**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 «Математическая кибернетика и информационные технологии»

Отчет по лабораторной работе №1

по дисциплине «Системы машинного зрения»

на тему:

«Разработка нейросетевых функций. Операция Convolution 2D»

Выполнил:

студент группы БВТ2003

Ушаков М.С.

**Цель работы**

Разработать алгоритм, используя язык python, реализующий работу операции двумерной свертки.

**Задание**

При выполнении лабораторной работы необходимо:

* ознакомиться с описанием операции библиотеки PyTorch (https://pytorch.org/docs/stable/generated/torch.nn.Conv2d);
* используя язык программирования Python написать алгоритм, реализующий алгоритм свертки;
* составить отчет по лабораторной работе.

**Теоретические сведения**

Операция 2D-свертки (Convolution 2D) является ключевой процедурой в разработке нейронных сетей, применяемых в глубоком обучении и искусственном интеллекте. Эта математическая операция применяется к двумерным массивам данных, таким как изображения, с целью извлечения существенных признаков из входных данных.

В рамках нейронных сетей операция 2D свертки выполняет фильтрацию входного изображения, используя матрицу фильтров, которая обучается для выявления определенных характеристик изображения. Это обеспечивает автоматическое обнаружение значимых признаков, вместо использования заранее заданных функций.

Шаги операции 2D свертки включают в себя:

Загрузка входных данных: Двумерное изображение преобразуется в одномерный массив (вектор), который затем служит в качестве входного сигнала для операции свертки.

Определение фильтров: Матрица фильтров представляет собой двумерный массив чисел, применяемый для фильтрации входного изображения. Эта матрица обучается на основе обучающего набора данных для выделения определенных особенностей изображений.

Выполнение свертки: Каждый фильтр применяется к входному изображению, перемещаясь по всему изображению. Каждый раз, когда фильтр охватывает конкретную область изображения, выполняется операция умножения между значениями фильтра и соответствующими значениями входного изображения, за которой следует суммирование результатов умножения. Этот процесс повторяется для каждого фильтра, формируя карты признаков.

Нелинейное преобразование: После свертки для каждого фильтра часто применяется нелинейная функция (например, сигмоид или ReLU), чтобы увеличить выразительность модели.

Построение выходных данных: Выходные данные обычно представляют собой вектор, где каждый компонент соответствует одному фильтру. Этот вектор затем используется в последующих слоях нейронной сети, таких как полносвязные слои или слои активации.

Операция 2D свертки играет ключевую роль в обработке изображений и глубоком обучении, автоматически извлекая значимые признаки из входных изображений, что упрощает процесс обучения и придает модели большую гибкость.

**Выполнение работы**

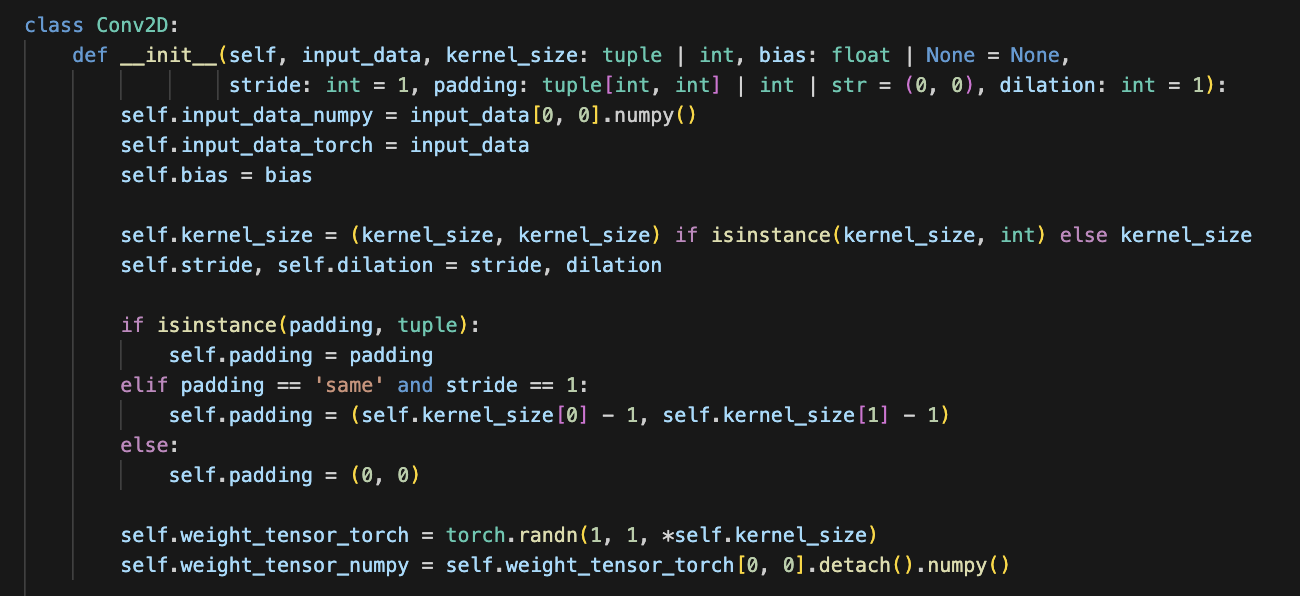
Создадим класс conv2D, который будет реализовывать операцию двумерной свертки. В этом классе опишем все необходимые функции. Начнем с создания функции для проверки соответствия входных данных (параметров) типу класса. Функция для инициализации объекта класса представлена в следующем блоке кода. 

Рисунок 1 – Инициализация объектов.

Ниже представлен сам алгоритм свертки, а также функция теста, по которой мы будем сравнивать значения нашей рукописной свертки с оригинальной сверткой из библиотеки torch: 

Рисунок 2 – Алгоритм двумерной свертки и функция тестирования

На рисунке 3 представлены результаты непосредственно самого тестирования для обычной свертки, для свертки с padding “same”, для свертки с большим размером kernelsize, с добавлением dilation и с добавлением stride:

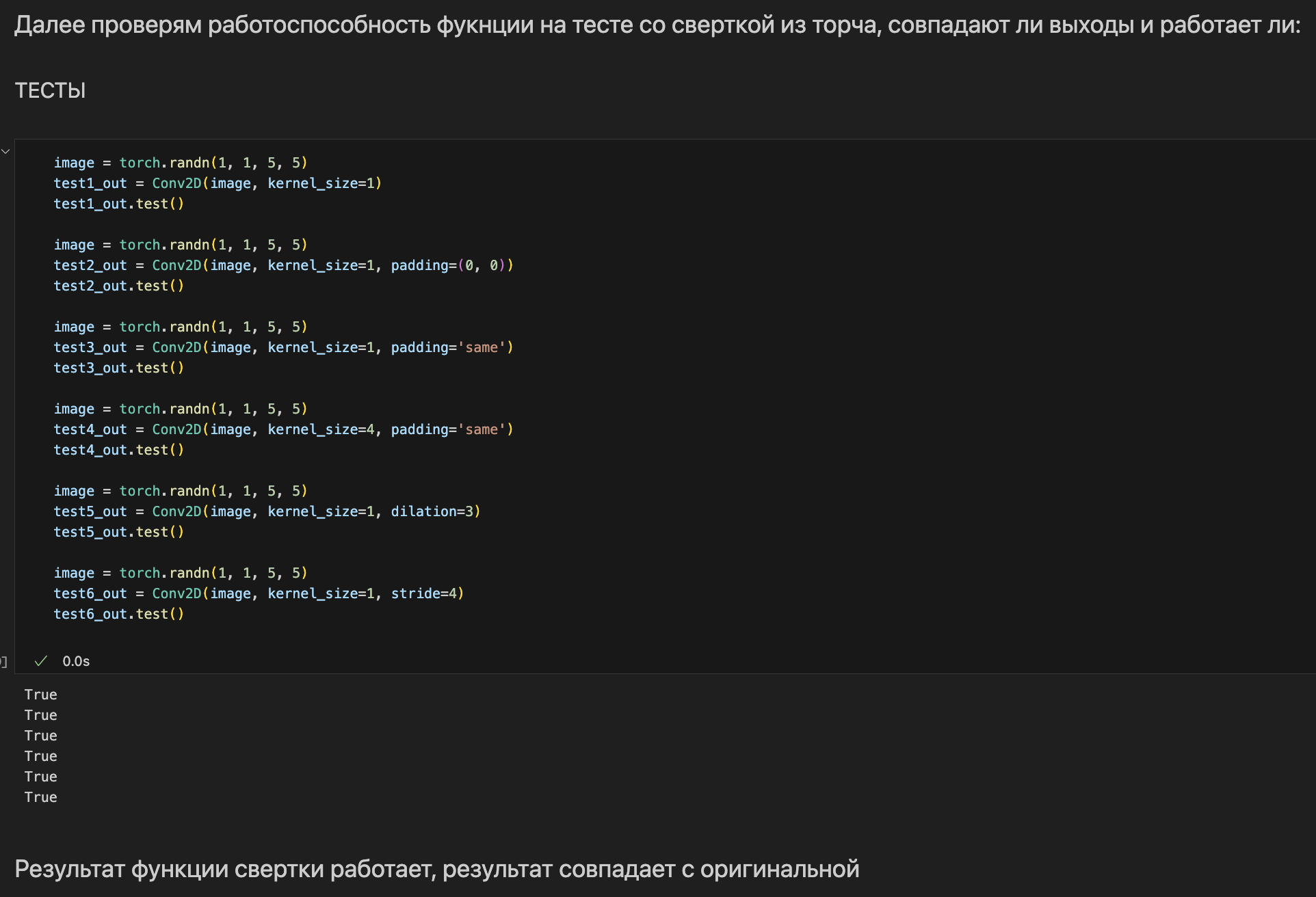
****

Рисунок 3 – Результаты всех шести тестов.

**Вывод**

В ходе выполнения данной лабораторной работы я разработал алгоритм на языке Python, который реализует операцию двумерной свертки. Также я создал тесты для проверки корректности работы алгоритма.

Ссылка на git-репозиторий: https://github.com/UshakovMatt/SMZ