

Практическое задание №3

Задача №6

Условие: Написать функцию, которая возвращает тензор представляющий изображение круга с заданным цветом и радиусом в схеме rgb на черном фоне.

Выполнение: Была написана функция `create_tensor_v_1(data)`. Она принимает список параметров `data`. Первый элемент `data` – радиус окружности, остальные 3 элемента – код цвета в rgb. С помощью библиотеки `torch` и метода `torch.randint(0, 1, (size, size, 3))` создается 4-х мерный тензор, который представляет собой квадрат черного цвета со стороной длины `size`. Далее с помощью двойного цикла и формулы окружности выбираются те пиксели, которые надо перекрасить, возвращает функция искомый тензор с закрасенным кругом на черном фоне. Однако эта функция работает достаточно медленно, особенно при большом радиусе круга.

Поэтому была написана функция `create_tensor_v_2(data)`. Она принимает то же самое, что и предыдущая функция, таким же образом создает тензор в виде черного квадрата. Однако далее мы, не используя циклы, создаем маску, которая определяет какие пиксели необходимо закрасить. Функция возвращает используемый тензор и работает в разы быстрее предыдущей функции. Об этом может сказать небольшой бенчмарк, который показывает, что для радиуса в 300, функция `create_tensor_v_1(data)` выполняется за 4.7447689, а функция `create_tensor_v_2(data)` выполняется за 0.04461220000000021.

Информация о радиусе круга и цвете в rgb считывается с файла `input_file.txt` с помощью `np.fromfile()`, данные разделяются пробелами. Искомый тензор выводится в бинарный файл `bin_tensor.npy` с помощью `np.save()`. Тензор из этого файла можно загрузить в программу с помощью `np.load()`.

Пример работы программы приведен ниже. В качестве входных данных использовались следующие параметры [300 255 125 0].

