

## ▼ dependancys

```

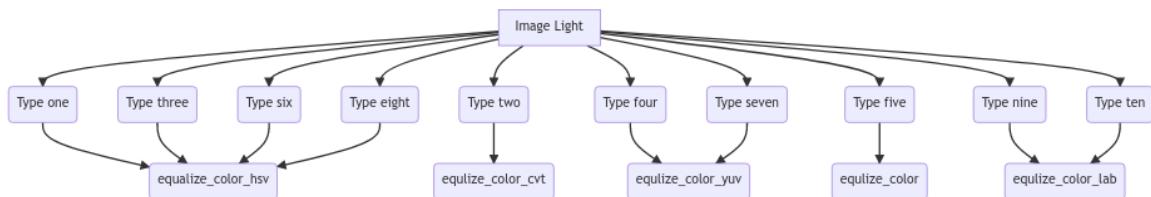
1 from itertools import combinations_with_replacement
2 from collections import defaultdict
3 import cv2
4 import numpy as np
5 from numpy.linalg import inv
6 from google.colab.patches import cv2_imshow
7 import pandas as pd
8 import glob
9 import os
10 from google.colab import drive
11 drive.mount('/content/drive')

```

Drive already mounted at /content/drive; to attempt to forcibly remount, call

## Histogram equilization

is the way we choose to equilize the intencety of the images, [OpenCV](#) has Impalimentaion of it wich handles gray images, so acording to this [StackoverFlow post](#) in RGB images each chanale represents the **intensity of the related color** so we saprated the **Brightness** of the image, By using **standardized colorspace**s that encode brightness and color **separately** like (YCbCr, HSV, ect ).



and then **Applay simple Histogram**, because Histogram lacking results, due to it's lack of ragrad to outliers and the location of the pixel, we chose to implemint Contrast Limited Adaptive HE witch have better slitly better reusltus.

## results





## ▼ Filters and Sparte Chanales

```
1
2 # FILters and Sprate Chanles.
3 clip=6.0
4 def equalize_clahe_color_hsv(img):
5     cla = cv2.createCLAHE(clipLimit=clip) # creatin Object with clip ie. constr
```

```

6     H, S, V = cv2.split(cv2.cvtColor(img, cv2.COLOR_BGR2HSV))
7     eq_V = cla.apply(V)
8     eq_image = cv2.cvtColor(cv2.merge([H, S, eq_V]), cv2.COLOR_HSV2BGR)
9     return eq_image
10    def equalize_clahe_color_lab(img):
11        cla = cv2.createCLAHE(clipLimit=clip)
12        L, a, b = cv2.split(cv2.cvtColor(img, cv2.COLOR_BGR2Lab))
13        eq_L = cla.apply(L)
14        eq_image = cv2.cvtColor(cv2.merge([eq_L, a, b]), cv2.COLOR_Lab2BGR)
15        return eq_image
16    def equalize_clahe_color_yuv(img):
17        cla = cv2.createCLAHE(clipLimit=clip)
18        Y, U, V = cv2.split(cv2.cvtColor(img, cv2.COLOR_BGR2YUV))
19        eq_Y = cla.apply(Y)
20        eq_image = cv2.cvtColor(cv2.merge([eq_Y, U, V]), cv2.COLOR_YUV2BGR)
21        return eq_image
22    def equalize_clahe_color(img):
23        cla = cv2.createCLAHE(clipLimit=clip)
24        channels = cv2.split(img)
25        eq_channels = []
26        for ch in channels:
27            eq_channels.append(cla.apply(ch))
28        eq_image = cv2.merge(eq_channels)
29        return eq_image
30
31    def equalize_cvtColor(img):
32        img_hsv = cv2.cvtColor(img, cv2.COLOR_RGB2HSV)
33
34        # Histogram equalisation on the V-channel
35        img_hsv[:, :, 2] = cv2.equalizeHist(img_hsv[:, :, 2])
36        img2 = cv2.cvtColor(img_hsv, cv2.COLOR_HSV2RGB)
37        return img2
38
39
40
41    # Loading Images based on Light types
42
43    os.chdir("/content/drive/MyDrive/Cv_ProjectData/Images/Images") # Changing di
44    path = "/content/drive/MyDrive/Cv_ProjectData/Fillterd/train/" # path saving
45
46    imageLabel = pd.read_csv('/content/drive/MyDrive/Cv_ProjectData/TrainImageLab
47
48    # Loading Rows of each image light type
49    image1 = imageLabel.loc[imageLabel["LightType"] == 1]
50    image2 = imageLabel.loc[imageLabel["LightType"] == 2]
51    image3 = imageLabel.loc[imageLabel["LightType"] == 3]
52    image4 = imageLabel.loc[imageLabel["LightType"] == 4]
53    image5 = imageLabel.loc[imageLabel["LightType"] == 5]
54    image6 = imageLabel.loc[imageLabel["LightType"] == 6]
55    image7 = imageLabel.loc[imageLabel["LightType"] == 7]
56    image8 = imageLabel.loc[imageLabel["LightType"] == 8]
57    image9 = imageLabel.loc[imageLabel["LightType"] == 9]
58    image10 = imageLabel.loc[imageLabel["LightType"] == 10]
59    # Filtering Rows and keeping only the pass of the intended Images.
60    imageName = image1["ImageName"]

```

```
61 image2Name = image2["ImageName"]
62 image3Name = image3["ImageName"]
63 image4Name = image4["ImageName"]
64 image5Name = image5["ImageName"]
65 image6Name = image6["ImageName"]
66 image7Name = image7["ImageName"]
67 image8Name = image8["ImageName"]
68 image9Name = image9["ImageName"]
69 image10Name = image10["ImageName"]
70 # converting data fram to list
71 imageName = imageName.to_list()
72 image2Name = image2Name.to_list()
73 image3Name = image3Name.to_list()
74 image4Name = image4Name.to_list()
75 image5Name = image5Name.to_list()
76 image6Name = image6Name.to_list()
77 image7Name = image7Name.to_list()
78 image8Name = image8Name.to_list()
79 image9Name = image9Name.to_list()
80 image10Name = image10Name.to_list()
81
82 # calling Fillters By Itrating on every Image
83 ## Type one
84 count = 5 # count of images keep it 1 to itrate over all images
85 for i in imageName :
86     img = cv2.imread(i) # Reading Images
87     cv2_imshow(img) # showing image befor
88     img = equalize_clahe_color_hsv(img) # Equlizing the image
89     cv2_imshow(img) # showing image after enhancement
90     # cv2.imwrite(path+i , img) # writing the filterd image into dis ps concatni
91     count-=1 # counter for pringting 5 image only and converting 5
92     if not count:
93         print("IMage type 1 ")
94         print()
95         print()
96         break
97
98
99
100 ## Type two
101 count = 5 # count of images keep it 1 to itrate over all images
102 for i in image2Name :
103     img = cv2.imread(i) # Reading Images
104     cv2_imshow(img) # showing image befor
105     img= equalize_cvtColor(img) # Equlizing the image
106     cv2_imshow(img) # showing image after enhancement
107     # cv2.imwrite(path+i , img) # writing the filterd image into dis ps concatni
108     count-=1 # counter for pringting 5 image only and converting 5
109     if not count:
110         print("IMage type 2 ")
111         print()
112         print()
113         break
114
115 ## Type three
```

```
116 count = 5 # count of images keep it 1 to iterate over all images
117 for i in image3Name :
118     img = cv2.imread(i) # Reading Images
119     cv2_imshow(img) # showing image before
120     img = equalize_clahe_color_hsv(img) # Equalizing the image
121     cv2_imshow(img) # showing image after enhancement
122     # cv2.imwrite(path+i , img) # writing the filtered image into this ps concatenation
123     count-=1 # counter for printing 5 image only and converting 5
124 if not count:
125     print("Image type 1 ")
126     print()
127     print()
128     break
129
130 ## Type four
131 count = 5 # count of images keep it 1 to iterate over all images
132 for i in image4Name :
133     img = cv2.imread(i) # Reading Images
134     cv2_imshow(img) # showing image before
135     img = equalize_clahe_color_yuv(img) # Equalizing the image
136     cv2_imshow(img) # showing image after enhancement
137     # cv2.imwrite(path+i , img) # writing the filtered image into this ps concatenation
138     count-=1 # counter for printing 5 image only and converting 5
139 if not count:
140     print("Image type 4 ")
141     print()
142     print()
143     break
144
145 ## Type five
146 count = 5 # count of images keep it 1 to iterate over all images
147 for i in image5Name :
148     img = cv2.imread(i) # Reading Images
149     cv2_imshow(img) # showing image before
150     img = equalize_clahe_color(img) # Equalizing the image
151     cv2_imshow(img) # showing image after enhancement
152     # cv2.imwrite(path+i , img) # writing the filtered image into this ps concatenation
153     count-=1 # counter for printing 5 image only and converting 5
154 if not count:
155     print("Image type 5 ")
156     print()
157     print()
158     break
159
160 ## Type six
161 count = 5 # count of images keep it 1 to iterate over all images
162 for i in image6Name :
163     img = cv2.imread(i) # Reading Images
164     cv2_imshow(img) # showing image before
165     img = equalize_clahe_color_hsv(img) # Equalizing the image
166     cv2_imshow(img) # showing image after enhancement
167     # cv2.imwrite(path+i , img) # writing the filtered image into this ps concatenation
168     count-=1 # counter for printing 5 image only and converting 5
169 if not count:
170     print("Image type 6 ")
```

```
1/1      print()
172      print()
173      break
174
175  ## Type siven
176  count = 5 # count of images keep it 1 to itrate over all images
177  for i in image7Name :
178      img = cv2.imread(i) # Reading Images
179      cv2_imshow(img) # showing image befor
180      img = equalize_clahe_color_hsv(img) # Eqlizing the image
181      cv2_imshow(img) # showing image after enhancement
182      # cv2.imwrite(path+i , img) # writing the filterd image into dis ps concatni
183      count-=1 # counter for pringting 5 image only and converting 5
184  if not count:
185      print("IMage type 7 ")
186      print()
187      print()
188      break
189
190  ## Type eight
191  count = 5 # count of images keep it 1 to itrate over all images
192  for i in image8Name :
193      img = cv2.imread(i) # Reading Images
194      cv2_imshow(img) # showing image befor
195      img = equalize_clahe_color_hsv(img) # Eqlizing the image
196      cv2_imshow(img) # showing image after enhancement
197      # cv2.imwrite(path+i , img) # writing the filterd image into dis ps concatni
198      count-=1 # counter for pringting 5 image only and converting 5
199  if not count:
200      print("IMage type 8 ")
201      print()
202      print()
203      break
204
205  ## Type nine
206  count = 5 # count of images keep it 1 to itrate over all images
207  for i in image9Name :
208      img = cv2.imread(i) # Reading Images
209      cv2_imshow(img) # showing image befor
210      img = equalize_clahe_color_hsv(img) # Eqlizing the image
211      cv2_imshow(img) # showing image after enhancement
212      # cv2.imwrite(path+i , img) # writing the filterd image into dis ps concatni
213      count-=1 # counter for pringting 5 image only and converting 5
214  if not count:
215      print("IMage type 9 ")
216      print()
217      print()
218      break
219
220  ## Type ten
221  count = 5 # count of images keep it 1 to itrate over all images
222  for i in image10Name :
223      img = cv2.imread(i) # Reading Images
224      cv2_imshow(img) # showing image befor
225      img = equalize_clahe_color_lab(img) # Eqlizing the image
226      # cv2.imwrite(path+i , img) # writing the filterd image into dis ps concatni
```

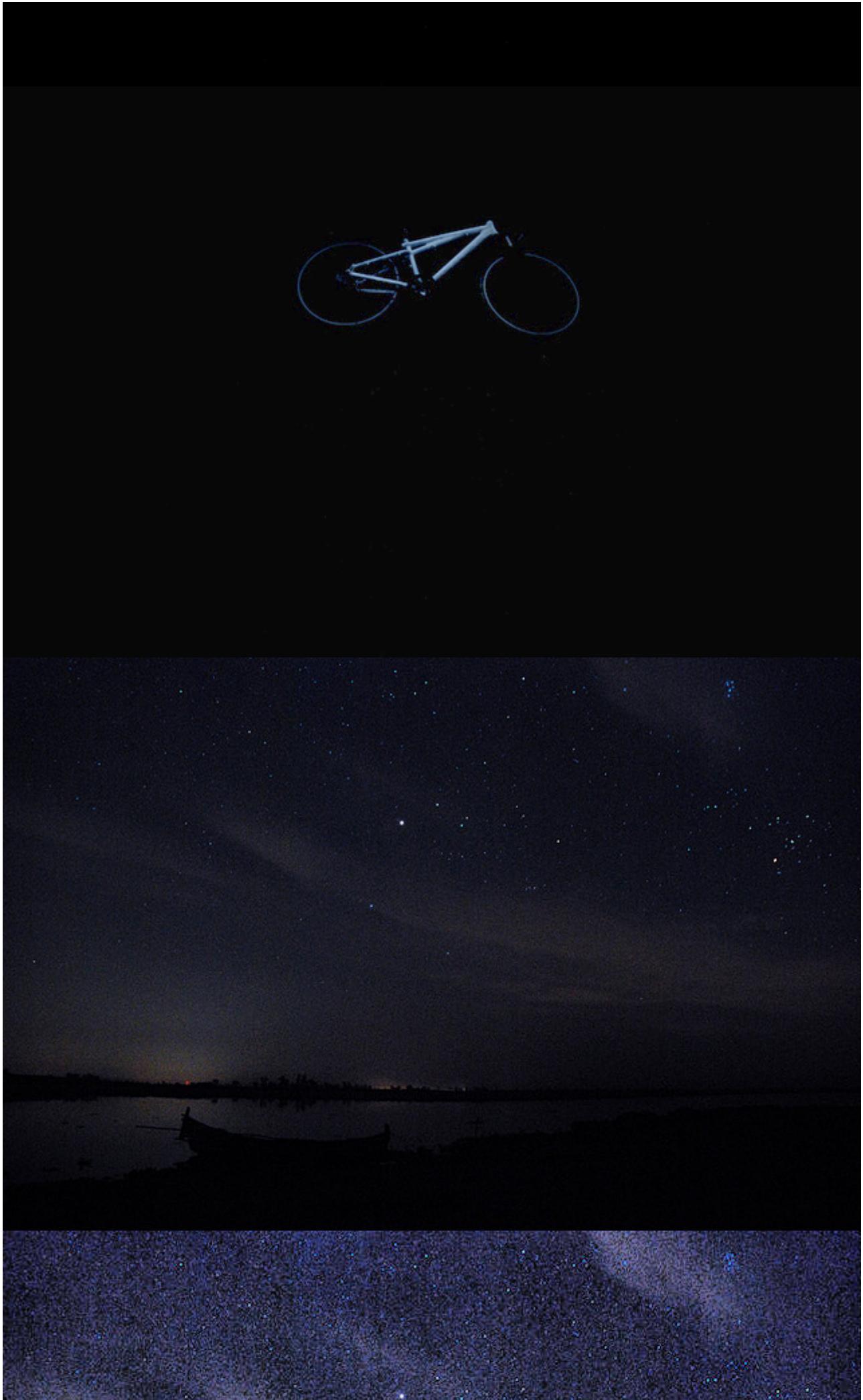
5/31/22, 4:35 AM

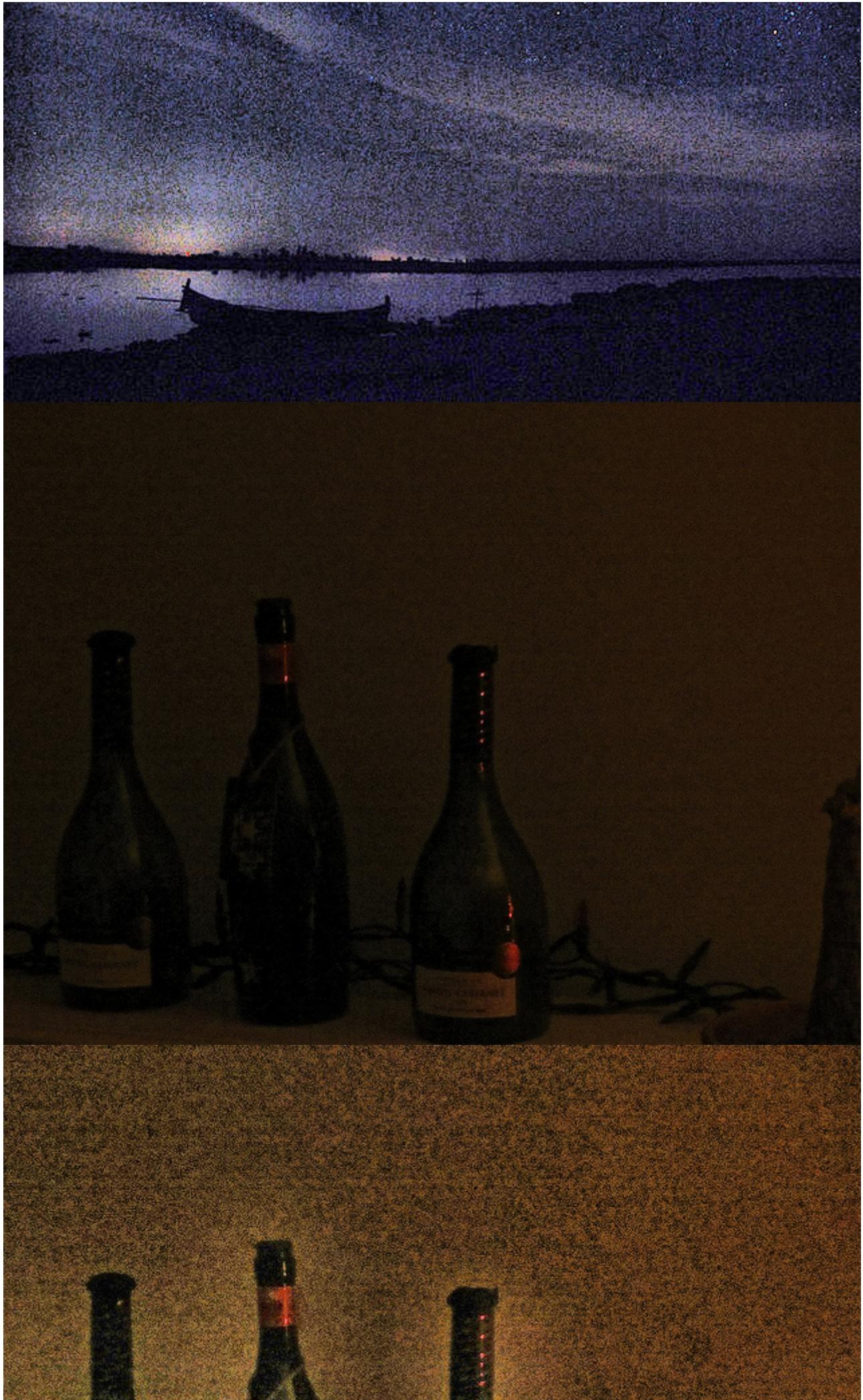
CV\_intesity\_improveTry.ipynb - Colaboratory

```
220     cv2_imshow(img) # showing image after enhancement
221     # cv2.imwrite(path+i , img) # writing the filtered image into this ps concatenated
222     count-=1 # counter for printing 5 image only and converting 5
223     if not count:
224         print("IMage type 10 ")
225         print()
226         print()
227         break
228
229
230
231
232
233
234
235
236
237
```

-- INSERT --







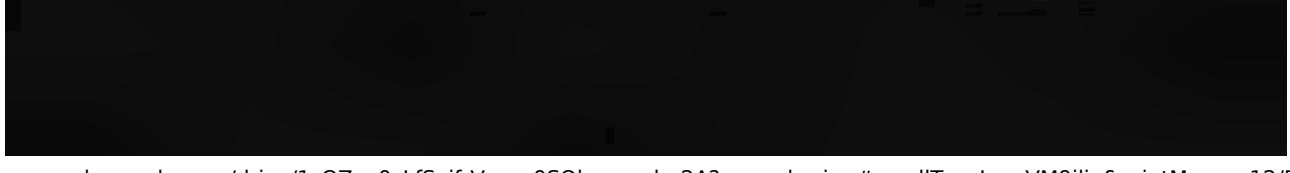
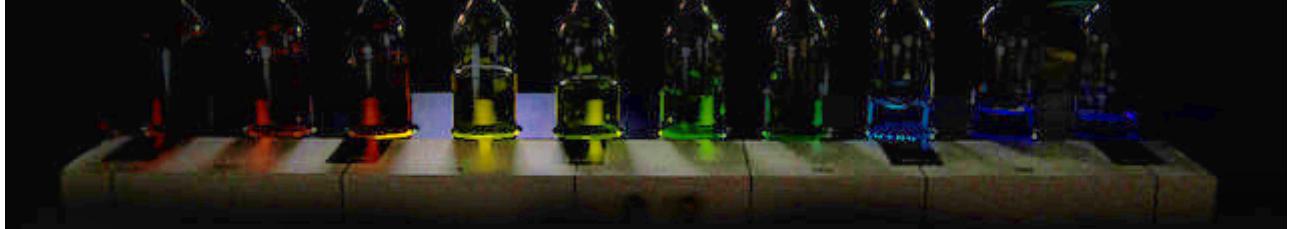
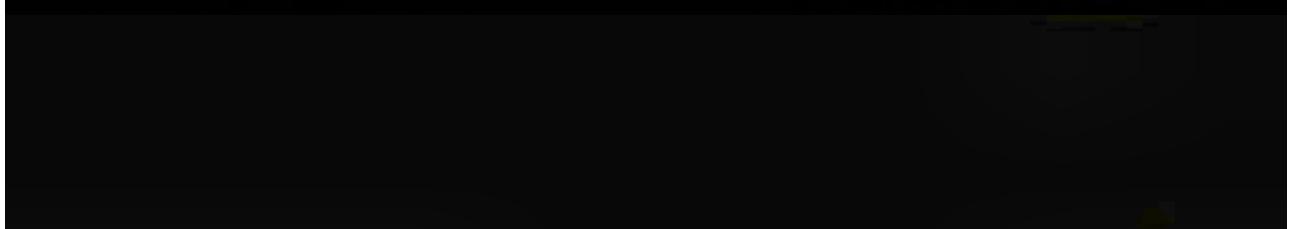
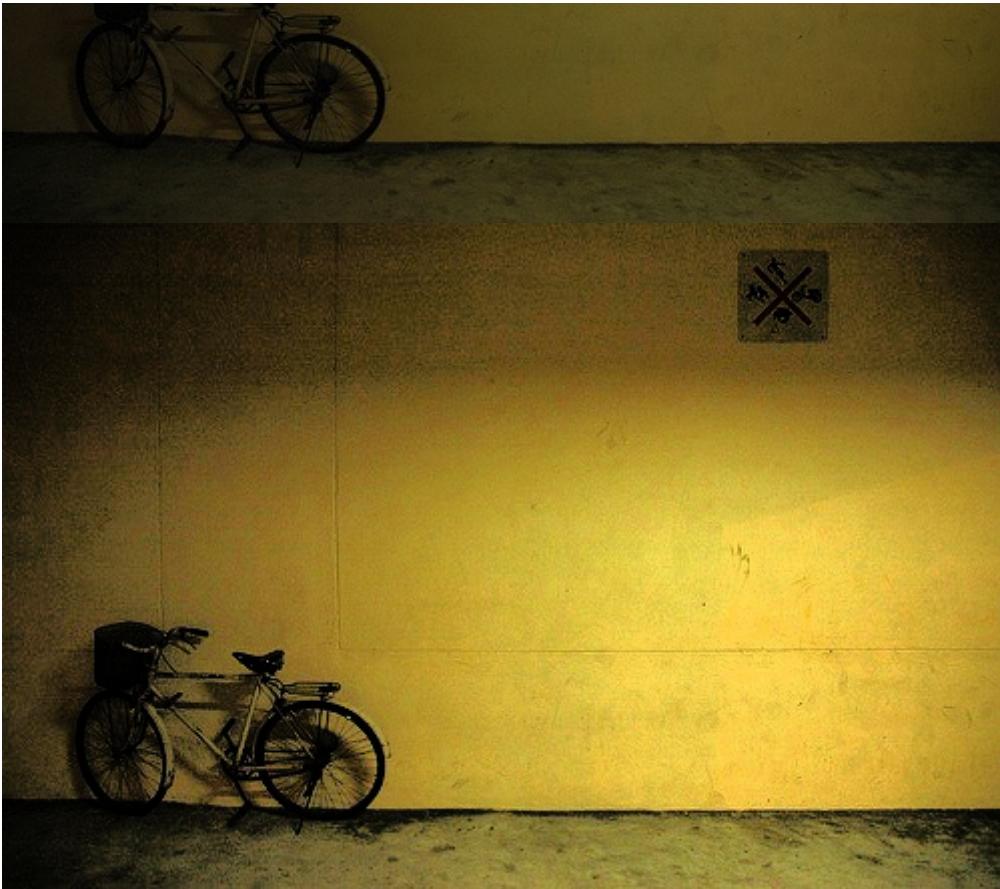
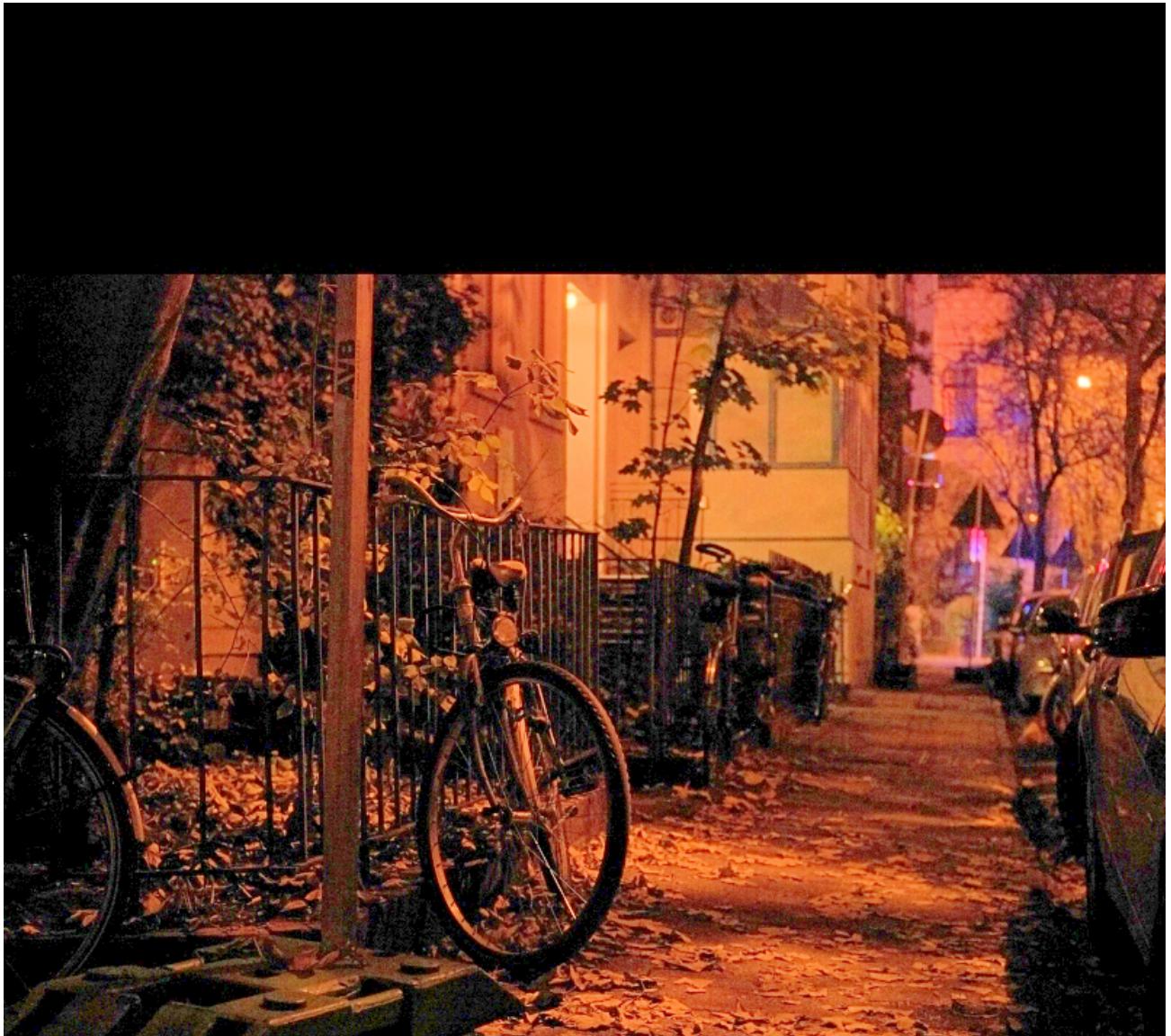


IMage type 1







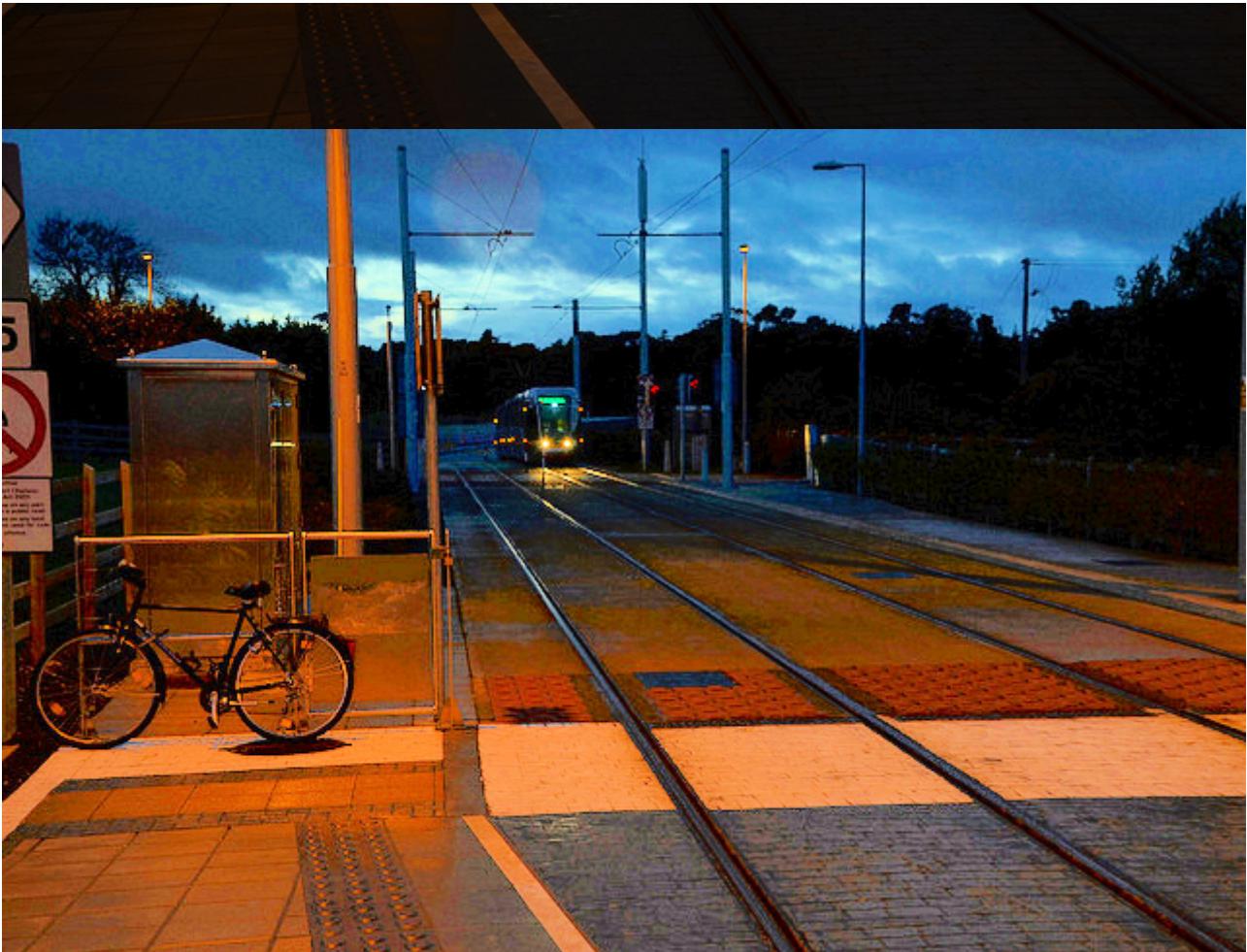




IMage type 2







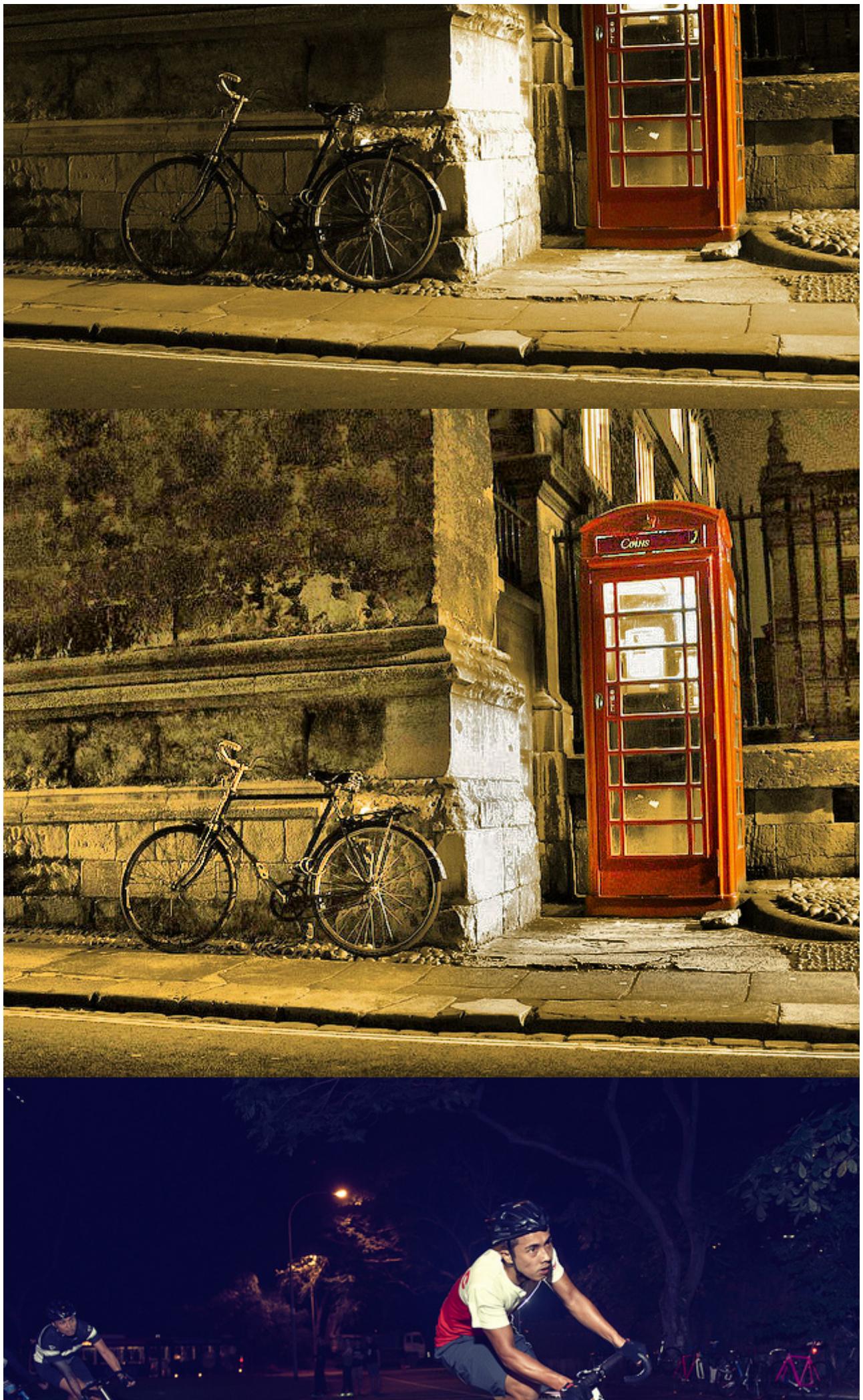




IMage type 1







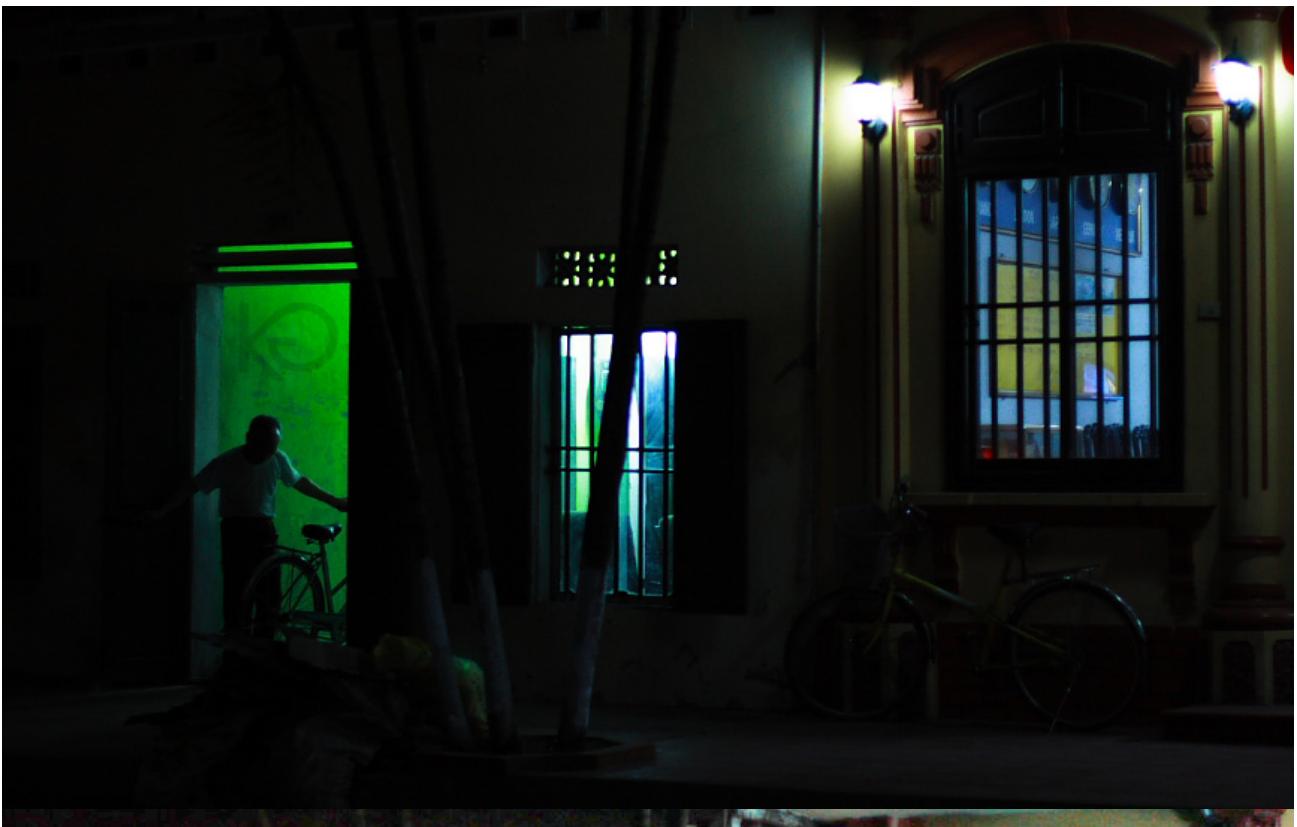






IMage type 4







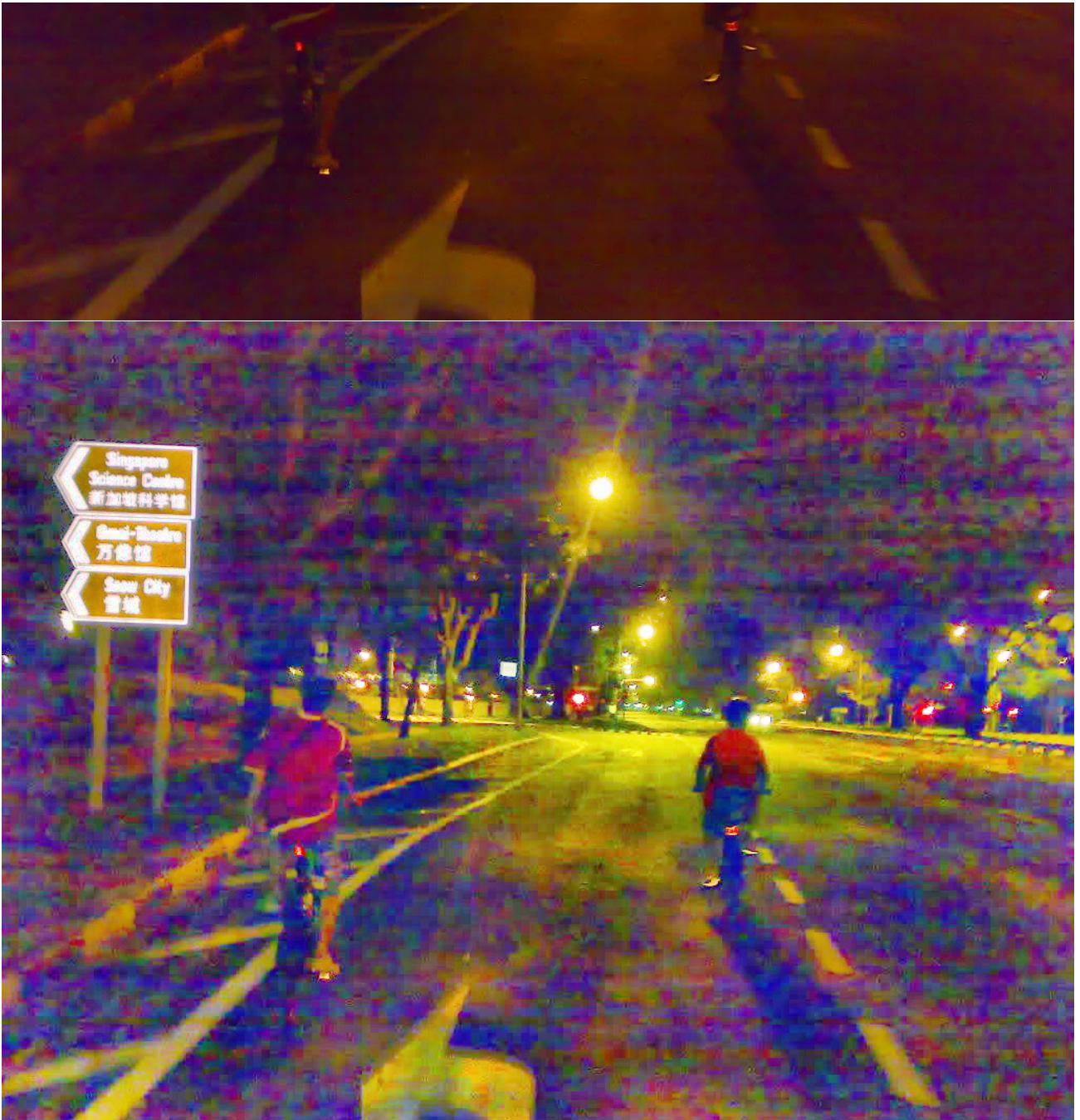




IMage type 5

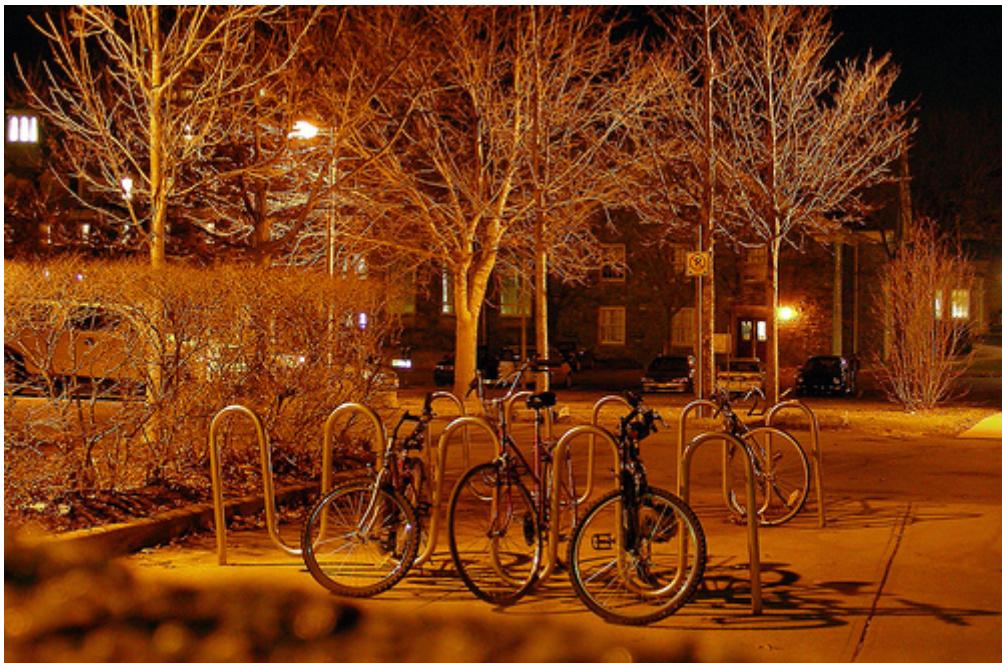










IMage type 6











IMage type 7









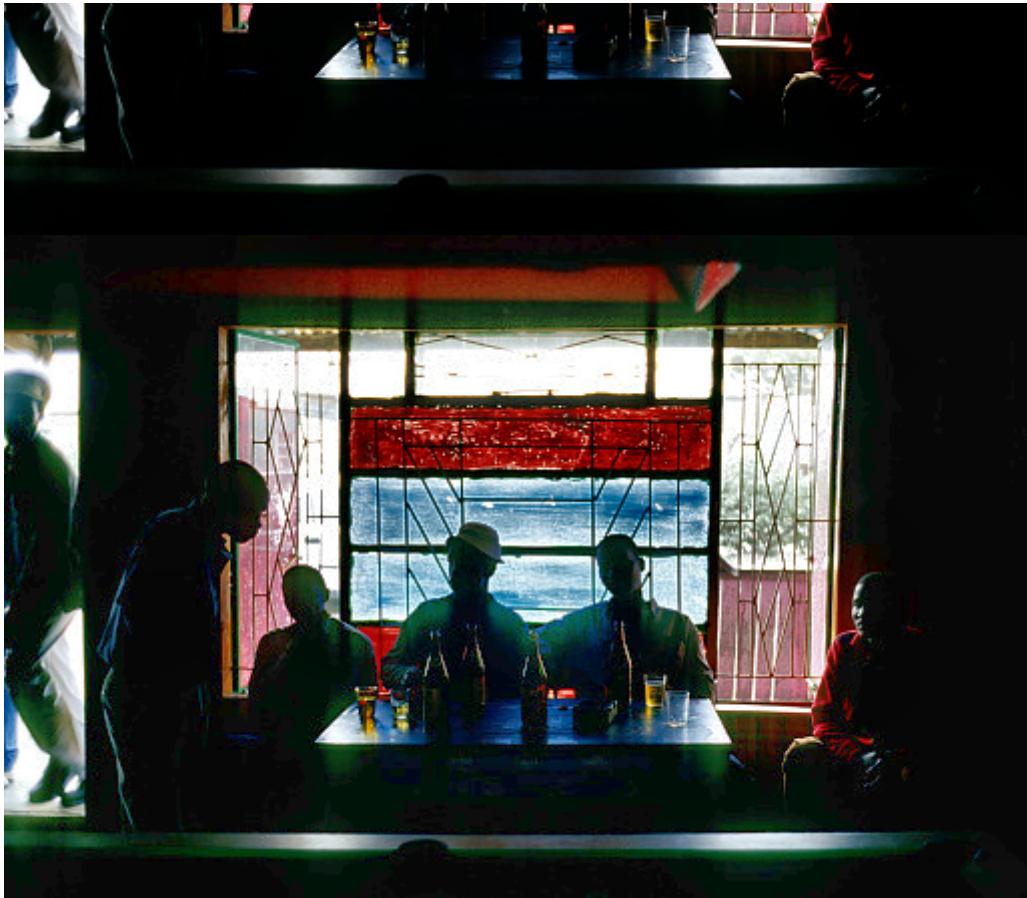




IMage type 8







