cnn_likecopy

January 6, 2020

Basic concept for feature extraction, used in Convolutional Neural Netwrok

[1]:

```
import skimage.data
from PIL import Image
from matplotlib import pyplot as plt
import numpy as np
import scipy.signal as ss
```

Create image with numpy

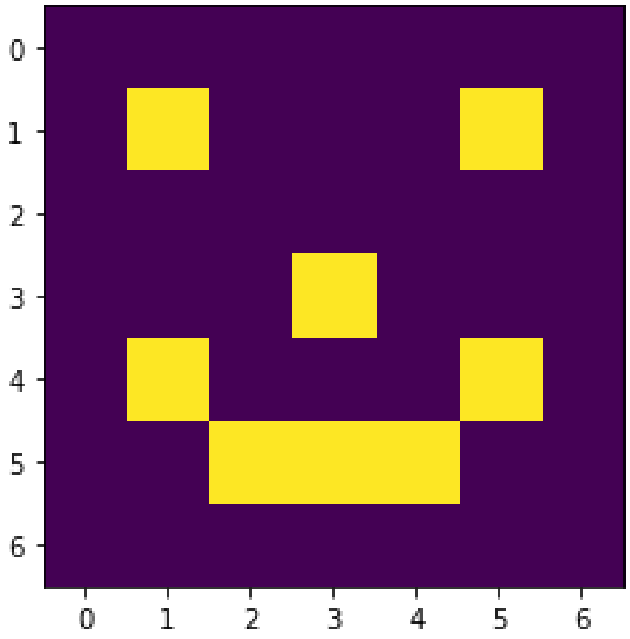
[2]:

```
img = np.array([[0,0,0,0,0,0,0],
                [0,1,0,0,0,1,0],
                [0,0,0,0,0,0,0],
                [0,0,0,1,0,0,0],
                [0,1,0,0,0,1,0],
                [0,0,1,1,1,0,0],
                [0,0,0,0,0,0,0]
               ])
print(img)
```

```
[[0 0 0 0 0 0 0]
 [0 1 0 0 0 1 0]
 [0 0 0 0 0 0 0]
 [0 0 0 1 0 0 0]
 [0 1 0 0 0 1 0]
 [0 0 1 1 1 0 0]
 [0 0 0 0 0 0 0]]
```

[3]:

```
plt.imshow(img,interpolation='nearest')
plt.show()
```



[4]:
```
img.shape
```

[4]:
```
(7, 7)
```

Perform Convolution Operation with kernel size–>3*3 and stride=1

If used convolve2d—>Note that the kernal has to be reversed. Otherwise usesignal.correlate2d.

[5]:
```
def apply_kernel(img,kernel):
    return(ss.correlate2d(img,kernel,mode='valid'))
```

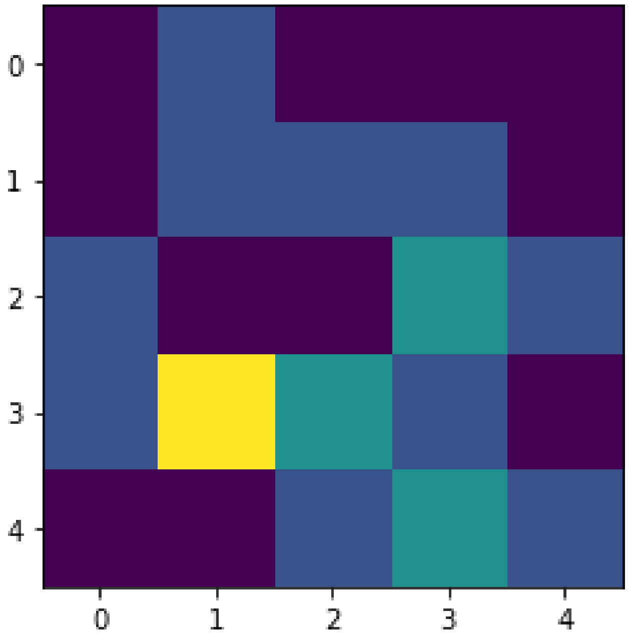
[6]:
```
kernel = np.array([[0,0,1],[1,0,0],[0,1,1]])
img_convolve = apply_kernel(img,kernel)
print(img_convolve)
```

```
[[0 1 0 0 0]
 [0 1 1 1 0]
 [1 0 0 2 1]
 [1 4 2 1 0]
 [0 0 1 2 1]]
```

[7]:
```
print(img_convolve.shape)
```

```
(5, 5)
```

[8]:
```
plt.imshow(img_convolve,interpolation='nearest')
plt.show()
```



Perfrom pooling operation(in this case maxpool) with size—>2*2 and stride=2

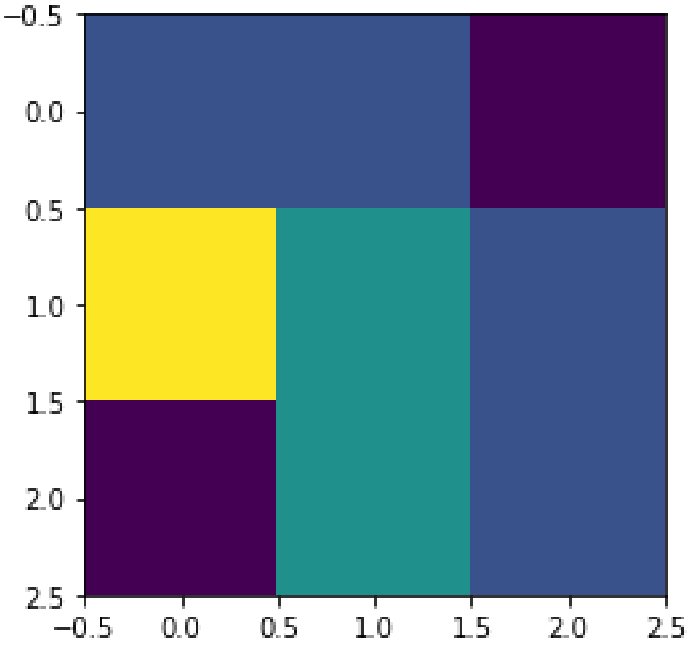
[9]:
```
pool_image = skimage.measure.block_reduce(img_convolve,(2,2),np.max)
print(pool_image)
```

```
[[1 1 0]
 [4 2 1]
 [0 2 1]]
```

[10]:
```
print(pool_image.shape)
```

```
(3, 3)
```

[11]:
```
plt.imshow(pool_image,interpolation='nearest')
plt.show()
```



Make a single vector for futher processing

[12]:
```
flattened_image = pool_image.reshape(-1,1)
print(flattened_image)
```

```
[[1]
 [1]
 [0]
 [4]
 [2]
 [1]
 [0]
 [2]
 [1]]
```

[13]:
```
flattened_image.shape
```

[13]:
```
(9, 1)
```

Pass Vector to artificial neural network..... Thanks for Viewing this notebook. Hope this provide a basic concept for feature extraction, used in Convolutional Neural Network