https://www.youtube.com/watch?v=eoGJhf4Zsuc

Max Pooling and Convolution from scratch.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

+ Code + Text

Reconnect Editing

Importing Libraries and Image

```
import numpy as np
from scipy import misc
i = misc.face(gray=True)
```

Color image convert into grey scale

Library store in "i"

```
[ ] import matplotlib.pyplot as plt
    plt.grid(False)
    plt.gray()
    plt.axis('off')
    plt.imshow(i)
    plt.show()
```

```
[ ] i_transformed = np.copy(i)
    size_x = i_transformed.shape[0]
    size_y = i_transformed.shape[1]
```

Apply Filter On image

Importing Libraries and Image

+ Code + Text

RAM Disk

[2]

```
i_transformed = np.copy(i)
size_x = i_transformed.shape[0]
size_y = i_transformed.shape[1]
```

+ Code + Text

Apply Filter On image

[ ] 3 cells hidden

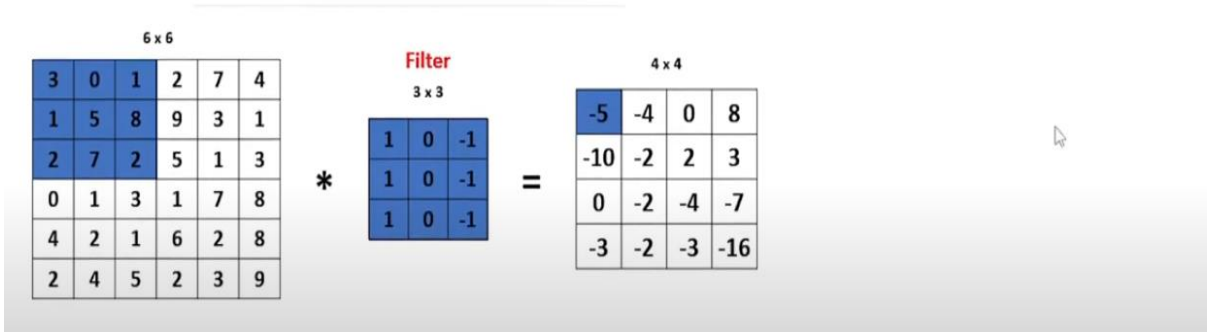
Max Pooling

Apply Filter On image

```
#Experiment with different values for fun effects.
#filter = [ [0, 1, 0], [1, -4, 1], [0, 1, 0]]

# A couple more filters to try for fun!
#filter = [ [-1, -2, -1], [0, 0, 0], [1, 2, 1]]
#filter = [ [-1, 0, 1], [-2, 0, 2], [-1, 0, 1]]

filter = [[1,0,-1],
          [1,0,-1],
          [1,0,-1]]
```



Details in vide https://www.youtube.com/watch?v=eoGJhf4Zsuc on 08:20

Consider as row consider as column row column

```
for x in range(1,size_x-1):
  for y in range(1,size_y-1):
      convolution = 0.0
      convolution = convolution + (i[x-1, y-1] * filter[0][0])
      convolution = convolution + (i[x-1, y] * filter[0][1])
      convolution = convolution + (i[x-1, y+1] * filter[0][2])
      convolution = convolution + (i[x, y-1] * filter[1][0])
      convolution = convolution + (i[x, y] * filter[1][1])
      convolution = convolution + (i[x, y+1] * filter[1][2])
      convolution = convolution + (i[x+1, y-1] * filter[2][0])
      convolution = convolution + (i[x+1, y] * filter[2][1])
      convolution = convolution + (i[x+1, y+1] * filter[2][2])
      if(convolution<0):
        convolution=0
      if(convolution>255):
        convolution=255
      i_transformed[x, y] = convolution
```

```
[ ] # Plot the image. Note the size of the axes -- they are 512 by 512
    plt.gray()
    plt.grid(False)
    plt.imshow(i_transformed)
    #plt.axis('off')
    plt.show()
```

Max Pooling

```
[ ] new_x = int(size_x/2)
    new_y = int(size_y/2)
    newImage = np.zeros((new_x, new_y))
    for x in range(0, size_x, 2):
      for y in range(0, size_y, 2):
        pixels = []
        pixels.append(i_transformed[x, y])
        pixels.append(i_transformed[x+1, y])
        pixels.append(i_transformed[x, y+1])
        pixels.append(i_transformed[x+1, y+1])
        newImage[int(x/2),int(y/2)] = max(pixels)
```

np= numpy

```
new_x = int(size_x/2)
new_y = int(size_y/2)
newImage = np.zeros((new_x, new_y))
for x in range(0, size_x, 2):
  for y in range(0, size_y, 2):
    pixels = []
    pixels.append(i_transformed[x, y])
    pixels.append(i_transformed[x+1, y])
    pixels.append(i_transformed[x, y+1])
    pixels.append(i_transformed[x+1, y+1])
    newImage[int(x/2),int(y/2)] = max(pixels)

# Plot the image. Note the size of the axes -- now 256 pixels instead of 512
plt.gray()
plt.grid(False)
plt.imshow(newImage)
#plt.axis('off')
plt.show()
```

Situration range 0 to 255
If the value is more than 255 than the color is white
Less than 0 than the color is black