Пензенский государственный университет

Кафедра «Вычислительная техника»

**ОТЧЕТ**

по лабораторной работе № 2

по курсу «Цифровая обработка изображение»

Выполнил:  
студент группы 22ВВИм1:

Милованов А.С.

Приняли:

Митрохин М.А  
Панков А.А.

Пенза 2023

# Цель работы: ознакомиться с библиотекой OpenCV и получить навык её использования. Реализовать эквализацию изображения.

# Порядок выполнения работы:

# Формула эквализации изображения

# 

# Где Hist гистограмма для цветового канала

# Задание:

# Преобразовать/улучшить исходное изображение

# 

Рисунок 1 Исходное изображение

# Листинг программы:

#!/usr/bin/env python  
# coding: utf-8  
  
# In[67]:  
  
  
import numpy as np  
import cv2 as cv  
import matplotlib.pyplot as plt  
  
  
# In[68]:  
  
  
image = cv.imread('croped\_image.jpg')  
plt.imshow(image)  
  
  
# In[74]:  
  
  
histSize = [256]  
range\_for\_hist = [0, 256]  
  
#функция получения гистограммы  
def get\_hist(image, histSize, range\_for\_hist,channel):  
 histSize = [256]  
 range = [0, 256]  
 hist = cv.calcHist([image], [channel], None, histSize, range\_for\_hist)  
 return(hist)   
  
#функия отображения гистограммы по трем цветовым каналам  
def plot\_rgb\_hist(image, histSize, range):  
 histSize = [256]  
 range = [0, 256]  
 for i, col in enumerate(['r', 'g', 'b']):  
 hist = cv.calcHist([image], [i], None, histSize, range)  
 plt.plot(hist, color=col, marker='o', linewidth=0)  
 plt.xlim(range)  
  
  
# In[75]:  
  
  
#Функция эквализации (реализованная)  
def def\_lut (i,hist\_for\_channel):  
 array = i.reshape(-1)  
 lut = np.empty(array.shape[0], np.uint8)  
 i = 0  
 divider = sum(hist\_for\_channel)  
 for element in array:  
 dividend=0  
 for j in range(element):   
 dividend+=hist\_for\_channel[j]  
 lut[i]=255 \* (dividend /divider)  
 i+=1  
 return lut.reshape(256,386)   
   
result\_image = np.empty(np.shape(image), np.uint8)   
  
hist\_by\_r\_channel = get\_hist(image[:,:,0],histSize,range\_for\_hist,0)  
hist\_by\_g\_channel = get\_hist(image[:,:,1],histSize,range\_for\_hist,0)  
hist\_by\_b\_channel = get\_hist(image[:,:,2],histSize,range\_for\_hist,0)  
result\_image[:,:,0] = def\_lut(image[:,:,0],hist\_by\_r\_channel)  
result\_image[:,:,1] = def\_lut(image[:,:,1],hist\_by\_g\_channel)  
result\_image[:,:,2] = def\_lut(image[:,:,2],hist\_by\_b\_channel)  
  
  
gs = plt.GridSpec(2, 2)  
plt.figure(figsize=(10, 8))  
plt.subplot(gs[0])  
plt.imshow(image)  
plt.subplot(gs[1])  
plt.imshow(result\_image)  
plt.subplot(gs[2])  
plot\_rgb\_hist(image, histSize, range)  
plt.subplot(gs[3])  
plot\_rgb\_hist(result\_image, histSize, range)  
plt.show()  
  
  
# In[85]:  
  
  
#Эквализация с помощью библиотеки  
  
img = image  
  
result\_image = np.empty(np.shape(img), np.uint8)   
result\_image[:,:,0] = cv2.equalizeHist(img[:,:,0])  
result\_image[:,:,1] = cv2.equalizeHist(img[:,:,1])  
result\_image[:,:,2] = cv2.equalizeHist(img[:,:,2])  
  
  
gs = plt.GridSpec(1, 2)  
plt.figure(figsize=(10, 8))  
plt.subplot(gs[0])  
plt.imshow(result\_image)  
plt.subplot(gs[1])  
plot\_rgb\_hist(result\_image, histSize, range)  
  
  
# In[ ]:

# Результат работы программы:

# 

Рисунок 2 Исходное изображение в цветовом пространстве BGR

# 

Рисунок 3 Исходное изображение в цветовом пространстве RGB

# 

Рисунок 4 Результат применения матрицы сдвига в направление оси X для исходного изображения

# 

Рисунок 6 Исходное изображение в цветовом пространстве GRAY

# 

Рисунок 7 Кадрирование изображение

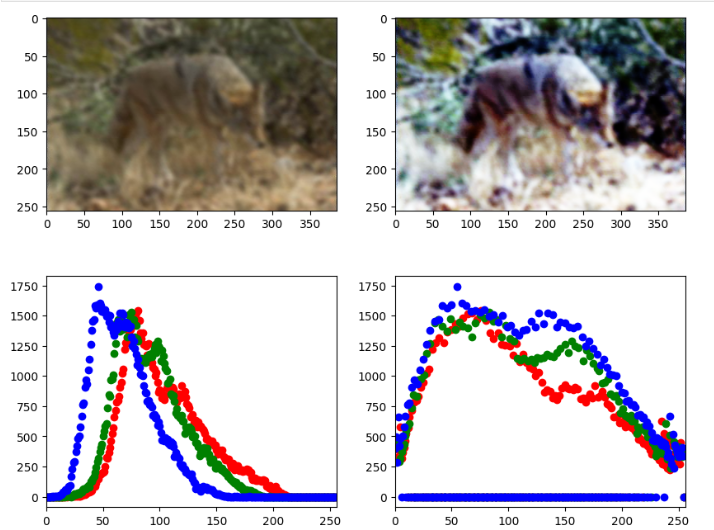


Рисунок 8 Результат применения эквализации к изображению

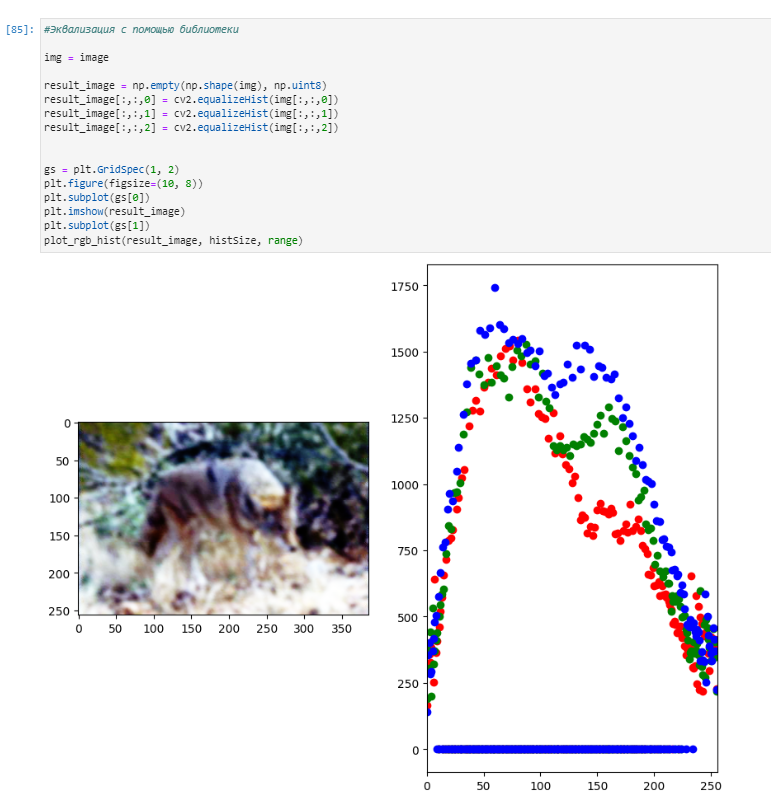


Рисунок 9 Эквализации изображения с помощью библиотеки OpenCV

# Вывод: В ходе выполнения лабораторной работы были получены навыки использования библиотеки OpenCV и реализована эквализация изображения.

# 