Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

**«КУБАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»**

**(ФГБОУ ВО «КубГУ»)**

**Факультет компьютерных технологий и прикладной математики**

**Кафедра вычислительных технологий**

**ЛАБОРАТОРНАЯ РАБОТА №1**

**Дисциплина: Алгоритмы цифровой обработки мультимедиа**

Работу выполнил: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_М.С. Прозоров

Направление подготовки: 02.03.02 Фундаментальная информатика и информационные технологии

Преподаватель: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_А.А. Крамаренко

Лабораторная работа №1

Тестирование вывода изображений и видео на экран, запись в файл, Формат

изображения HSV, определение цвета, построение надписей и доп

изображений на рисунке

**Задание 1**. Установить библиотеку OpenCV.

**Задание 2**. Вывести на экран изображение. Протестировать три

возможных расширения, три различных флага для создания окна и три

различных флага для чтения изображения.

Результат на рисунке 1.

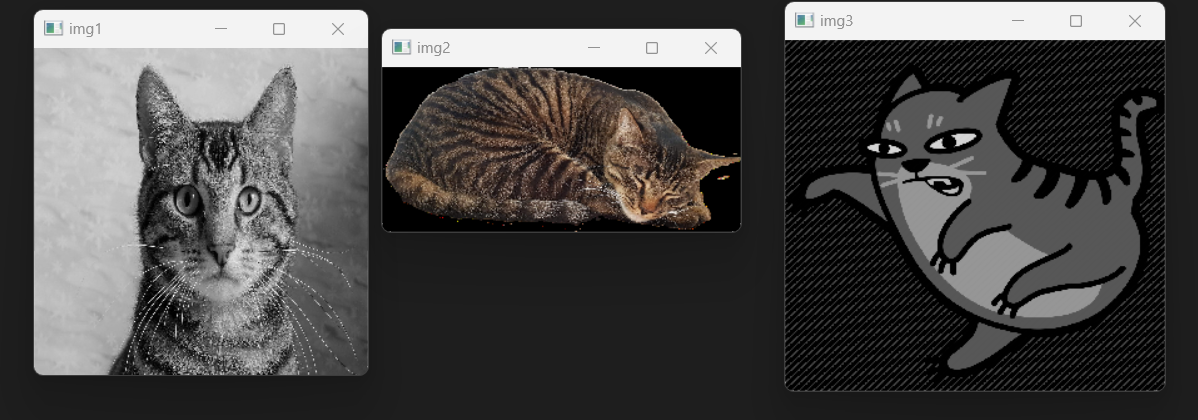


Рисунок 1 – Три изображения.

**Задание 3**. Отобразить видео в окне. Рассмотреть методы класса VideoCapture и попробовать отображать видео в разных форматах, в частности размеры и цветовая гамма

Результат на рисунке 2.

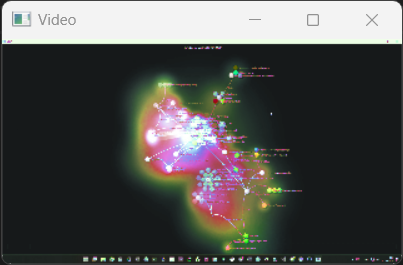


Рисунок 2 – Видео.

**Задание 4**. Записать видео из файла в другой файл.

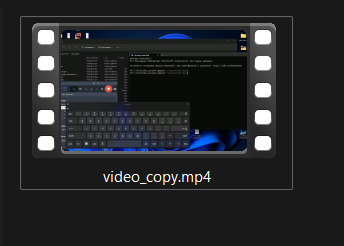


Рисунок 3 – Скопированное видео.

**Задание 5**. Прочитать изображение, перевести его в формат HSV. Вывести на экран два окна, в одном изображение в формате HSV, в другом – исходное изображение.

Результат на рисунке 4.

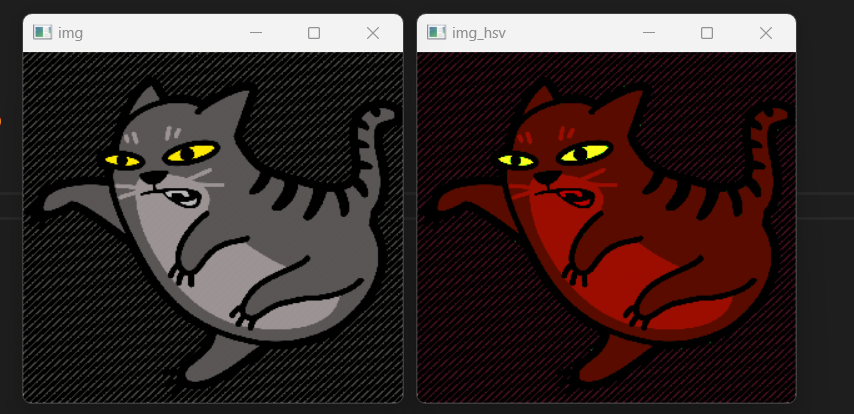


Рисунок 4 – Фото в формате HSV.

**Задание 6**. Прочитать изображение с камеры. Вывести в центре на экране Красный крест в формате, как на изображении. Указать команды, которые позволяют это сделать. Результат на рисунке 5.

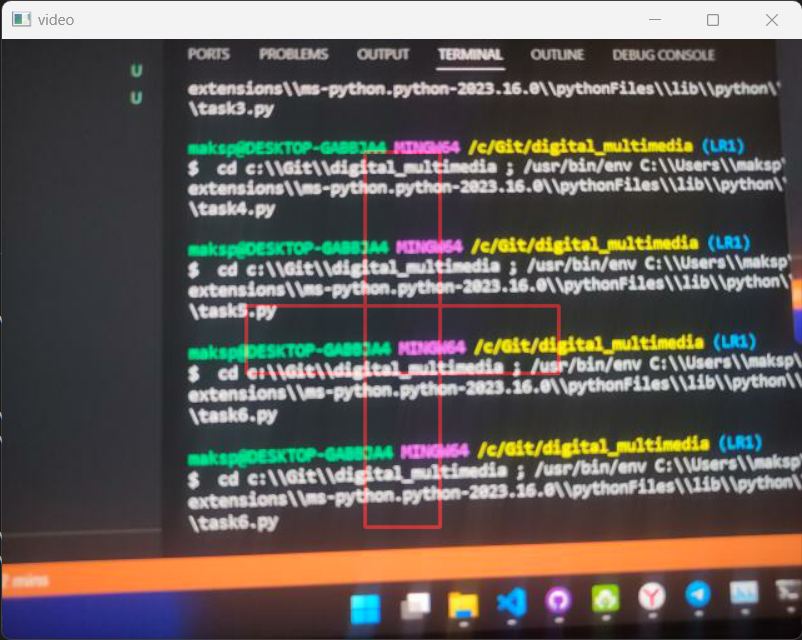


Рисунок 5 – Красный крест.

**Задание 7**. Отобразить информацию с вебкамеры, записать видео в файл, продемонстрировать видео. Результат на рисунке 6.

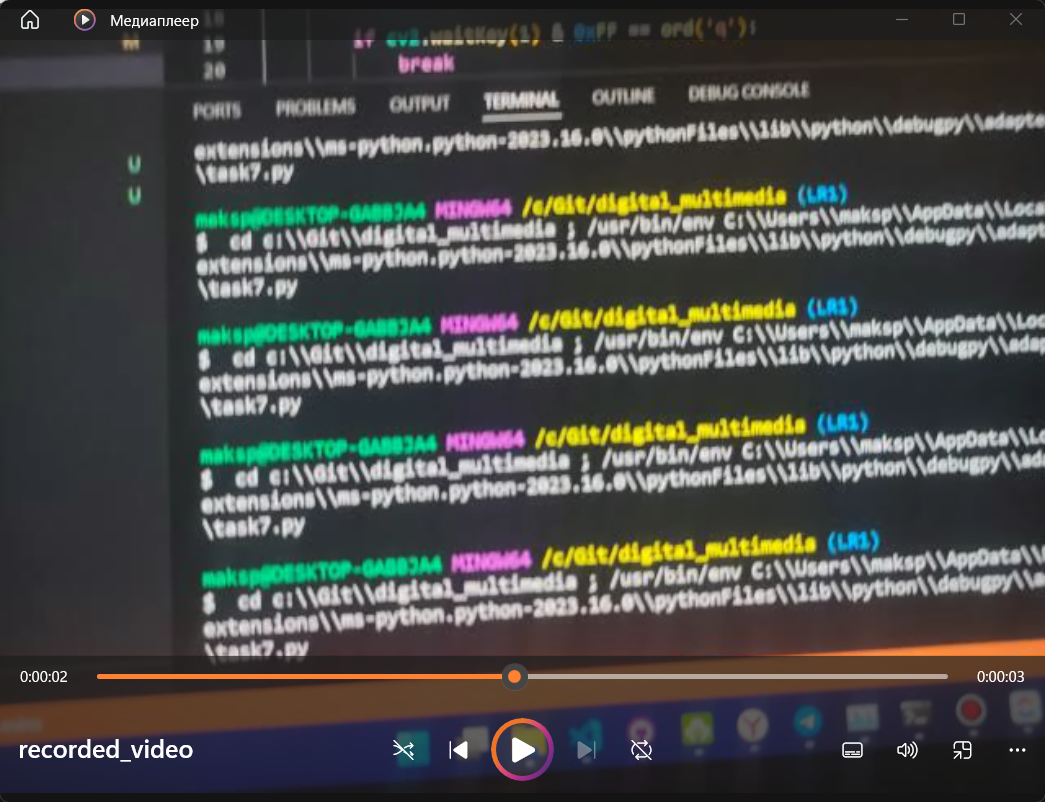


Рисунок 6 – Запись с Вебкамеры.

**Задание 8**. Залить крест одним из 3 цветов – красный, зеленый, синий по следующему правилу: НА ОСНОВАНИИ ФОРМАТА RGB определить, центральный пиксель ближе к какому из цветов красный, зеленый, синий и таким цветом заполнить крест.

Результат на рисунке 7.

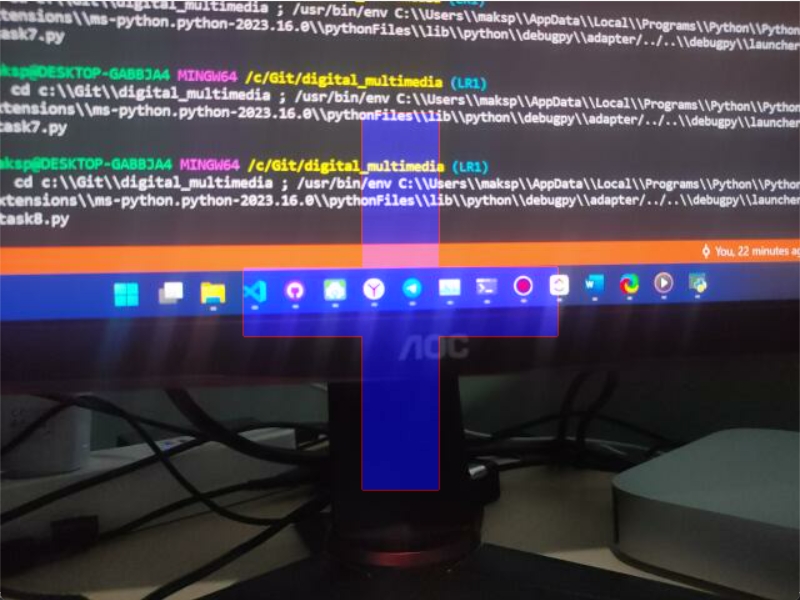


Рисунок 7 – Цветной крест.

**Задание 9**. Подключите телефон, подключитесь к его камере, выведете на экран видео с камеры.

Результат на рисунке 8.

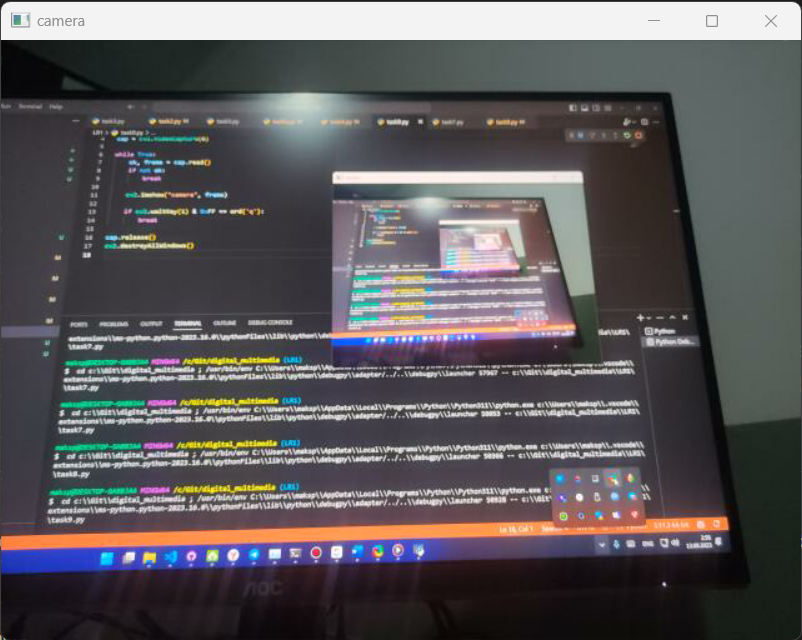


Рисунок 8 – Видео с телефона.

**Листинг программы**

**Файл digital\_multimedia\LR1\task0.py**

import cv2

def print\_cam():

cap = cv2.VideoCapture(0)

cap.set(3, 640)

cap.set(4, 480)

while True:

ret, frame = cap.read()

gray = cv2.cvtColor(frame, cv2.COLOR\_BGR2GRAY)

cv2.imshow('frame', gray)

if cv2.waitKey(1) & 0xFF == 27:

break

cap.release()

cv2.destroyAllWindows()

print\_cam()

**Файл digital\_multimedia\LR1\task2.py**

import cv2

img1 = cv2.imread(r'.\LR1\src\img1.jpeg',cv2.IMREAD\_GRAYSCALE)

img2 = cv2.imread(r'.\LR1\src\img2.png',cv2.IMREAD\_REDUCED\_COLOR\_8)

img3 = cv2.imread(r'.\LR1\src\img3.webp',cv2.IMREAD\_ANYDEPTH)

cv2.namedWindow('img1', cv2.WINDOW\_FREERATIO)

cv2.namedWindow('img2', cv2.WINDOW\_AUTOSIZE)

cv2.namedWindow('img3', cv2.WINDOW\_NORMAL)

cv2.imshow('img1', img1)

cv2.imshow('img2', img2)

cv2.imshow('img3', img3)

cv2.waitKey(0)

cv2.destroyAllWindows()

**Файл digital\_multimedia\LR1\task3.py**

import cv2

cap = cv2.VideoCapture(r'.\LR1\src\video.mp4', cv2.CAP\_ANY)

w = 320

h = 180

while True:

ok, frame = cap.read()

if not ok: break

frame = cv2.cvtColor(cv2.resize(frame, (w, h)), cv2.COLOR\_HLS2RGB)

cv2.imshow('Video', frame)

if cv2.waitKey(1) & 0xFF == 27:

break

**Файл digital\_multimedia\LR1\task4.py**

import cv2

def copyVideo():

video = cv2.VideoCapture(r'.\LR1\src\video.mp4', cv2.CAP\_ANY)

ok, vid = video.read()

w = int(video.get(cv2.CAP\_PROP\_FRAME\_WIDTH))

h = int(video.get(cv2.CAP\_PROP\_FRAME\_HEIGHT))

fourcc = cv2.VideoWriter\_fourcc(\*'mp4v') # type: ignore

video\_writer = cv2.VideoWriter("./LR1/Output/video\_copy.mp4", fourcc, 25, (w, h))

while (True):

ok, vid = video.read()

if not ok: break

# cv2.imshow('Video', vid)

video\_writer.write(vid)

if cv2.waitKey(1) & 0xFF == ord('q'):

break

cv2.destroyAllWindows()

copyVideo()

**Файл digital\_multimedia\LR1\task5.py**

import cv2

img1 = cv2.imread(r'.\LR1\src\img3.webp')

cv2.namedWindow('img', cv2.WINDOW\_NORMAL)

cv2.namedWindow('img\_hsv', cv2.WINDOW\_NORMAL)

cv2.imshow('img', img1)

hsv = cv2.cvtColor(img1, cv2.COLOR\_BGR2HSV)

cv2.imshow('img\_hsv', hsv)

cv2.waitKey(0)

cv2.destroyAllWindows()

**Файл digital\_multimedia\LR1\task6.py**

import cv2

import numpy as np

cap = cv2.VideoCapture(0)

while True:

ret, frame = cap.read()

if not ret:

break

height, width, \_ = frame.shape

cross\_image = np.zeros((height, width, 3), dtype=np.uint8)

vertical\_line\_width = 60

vertical\_line\_height = 300

cv2.rectangle(cross\_image,

(width // 2 - vertical\_line\_width // 2, height // 2 - vertical\_line\_height // 2),

(width // 2 + vertical\_line\_width // 2, height // 2 + vertical\_line\_height // 2),

(0, 0, 255), 2)

horizontal\_line\_width = 250

horizontal\_line\_height = 55

cv2.rectangle(cross\_image,

(width // 2 - horizontal\_line\_width // 2, height // 2 - horizontal\_line\_height // 2),

(width // 2 + horizontal\_line\_width // 2, height // 2 + horizontal\_line\_height // 2),

(0, 0, 255), 2)

result\_frame = cv2.addWeighted(frame, 1, cross\_image, 0.5, 0)

cv2.imshow("video", result\_frame)

if cv2.waitKey(1) & 0xFF == ord('q'):

break

cap.release()

cv2.destroyAllWindows()

**Файл digital\_multimedia\LR1\task7.py**

import cv2

def readIPWriteTOFile():

video = cv2.VideoCapture(0)

ok, vid = video.read()

w = int(video.get(cv2.CAP\_PROP\_FRAME\_WIDTH))

h = int(video.get(cv2.CAP\_PROP\_FRAME\_HEIGHT))

fourcc = cv2.VideoWriter\_fourcc(\*'mp4v') # type: ignore

video\_writer = cv2.VideoWriter("./LR1/Output/recorded\_video.mp4", fourcc, 25, (w, h))

while (True):

ok, vid = video.read()

cv2.imshow('Video', vid)

video\_writer.write(vid)

if cv2.waitKey(1) & 0xFF == ord('q'):

break

video.release()

video\_writer.release()

cv2.destroyAllWindows()

readIPWriteTOFile()

**Файл digital\_multimedia\LR1\task8.py**

import cv2

import numpy as np

cap = cv2.VideoCapture(0)

while True:

ret, frame = cap.read()

if not ret:

break

height, width, \_ = frame.shape

cross\_image = np.zeros((height, width, 3), dtype=np.uint8)

vertical\_line\_width = 60

vertical\_line\_height = 300

cv2.rectangle(cross\_image,

(width // 2 - vertical\_line\_width // 2, height // 2 - vertical\_line\_height // 2),

(width // 2 + vertical\_line\_width // 2, height // 2 + vertical\_line\_height // 2),

(0, 0, 255), 2)

rect\_start\_v = (width // 2 - vertical\_line\_width // 2, height // 2 - vertical\_line\_height // 2)

rect\_end\_v = (width // 2 + vertical\_line\_width // 2, height // 2 + vertical\_line\_height // 2)

horizontal\_line\_width = 250

horizontal\_line\_height = 55

cv2.rectangle(cross\_image,

(width // 2 - horizontal\_line\_width // 2, height // 2 - horizontal\_line\_height // 2),

(width // 2 + horizontal\_line\_width // 2, height // 2 + horizontal\_line\_height // 2),

(0, 0, 255), 2)

rect\_start\_h = (width // 2 - horizontal\_line\_width // 2, height // 2 - horizontal\_line\_height // 2)

rect\_end\_h = (width // 2 + horizontal\_line\_width // 2, height // 2 + horizontal\_line\_height // 2)

central\_pixel\_color = frame[height // 2, width // 2]

color\_distances = [

np.linalg.norm(central\_pixel\_color - np.array([0, 0, 255])),

np.linalg.norm(central\_pixel\_color - np.array([0, 255, 0])),

np.linalg.norm(central\_pixel\_color - np.array([255, 0, 0]))

]

closest\_color\_index = np.argmin(color\_distances)

if closest\_color\_index == 0:

cv2.rectangle(cross\_image, rect\_start\_h, rect\_end\_h, (0, 0, 255), -1)

elif closest\_color\_index == 1:

cv2.rectangle(cross\_image, rect\_start\_h, rect\_end\_h, (0, 255, 0), -1)

else:

cv2.rectangle(cross\_image, rect\_start\_h, rect\_end\_h, (255, 0, 0), -1)

if closest\_color\_index == 0:

cv2.rectangle(cross\_image, rect\_start\_v, rect\_end\_v, (0, 0, 255), -1)

elif closest\_color\_index == 1:

cv2.rectangle(cross\_image, rect\_start\_v, rect\_end\_v, (0, 255, 0), -1)

else:

cv2.rectangle(cross\_image, rect\_start\_v, rect\_end\_v, (255, 0, 0), -1)

result\_frame = cv2.addWeighted(frame, 1, cross\_image, 0.5, 0)

cv2.imshow("video", result\_frame)

if cv2.waitKey(1) & 0xFF == ord('q'):

break

cap.release()

cv2.destroyAllWindows()

**Файл digital\_multimedia\LR1\task9.py**

import cv2

import numpy as np

cap = cv2.VideoCapture(0)

while True:

ok, frame = cap.read()

if not ok:

break

cv2.imshow("camera", frame)

if cv2.waitKey(1) & 0xFF == ord('q'):

break

cap.release()

cv2.destroyAllWindows()