

preliminary_data

March 26, 2019

```
In [23]: import cv2
        import numpy as np
        import matplotlib
        from matplotlib import pyplot as plt
%matplotlib inline
matplotlib.rcParams['figure.figsize'] = [10, 10]

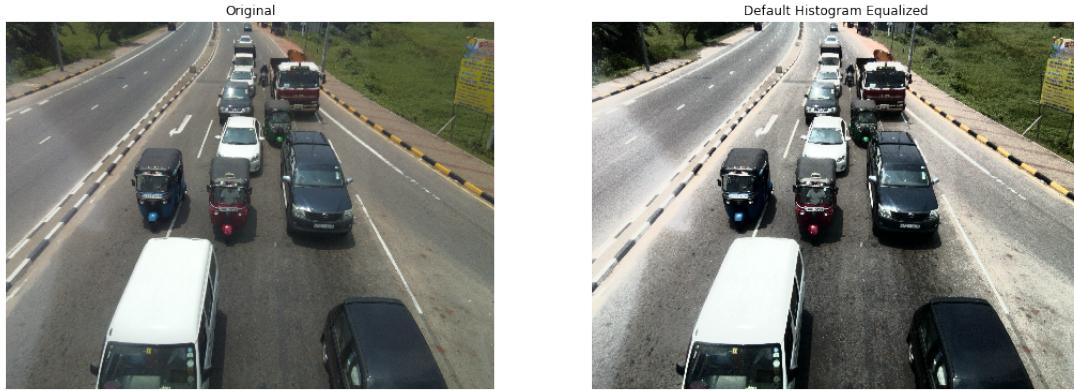
def show_img(img, title=''):
    img_rgb = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
    plt.imshow(img_rgb)
    plt.title(title)
    plt.show()
```

```
In [24]: path1 = r"D:\FYP\data\preliminary_data\hi_res\mid_day\5.png"
        img1 = cv2.imread(path1)
        show_img(img1, 'Original')
```



```
In [25]: # Hist Eq with global contrast (default)
img1_yuv = cv2.cvtColor(img1, cv2.COLOR_BGR2YUV)
img1_yuv[:, :, 0] = cv2.equalizeHist(img1_yuv[:, :, 0])
img1_heq = cv2.cvtColor(img1_yuv, cv2.COLOR_YUV2BGR)

f, axarr = plt.subplots(1,2)
f.set_figheight(18)
f.set_figwidth(18)
axarr[0].imshow(cv2.cvtColor(img1, cv2.COLOR_BGR2RGB))
axarr[0].set_title("Original")
axarr[0].axis('off')
axarr[1].imshow(cv2.cvtColor(img1_heq, cv2.COLOR_BGR2RGB))
axarr[1].set_title("Default Histogram Equalized")
axarr[1].axis('off')
plt.show()
```



```
In [26]: path2 = r"D:\FYP\data\preliminary_data\hi_res\night\b (3).png"
img2 = cv2.imread(path2)
show_img(img2, 'Original Night Image')
```



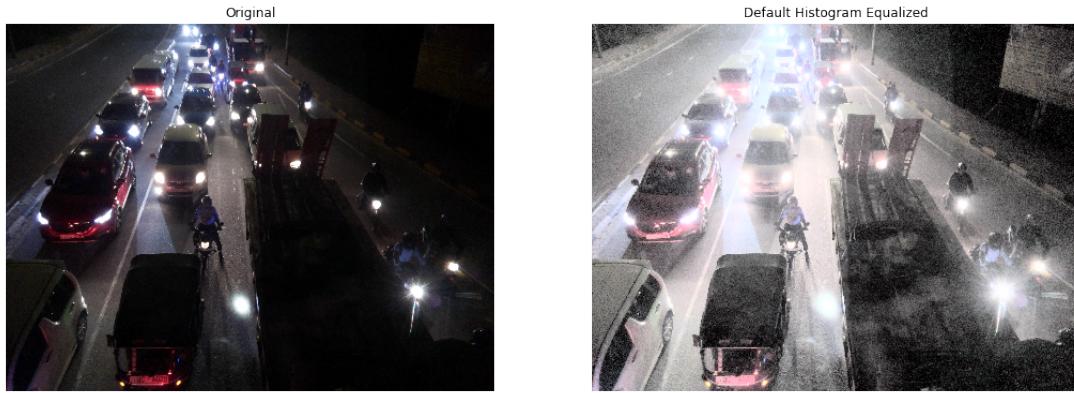
```
In [27]: # Hist Eq with global contrast (default)
img2_yuv = cv2.cvtColor(img2, cv2.COLOR_BGR2YUV)
```

```

img2_yuv[:, :, 0] = cv2.equalizeHist(img2_yuv[:, :, 0])
img2_heq = cv2.cvtColor(img2_yuv, cv2.COLOR_YUV2BGR)

f, axarr = plt.subplots(1,2)
f.set_figheight(18)
f.set_figwidth(18)
axarr[0].imshow(cv2.cvtColor(img2, cv2.COLOR_BGR2RGB))
axarr[0].set_title("Original")
axarr[0].axis('off')
axarr[1].imshow(cv2.cvtColor(img2_heq, cv2.COLOR_BGR2RGB))
axarr[1].set_title("Default Histogram Equalized")
axarr[1].axis('off')
plt.show()

```



```
In [28]: clahe = cv2.createCLAHE(clipLimit=2.0, tileGridSize=(8,8))
```

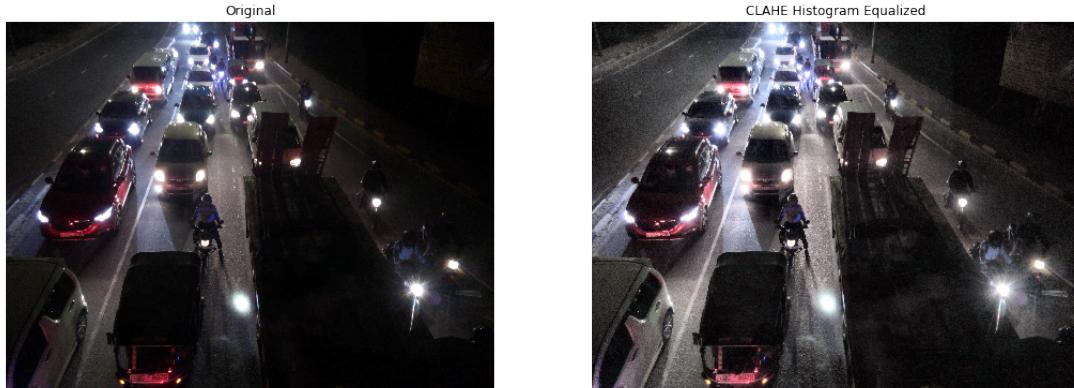
```

img2_yuv = cv2.cvtColor(img2, cv2.COLOR_BGR2YUV)
img2_yuv[:, :, 0] = clahe.apply(img2_yuv[:, :, 0])
img2_heq = cv2.cvtColor(img2_yuv, cv2.COLOR_YUV2BGR)

# Show

f, axarr = plt.subplots(1,2)
f.set_figheight(18)
f.set_figwidth(18)
axarr[0].imshow(cv2.cvtColor(img2, cv2.COLOR_BGR2RGB))
axarr[0].set_title("Original")
axarr[0].axis('off')
axarr[1].imshow(cv2.cvtColor(img2_heq, cv2.COLOR_BGR2RGB))
axarr[1].set_title("CLAHE Histogram Equalized")
axarr[1].axis('off')
plt.show()

```



```
In [29]: path3 = r"D:\FYP\data\preliminary_data\hi_res\night\a (4).png"
img3 = cv2.imread(path3)
show_img(img3, 'Original')
```



```
In [30]: clahe = cv2.createCLAHE(clipLimit=2.0, tileGridSize=(8,8))
```

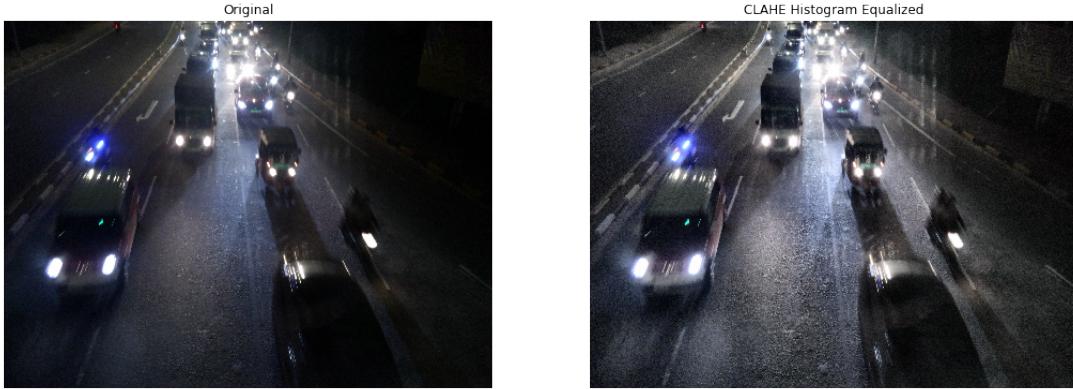
```

img3_yuv = cv2.cvtColor(img3, cv2.COLOR_BGR2YUV)
img3_yuv[:, :, 0] = clahe.apply(img3_yuv[:, :, 0])
img3_heq = cv2.cvtColor(img3_yuv, cv2.COLOR_YUV2BGR)

# Show

f, axarr = plt.subplots(1,2)
f.set_figheight(18)
f.set_figwidth(18)
axarr[0].imshow(cv2.cvtColor(img3, cv2.COLOR_BGR2RGB))
axarr[0].set_title("Original")
axarr[0].axis('off')
axarr[1].imshow(cv2.cvtColor(img3_heq, cv2.COLOR_BGR2RGB))
axarr[1].set_title("CLAHE Histogram Equalized")
axarr[1].axis('off')
plt.show()

```



In [31]: # Unsharp Mask

```

kernel = np.array([[-1,-1,-1], [-1,9,-1], [-1,-1,-1]])
img3_sh = np.copy(img3)
for i in range(3):
    img3_sh[:, :, i] = cv2.filter2D(img3_sh[:, :, i], -1, kernel)

# Show

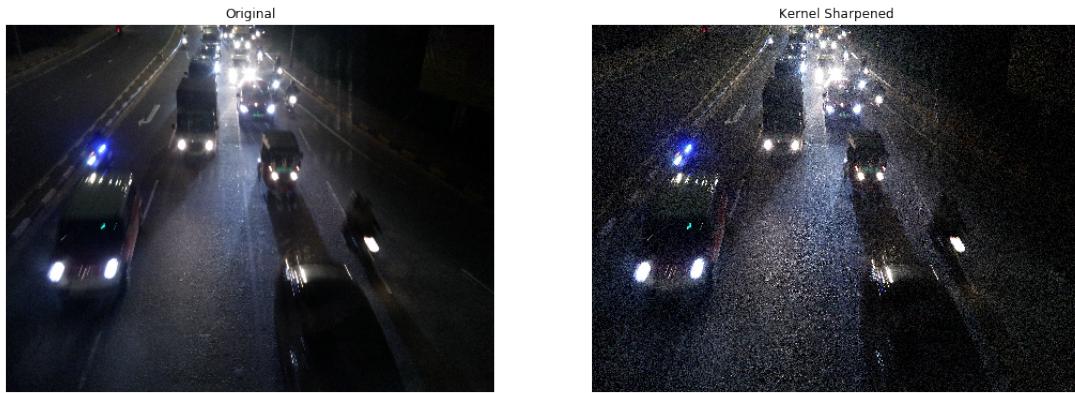
f, axarr = plt.subplots(1,2)
f.set_figheight(18)
f.set_figwidth(18)
axarr[0].imshow(cv2.cvtColor(img3, cv2.COLOR_BGR2RGB))
axarr[0].set_title("Original")
axarr[0].axis('off')
axarr[1].imshow(cv2.cvtColor(img3_sh, cv2.COLOR_BGR2RGB))

```

```

axarr[1].set_title("Kernel Sharpened")
axarr[1].axis('off')
plt.show()

```



In [33]: # Sharpening

```

gaussian_3 = cv2.GaussianBlur(img3, (9,9), 10.0)
img3_sh = cv2.addWeighted(img3, 1.8, gaussian_3, -0.8, 0, img3_sh)

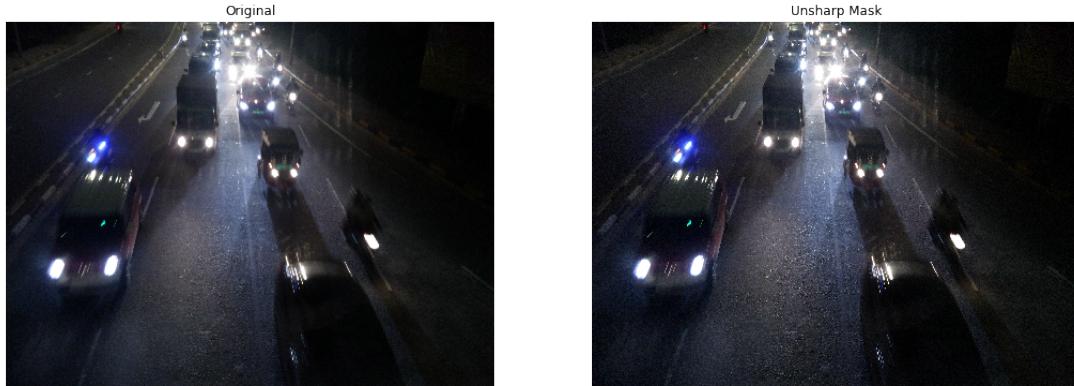
```

Show

```

f, axarr = plt.subplots(1,2)
f.set_figheight(18)
f.set_figwidth(18)
axarr[0].imshow(cv2.cvtColor(img3, cv2.COLOR_BGR2RGB))
axarr[0].set_title("Original")
axarr[0].axis('off')
axarr[1].imshow(cv2.cvtColor(img3_sh, cv2.COLOR_BGR2RGB))
axarr[1].set_title("Unsharp Mask")
axarr[1].axis('off')
plt.show()

```



In []: