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**

**Отчёт по лабораторной №1**

**Дисциплина: АЛГОРИТМЫ ЦИФРОВОЙ ОБРАБОТКИ МУЛЬТИМЕДИА**

Работу выполнил \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Дука В.А.

Направление подготовки 02.03.02 Фундаментальная информатика и

информационные технологии

Старший преподаватель \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Крамаренко А.А.

Краснодар

2025

**Тема:** Тестирование вывода изображений и видео на экран, запись в файл, Формат изображения HSV, определение цвета, построение надписей и доп. изображений на рисунке.

**Постановка задачи:**

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.

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

*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.

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

*Задание 7* (самостоятельно). Отобразить информацию с вебкамеры, записать видео в файл, продемонстрировать видео.

*Задание 8* (самостоятельно). 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.

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

**Ход работы:**

**Задание 1.**

Установка библиотеки: import cv2 as cv

**Результат вывода изображения:**

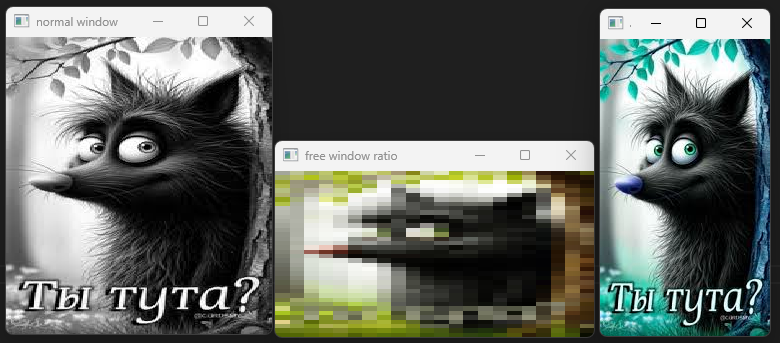
Изображение выглядит как текст, млекопитающее, Мелкие и средние кошки, снимок экрана

Содержимое, созданное искусственным интеллектом, может быть неверным.

**Задание 2.**

Работа с методом *imread -* читает файл с диска, декодирует его в массив пикселей и возвращает OpenCV массив

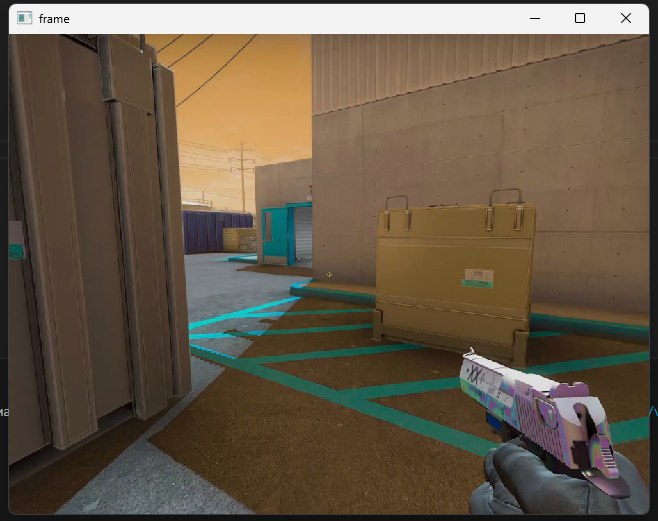
**Результат:**

****

**Задание 3.**

Работа с *VideoCapture -* высокоуровневая оболочка OpenCV для получения видеокадров из **источника**: файла (mp4, mkv и т.п.), веб-камеры IP-потока.

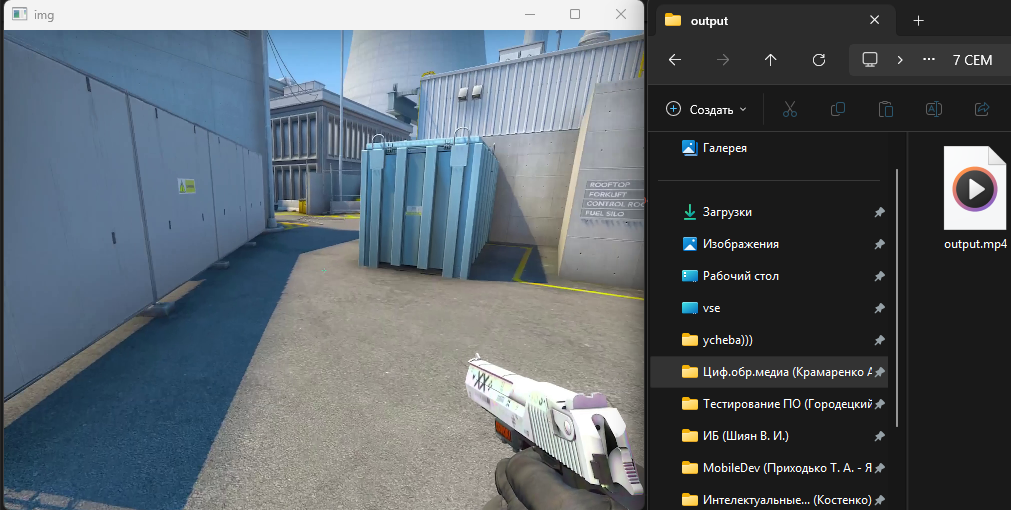
**Результат:**

****

**Задание 4.**

Запись из одного видеофайла в другой с помощью *VideoWriter* – создаёт соответствующий кодек с параметрами (размер кадра, fps), затем принимает frame и конвертирует его в нужный формат цвета (например из BGR в YUV), кодек преобразует массив фреймов в сжатый пакет и передаёт все в контейнер (MP4/AVI/MKV), который сохраняет файл с соответствующим расширением

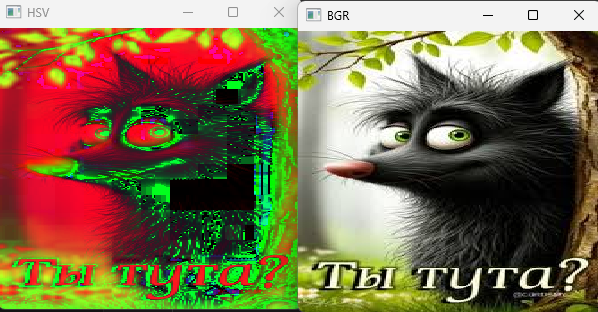
**Результат** – файл output.mp4:



**Задание 5.**

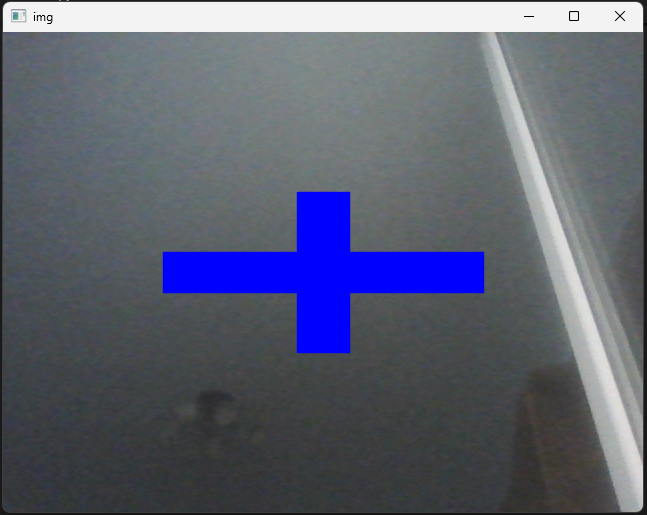
Сравнение стандартного BGR и HSV с помощью cvtColor() - преобразует изображение из одного цветового пространства в другое - сначала нормируют компоненты, затем используют стандартные формулы для Hue/Saturation/Value

**Результат:**

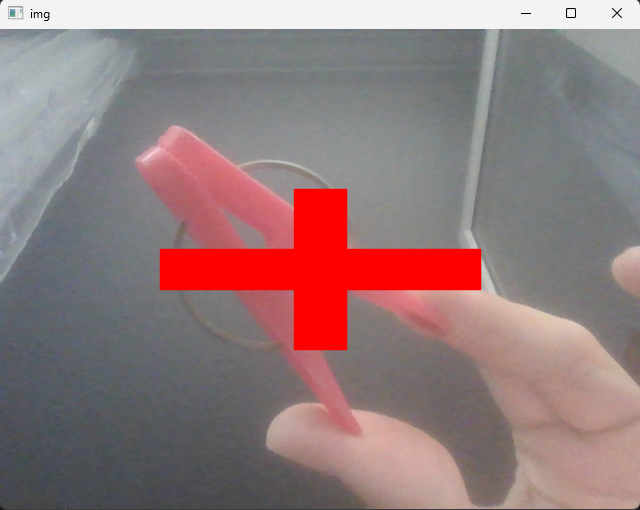


**Задание 6-8.**

Получение доступа к кадрам из веб-камеры с помощью CAP\_DSHOW, запись видео из этого потока в файл и отрисовка креста, принимающего цвет из RGB наиболее близкий к центральному пикселю:



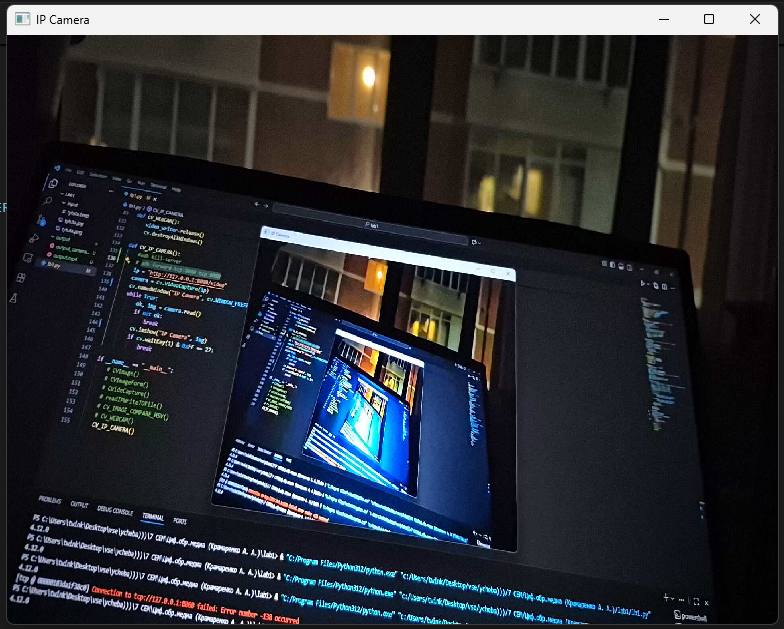




**Задание 9.**

Демонстрация видео с помощью IP Camera приложения с телефона на ноутбук – считывание видеопотока с URL локальной машины.

**Результат работы:**



**Вывод:** в ходе работы удалось освоить базовые навыки с методами работы с медиа потоками с помощью библиотеки OpenCV и её методов.

Контрольные вопросы:

1. Формат представления изображений в OpenCV.  
В OpenCV изображение хранится как прямоугольная таблица пикселей: по строкам и столбцам, и в каждой ячейке лежат значения каналов цвета. Проще: изображение — это обычный трёхмерный массив чисел, где первое измерение — количество строк (высота), второе — количество столбцов (ширина), а третье — значения цвета для каждого пикселя (обычно три числа — синий, зелёный, красный). Каждый числовой канал обычно представлен целым числом от 0 до 255. По умолчанию порядок каналов в OpenCV — сначала синий, потом зелёный, потом красный, а не привычный RGB; это важно помнить при обработке.

2. Принцип отображения окон в OpenCV.  
Когда вы вызываете команду, которая показывает изображение в окне imshow, OpenCV помещает изображение в специальный буфер для окна, а фактическая отрисовка и обработка событий (клавиш, мыши, перерисовки) происходит во время вызова функции ожидания событий. Проще: imshow только отправляет картинку в окно, а waitKey даёт время программе обработать окно и показать картинку на экране; без waitKey окно либо не отрисуется, либо сразу закроется.

3. Управление параметрами отображения окон в OpenCV.  
Сначала нужно создать окно с именем и параметрами, затем можно менять его поведение: запретить изменение размера окна, разрешить изменять размер, установить полноэкранный режим, переместить окно на экран или изменить его размер программно. Также можно добавить полосы прокрутки (ползунки) и обработчик мыши, чтобы реагировать на клики и движения в окне. Все эти действия выполняются специальными командами, которые нужно вызывать перед показом или во время работы с окном.

4. Какой класс отвечает за работу с видеопотоком.  
За получение изображений из видео или с камеры в OpenCV отвечает класс, доступный в Python как VideoCapture. Этот класс открывает источник (файл, камеру или сетевой поток), управляет соединением и предоставляет кадры по запросу.

5. Откуда возможно получение видеопотока.  
Видеопоток можно брать из локальной веб-камеры ноутбука, из файла с диска (например, .mp4 или .avi), из сетевого потока с телефона или IP-камеры (по протоколам HTTP/RTSP), или из виртуальной камеры (специальный драйвер, который делает из телефона «камеру» для системы). Главное, что OpenCV видит — это какой-то источник, откуда можно последовательно читать кадры.

6. Общие принципы работы с изображениями в потоке. Понятие frame и принцип работы метода read().  
Frame (кадр) — это одно изображение из потока, то есть один «снимок» времени. Метод read() берёт следующий кадр из источника и возвращает две вещи: флаг успешности (удалось ли получить кадр) и сам кадр (изображение). Под капотом обычно выполняются две операции: сначала программа захватывает (получает) следующий кусок данных из потока, а затем декодирует этот кусок в формат изображения, пригодный для показа или обработки. Если вы используете камеру в реальном времени, то драйвер может накапливать или сбрасывать кадры в буфер, поэтому иногда полезно отдельно «схватить» кадр и отдельно «извлечь» его для обработки, чтобы не задерживать поток.

7. Что такое FourCC и зачем применяется.  
FourCC — это короткий код из четырёх символов, который идентифицирует видеокодек, то есть алгоритм сжатия/кодирования видео. Этот код нужен, чтобы при записи видео в файл выбрать, каким именно алгоритмом нужно сжимать кадры. Контейнер файла (например .mp4 или .avi) затем хранит сжатые кадры и служебную информацию о том, какой codec использовался.

8. Основные особенности класса VideoWriter.  
Класс VideoWriter используется для записи видеопотока в файл. При создании нужно указать имя файла, кодек (FourCC), частоту кадров, размер кадра и цветность. Когда вы передаёте кадр в запись, библиотека конвертирует изображение в формат, который требует кодек (например в формат яркости+хрома), сжимает его и записывает в файл вместе с меткой времени. Важно, чтобы размер кадра, который вы записываете, совпадал с указанным при создании VideoWriter, иначе запись будет не работать.

9. Что означает проверка if (cv2.waitKey(1) & 0xFF == 27).  
Функция, которая ждёт клавиатуру, возвращает числовой код нажтой клавиши. Оператор & 0xFF берёт только младший байт этого числа, потому что в разных системах старшие биты могут содержать дополнительную информацию. Число 27 — это ASCII-код клавиши Escape. Таким образом, эта проверка означает: «если пользователь нажал клавишу Escape, то выполнить выход».

10. Почему применяется формат HSV, значения параметров, формулы перевода из HSV в RGB и обратно, и геометрический смысл преобразований.  
Формат HSV разделяет цвет на три простые характеристики, которые удобнее использовать для задач обработки, чем простые три канала RGB. Первая характеристика Hue (оттенок) говорит, какой это цвет: красный, зелёный, синий и т.п.; в HSV это угол на цветовом круге. Вторая характеристика Saturation (насыщенность) показывает, насколько «чистый» цвет — близок ли он к яркому цвету или ближе к серому. Третья характеристика Value (значение, светлота) показывает, насколько цвет светлый или тёмный. В OpenCV Hue обычно кодируется числом от 0 до 179, а Saturation и Value — от 0 до 255, потому что значения упакованы в один байт для эффективности. Для перевода из RGB в HSV сперва нормируют компоненты красного, зелёного и синего в диапазон 0–1, затем вычисляют максимум и минимум этих трёх значений, их разницу и по этим величинам получают Value и Saturation; Hue вычисляется как угол, зависящий от того, какой канал был максимальным, и от отношений между каналами. Обратное преобразование берёт угол Hue, рассчитывает соответствующие по формуле промежуточные значения и восстанавливает красный/зелёный/синий. Геометрически представление HSV похоже на цилиндр или конус: вокруг оси идёт оттенок (угол), от центра к краю радиально растёт насыщенность (чем дальше от центра, тем «чище» цвет), а по вертикали идёт светлота. Это представление удобно, когда нужно, например, выделить все пиксели примерно красного оттенка независимо от того, светлые они или тёмные.