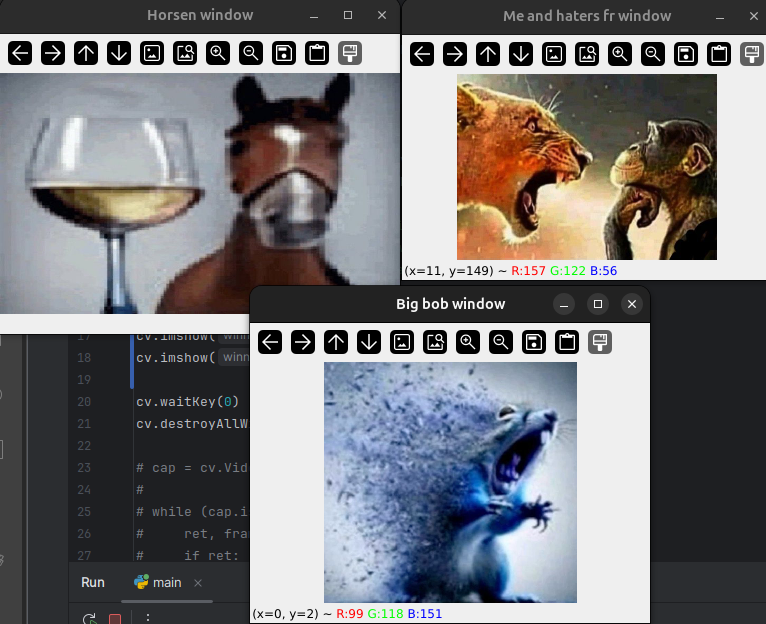
**Лабораторная работа №1.**

**Иванченко Павла ФИ 41/1**

1. Установим библиотеку opencv и импортируем ее в проект:
2. Загрузим изображения разных форматов, использовав метод imread. Для управления параметрами цвета изображения используем флаги IMREAD\_GRAYSCALE, IMREAD\_COLOR\_RGB, IMREAD\_REDUCED\_COLOR\_2.

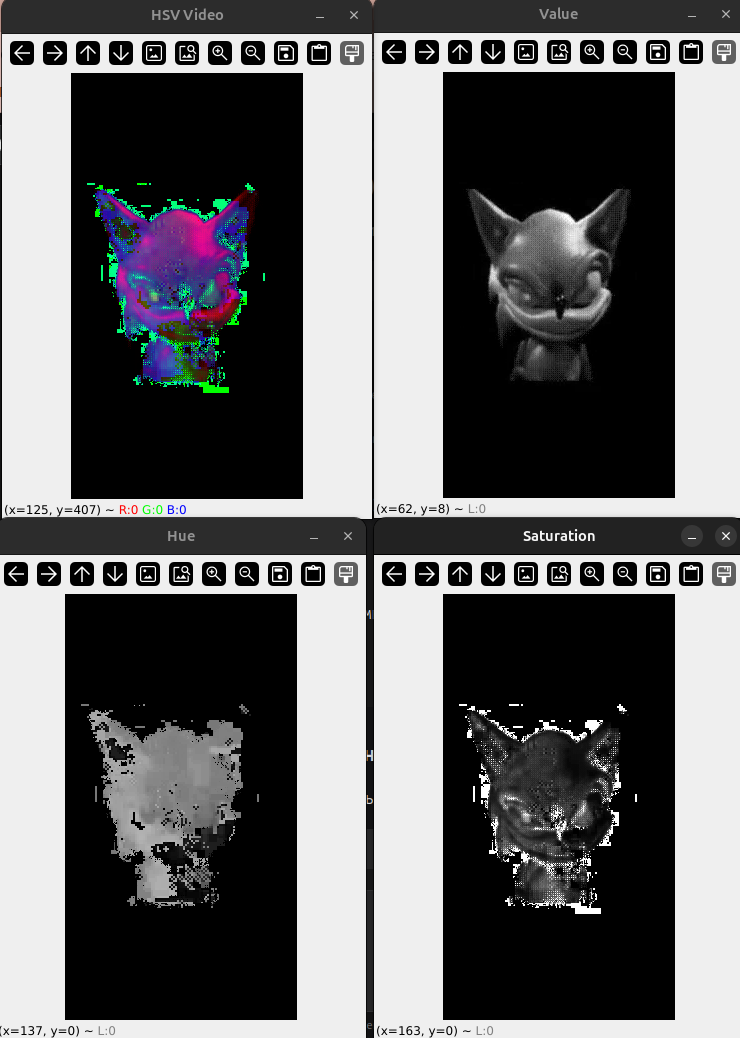
Для управления форматом отображения настроим окно вывода. Для этого создадим отдельные окна с помощью команды namedWindow и выставим им флаги WINDOW\_FREERATIO, WINDOW\_KEEPRATIO и WINDOW\_AUTOSIZE.

Выведем изображения в окнах на экран, использовав команду imshow. Добавим также waitKey и destroyAllWindows, чтобы контролировать отображение окон.

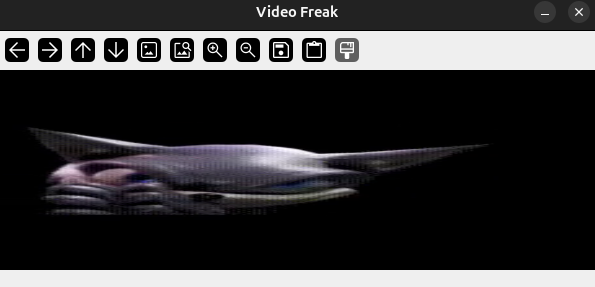


1. Для того, чтобы отобразить видео, создадим экземпляр класса VideoCapture. С помощью цикла и метода isOpened запустим видео, которое будет проигрываться заново каждый раз, пока мы не нажмем установленный waitKey. Цикл осуществляется с помощью метода класса s et(), с помощью которого видео перематывается в начало.

С помощью метода cvtColor преобразуем изображения из стандартного формата BGR в HSV – оттенок, насыщенность, яркость. После этого, можно разделить видео на отдельные потоки HSV.

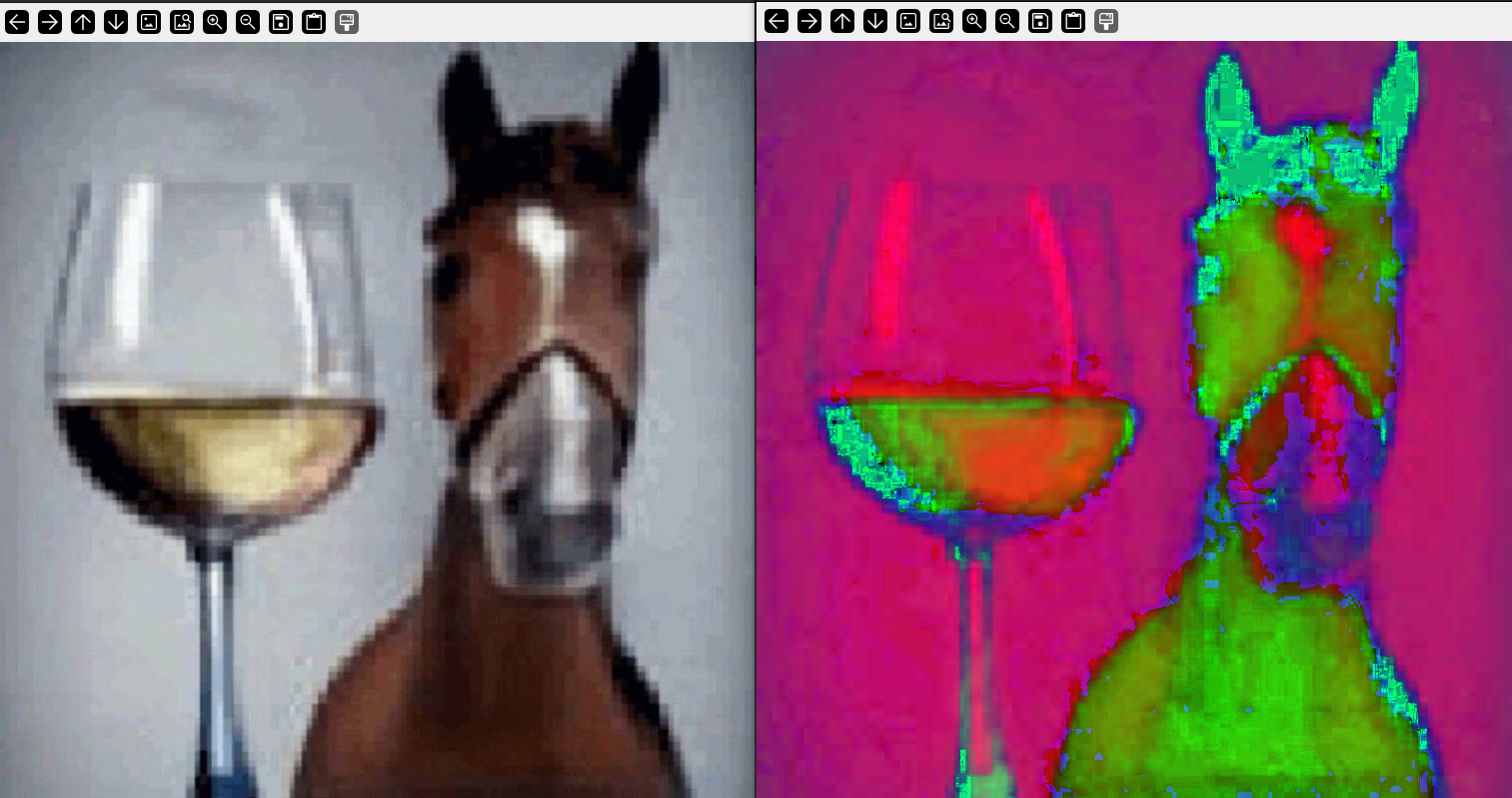


Также можем поменять размер изображения с помощью cv.resize().



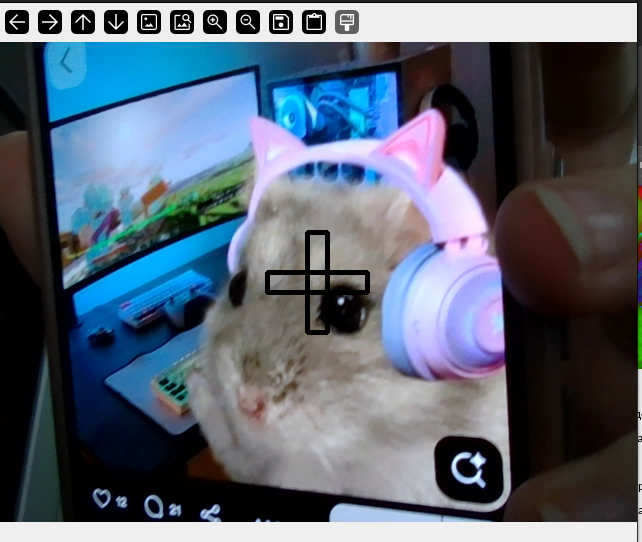
4. Запишем видео из файла в другой файл. Для этого из экземпляра класса VideoCapture, который был создан ранее, прочтем fps, ширину и высоту кадров, а также создадим экземпляр класса VideoWriter. После этого в цикле обработаем каждый кадр видео, записав в выходной файл.

5. Ранее прочитанное изображение img1 переведем в hsv формат с помощью cvtColor и флага BGR2HSV.

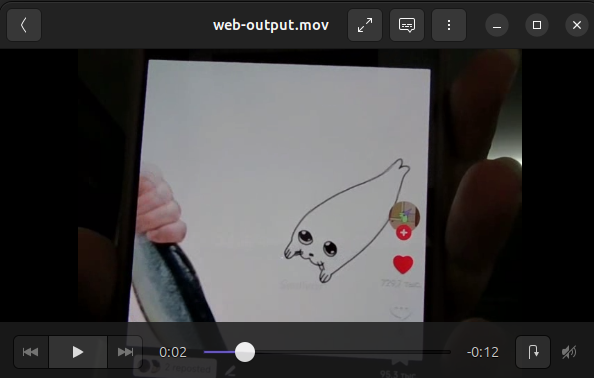


6. Процесс чтения изображения с камеры аналогичен процессу чтения видео из файла, однако при создании экземпляра VideoCapture необходимо передать не название файла, а код камеры (0 для устройства по умолчанию).

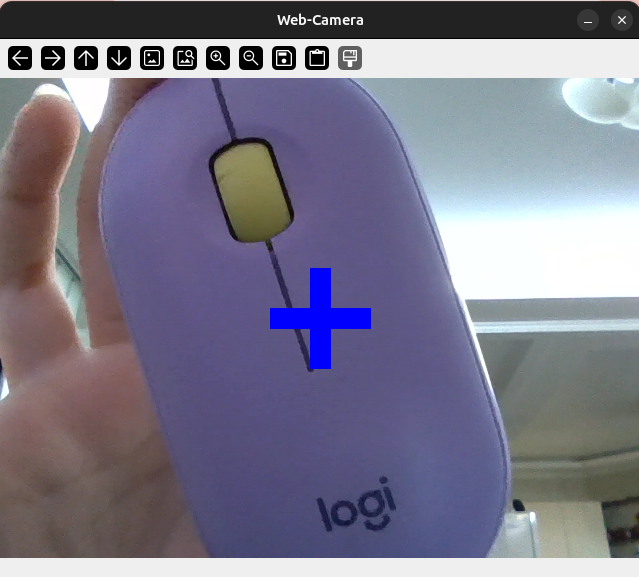
Для того, чтобы вывести красный крест по центру кадра, необходимо построить два прямоугольника. Получим ширину и высоту кадра, на основе этих данных зададим левую верхнюю и правую нижнюю точки, по которым и построим фигуру командой cv.rectangle. Проделаем это дважды для горизонтального и вертикального прямоугольников.



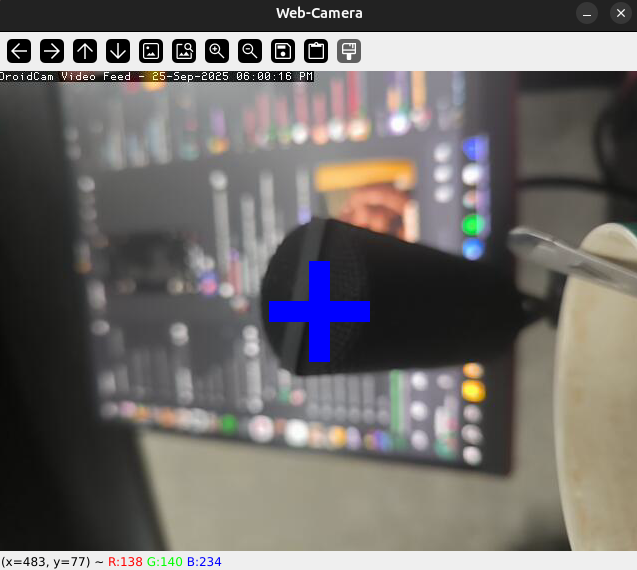
7. Для видеопотока с камеры повторим шаги из пункта 4, результат запишем в файл web-output.mov.



8. Определим цвет центрального пикселя кадра (по умолчанию в формате BGR), разбив его на 3 части – b, g и r. Чем больше значение какой-либо из частей, тем сильнее в этом пикселе преобладает этот цвет. Сравним значения каждой из частей и на основе сравнения выберем цвет креста. Чтобы залить крест цветом, в параметре толщины укажем значение «-1».



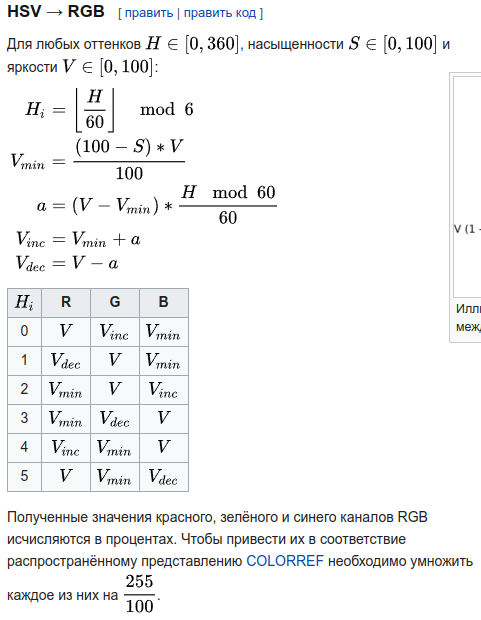
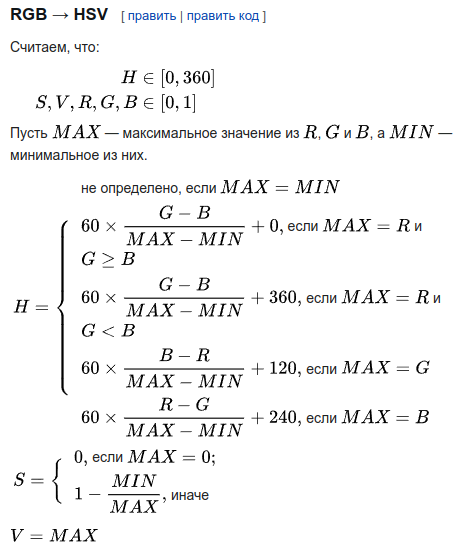
9. Для того, чтобы считать видеопоток с камеры телефона, необходимо подключиться к камере по IP. IP необходимо передать в экземпляр класса VideoCapture.



Ответы на вопросы:

1. Изображения в OpenCV хранятся в матрицах, причем тип изображения определяется содержанием изображения, а не его расширением. Если изображение не может быть считано, возвращается пустая матрица.
2. Окна создаются отдельно для каждого изображения, причем, если не стоит никаких флагов, изображение подстраивается под размер окна.
3. Создавать и управлять флагами окон можно с помощью cv.namedWindow(), изменять размер с помощью resizeWindow(), изменять положение с помощью moveWindow(). Основные флаги: WINDOW\_NORMAL, WINDOW\_AUTOSIZE, WINDOW\_FREERATIO, WINDOW\_KEEPRATIO, WINDOW\_GUI\_NORMAL, WINDOW\_GUI\_EXPANDED.
4. Класс VideoCapture().
5. Из пути к файлу, индекса веб-камеры, по url потока.
6. Видео обрабатывается покадрово, каждый кадр записывается/изменяется, если он успешно прочитан. Используя функцию read(), мы берем кадры, начая с начала, и задаем 2 переменные: ret – булевая переменная, прочитан ли кадр успешно, frame – кадр как изображение, записанное в виде массива.
7. fourcc – Four Character Code - 4-символьный код видеокодека, который определяет формат сжатия видео (‘XVID’, ‘MJPG’, ‘H264’).
8. Класс VideoWriter является API для записи видео файлов и последовательностей изображений. В конструктор экземпляра обычно передаются параметры: название файла вывода, видеокодек, фпс (количество кадров в секунду), размер кадра, цветное ли изображение.
9. cv2.waitKey(1) ожидает нажатия клавиши, а 0xFF == 27 проверяет, что нажата клавиша Esc.
10. HSV формат нужен для альтернативного представления изображения, которое опирается не на количество красного, синего и зеленого в изображении, а на оттенок, насыщенность и яркость. HSV представляет цвета в цилиндрической системе координат: H – угол на цветовом круге, S – радиус, V – яркость.

Формулы преобразования:



Преобразование из RGB в HSV – это афинное преобразование.