Project

Creating a Multi-Image Window

We will trying to create a Module which will help us to merge multiple individal images to single canvas. This Module is going to be very helpful as we can use this through out the entire image processing course as we might often be requiring to have multiple images shown simultaneously in separate windows. So this will help us with the hassle

In [1]:

```
from cv2 import cv2
import numpy as np

def combine(img_arr):
    '''
    DOC STRING :\n
    The function takes in image array as a list and displays them as a single image in
 a grid like formation.\n
    The function can take in 1-9 images\n
    The function call the cv2.imshow() function\n
    The user has to impliment the waitKey() and the destroyAllwindows function manually
\n
    '''
    count = len(img_arr)
    print(count)

    if count <= 6:
        line = np.full((400,20,3),0, dtype = 'uint8')
        black = np.zeros((400,400,3),dtype = 'uint8')

        for i in range(count):
            img_arr[i] = cv2.resize(img_arr[i],(400,400))
            if img_arr[i].ndim == 2:
                img_arr[i] = cv2.cvtColor(img_arr[i],cv2.COLOR_GRAY2BGR)
    else:
        line = np.full((300,15,3),0, dtype = 'uint8')
        black = np.zeros((300,300,3),dtype = 'uint8')

        for i in range(count):
            img_arr[i] = cv2.resize(img_arr[i],(300,300))
            if img_arr[i].ndim == 2:
                img_arr[i] = cv2.cvtColor(img_arr[i],cv2.COLOR_GRAY2BGR)


    if count == 1:
        cv2.imshow('combine',img_arr[0])

    elif count == 2:
        img = np.hstack((img_arr[0],line))
        img = np.hstack((img,img_arr[1]))
        cv2.imshow('combine',img)

    elif count == 3:
        img = np.hstack((img_arr[0],line))
        img = np.hstack((img,img_arr[1]))
        img = np.hstack((img,line))
        img = np.hstack((img,img_arr[2]))
        cv2.imshow('combine',img)

    elif count == 4:
        img1 = np.hstack((img_arr[0],line))
        img1 = np.hstack((img1,img_arr[1]))

        img2 = np.hstack((img_arr[2],line))
        img2 = np.hstack((img2,img_arr[3]))

        img  = np.vstack((img1,img2))
        cv2.imshow('combine',img)
    elif count == 5:
        img1 = np.hstack((img_arr[0],line))
        img1 = np.hstack((img1,img_arr[1]))
        img1 = np.hstack((img1,line))
        img1 = np.hstack((img1,img_arr[2]))

        img2 = np.hstack((img_arr[3],line))
        img2 = np.hstack((img2,img_arr[4]))
        img2 = np.hstack((img2,line))
        img2 = np.hstack((img2,black))

        img  = np.vstack((img1,img2))
        cv2.imshow('combine',img)

    elif count == 6:
        img1 = np.hstack((img_arr[0],line))
        img1 = np.hstack((img1,img_arr[1]))
        img1 = np.hstack((img1,line))
        img1 = np.hstack((img1,img_arr[2]))

        img2 = np.hstack((img_arr[3],line))
        img2 = np.hstack((img2,img_arr[4]))
        img2 = np.hstack((img2,line))
        img2 = np.hstack((img2,img_arr[5]))

        img  = np.vstack((img1,img2))
        cv2.imshow('combine',img)

    elif count == 7:
        img1 = np.hstack((img_arr[0],line))
        img1 = np.hstack((img1,img_arr[1]))
        img1 = np.hstack((img1,line))
        img1 = np.hstack((img1,img_arr[2]))

        img2 = np.hstack((img_arr[3],line))
        img2 = np.hstack((img2,img_arr[4]))
        img2 = np.hstack((img2,line))
        img2 = np.hstack((img2,img_arr[5]))

        img3 = np.hstack((img_arr[6],line))
        img3 = np.hstack((img3,black))
        img3 = np.hstack((img3,line))
        img3 = np.hstack((img3,black))

        img  = np.vstack((img1,img2))
        img = np.vstack((img,img3))

        cv2.imshow('combine',img)

    elif count == 8:
        img1 = np.hstack((img_arr[0],line))
        img1 = np.hstack((img1,img_arr[1]))
        img1 = np.hstack((img1,line))
        img1 = np.hstack((img1,img_arr[2]))

        img2 = np.hstack((img_arr[3],line))
        img2 = np.hstack((img2,img_arr[4]))
        img2 = np.hstack((img2,line))
        img2 = np.hstack((img2,img_arr[5]))

        img3 = np.hstack((img_arr[6],line))
        img3 = np.hstack((img3,img_arr[7]))
        img3 = np.hstack((img3,line))
        img3 = np.hstack((img3,black))

        img  = np.vstack((img1,img2))
        img = np.vstack((img,img3))

        cv2.imshow('combine',img)

    elif count == 9:
        img1 = np.hstack((img_arr[0],line))
        img1 = np.hstack((img1,img_arr[1]))
        img1 = np.hstack((img1,line))
        img1 = np.hstack((img1,img_arr[2]))

        img2 = np.hstack((img_arr[3],line))
        img2 = np.hstack((img2,img_arr[4]))
        img2 = np.hstack((img2,line))
        img2 = np.hstack((img2,img_arr[5]))

        img3 = np.hstack((img_arr[6],line))
        img3 = np.hstack((img3,img_arr[7]))
        img3 = np.hstack((img3,line))
        img3 = np.hstack((img3,img_arr[8]))

        img  = np.vstack((img1,img2))
        img = np.vstack((img,img3))

        cv2.imshow('combine',img)


    else:
        pass
```

In [ ]: