МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

**«КУБАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»**

**(ФГБОУ ВО «КУБГУ»)**

**Факультет компьютерных технологий и прикладной математики**

**Кафедра вычислительных технологий**

**Отчет**

**по лабораторной работе №1 по курсу**

**«АЛГОРИТМЫ ЦИФРОВОЙ ОБРАБОТКИ МУЛЬТИМЕДИА»**

Работу выполнил

Студент 49 группы

Иванова В. А

Преподаватель:

Крамаренко А. А.

Краснодар

2024

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

**Ход работы:**

**Задание №1**. Вывести на экран изображение. Протестировать три возможных расширения, три различных флага для создания окна и три различных флага для чтения изображения.

Для тестирования используем jpg-изображение ‘sleepy\_cat’, которое сперва выведем на экран.

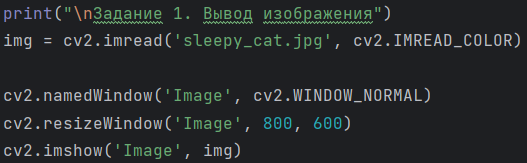


Рисунок 1 – код для вывода изображения

В img записывается изображение с использованием флага IMREAD\_COLOR. Далее создается новое окно для отображения (‘Image’), устанавливается размер и через imshow отображается добавленное фото в окне.

Для тестирования расширений и флагов будут использованы:

* IMREAD\_COLOR (загрузка цветного изображения);
* IMREAD\_GRAYSCALE (загрузка черно-белого изображения);
* IMREAD\_UNCHANGED (загрузка изображения без изменений формата).

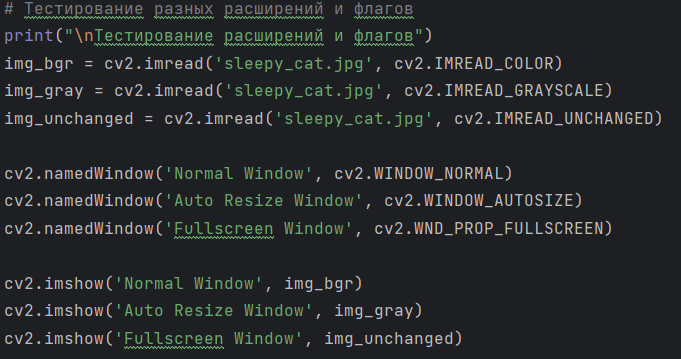


Рисунок 2 – загрузка изображения с флагами и их отображение

В namedWindow используем три варианта для формата вывода изображения, которые продемонстрированы ниже.

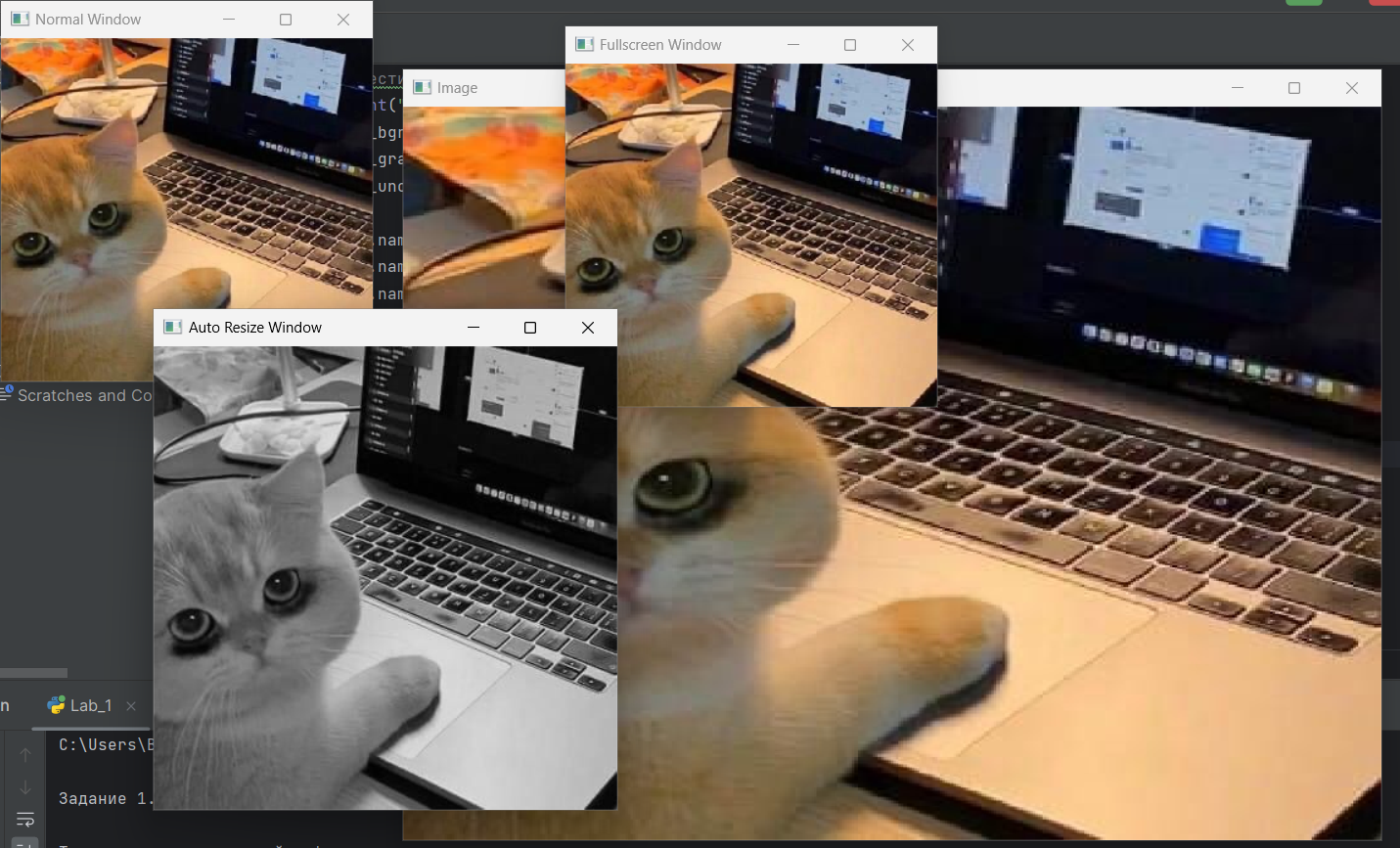


Рисунок 3 – результат работы программы

**Задание №2**. Отобразить видео в окне. Рассмотреть методы класса VideoCapture и попробовать отображать видео в разных форматах, в частности размеры и цветовая гамма.

В cap через VideoCapture читаем mp4-файл ‘Seoul\_street’.



Рисунок 4 – кадр из используемого файла

Для вывода файла будем использовать цикл, который работает до тех пор, пока видео не закончится. cap.read() читает следующие кадры, а ret, если равняется False, обозначает конец, после чего показ завершается.

В resize\_frame с помощью resize масштабируем до нужного нам размера, а в gray\_frame через cvtColor, используя флажок BGR2GRAY записываем видео в черно-белом формате. Выводим оба видео одновременно и, если нажата клавиша «e», программа завершается.

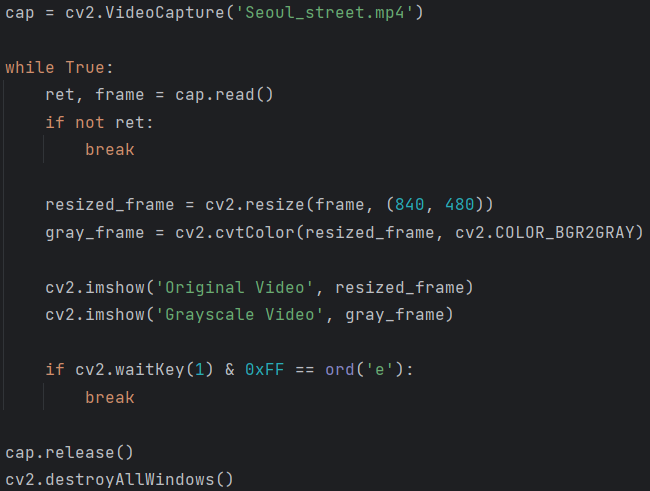


Рисунок 5 – код для работы с видео

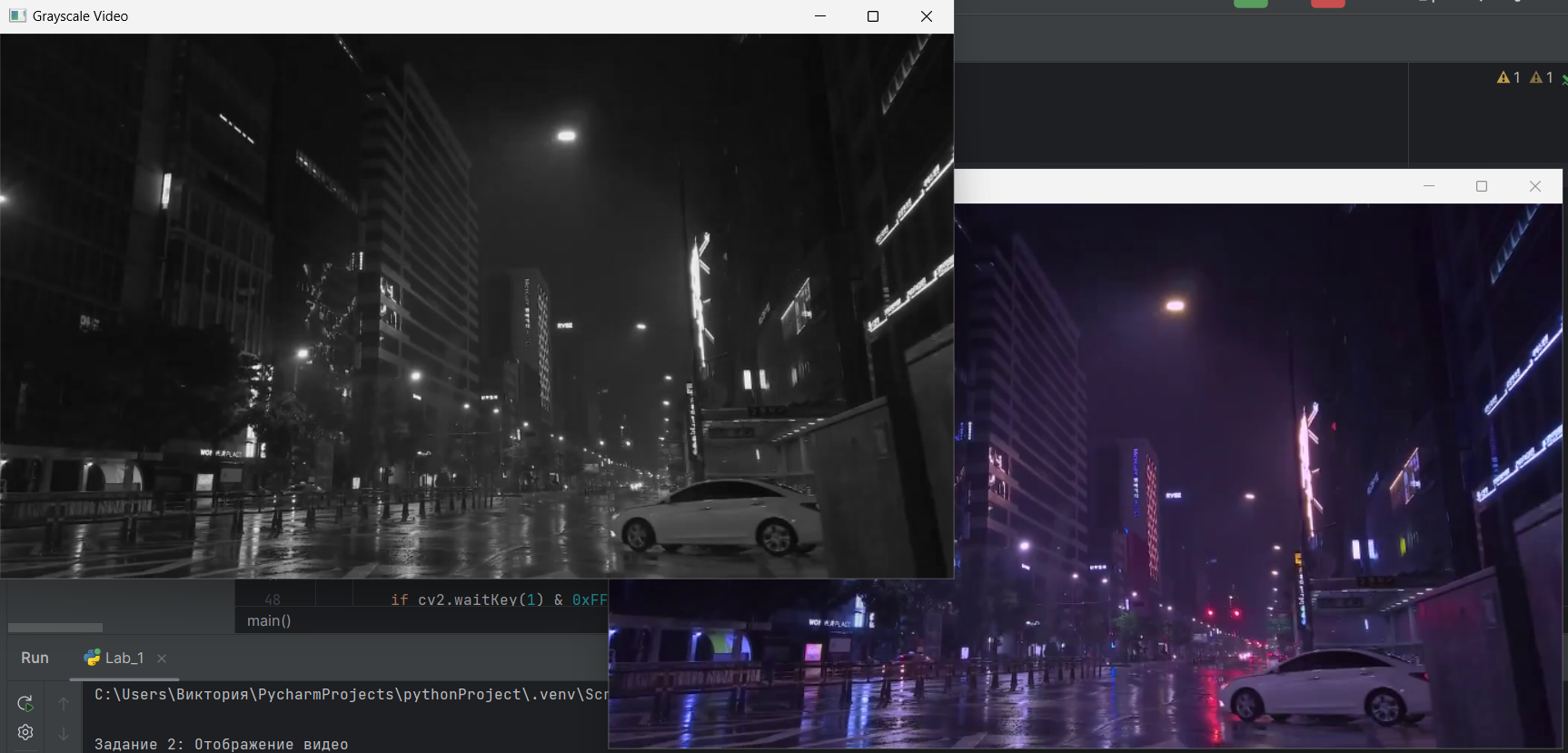


Рисунок 6 – результат работы программы

Задание № 3. Записать видео из файла в другой файл.

Для записи используем тот же файл, что указан в задании выше, но сохраним его в черно-белом формате.

Через get(cv2.CAP\_PROP\_FRAME\_WIDTH) и get(cv2.CAP\_PROP\_FRAME\_HEIGHT) сохраняем разрешение (размер) видео, а через cap.get(cv2.CAP\_PROP\_FPS) не допускаем искажение кадров.

Производим запись в файл формата mp4 с названием ‘output\_video’ с помощью цикла и сохраняем, после чего в папке появляется указанный файл.

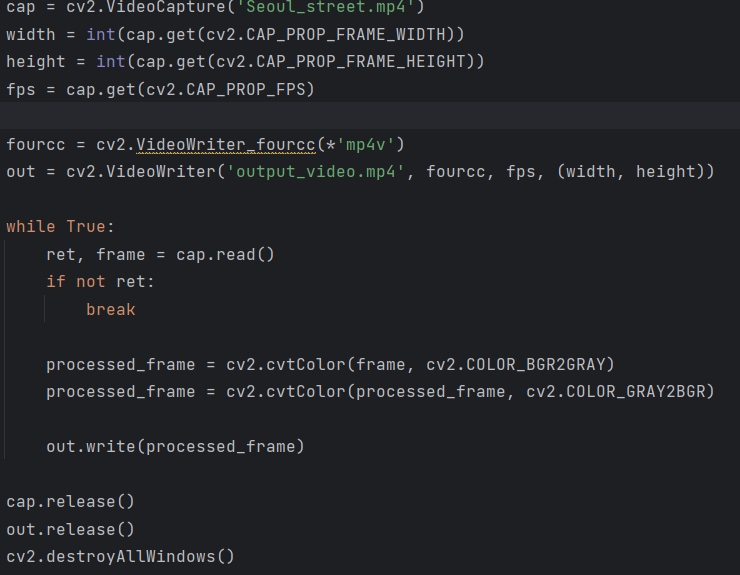


Рисунок 7 – код для записи нового файла



Рисунок 8 – результат работы программы

**Задание №4**. Прочитать изображение, перевести его в формат HSV. Вывести на экран два окна, в одном изображение в формате HSV, в другом – исходное изображение.

Для задания используем изображение из первого задания (‘sleepy\_cat.jpg’). В hsv\_img с помощью флажка COLOR\_BGR2HSV переводим формат цвета в HSV. Сохраняем размер окна согласно размеру изображения и выводим.

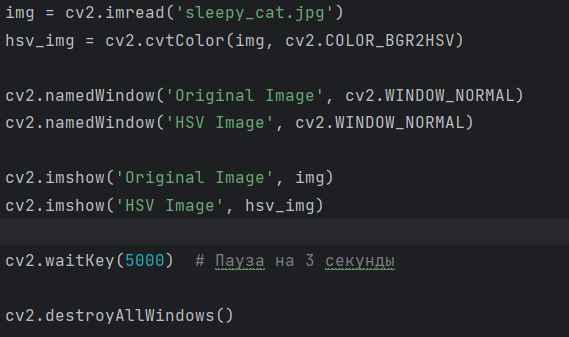


Рисунок 9 – код для перевода в HSV формат

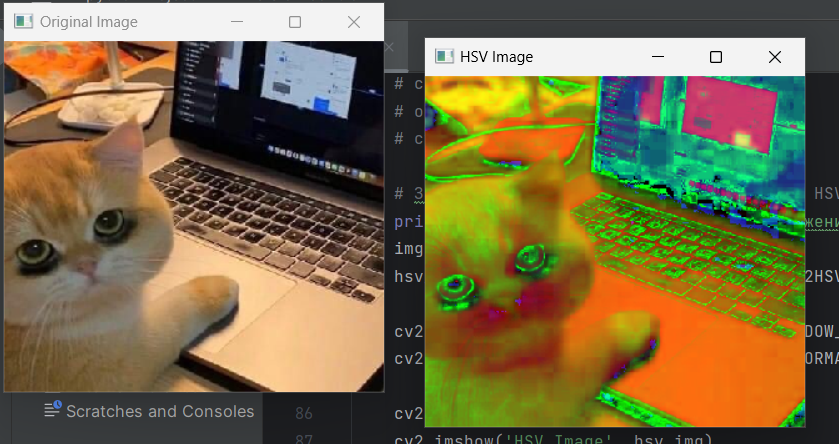


Рисунок 10 – результат работы программы

**Задание №5**. Прочитать изображение с камеры. Вывести в центре на экране Красный крест в формате, как на изображении. Указать команды, которые позволяют это сделать.

**Задание №6**. Отобразить информацию с вебкамеры, записать видео в файл, продемонстрировать видео. (Задания 5 и 6 объединены для удобства решения)

Используя необходимые переменные считываем изображение с камеры, применяя цикл, открываем его в новом окне с сохранением разрешения и сохраняем в файл формата mp4 ‘camera\_video’.

Для отображения креста, указанного в задании, используем запись координат. Вычисляем центр (center\_x = int(width / 2), center\_y = int(height / 2)) и с помощью line() добавляем линию с нужными координатами (существует вариант отрисовки через прямоугольники, выбраны отрезки).

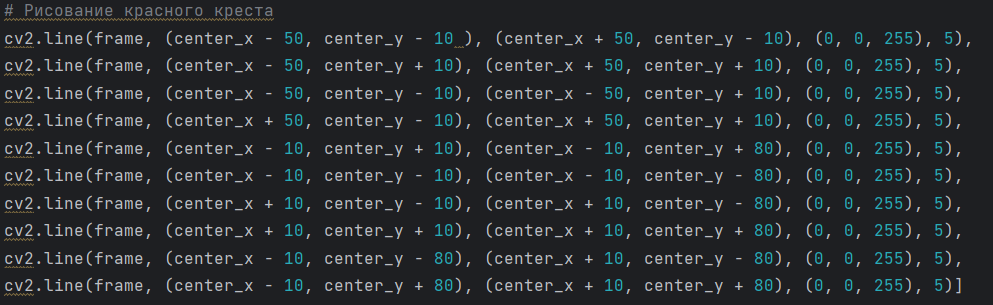


Рисунок 11 – запись координат креста

Для отображения кадра используем imshow, записываем фрейм и с помощью waitkey обозначаем возможность завершения трансляции с камеры через клавишу ‘e’.



Рисунок 12 – результат работы программы

**Задание №7**. 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.

Берем фрагмент кода из задания 5 и 6, определяем значения для закраски креста. Высчитываем центр креста, определяем доминирующий цвет через max.

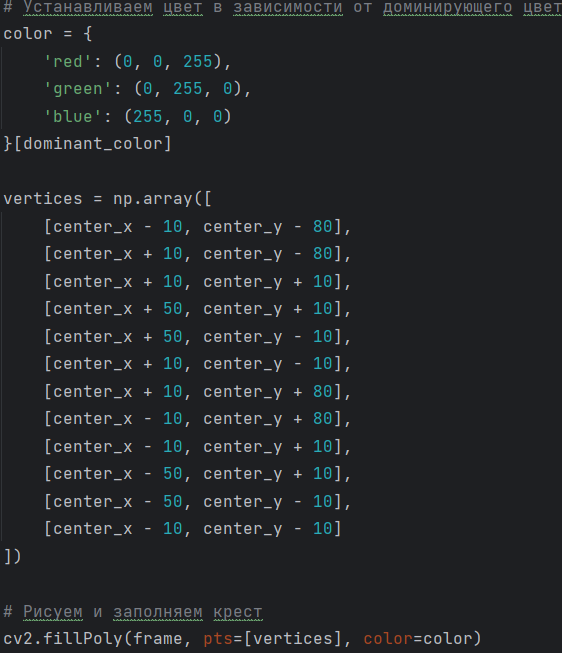


Рисунок 13 – определение цвета и отрисовка

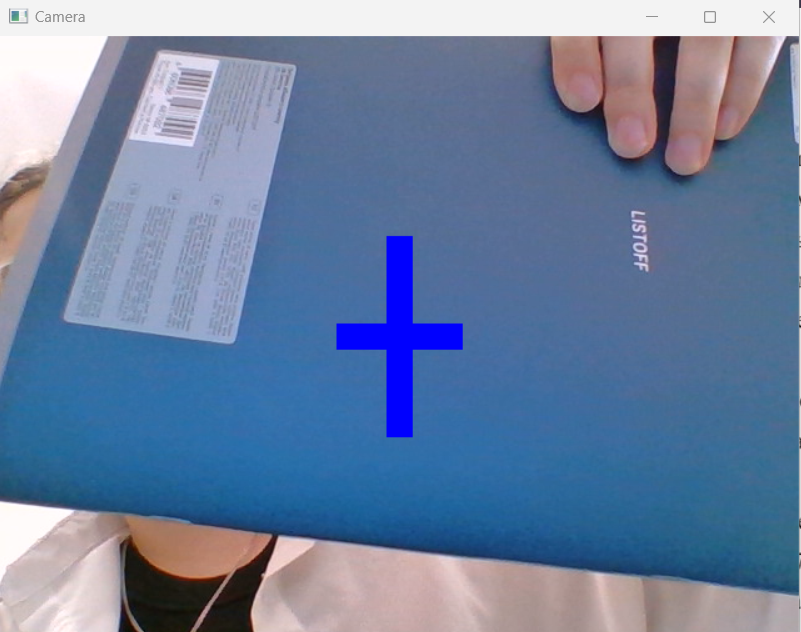
 

Рисунок 14 – результат работы программы

Задание №7. Подключите телефон, подключитесь к его камере, выведете на экран видео с камеры. Продемонстрировать процесс на ноутбуке преподавателя и своем телефоне.

Записываем в url адрес, по которому можно подключиться к камере телефона (используется приложение IP Webcam). Через цикл считываем кадр, выводим его в окно с нужным размером и, если необходимо, завершаем трансляцию через клавишу ‘q’.

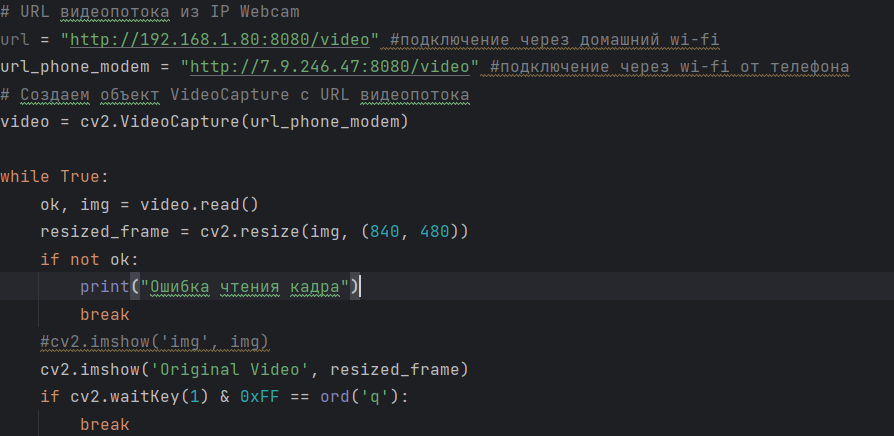


Рисунок 15 – вывод кадра с телефона

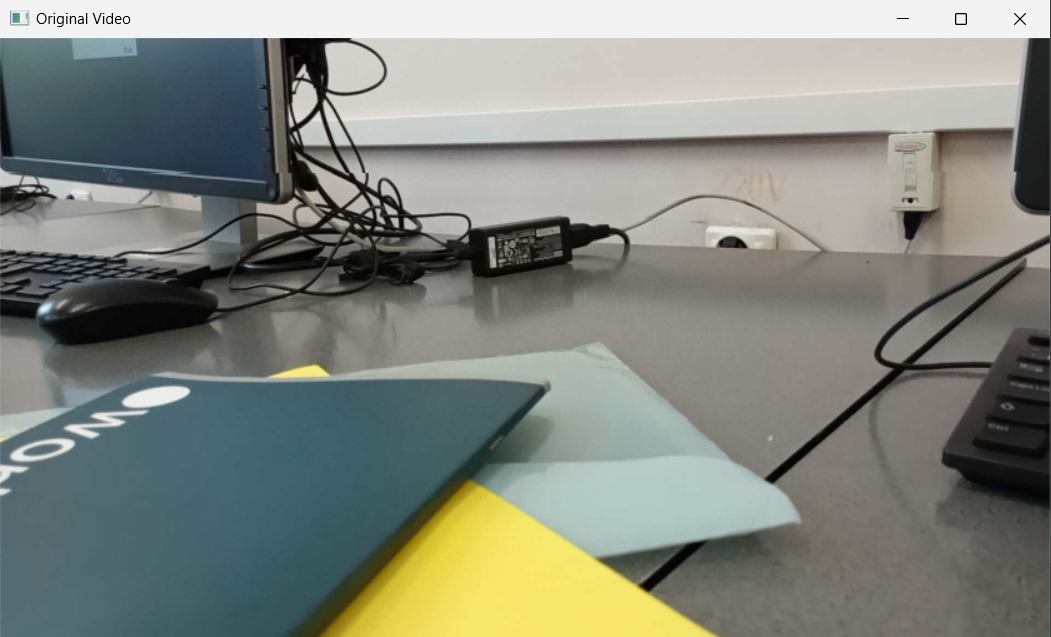


Рисунок 16 – результат работы программы