

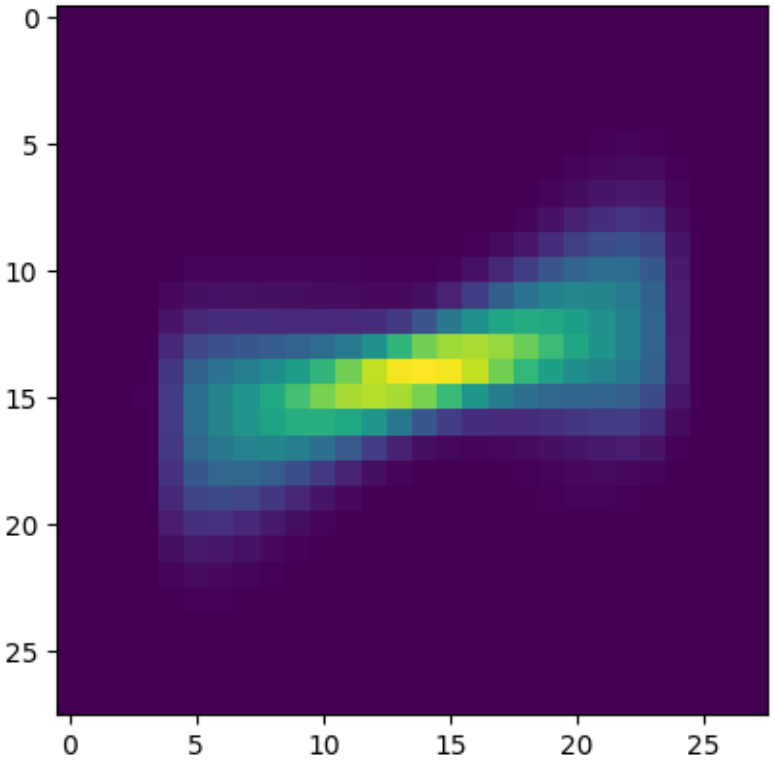
Label-1



Minus_lambd1

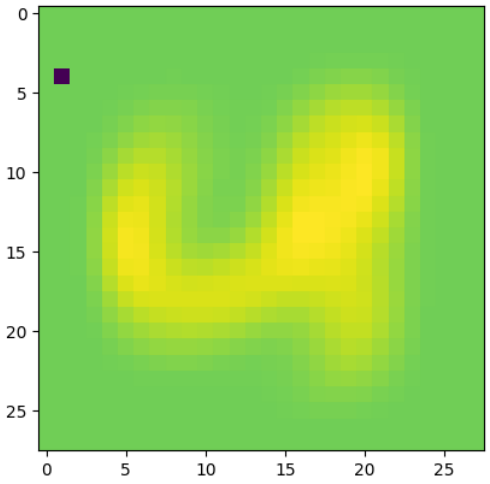


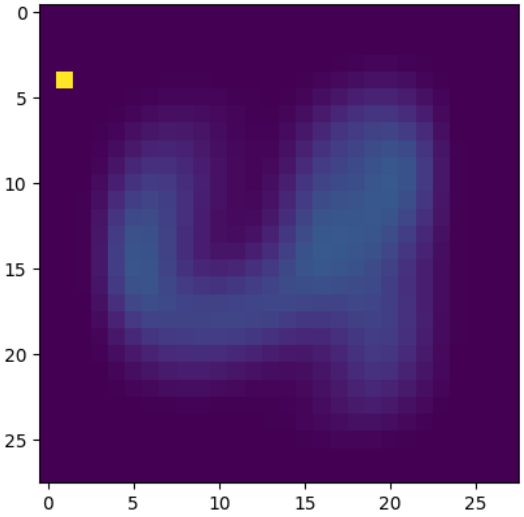
plus_lambd1

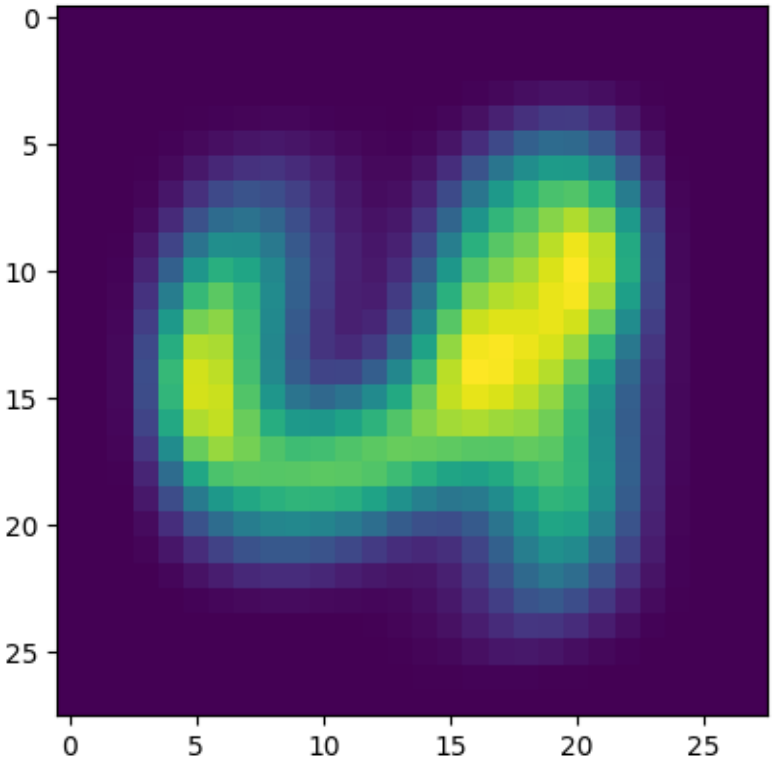


Actual mean

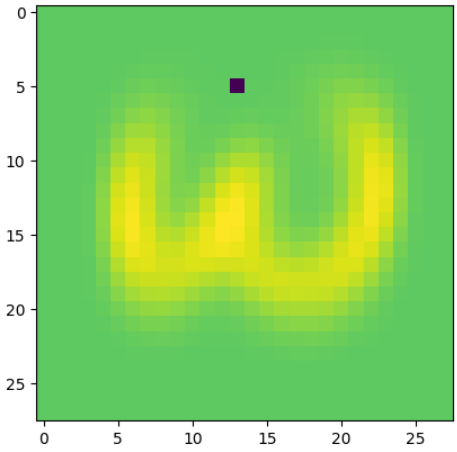
Label-2

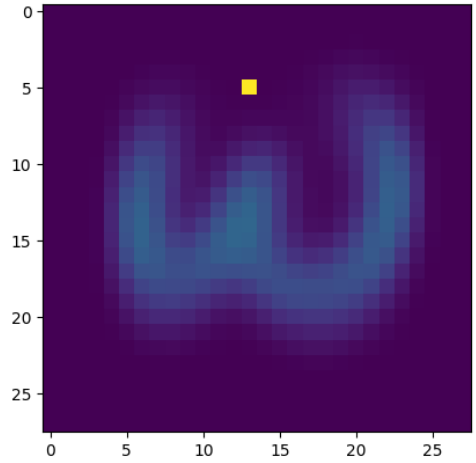

0
5
10
15
20
25
0
5
10
15
20
25

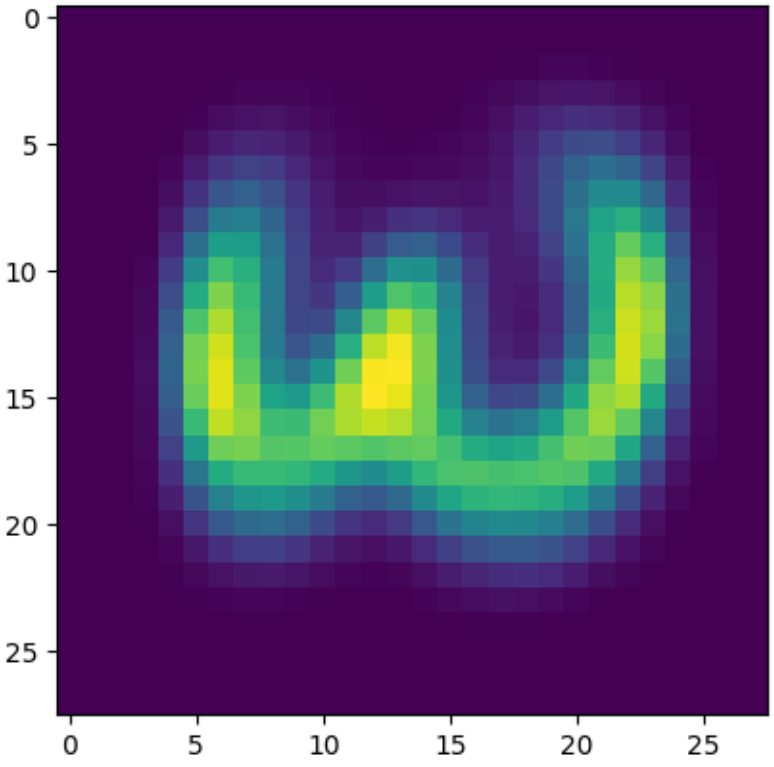

0
5
10
15
20
25
0
5
10
15
20
25


0
5
10
15
20
25
0
5
10
15
20
25

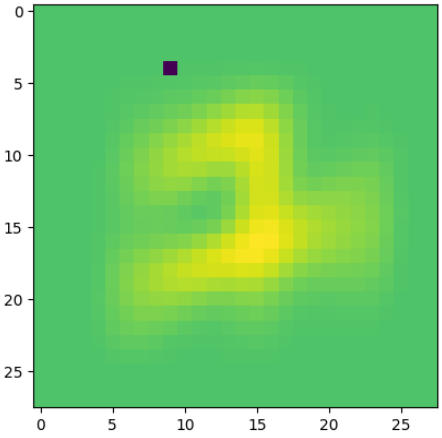
Label-3

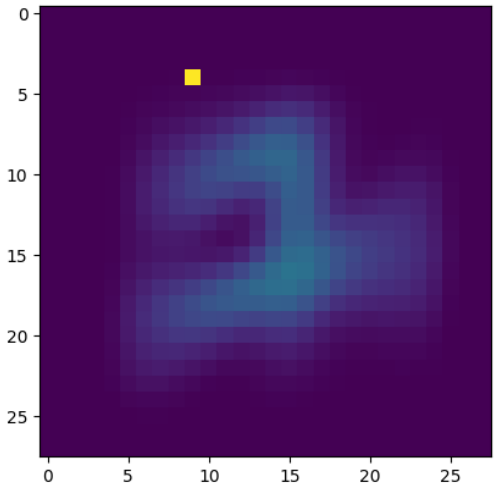

0
5
10
15
20
25
0
5
10
15
20
25

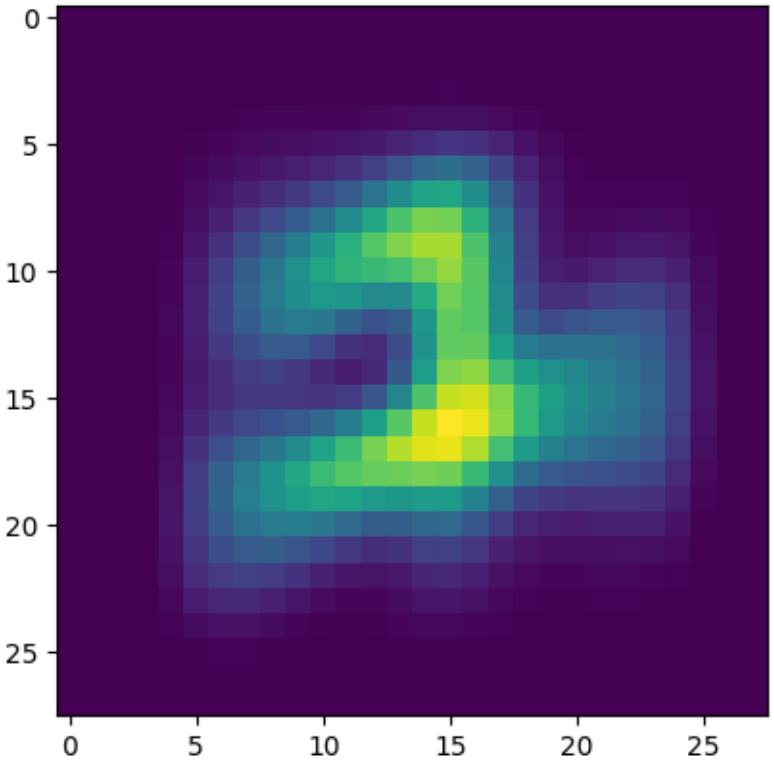

0
5
10
15
20
25
0
5
10
15
20
25


0
5
10
15
20
25
0
5
10
15
20
25

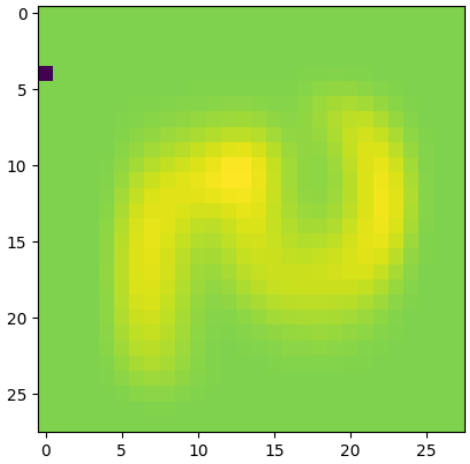
Label-4

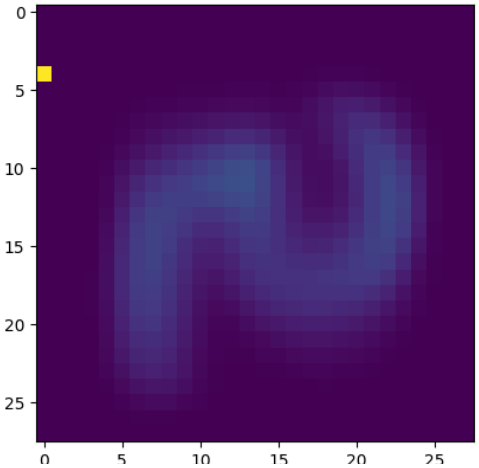

0
5
10
15
20
25
0
5
10
15
20
25

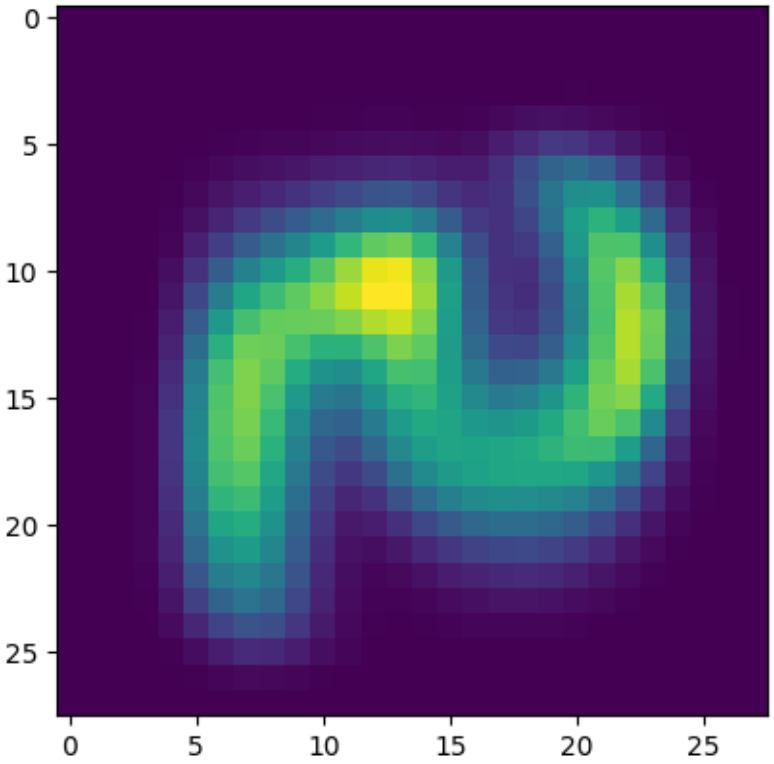

0
5
10
15
20
25
0
5
10
15
20
25


0
5
10
15
20
25
0
5
10
15
20
25

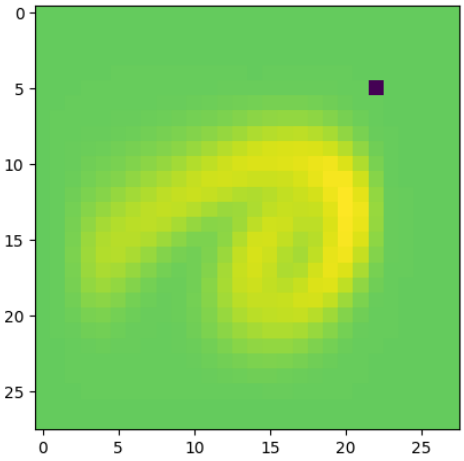
Label-5

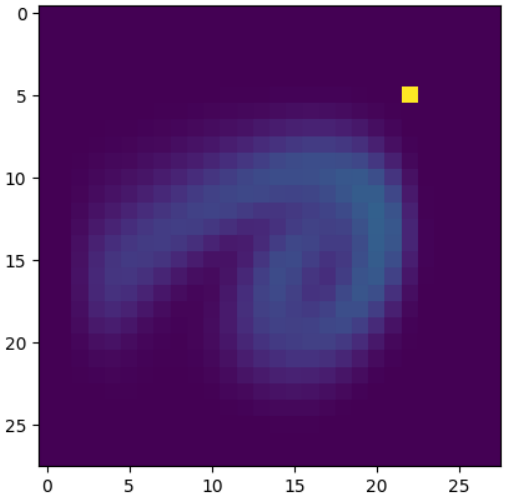

0
5
10
15
20
25
0
5
10
15
20
25

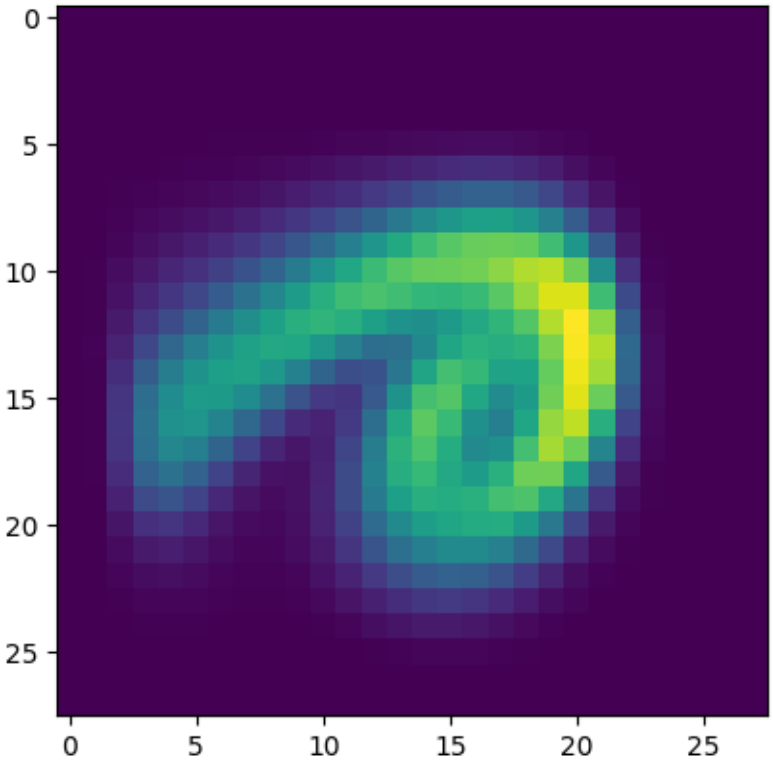

0
5
10
15
20
25
0
5
10
15
20
25


0
5
10
15
20
25
0
5
10
15
20
25

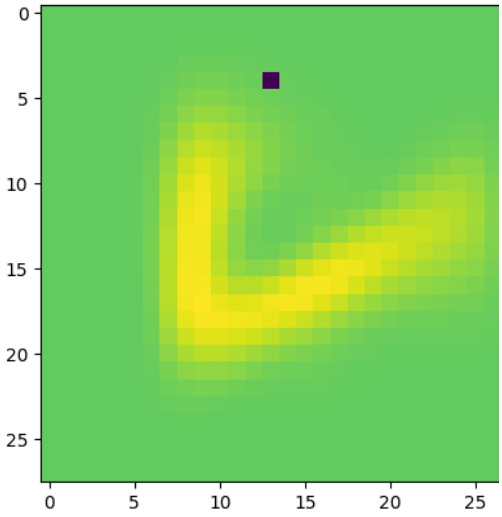
Label-6

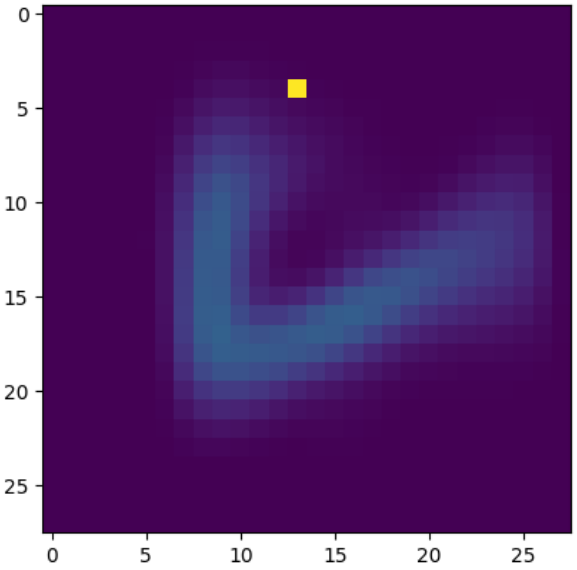

0
5
10
15
20
25
0
5
10
15
20
25

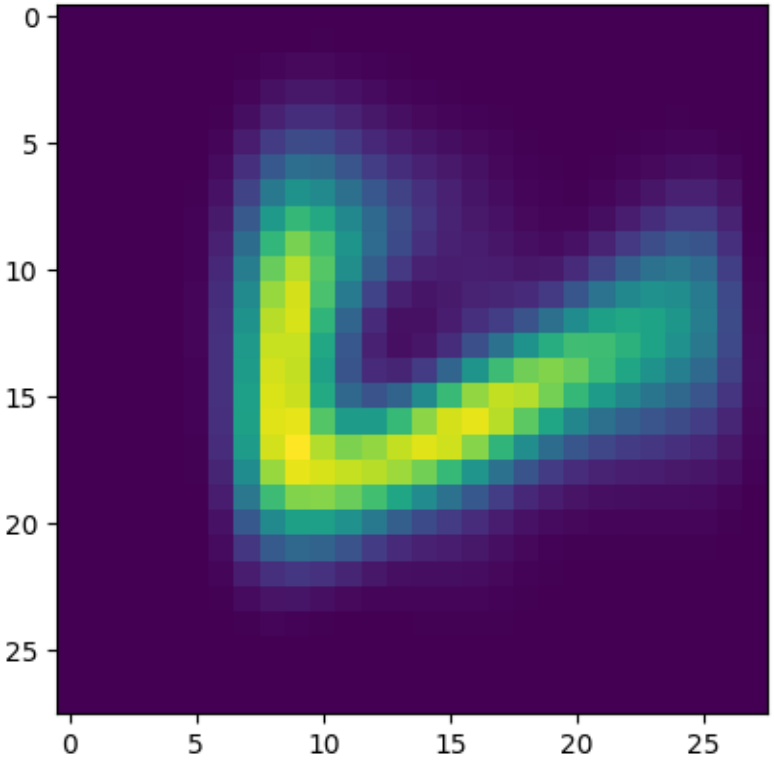

0
5
10
15
20
25
0
5
10
15
20
25


0
5
10
15
20
25
0
5
10
15
20
25

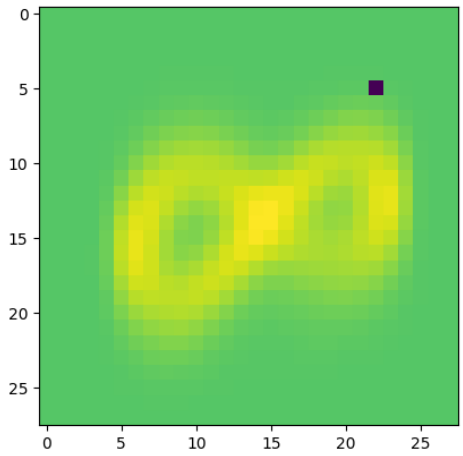
Label-7

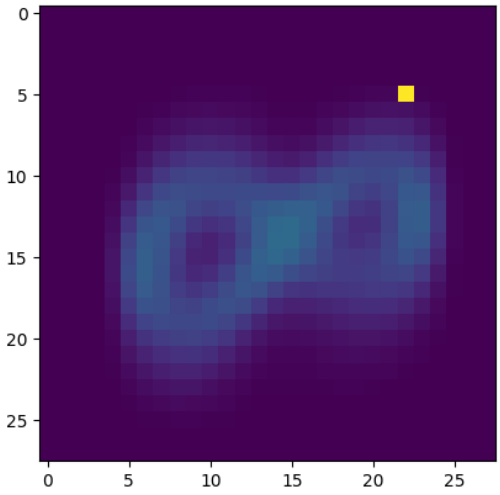

0
5
10
15
20
25
0
5
10
15
20
25

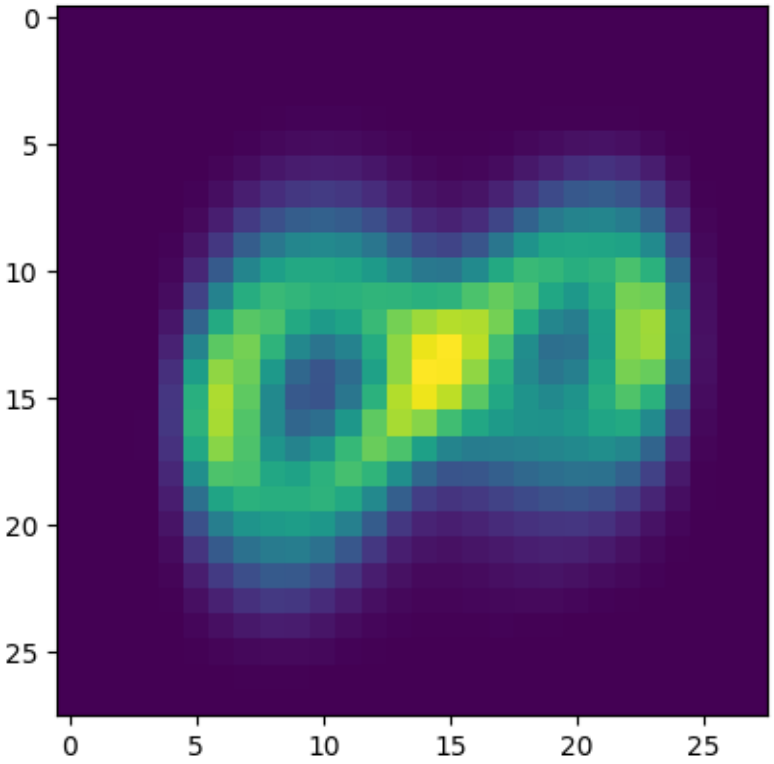

0
5
10
15
20
25
0
5
10
15
20
25


0
5
10
15
20
25
0
5
10
15
20
25

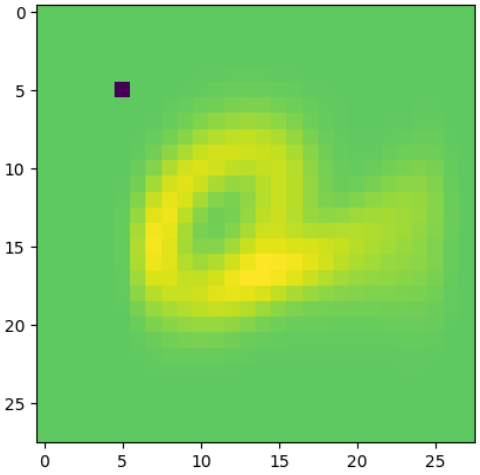
Label-8

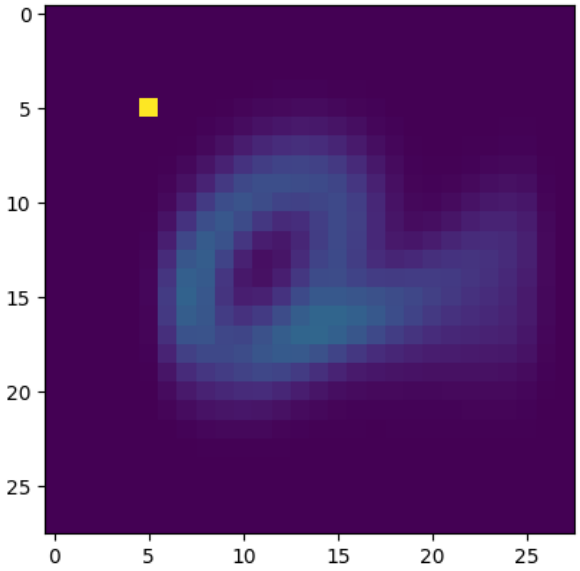

0
5
10
15
20
25
0
5
10
15
20
25

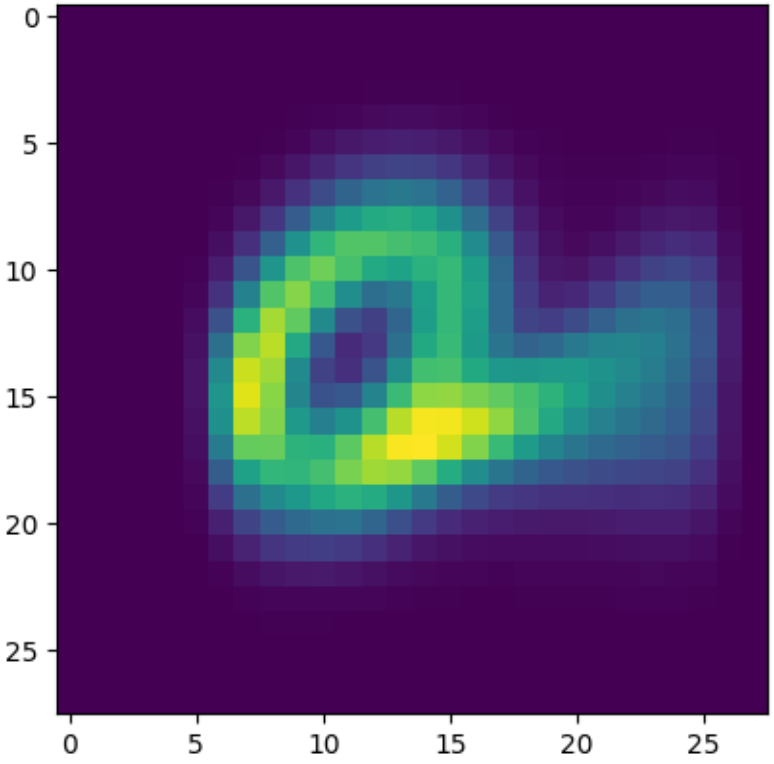

0
5
10
15
20
25
0
5
10
15
20
25


0
5
10
15
20
25
0
5
10
15
20
25

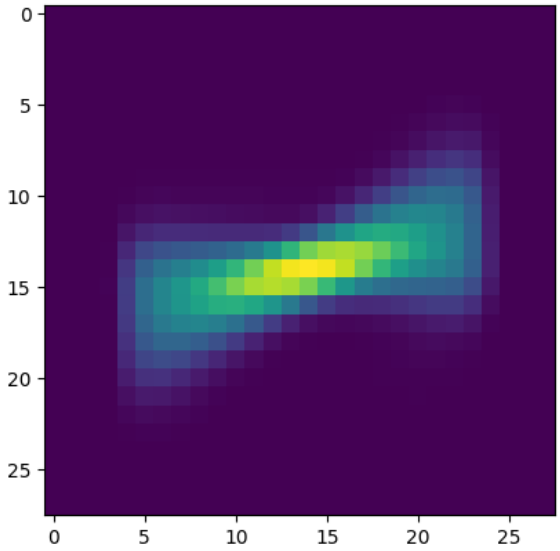
Label-9


0
5
10
15
20
25
0
5
10
15
20
25


0
5
10
15
20
25
0
5
10
15
20
25


0
5
10
15
20
25
0
5
10
15
20
25

From the plot of digit 1 , one conclude that the people generally write the digit 1 as a straight line with different orientation of the straight line.



Instructions to run the code

Please move to the Q4 directory and

- python3 ./code/q4.py will run the q4.py and plot a total of 30 different plots with the naming formats “mean_label{i}.png” , “minus_lambda1_label{i}.png” and “plus_lambda1_label{i}.png”,
- The above command takes a time of approximately 2 mins, as the data we are dealing is large, nevertheless the program generates all the plots and print the required results.