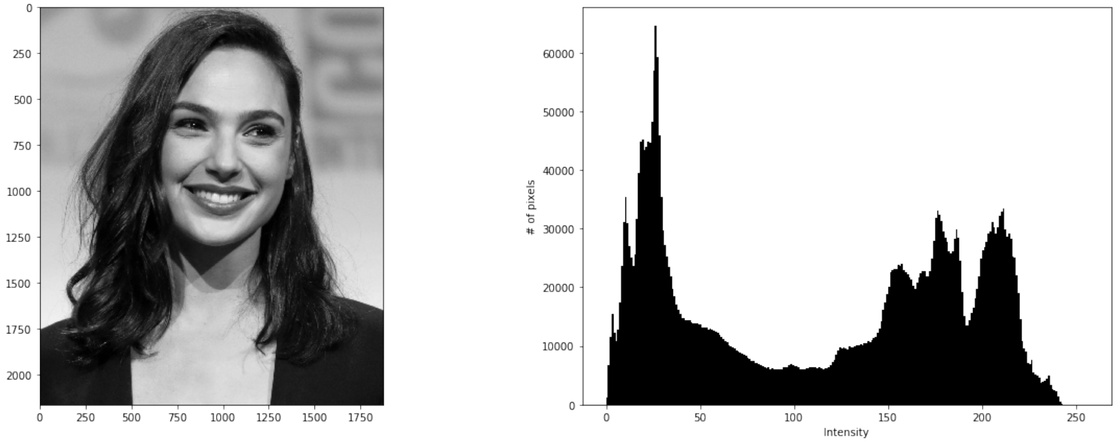
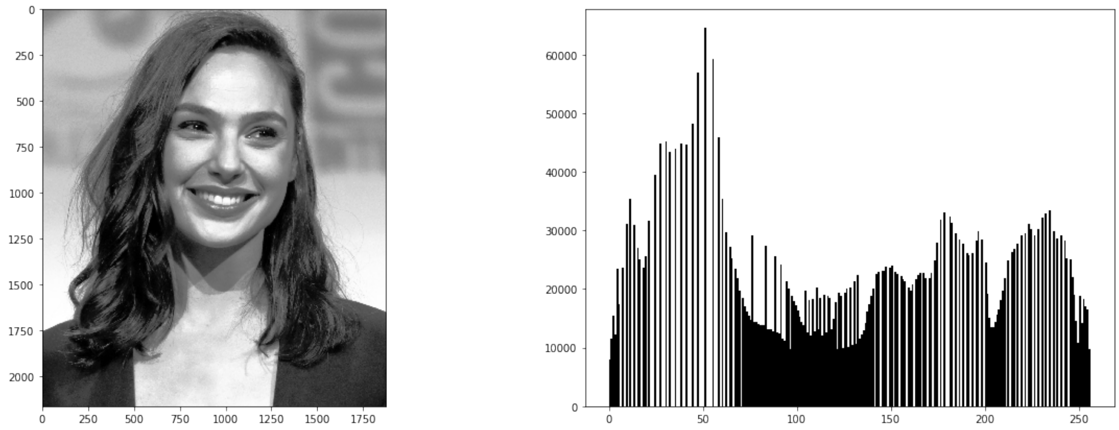
ASSIGNMENT_1

March 1, 2021

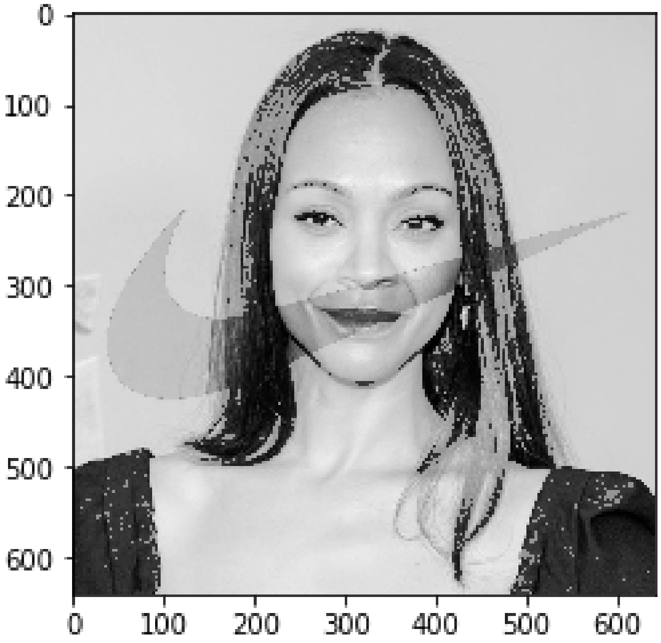
```
In [1]: %matplotlib inline
        import cv2 as cv
        import numpy as np
        import matplotlib.pyplot as plt
        img = cv.imread('../images/gal_gadot.jpg',cv.IMREAD_GRAYSCALE)
        fig, axes = plt.subplots(1,2,squeeze = False,figsize = (20,7), sharex = False)
        axes[0,0].imshow(img,cmap='gray')
        axes[0,1].hist(img.ravel(),256,[0,256],color ='black');
        axes[0,1].set_xlabel("Intensity")
        axes[0,1].set_ylabel("# of pixels")
        plt.show()
```



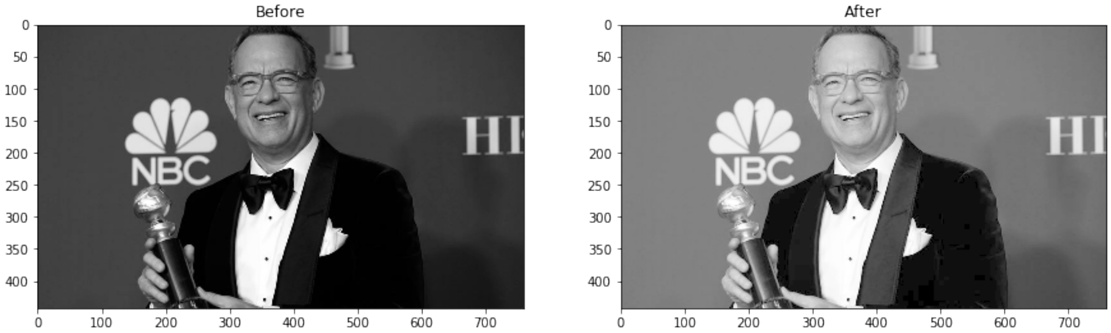
```
In [55]: eq_img = cv.equalizeHist(img)
         fig_2, axes_2= plt.subplots(1,2,figsize=(20,7))
         axes_2[0].imshow(eq_img,cmap='gray')
         axes_2[1].hist(eq_img.ravel(),256,[0,256],color='black')
         plt.show()
```



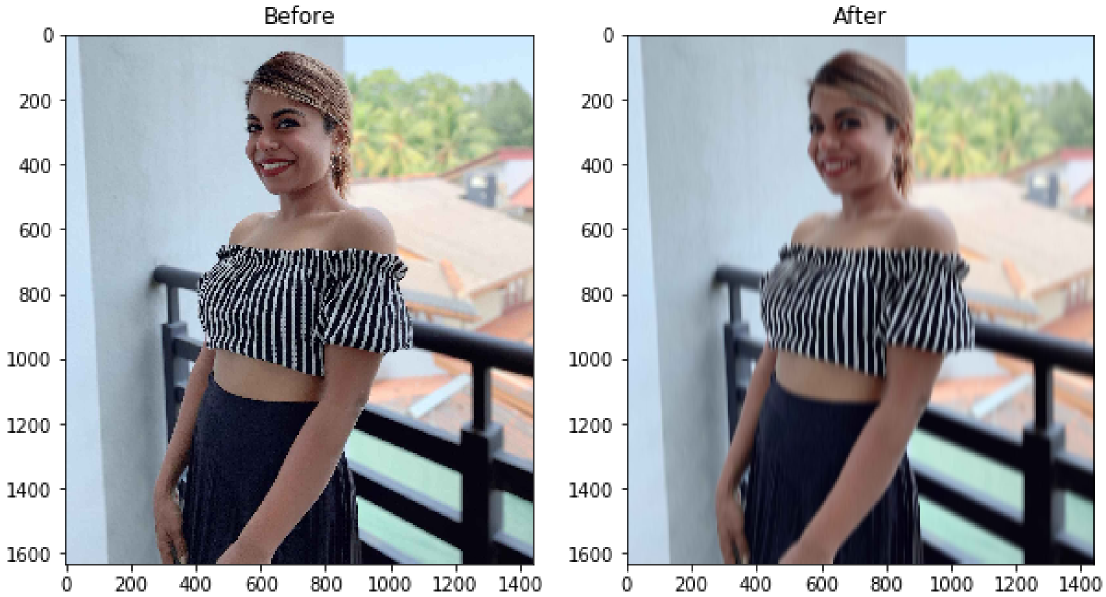
```
In [93]: alpha =0.2
         img_1 = cv.imread('../images/zoe.jpg',cv.IMREAD_GRAYSCALE)
         img_2 = cv.imread('../images/nike.jpg',cv.IMREAD_GRAYSCALE)
         #plt.imshow(img_2,'gray')
         np.shape(img_1)[1]
         np.shape(img_2)
         for i in range(0,np.shape(img_2)[0]):
             for j in range(0,np.shape(img_2)[1]):
                 val = (1-alpha)*img_2[i][j]+img_1[i][j]
                 if (val >255):
                     img_1[i][j] = (val/510)*255

         plt.imshow(img_1,'gray')
         plt.show()
```



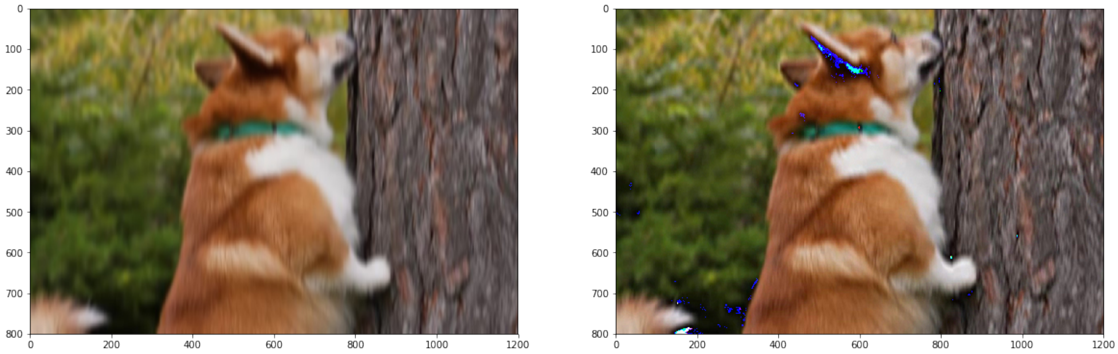
```
In [202]: gamma = 0.5
          tom = cv.imread('../images/tom_hanks.jpg',cv.IMREAD_GRAYSCALE)
          LUT = np.empty((1,256), np.uint8)
          for i in range(256):
              LUT[0,i] = np.clip(pow(i / 255.0, gamma) * 255.0, 0, 255)
          result = cv.LUT(tom, LUT)
          fig, axes = plt.subplots(1,2,figsize=(15,5))
          axes[0].imshow(tom,'gray')
          axes[1].imshow(result,'gray')
          axes[0].set_title("Before")
          axes[1].set_title("After")
          plt.show()
```



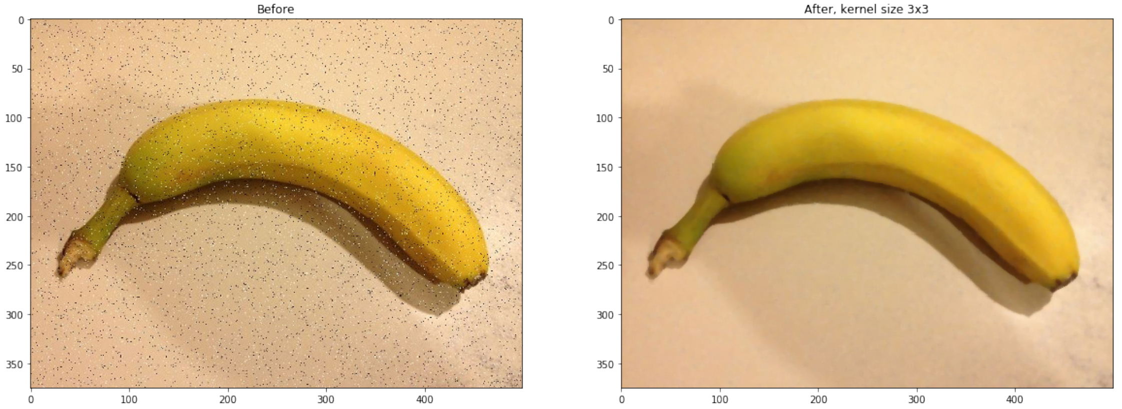
```
In [180]: umaria = cv.imread('../images/umaria.jpg',cv.IMREAD_COLOR)
          umaria = cv.cvtColor(umaria,cv.COLOR_BGR2RGB)
          kernel = cv.getGaussianKernel(31,10)
          result_gau_smth = cv.filter2D(umaria,-1,kernel)
          fig, axes = plt.subplots(1,2,figsize=(10,20))
          axes[0].imshow(umaria)
          axes[1].imshow(result_gau_smth)
          axes[0].set_title("Before")
          axes[1].set_title("After")
          plt.show()
```



```
In [197]: corgi = cv.imread('../images/corgi_blurred.jpg',cv.IMREAD_COLOR) #blurred image
          corgi = cv.cvtColor(corgi,cv.COLOR_BGR2RGB)
          #create a blurred image and then subtract it from the original
          kernel_gaussian = cv.getGaussianKernel(31,10)
          intermediate = corgi - cv.filter2D(corgi,-1,kernel_gaussian)
          #add the intermediate to the original image with a coefficient
          coefficient = 2
          result = corgi + coefficient * intermediate
          fig,axes = plt.subplots(1,2,figsize=(20,10))
          axes[0].imshow(corgi)
          axes[1].imshow(result)
          plt.show()
```



```
In [214]: banana = cv.imread('../images/banana.jpg',cv.IMREAD_COLOR)
          banana = cv.cvtColor(banana,cv.COLOR_BGR2RGB)
          median_result = cv.medianBlur(banana,5)
          fig, axes = plt.subplots(1,2,figsize=(20,20))
          axes[0].imshow(banana)
          axes[1].imshow(median_result)
          axes[0].set_title("Before")
          axes[1].set_title("After, kernel size 3x3")
          plt.show()
```



```
In [224]: banana_2 = cv.imread('../images/banana.jpg',cv.IMREAD_COLOR)
          banana_2 = cv.cvtColor(banana_2,cv.COLOR_BGR2RGB)
          bilateral_result = cv.bilateralFilter(banana_2,3,2,2)
          fig, axes = plt.subplots(1,2,figsize=(20,20))
          axes[0].imshow(banana)
          axes[1].imshow(median_result)
          axes[0].set_title("Before")
          axes[1].set_title("After, kernel size 3x3")
          plt.show()
```