# VishwaCTF

CHALLENGE NAME : HOW CNN'S SEE

DEV : NAZIYA MAHIMKAR

CATEGORY : MISCELLANEOUS

LEVEL : MEDIUM




**2024**

DESCRIPTION

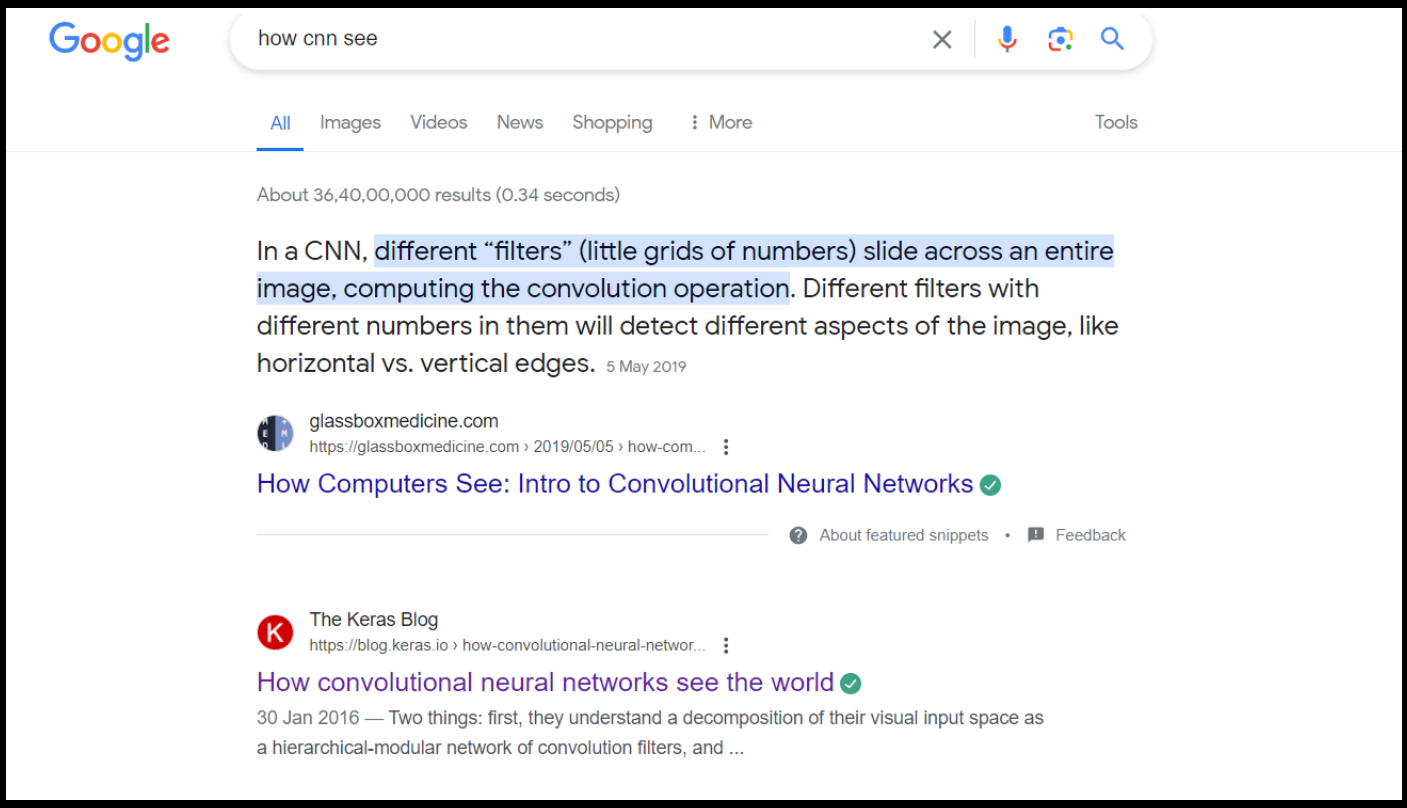
Have a look inside my number recognizer black box but be careful better not mess with the numbers.

Flag format: VishwaCTF{}

**SOLUTION:**

**Hint 1**

If you look carefully the name of the challenge itself a hint. Googling it we come across this article "How convolutional neural networks see the world" which talks about the filters in a Convolutional Neural Network and plotting them.



Hint 2

We are provided with a images folders which consists of 4 images. A neural Network cannot be trained with just 4 images further the description itself says don't mess the numbers hence no training required.


If you look into the names of the images the represent something it is:

2 , 6 , 10

They are a hint to the layers that contain flag. You will understand further if you will look into the architecture of the flag.

```
model.summary()
```

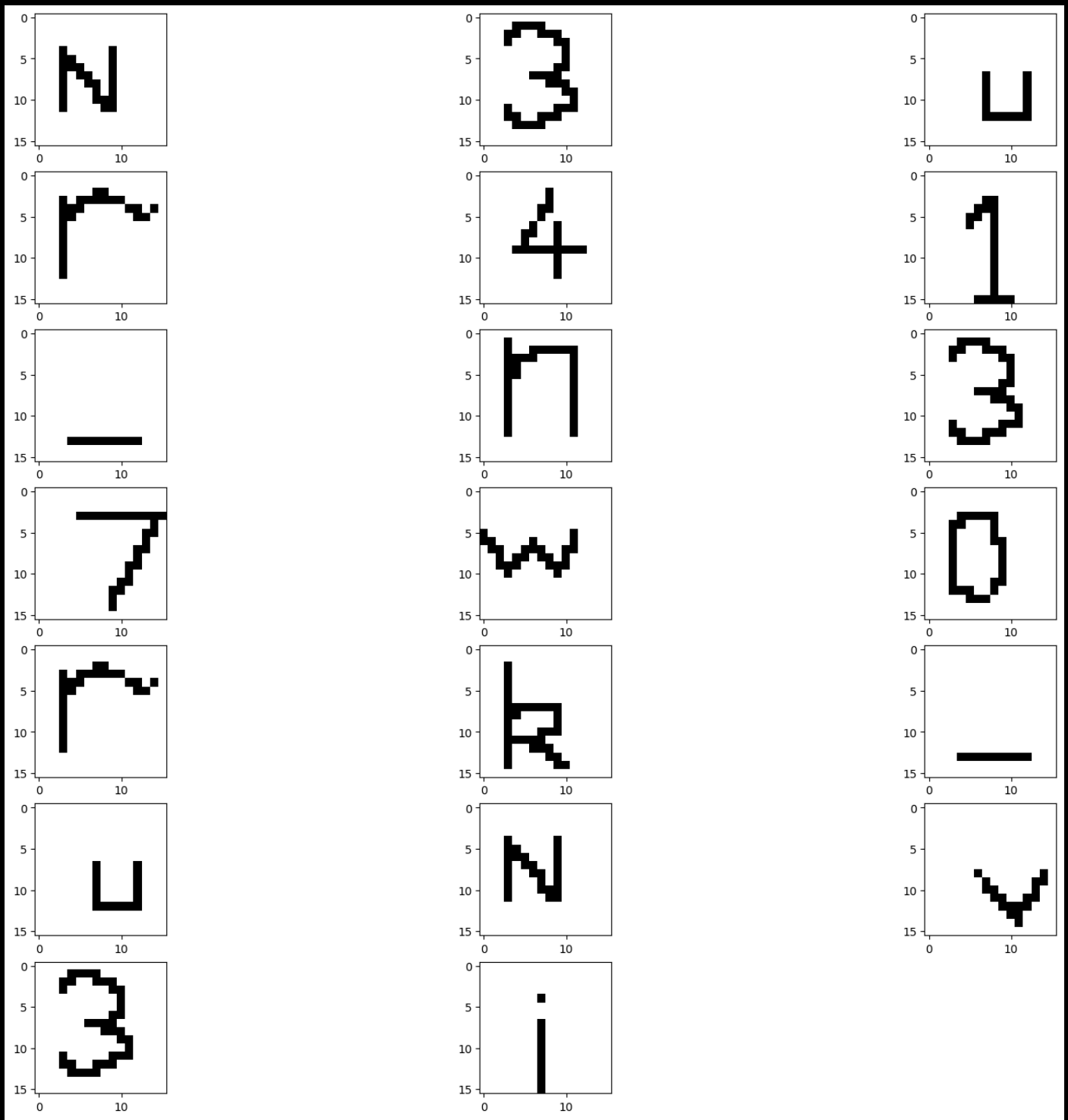
```
Model: "model_28"
_________________________________________________________________
 Layer (type)                Output Shape              Param #   
=================================================================
0 input_37 (InputLayer)       [(None, 28, 28, 1)]       0         
1 resizing_32 (Resizing)      (None, 103, 103, 1)       0         
2 conv2d_184 (Conv2D)         (None, 88, 88, 20)        5140      
3 max_pooling2d_180 (MaxPooling2  (None, 88, 88, 20)    0         
  D)                                                             
4 conv2d_185 (Conv2D)         (None, 73, 73, 1)         5121      
5 max_pooling2d_181 (MaxPooling2  (None, 73, 73, 1)     0         
  D)                                                             
6 conv2d_186 (Conv2D)         (None, 58, 58, 20)        5140      
7 max_pooling2d_182 (MaxPooling2  (None, 58, 58, 20)    0         
  D)                                                             
8 conv2d_187 (Conv2D)         (None, 43, 43, 1)         5121      
9 max_pooling2d_183 (MaxPooling2  (None, 43, 43, 1)     0         
  D)                                                             
10 conv2d_188 (Conv2D)        (None, 28, 28, 20)        5140      
```

Now we just have to plot the filters in the above layer. Using the below code;

```
# retrieve weights from the second hidden layer
import matplotlib.pyplot as pyplot
filters , bias = model.layers[2].get_weights()
f_min, f_max = filters.min(), filters.max()
filters = (filters - f_min) / (f_max - f_min)
n_filters =20
ix=1
fig = pyplot.figure(figsize=(20,50))
for i in range(n_filters):
    # get the filters
    f = filters[:,:,:,i]
    for j in range(1):
        # subplot for 6 filters and 3 channels
        pyplot.subplot(n_filters,3,ix)
        pyplot.imshow(f[:,:,j] ,cmap='gray')
        ix+=1
#plot the filters
pyplot.show()
```

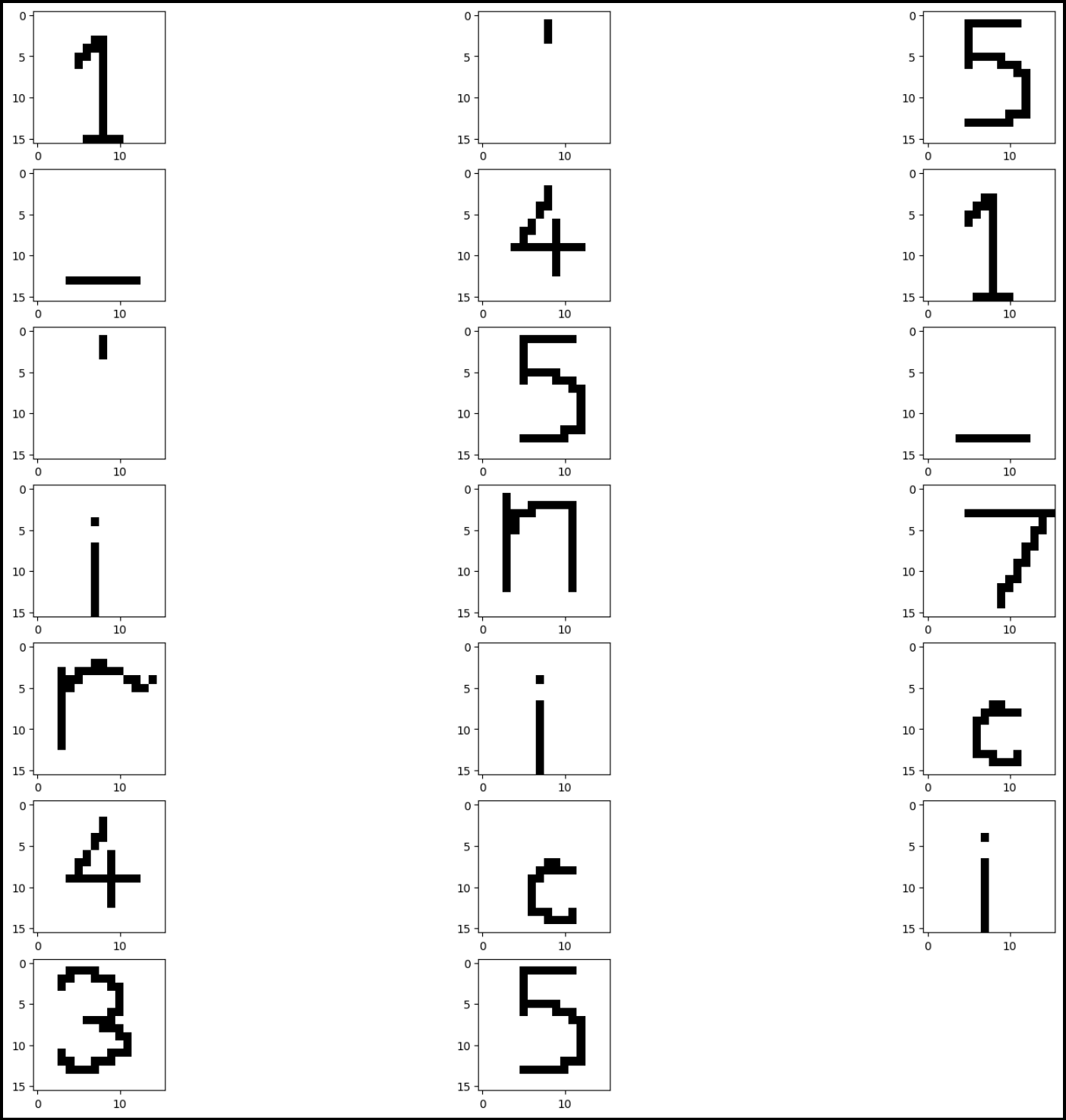
You get the below plot for the filters:

Layer 2

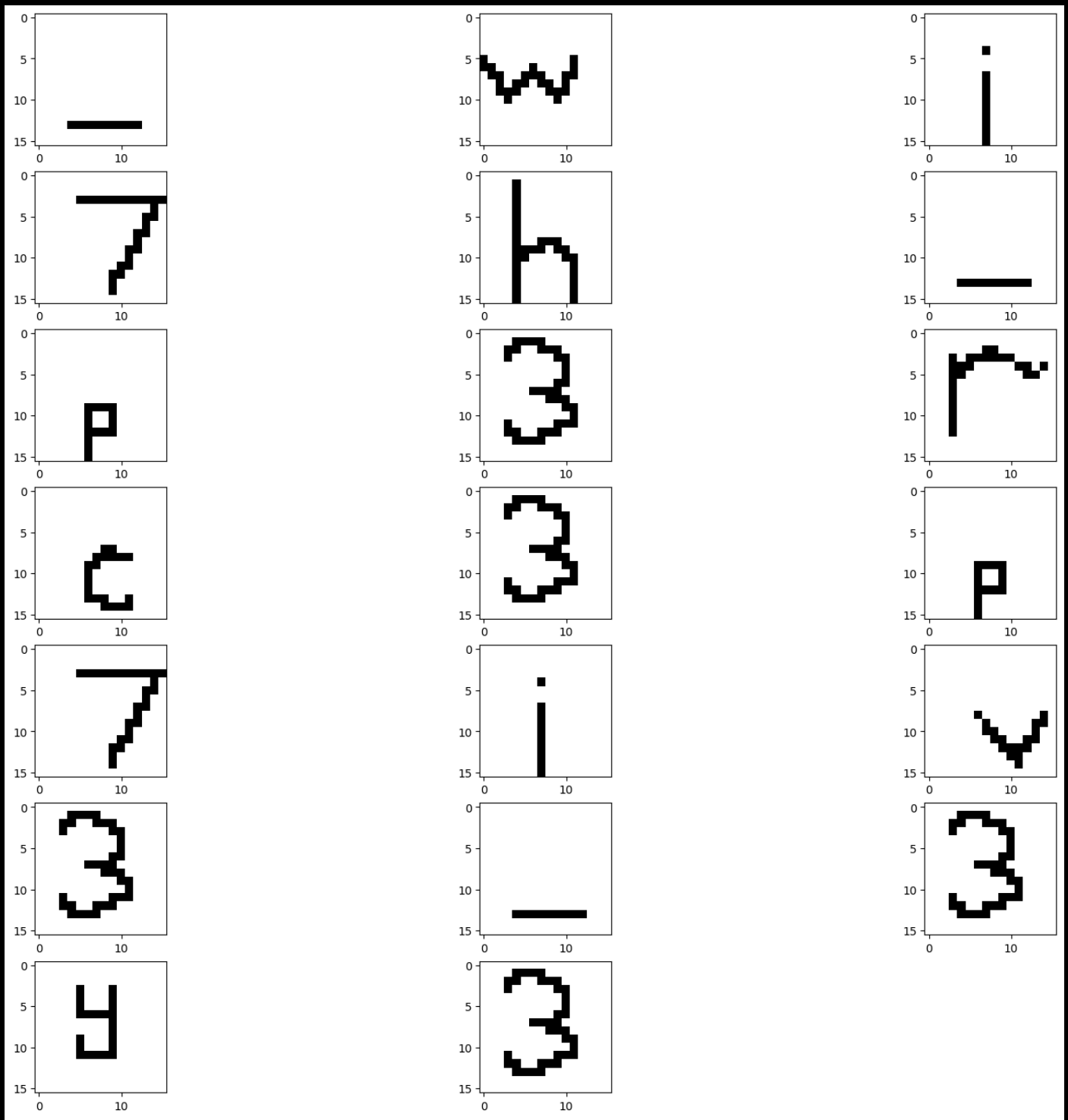


Similarly for layer 6 and 10 are as below

Layer 6



Layer 10



Putting all the plotted characters together you get the flag:

“N3ur41_n37w0rk_uNv3i1'5_41'5_in7ric4ci35_wi7h_p3rc3p7iv3_3y3”