Пензенский государственный университет

Кафедра «Вычислительная техника»

**ОТЧЕТ**

по лабораторной работе № 3

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. Применить теоретические знания об улучшении изображения

# Порядок выполнения работы:

# Задание:

# Преобразовать/улучшить исходное изображение

# 

Рисунок Исходное изображение

# Листинг программы:

#!/usr/bin/env python  
# coding: utf-8  
  
# In[1]:  
  
  
import numpy as np  
import cv2 as cv  
import matplotlib.pyplot as plt  
  
  
# In[4]:  
  
  
image = cv.imread('image2.jpg')  
  
  
# In[5]:  
  
  
plt.imshow(image)  
  
  
# ## Преобразование из brg в rgb  
  
# In[6]:  
  
  
#Преобразуем изображение в rgb  
rgb\_image = cv.cvtColor(image, cv.COLOR\_BGR2RGB)  
plt.imshow(rgb\_image)  
  
  
# ## Применение матрицы сдвига в направление оси X  
  
# In[7]:  
  
  
(h, w, d) = rgb\_image.shape  
  
M = np.float32([ [1, 0.5, 0],  
 [0, 1, 0],  
 [0, 0 , 1] ])  
warped = cv.warpPerspective(rgb\_image,M,(w,h))  
plt.imshow(warped)  
  
  
# ## Кадрирование изображение (избавление от черного фона)  
  
# In[10]:  
  
  
gray\_image = cv.cvtColor(warped,cv.COLOR\_RGB2GRAY)  
  
plt.imshow(gray\_image, cmap='gray', vmin = 0, vmax = 255)  
  
  
# In[11]:  
  
  
# Преобразование в двоичный формат  
\_,thresh = cv.threshold(gray\_image,30,255,cv.THRESH\_BINARY)  
  
#Поиск конутра   
contours,hierarchy = cv.findContours(thresh,cv.RETR\_EXTERNAL,cv.CHAIN\_APPROX\_SIMPLE)  
cnt = contours[0]  
  
#Кадрирование  
x,y,w,h= cv.boundingRect(cnt)  
croped\_image = warped[y:y+h,x:x+w]  
cv.imwrite("croped\_image.jpg", croped\_image)  
  
plt.imshow(croped\_image)  
  
  
# In[14]:  
  
  
histSize = [256]  
range = [0, 256]  
  
  
def plot\_rgb\_hist(image, histSize, range):  
 histSize = [256]  
 range = [0, 256]  
 for i, col in enumerate(['b', 'g', 'r']):  
 hist = cv.calcHist([image], [i], None, histSize, range)  
 plt.plot(hist, color=col)  
 plt.xlim(range)  
  
  
# In[15]:  
  
  
croped\_image  
clahe = cv.createCLAHE(clipLimit=2.0, tileGridSize=(8, 8))  
  
rgb\_result\_image = np.empty(np.shape(croped\_image), np.uint8)  
rgb\_result\_image[:, :, 1] = clahe.apply(croped\_image[:, :, 1])  
rgb\_result\_image[:, :, 2] = clahe.apply(croped\_image[:, :, 2])  
rgb\_result\_image[:, :, 0] =clahe.apply(croped\_image[:, :, 0])  
  
  
gs = plt.GridSpec(2, 2)  
plt.figure(figsize=(10, 8))  
plt.subplot(gs[0])  
plt.imshow(croped\_image)  
plt.subplot(gs[1])  
plt.imshow(rgb\_result\_image)  
plt.subplot(gs[2])  
plot\_rgb\_hist(croped\_image, histSize, range)  
plt.subplot(gs[3])  
plot\_rgb\_hist(rgb\_result\_image, histSize, range)  
  
cv.imwrite("rgb\_result\_image.jpg", rgb\_result\_image)  
plt.show()  
  
  
# # Улучшение изображения  
  
# In[16]:  
  
  
kernel1 = np.asarray([[-1, -1, -1], [-1, 9, -1], [-1, -1, -1]])  
filtered\_image = cv.filter2D(rgb\_result\_image, -1, kernel1)  
  
gs = plt.GridSpec(1, 2)  
plt.figure(figsize=(10, 8))  
plt.subplot(gs[0])  
plt.imshow(rgb\_result\_image)  
plt.subplot(gs[1])  
plt.imshow(filtered\_image)  
  
plt.show()  
cv.imwrite("result\_image.jpg", filtered\_image)  
  
  
# 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

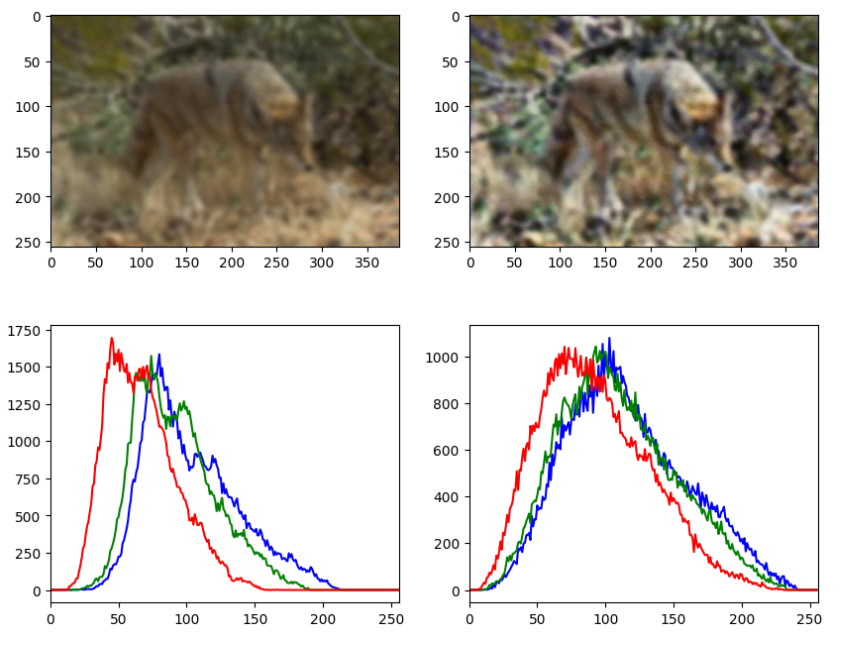


Рисунок 8 Выравнивание гистограмм по трем каналам изображения

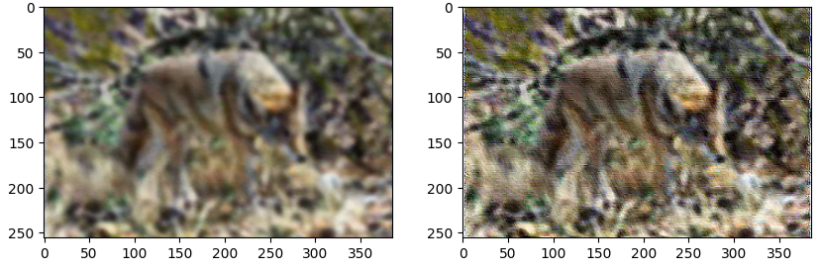


Рисунок 9 Применение матрицы повышения резкости к изображению

# Вывод:

# В ходе выполнения лабораторной работы были получены навыки использования библиотеки OpenCV.

# 