

МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра Систем автоматического управления

ОТЧЕТ
по лабораторной работе №1
по дисциплине «Техническое зрение»
Тема: «Операции с изображениями и видео»

Студент гр. 1492

Ляховский А.С.

Преподаватель

Федоркова А.О.

Санкт-Петербург

2024

Цель работы.

Научиться использовать библиотеки OpenCV и NumPy для работы с изображениями и видео

Ход работы.

Задание 1.

Напишите программу, которая будет выводить изображение на экран следующим образом:

1. в цвете в полном размере на 5 секунд, затем закрыть;
2. в оттенках серого в полном размере на 7 секунд, затем закрыть;
3. в цвете в 2 раза меньше, чем исходный размер, на 9 секунд, затем закрыть;
4. в оттенках серого в 4 раза меньше, чем исходный размер, на 11 секунд, затем закрыть.
5. в цвете в полном размере, поменяв местами зелёный и красный каналы на 4 секунды.
6. Закрыть при нажатии на клавишу Esc.

Код программы:

```
import cv2
import numpy as np
path = 'photo.jpg'
window_name = 'figure 1'
while (True):
    img = cv2.imread(path, flags=cv2.IMREAD_COLOR) # Чтение в формате BGR
    cv2.namedWindow(window_name, flags=cv2.WINDOW_AUTOSIZE) #
    Создание окна
    cv2.imshow(window_name, img) # Вывод изображения на экран
    if cv2.waitKey(5000)==27: # Если в течение 5с нажат ESC, то
        выходим и закрываем окна
        cv2.destroyAllWindows(window_name) # Удаление окна
window_name
```

```

        break # Выход из цикла
img2 = cv2.cvtColor(img,cv2.COLOR_BGR2GRAY) # Из BGR в чб
cv2.imshow(window_name,img2)
if cv2.waitKey(7000)==27:
    cv2.destroyAllWindows(window_name)
    break
img3 = cv2.imread(path, flags=cv2.IMREAD_REDUCED_COLOR_2) #
Чтение bgr, уменьшенное в 2 раза
cv2.imshow(window_name,img3)
if cv2.waitKey(9000)==27:
    cv2.destroyAllWindows(window_name)
    break
img4 = cv2.imread(path, flags=cv2.IMREAD_REDUCED_GRAYSCALE_4) #
Чтение чб, уменьшенное в 4 раза
cv2.imshow(window_name,img4)
if cv2.waitKey(11000)==27:
    cv2.destroyAllWindows(window_name)
    break
b,g,r = cv2.split(img) # Делим на отдельные каналы
img_n = cv2.merge([b,r,g]) # Собираем BRG
cv2.imshow(window_name,img_n)
if cv2.waitKey(4000)==27:
    cv2.destroyAllWindows(window_name)
    break

```

Результат работы программы:



Рисунок 1 Изображение в цвете в полном размере

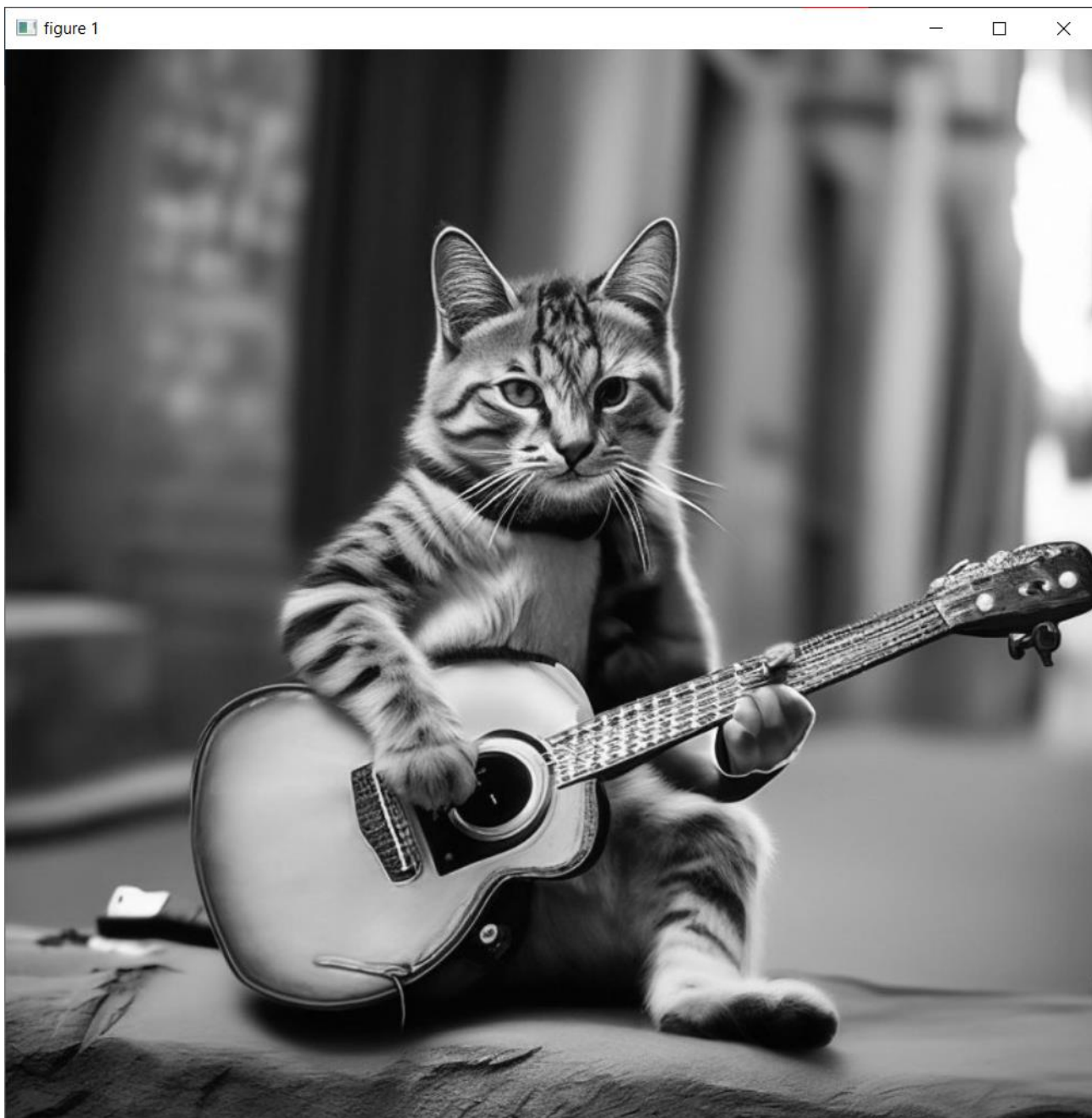


Рисунок 2 Изображение в оттенках серого в полном размере



Рисунок 3 Изображение в цвете в 2 раза меньше



Рисунок 4 Изображение в оттенках серого в 4 раза меньше

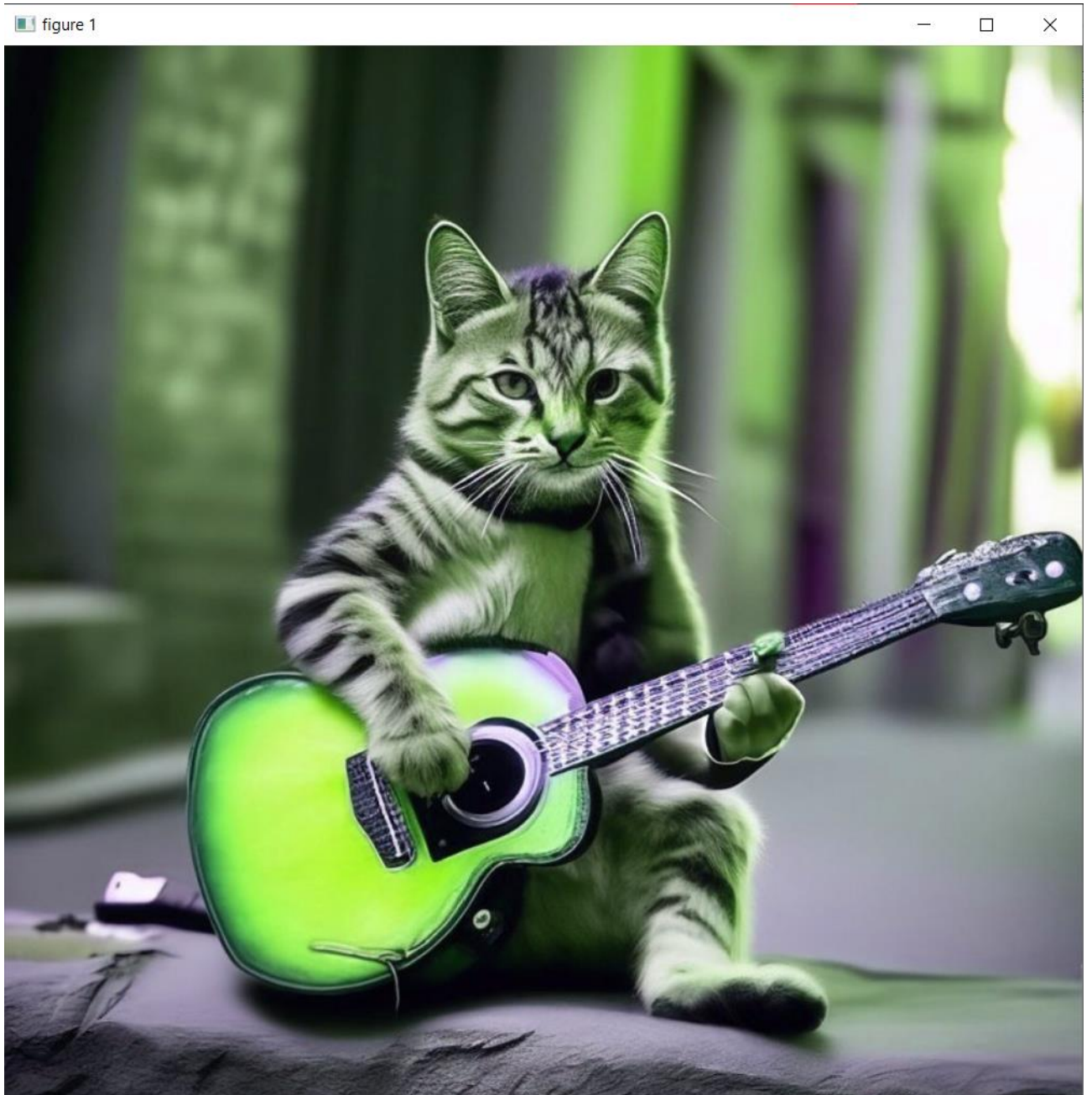


Рисунок 5 Изображение в цвете в полном размере, поменяны местами зелёный и красный

Задание 2.

С помощью средств библиотек OpenCV и NumPy создайте изображение. На белом фоне нарисуйте:

- окружность красным цветом;
- прямоугольник фиолетовым цветом;

- отрезок, соединяющий нижний левый и верхний правый углы изображения - голубым цветом.

Каждый элемент изображения подпишите черным цветом.

Код программы:

```
import cv2
import numpy as np
path = 'Paint.jpg'
window_name = 'figure 2'
font = cv2.FONT_HERSHEY_COMPLEX
fontScale = 0.8
img = np.full((500,500,3),255,dtype='uint8') # Создаём белый холст
cv2.line(img,pt1=(500,0),pt2=(0,500),color=(255,191,0),
thickness=3) # Рисуем голубую линию
cv2.putText(img,'Line',(50,380),font,fontScale,color=(0,0,0))
cv2.rectangle(img,pt1=(100,100),pt2=(200,150),
color=(255,0,139),thickness=3) # Рисуем фиолетовый прямоугольник
cv2.putText(img,'Rectangle',(90,90),font,fontScale,color=(0,0,0))
cv2.circle(img,center=(350,400),radius=50,color=(0,0,255),thickness=3) # Рисуем красный круг
cv2.putText(img,'Circle',(310,340),font,fontScale,color=(0,0,0))
cv2.namedWindow(window_name, flags=cv2.WINDOW_AUTOSIZE) # Создание окна
cv2.imshow(window_name,img) # Вывод изображения на экран
if cv2.waitKey(0)==121: # Если нажата "Y", то сохраняем
    cv2.destroyAllWindows() # Удалить окно
    cv2.imwrite(path,img) # Сохранить изображение
```

Результат работы программы:

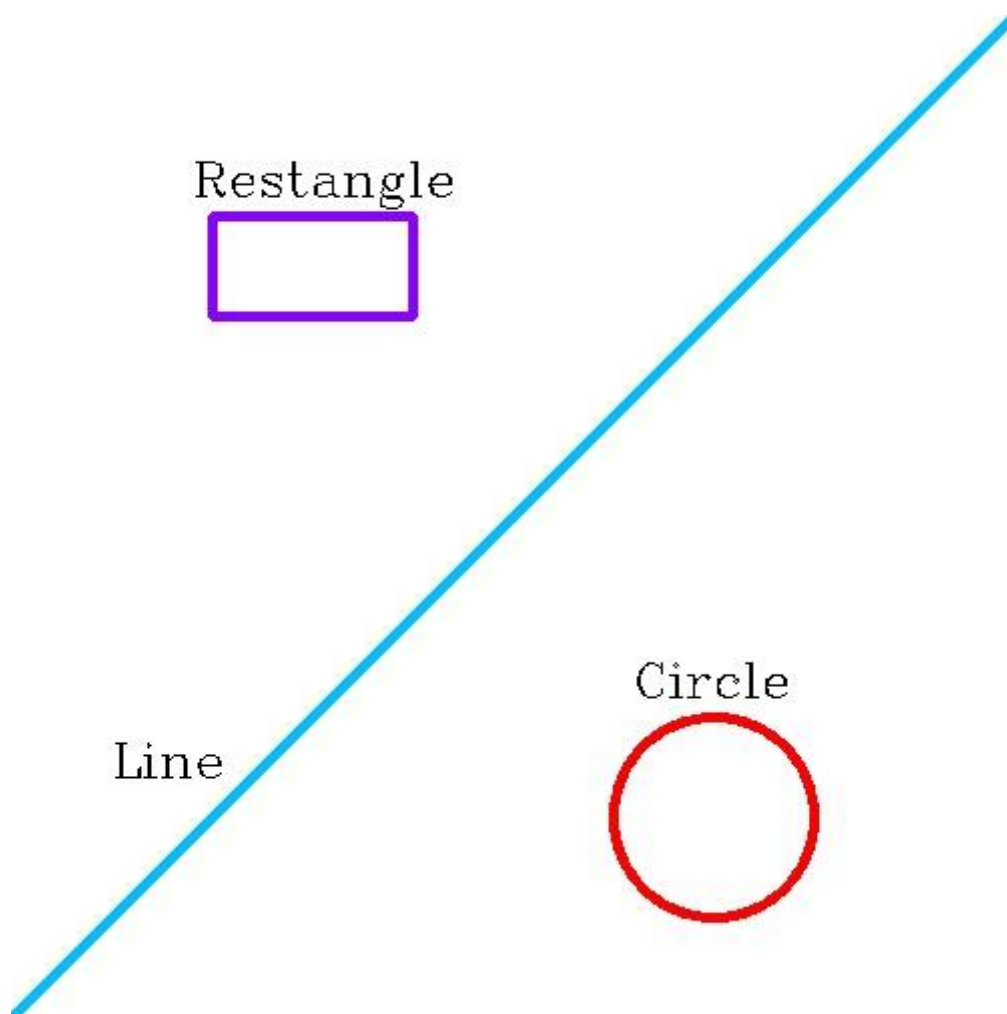


Рисунок 6 Результат работы программы

Задание 3.

С помощью средств библиотек OpenCV и NumPy создайте изображение шахматной доски шириной в 15 клеток и длиной в 40 клеток. Пусть верхний левый квадрат будет фиолетовым, а его сосед справа - белым. Дальше чередуйте эти цвета в шахматном порядке. Откройте изображение на 5 секунд.

Площадь каждого квадрата: 100 пикселей

Код программы:

```
import cv2
import numpy as np
path = 'Desk.jpg'
window_name= 'figure 3'
step=10
hight = 40*step
widht = 15*step
```

```

img = np.full((hight,widht,3),255,dtype='uint8') # Создаём белый холст
x=0
y=0
while(y<hight):
cv2.rectangle(img,pt1=(x,y),pt2=(x+step,y+step),
color=(255,0,139),thickness=-1) # Рисуем фиолетовый прямоугольник
x=x+step*2
y=y+x//widht*step
x=x%widht
cv2.namedWindow(window_name, flags=cv2.WINDOW_AUTOSIZE) # Создание окна
cv2.imshow(window_name,img) # Вывод изображения на экран
if cv2.waitKey(5000)==121: # Если нажата "Y", то сохраняем
cv2.destroyWindow(window_name) # Удалить окно figure 2
cv2.imwrite(path,img) # Сохранить изображение

```

Результат работы программы:

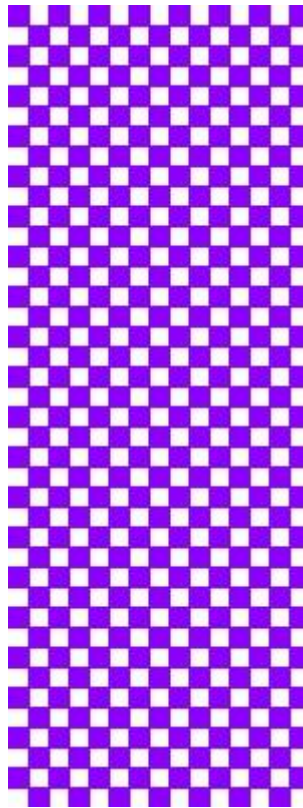


Рисунок 7 Результат работы программы

Вывод: в данной лабораторной работы были получены навыки работы с библиотеками OpenCV и NumPy для работы с изображениями. Была открыто изображение в разном масштабе, в цвете и в оттенках серого, а также с поменянными каналами, нарисована на белом фоне линия, квадрат и круг разных

цветов, которые потом были подписаны чёрным цветом. В конце работы была создана шахматная доска с фиолетовыми квадратами шириной в 15 квадратов и высотой в 40.