**Ход работы**

1. Я установил библиотеку OpenCV для Python путем ввода команды в консоль:

pip install opencv-python

2. Далее я проверил вывод изображений различных форматов в именованных окнах с различными флагами.



Рисунок 1 – Листинг задания 2

Метод imread считывает изображение по указанному пути и применяет к нему опцию, указанную флагом во втором параметре функции:

* IMREAD\_REDUCED\_COLOR\_8 – считывает изображение в цвете с уменьшением размера в 8 раз,
* IMREAD\_REDUCED\_GRAYSCALE – считывает изображение в чёрно-белом формате,
* IMREAD\_REDUCED\_RGB - считывает изображение в формате RGB.

Метод namedWindow создает новое именованное окно с различными флагами:

* WINDOW\_AUTOSIZE – создает окно автоматически определенного фиксированного размера,
* WINDOW\_KEEPRATIO – создает окно, сохраняющее пропорции изображения при изменении размера,
* WINDOW\_FREERATIO – создает окно со свободными пропорциями изображения.

Метод imshow выводит изображение в указанное именованное окно.

Чтобы окно не закрывалось сразу, с помощью метода waitKey ожидается ввод с любой кнопки.

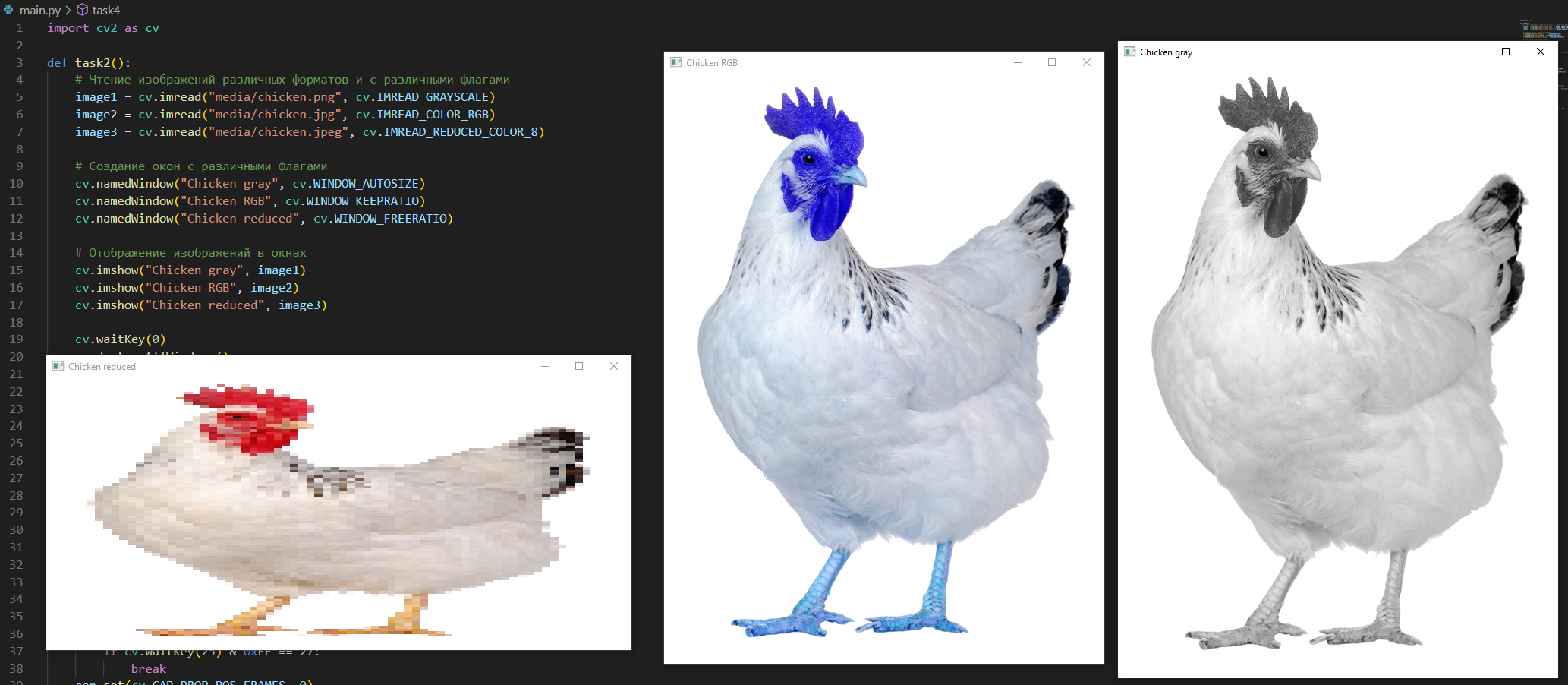


Рисунок 2 – Вывод изображений в различных форматах

3. Для отображения в окне видео в различных форматах я написал функцию, последовательно отображающую кадры считанного видео и применяющую указанные преобразования.



Рисунок 3 – Листинг задания 3

Видео считывается в объект класса VideoCapture, который предоставляет метод read, возвращающий булевое значение ret, обозначающее успешно ли считан кадр, и сам кадр, представляемый, как и обычное изображение, в виде массива NumPy в некотором формате кодирования цветов (по умолчанию BGR).

Метод waitKey ждет указанное число миллисекунд нажатия любой клавиши и возвращает при нажатии код клавиши, которая была нажата. Далее, благодаря побитовому умножению на FF берутся первые 8 бит, которые соответствуют стандартному ASCII-коду клавиши. В данном случае это 27 – Esc.

Таким образом, видео проигрывается в 40 FPS, и кадры могут быть вручную пропущены нажатием клавиши Esc.

4. Для считывания видео и перезаписи его в другой файл необходимо получить его параметры: широту, высоту и количество кадров в секунду, что можно сделать методом cv.get.

Эти параметры применяются классом VideoWriter, который принимает их в конструкторе. В целом, его конструктор принимает следующие параметры:

- filename – путь к выходному файлу

- fourcc (Four Character Code) – кодек, определяющий, как будут сжиматься и храниться кадры видео

- fps – кадры в секунду

- (width, height) – размер кадра в пикселях

- isColor – флаг, определяющий, цветное ли изображение (по умолчанию True)

При создании объекта класса VideoWriter, запись видео в файл происходит аналогично считыванию видео.

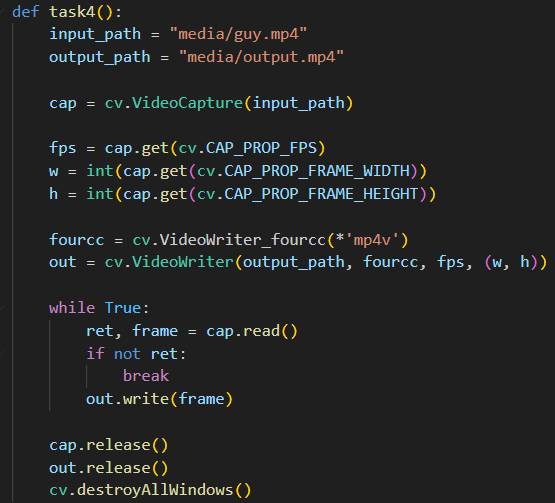


Рисунок 4 – Листинг задания 4

5. Для отображения изображения в стандартном формате и в формате HSV я использовал встроенный в OpenCV метод cvtColor, используемый для изменения цветового пространства, с параметром BGR2HSV.

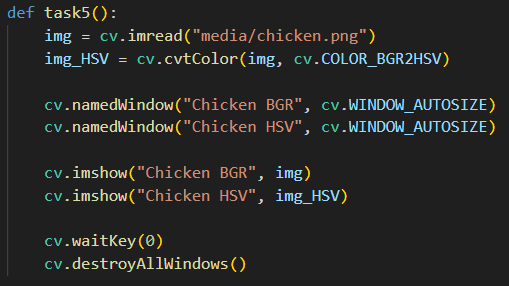


Рисунок 5 – Листинг задания 5

Таким образом, получились следующие изображения:



Рисунок 6 – Вывод изображения в формате BGR и HSV

6.