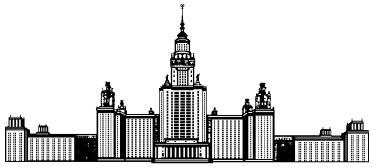
Московский государственный университет имени М. В. Ломоносова



Факультет Вычислительной Математики и Кибернетики

Кафедра Математических Методов Прогнозирования

ПРОВЕРКА ИССЛЕДОВАНИЯ

«Генерация пейзажей»

Домашнее задание №1 по курсу

«Прикладная алгебра»

студента 3 курса 317 группы Бессонова А.С.

Выполнил:

студент 3 курса 317 группы

Зиннуров Артём Ринатович

Научный руководитель:

к.ф-м.н.

Китов Виктор Владимирович

Москва, 2022

Содержание

1 Насколько хорошо выполнено задание

Задание выполнено очень качественно. Видно, что проделана большая работа: произведён разбор статьи, самостоятельно реализованы все части сети, используемой для генерации пейзажей, произведено обучение сети с применением аугментации данных, присутствуют многочисленные эксперименты, а также имеется возможность легко получить пейзаж по маске, нарисованной от руки в онлайн-редакторе, что сильно упрощает тестирование и анализ данного метода. В работе присутствует много исчерпывающих комментариев, разъясняющих назначение функций и классов, описывающих архитектурные составляющие сети. Кроме этого, приложены ссылки на код с обучением, данные и литературу.

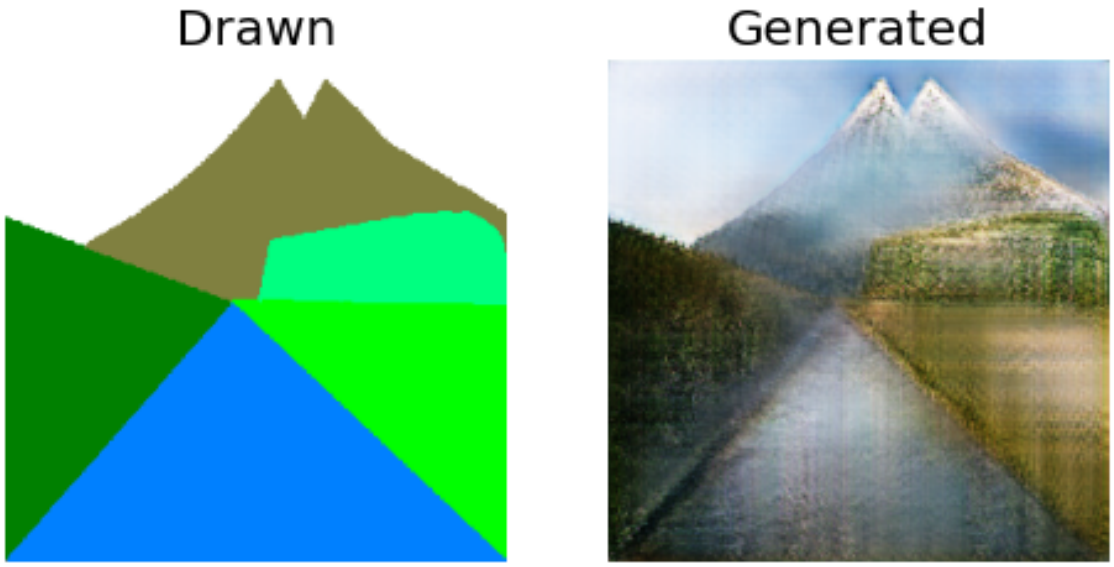


Рис. 1: Это мой собственный пейзаж!

2 Понятно ли Вам исследование

Исследование понятно. В самом начале статьи дана краткая постановка задачи, приложены картинки, демонстрирующие результат работы, тут же расположены ссылки на статью с реализуемой архитектурой. Но мне кажется, было бы здорово совсем коротко рассказать о самой задаче, о первых попытках её решения, достигнутых результатах на сегодняшний день и применении на практике. Тут же отмечу, что

имеет смысл в самом начале рассказать о структуре данных, с которыми работает нейросеть. Это положительно повлияло бы на понимании кода. Но это всё мелочи.

3 Убеждает ли оно Вас в чём-то

Тема данной работы в меньшей степени предполагает некоторое исследование, в большей - реализацию и тестирование. Но отвечая на поставленный вопрос: убеждает в возможности генерации пейзажей с помощью свёрточных нейронных сетей.

4 Корректны ли утверждения и выводы

Все утверждения корректны. В конце работы присутствует вывод, комментирующий получившиеся результаты.

5 Понятен ли Вам код, есть ли в нём ошибки

Код аккуратный и понятный. В большинстве нетривиальных мест присутствуют поясняющие комментарии (например, размер тензора, его назначение и т.д.), но не во всех. Возможно, в работах с большим числом функций и классов имеет смысл писать докстринги с размерами тензоров и краткими пояснениями. Уверен, читаемость и без того опрятного кода только увеличится.

В коде допущена небольшая ошибка: пропущен последний из семи **SPADE**-слоёв.

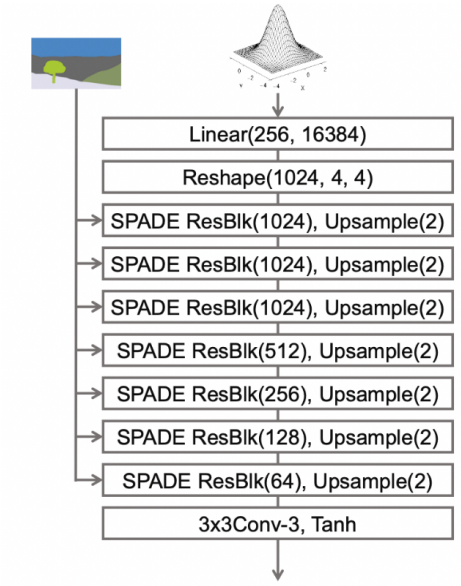


Рис. 2: В статье присутствует.

```python
class Generator(nn.Module):
  def __init__(self, mask_shape, latent_dim=256):
    super().__init__()
    self.latent = latent_dim
    self.mask_shape = mask_shape

    self.linear1 = nn.Linear(latent_dim, 16384)
    self.res1 = ResBlock(1024, (1024,4,4), mask_shape) # gets (linear1 reshaped to (-1,1024,4,4) , mask)

    self.upsample1 = nn.UpsamplingNearest2d(scale_factor=2)
    self.res2 = ResBlock(1024, (1024,8,8), mask_shape) # на входе изображение в 2 раза больше

    self.upsample2 = nn.UpsamplingNearest2d(scale_factor=2)
    self.res3 = ResBlock(1024, (1024,16,16), mask_shape) # на входе изображение в 2 раза больше

    self.upsample3 = nn.UpsamplingNearest2d(scale_factor=2)
    self.res4 = ResBlock(512, (1024,32,32), mask_shape) # на входе изображение в 2 раза больше -> (B, 512, 32,32)

    self.upsample4 = nn.UpsamplingNearest2d(scale_factor=2)
    self.res5 = ResBlock(256, (512,64,64), mask_shape) # на входе изображение в 2 раза больше

    self.upsample5 = nn.UpsamplingNearest2d(scale_factor=2)
    self.res6 = ResBlock(128, (256,128,128), mask_shape) # на входе изображение в 2 раза больше

    self.upsample6 = nn.UpsamplingNearest2d(scale_factor=2)
    self.leaky_relu = nn.LeakyReLU(0.2)

    self.conv = nn.Conv2d(128, 3, kernel_size=4, padding=0) # input = (B, 128, 256, 256)
    # used F.pad(X, (0, 3, 2, 1)) for same shape
    self.th = nn.Tanh()
```

Рис. 3: На деле отсутствует.

6 Понятны ли Вам рисунки и таблицы

Все рисунки подписаны и понятны. В работе активно используются рисунки из оригинальной статьи, что улучшает понимание происходящего.

7 Что на Ваш взгляд можно ещё сделать

Интересно было бы посмотреть на результаты сетей, обученных на разных количествах эпох. Если результат неудовлетворительный, то привести типичные ошибки. В общем случае обращать внимание и на отрицательные, и на положительные генерации определённых текстур (например, вода лучше передаётся на 75 эпохах, а деревья - на 30 и т.д.)

Было бы интересно сравниться с пейзажами лучших на текущий момент методов генерации пейзажей, которые используют большие специально подготовленные датасеты с высоким разрешением исходных картинок, чёткие маски сегментации и ещё более разнообразные аугментации.