Задание 0. Разметка изображений с помощью labelImg

Дана таблица **argo_classifier.csv** с перечисленными классами (41 класс объектов) для системы машинного зрения. Необходимо из известных датасетов (например MS COCO, Pascal VOC 2012, BDD100K и пр.) выбрать изображения с перечисленными классами и с помощью утилиты labelImg сверить верность разметки объектов в xml-файлах (если они есть), добавить нужные классы из таблицы при необходимости, а ненужные классы удалить. Либо заново создать файл разметки.

Важно: формат xml-файлов разметки необходимо выбрать **PascalVOC.**

Ссылка на labelImg с настроенными классами:
https://github.com/snoopy112/PythonLessons/blob/master/labelImg.zip

Ссылки на дасеты:
MS COCO: http://cocodataset.org/#home
Pascal VOC 2012: http://host.robots.ox.ac.uk/pascal/VOC/voc2012/#devkit
BDD100K: https://bdd-data.berkeley.edu/
Human Detection and Tracking in Agriculture:
https://www.nrec.ri.cmu.edu/solutions/agriculture/other-agriculture-projects/human-detection-and-tracking.html

Задание 1. Изучение Python, NumPy + OpenCV

Для каждой из задач:

- написать несколько вариантов кода различной эффективности. Один полностью векторизованный вариант и один вариант без векторизации. Варианты решения одной задачи должны содержаться в отдельном Python модуле;
- сравнить при помощи %timeit скорость работы на нескольких тестовых наборах разного размера (минимум 3);
- проанализировать полученные данные о скорости работы разных реализаций.

Задачи

Предполагается, что модуль numpy импортирован под названием np.

1) Подсчитать произведение ненулевых элементов на диагонали прямоугольной матрицы.
Для X = np.array([[1, 0, 1], [2, 0, 2], [3, 0, 3], [4, 4, 4]]), ответ: 3.

2) Дана матрица X и два вектора одинаковой длины i_idx и j_idx. Построить вектор np.array([X[i_idx[0], j_idx[0]], X[i_idx[1], j_idx[1]], . . . , X[i_idx[N-1], j_idx[N-1]]]).
Для X = np.array(range(4 * 5)).reshape(4, 5) + 1, i_idx = np.array([1, 3, 0, 2]), j_idx = np.array([0, 2, 3, 1]), ответ: [6 18 4 12].

3) Даны два вектора x и y. Проверить, задают ли они одно и то же мультимножество.

Для x = np.array([1, 2, 2, 4]), y = np.array([4, 2, 1, 2]) ответ: True.

4) Найти максимальный элемент в векторе x среди элементов, перед которымистоит нулевой. Для x = np.array([6, 2, 0, 3, 0, 0, 5, 7, 0]), ответ: 5.

5) Дано изображение (трёхмерный массив), размера (height, width, numChannels), а также вектор длины numChannels. Преобразуйте цветное изображение в оттенки серого, сложив каналы изображения с указанными весами, вернуть результат в виде матрицы размера (height, width) и вывести изображение на экран. Считать реальное изображение в numpy.array можно при помощи библиотеки OpenCV (cv2.imread(), но учитывайте, что в OpenCV цветовое пространство BGR, вместо RGB).

Матрица коэффициентов для RGB изображения: np.array([0.0012, 0.0004, 0.0023]).

6) Реализовать кодирование длин серий (Run-length encoding). Дан вектор x. Необходимо вернуть кортеж из двух векторов одинаковой длины. Первый содержит числа, а второй - сколько раз их нужно повторить.
Пример: x =np.array([2, 2, 2, 3, 3, 3, 5]), ответ: (np.array([2, 3, 5]), np.array([3, 3, 1])).

Замечание. Можно считать, что все указанные объекты непустые (к примеру, в задаче 1 на диагонали матрицы есть ненулевые элементы).

Полезные функции NumPy: np.zeros, np.ones, np.diag, np.eye, np.arange, np.linspace, np.meshgrid, np.random.random, np.random.randint, np.shape, np.reshape, np.transpose,np.any, np.all, np.nonzero, np.where, np.sum, np.cumsum, np.prod, np.diff, np.min, np.max,np.minimum, np.maximum, np.argmin, np.argmax, np.unique, np.sort, np.argsort, np.bincount, np.ravel, np.newaxis, np.dot, np.linalg.inv, np.linalg.solve. Многие из этих функций можно использовать так: x.argmin().