**Министерство цифрового развития, связи и массовых коммуникаций Российской Федерации**

**Ордена Трудового Красного Знамени**

**Федеральное государственное бюджетное образовательное учреждение высшего образования**

**«Московский технический университет связи и информатики»**

Кафедра математической кибернетики и информационных технологий

**Отчет по лабораторной работе №2**

по дисциплине «Системы машинного зрения»

на тему:

«Разработка нейросетевых функций. Операция Convolution3D»

Выполнил:

Студент группы БВТ2002

Дерябин С. С.

Содержание:

1 Введение….……………………………………………………………….3

2 Выполнение……………………………………………………………….4

3 Вывод………………………………………………………...……………6

**Введение**

В данной лабораторной работе требуется реализовать встроенную в библиотеку PyTorch функцию 3D-свертки Convolution 3D, а также сравнить готовую реализацию функции и написанную в рамках лабораторной работы с помощью группы тестов. Входные параметры функции должны совпадать с готовой реализацией.

Функция torch.nn.Conv3d в библиотеке PyTorch представляет собой операцию трехмерной свертки, предназначенную для обработки трехмерных данных, таких как видео-последовательности или трехмерные медицинские изображения. Она аналогична Conv2d, но работает с трехмерными тензорами

Отличия от Conv2d:

* Трехмерность: Главное отличие заключается в том, что Conv3d работает с трехмерными данными, в то время как Conv2d работает с двумерными данными (изображениями).
* Входные размеры: Conv3d принимает трехмерные тензоры, включая глубину данных (D), в то время как Conv2d работает с двумерными изображениями без глубины.
* Ядро и данные: Conv3d использует трехмерные ядра для свертки трехмерных данных.
* Выходные размеры: Результатом Conv3d является трехмерный тензор.

**Выполнение**

Согласно заданию лабораторной работы, мною было реализовано два Python файла. Первый – fu\_custom\_cov3d.py, представляет собой кастомную реализацию функции Conv3d, которая принимает на вход:

* input\_tensor – входной тензор,
* weight - ядро свертки,
* stride - шаг свертки,
* padding - значение паддинга для входного тензора,
* bias: флаг указывающий, должен ли использоваться смещение (по умолчанию True).

Листинг кода представлен на рисунке 1.

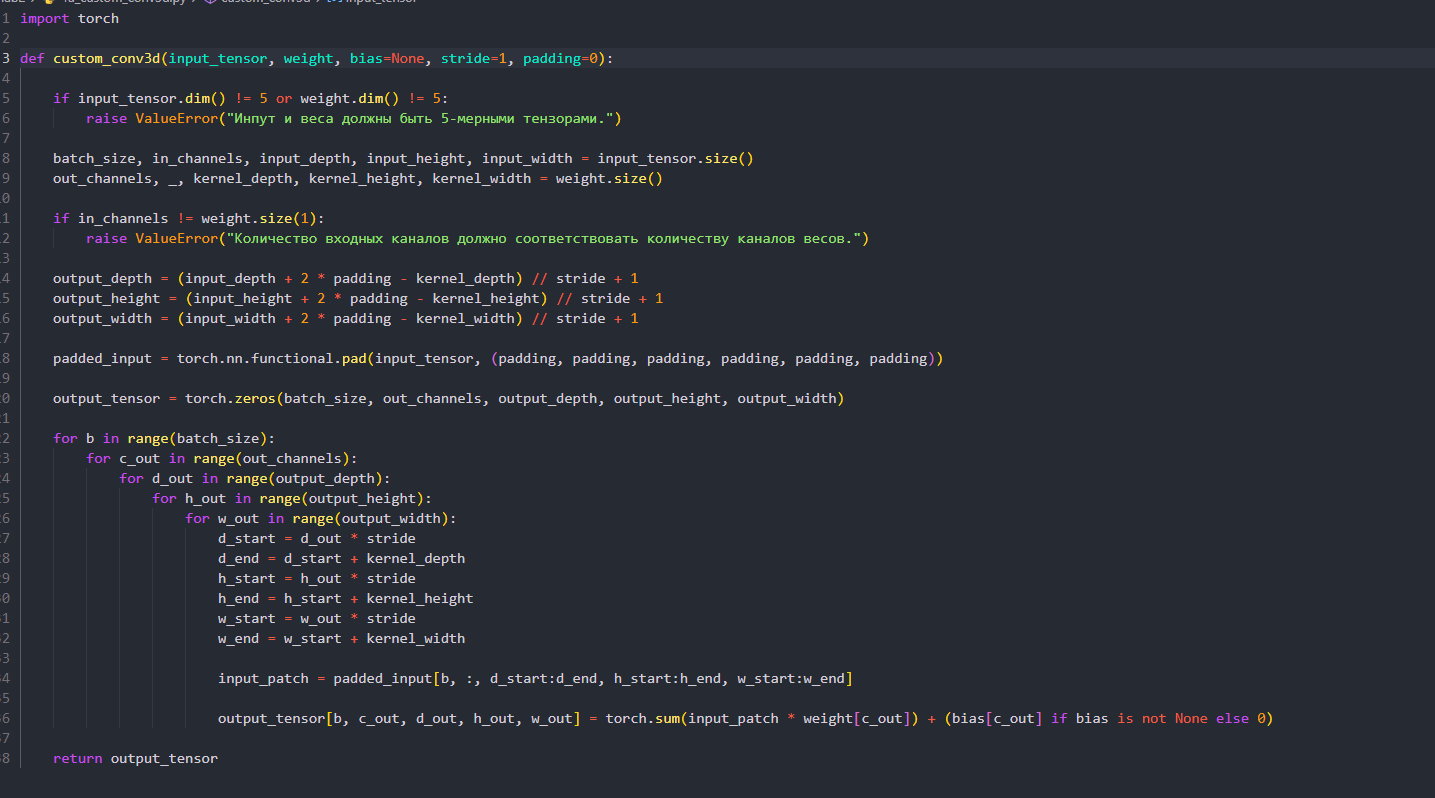


Рисунок 1 – реализация функции custom\_cov2d

Также для тестирования правильной работы функции были написаны unit-тесты, которые хранятся в файле – lab2.py. Структура тестов:

1. Первый тест описан для тестирования общего функционала на примерах 2-ух случайных тензеров.
2. Второй тест описан для тестирования функционала кастомной свертки без использования bias.
3. Третий тест описан для тестирования передачи больших данных на вход функции.

Листинг кода с тестами представлен на рисунке 2.

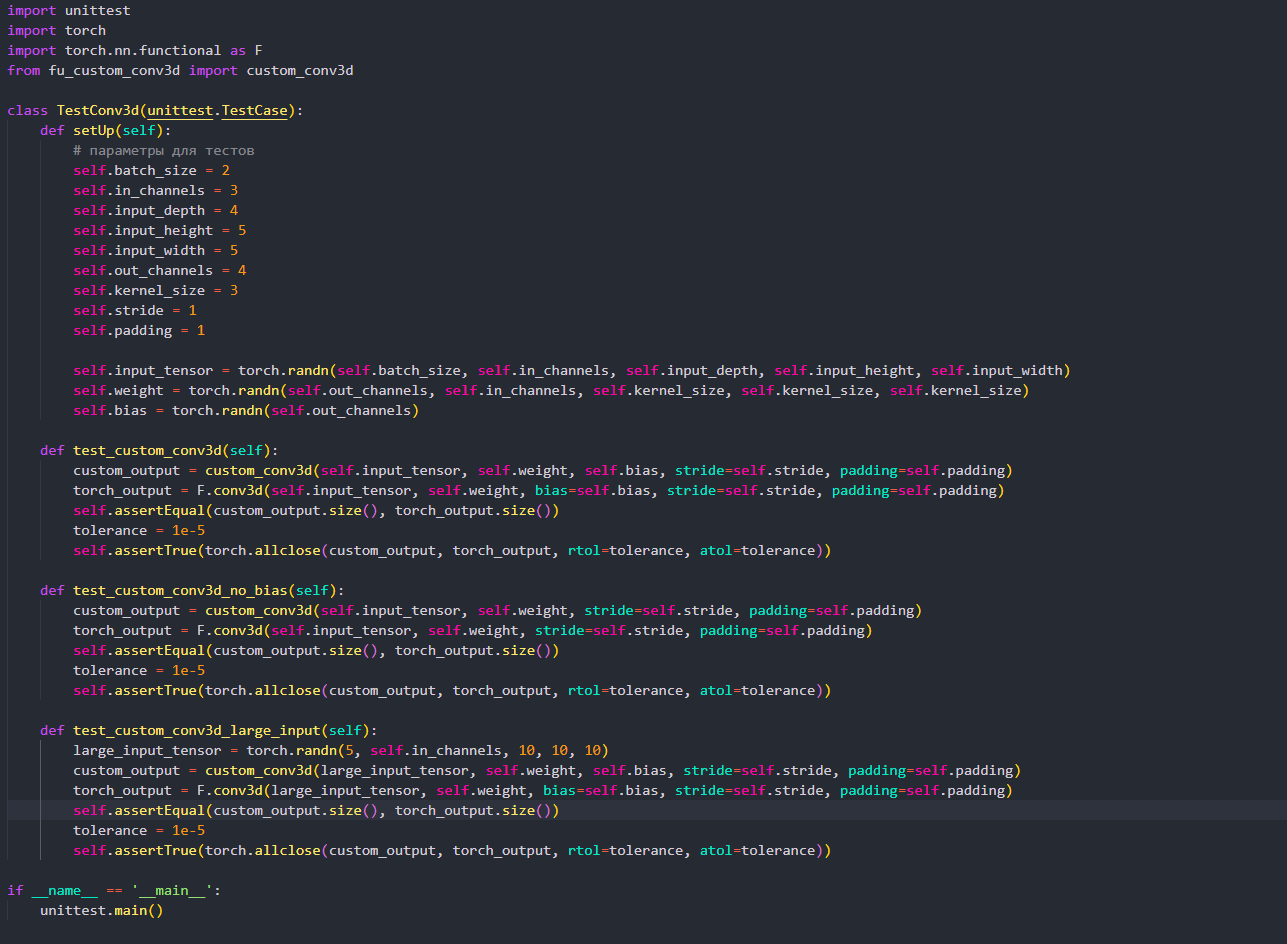
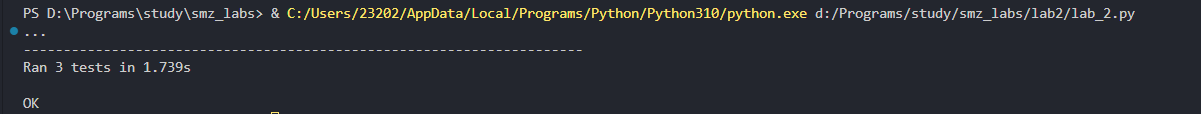


Рисунок 2 – реализация unit-тестов

Успешное прохождение тестов представлено на рисунке 3.



Ссылка на код лабораторной работы: https://github.com/vraniye/smz\_labs/tree/master/lab2

**Вывод**

В ходе выполнения данной лабораторной работы мной была реализована функция custom\_cov3d, которая, как и оригинальная функция conv3D из библиотеки PyTorch, осуществляет операцию трехмерной свёртки. Был произведён ряд тестов, в котором, в результате сравнения результата работы данных функций на одних и тех же данных, мной была выявлена корректность работы реализованной мной функции Convolution3D.