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: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_А.М.Нагалевский

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.

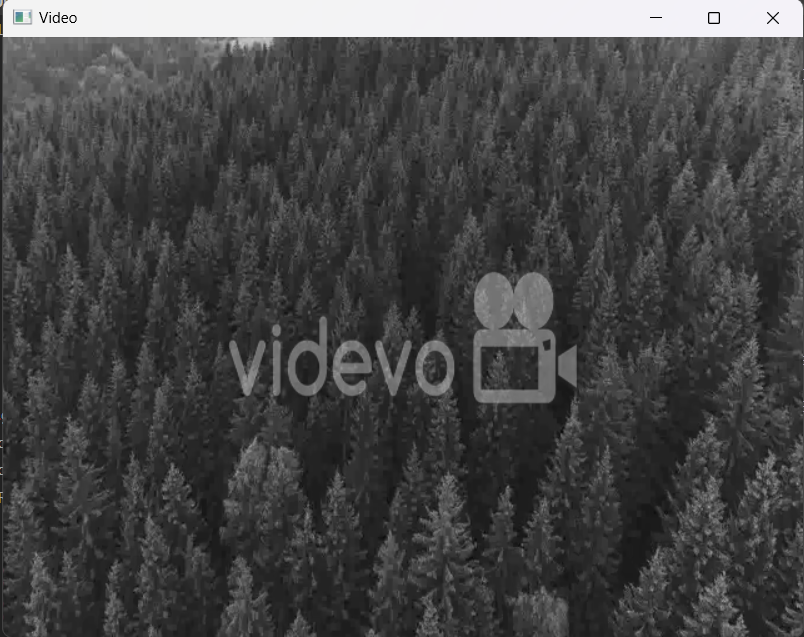


Рисунок 2 – Отображения видео.

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

**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 3.



Рисунок 3 – 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 4.

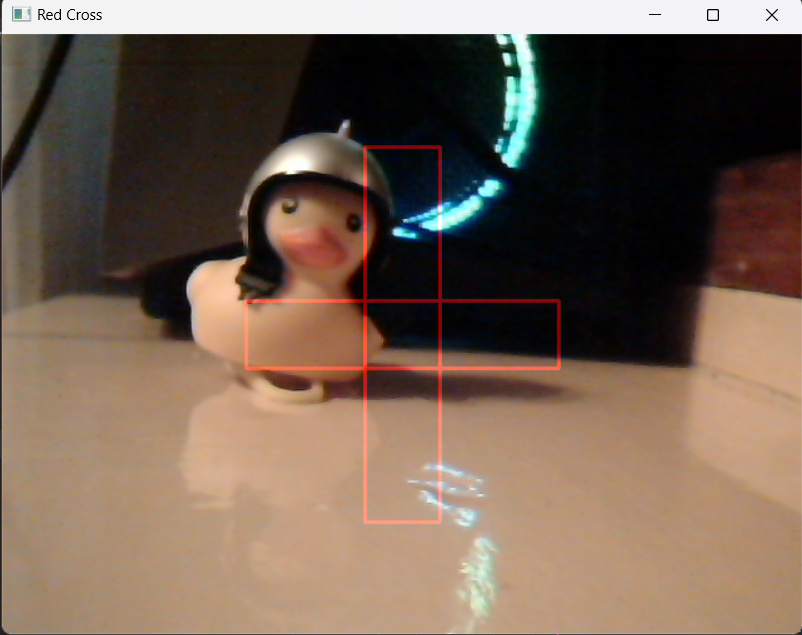


Рисунок 4 – Красный крест.

**Задание 7**. Отобразить информацию с вебкамеры, записать видео в файл, продемонстрировать видео. Результат на рисунке 5.

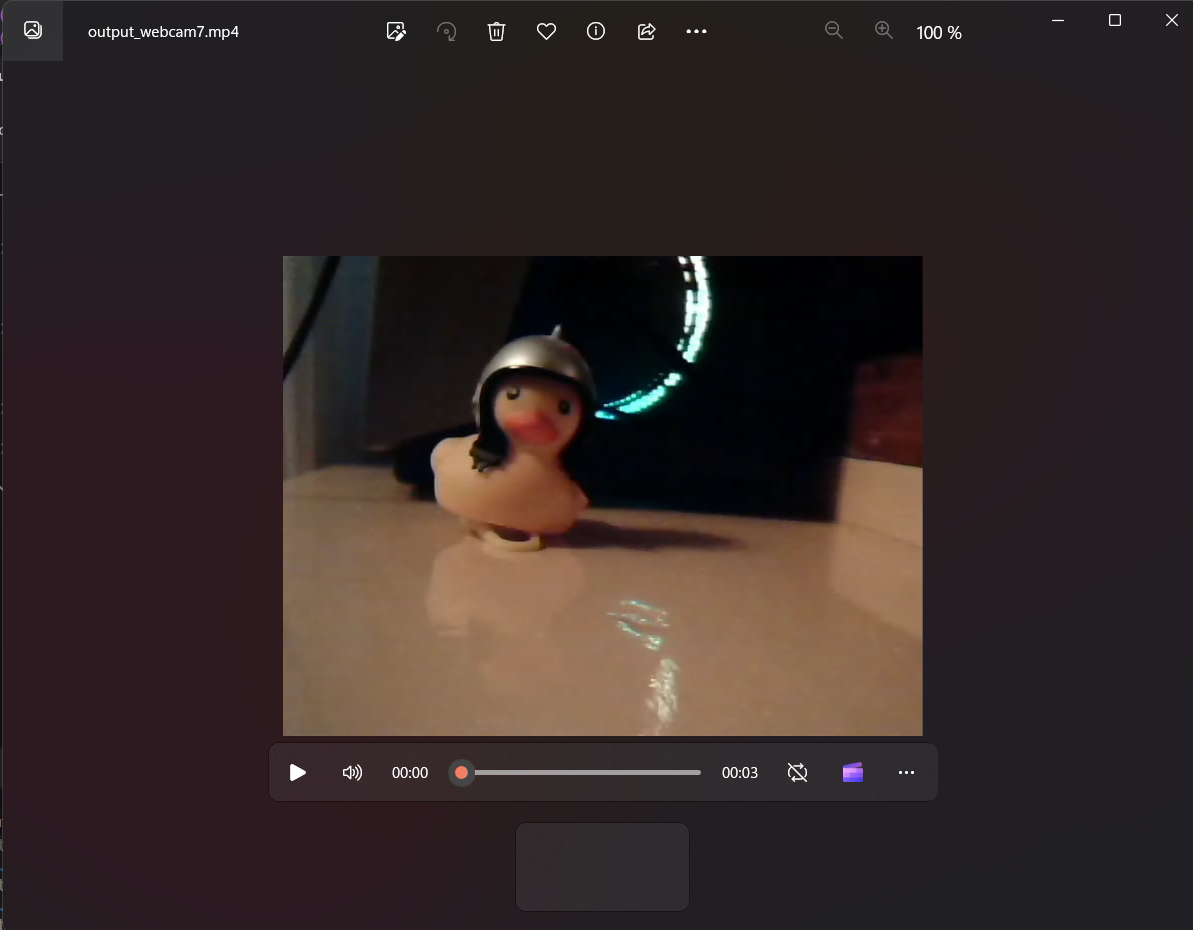


Рисунок 7 – Запись с Вебкамеры.

**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 8.

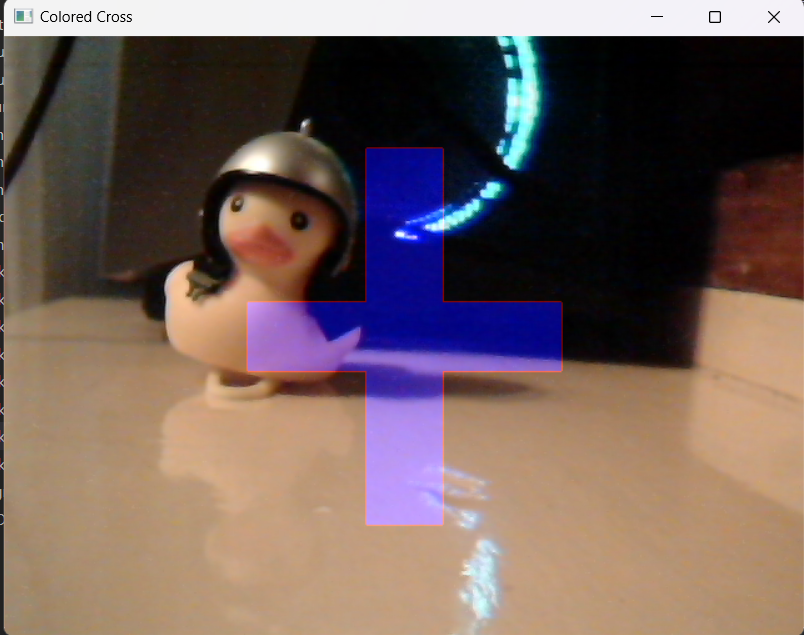


Рисунок 8 – Цветной крест.

**Задание 9**. Подключите телефон, подключитесь к его камере, выведете на экран видео с камеры. Продемонстрировать процесс на ноутбуке преподавателя и своем телефоне.

Результат на рисунке 9.

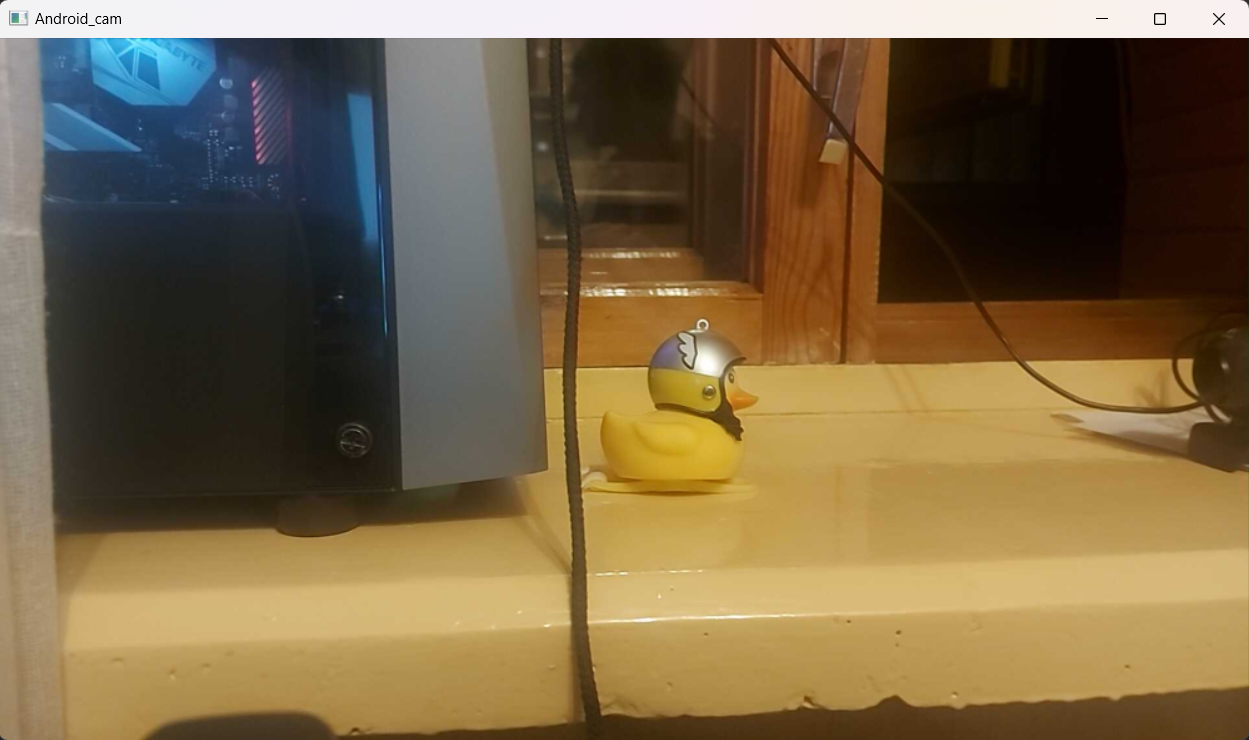


Рисунок 9 – Видео с телефона.

**Листинг программы**

Файл task2.py:

import cv2

img1 = cv2.imread(r'.\LR1\source\img1.jpg',cv2.IMREAD\_GRAYSCALE)

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

img3 = cv2.imread(r'.\LR1\source\img3.bmp',cv2.IMREAD\_ANYDEPTH)

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

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

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

cv2.imshow('gosling',img1)

cv2.imshow('ghost', img2)

cv2.imshow('nature', img3)

cv2.waitKey(0)

cv2.destroyAllWindows()

Файл task3.py:

import cv2

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

new\_width = 640

new\_height = 480

while **True**:

    ret, frame = cap.read()

    if not ret:

        exit()

    frame = cv2.resize(frame, (new\_width, new\_height))

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

    cv2.imshow('Video', gray\_frame)

    if cv2.waitKey(1) & 0xFF == 27:

        exit()

Файл task4.py:

import cv2

def readIPWriteTOFile():

    video = cv2.VideoCapture(r'.\LR1\source\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')

    video\_writer = cv2.VideoWriter("./LR1/Output/output3.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 task5.py:

import cv2

img1 = cv2.imread(r'.\LR1\source\img2.png')

img2 = cv2.imread(r'.\LR1\source\img2.png')

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

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

cv2.imshow('ghost',img1)

hsv = cv2.cvtColor(img2, cv2.COLOR\_BGR2HSV)

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

cv2.waitKey(0)

cv2.destroyAllWindows()

Файл 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("Red Cross", result\_frame)

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

        break

cap.release()

cv2.destroyAllWindows()

Файл 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')

    video\_writer = cv2.VideoWriter("./LR1/Output/output\_webcam7.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 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("Colored Cross", result\_frame)

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

        break

cap.release()

cv2.destroyAllWindows()

Файл task9.py:

import requests

import cv2

import numpy as np

import imutils

url = "http://192.168.0.136:8080/shot.jpg"

while **True**:

    img\_resp = requests.get(url)

    img\_arr = np.array(bytearray(img\_resp.content), *dtype*=np.uint8)

    img = cv2.imdecode(img\_arr, -1)

    img = imutils.resize(img, *width*=1000, *height*=1800)

    cv2.imshow("Android\_cam", img)

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

        break

cv2.destroyAllWindows()