

Лабораторная работа сжатие изображений

Полтораднев Кирилл, М4130

Что было сделано

- Полностью переписан процесс сборки **c++** библиотеки и создана обертка **EntropyCodec.py**. Это позволяет собирать проект на всех OS и под любую архитектуру. (Подробнее в [README](#))
- Написаны **pytest** тесты для всех моделей, что позволяет проверить верны ли размерности при изменении **encoder**, **decoder** в модели
- Проверены следующие гипотезы:
 - Strade вместо MaxPooling
 - VAE
 - ResidualBlock

Метрики

PSNR - измеряет насколько восстановленное изображение отличается от оригинала, выражая это в логарифмической шкале дБ (децибел)

$$\text{PSNR} = 10 \cdot \log_{10} \left(\frac{I_{\max}^2}{\text{MSE}} \right)$$

BPP - измеряет среднее количество битов, необходимых для кодирования одного пикселя изображения.

$$\text{BPP} = \frac{\text{Размер сжатого файла в битах}}{\text{Количество пикселей в изображении}}$$

Методология

Для всех экспериментов используется единый набор параметров для выявления наилучшего подхода при изменении архитектуры метода сжатия изображения

Параметры:

- Количество эпох (**epochs**): 3000
- Скорость обучения (**lr**): 0.003
- Размер батча: 64 (фактически 21, так как датасет состоит из 21 изображения)
- Валидация проводится каждые 300 шагов

Исходная модель autoencoder

					Оригинальное изображение
24.96	25.10	23.96	23.64	24.01	
					Автоэнкодер
23.64 0.54	23.09 0.46	22.49 0.50	22.63 0.53	22.38 0.48	
					Автоэнкодер и квантование с $b=2$
23.85 0.53	22.38 0.47	21.72 0.49	22.31 0.52	22.06 0.49	
					JPEG

Strade vs MaxPooling

Оригинал



PSNR: 25.12



PSNR: 25.22



PSNR: 24.55



PSNR: 24.14

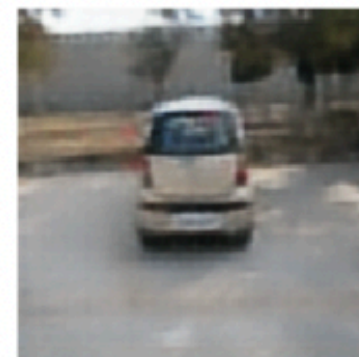


PSNR: 24.26

auto encoder



PSNR: 22.64 BPP: 0.48



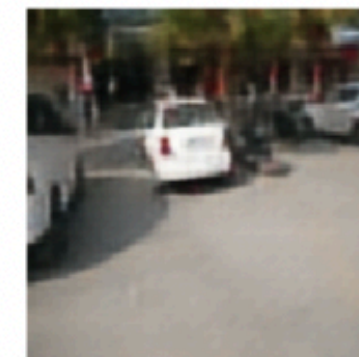
PSNR: 22.97 BPP: 0.45



PSNR: 23.12 BPP: 0.48



PSNR: 21.72 BPP: 0.44



PSNR: 22.65 BPP: 0.48

auto encoder с
квантованием 2



PNSR: 29.16 BPP: 0.48



PNSR: 28.83 BPP: 0.48



PNSR: 28.98 BPP: 0.48



PNSR: 28.63 BPP: 0.49



PNSR: 28.71 BPP: 0.48

jpeg



Использование сверток с шагом (stride) 2 вместо MaxPooling с ядром (kernel) 2 имеет несколько преимуществ:

- 1. Параметрическая оптимизация:** Свертки с шагом 2 добавляют дополнительные обучаемые параметры (веса)
- 2. Сохранение пространственной информации:** Свертки с шагом 2 могут лучше сохранять пространственную информацию, так как они учитывают значения всех пикселей при вычислении выхода
- 3. Гладкость градиентов:** Свертки с шагом 2 могут обеспечить более гладкие градиенты

Данные после 3000 эпох

VAE

Оригинал



PSNR: 12.60

PSNR: 14.01

PSNR: 14.11

PSNR: 13.12

vae



PSNR: 12.60 BPP: 0.02

PSNR: 14.01 BPP: 0.02

PSNR: 14.11 BPP: 0.02

PSNR: 13.12 BPP: 0.02

vae с
квантованием 2



PSNR: 0.00 BPP: 0.00

PSNR: 0.00 BPP: 0.00

PSNR: 0.00 BPP: 0.00

PSNR: 0.00 BPP: 0.00

jpeg

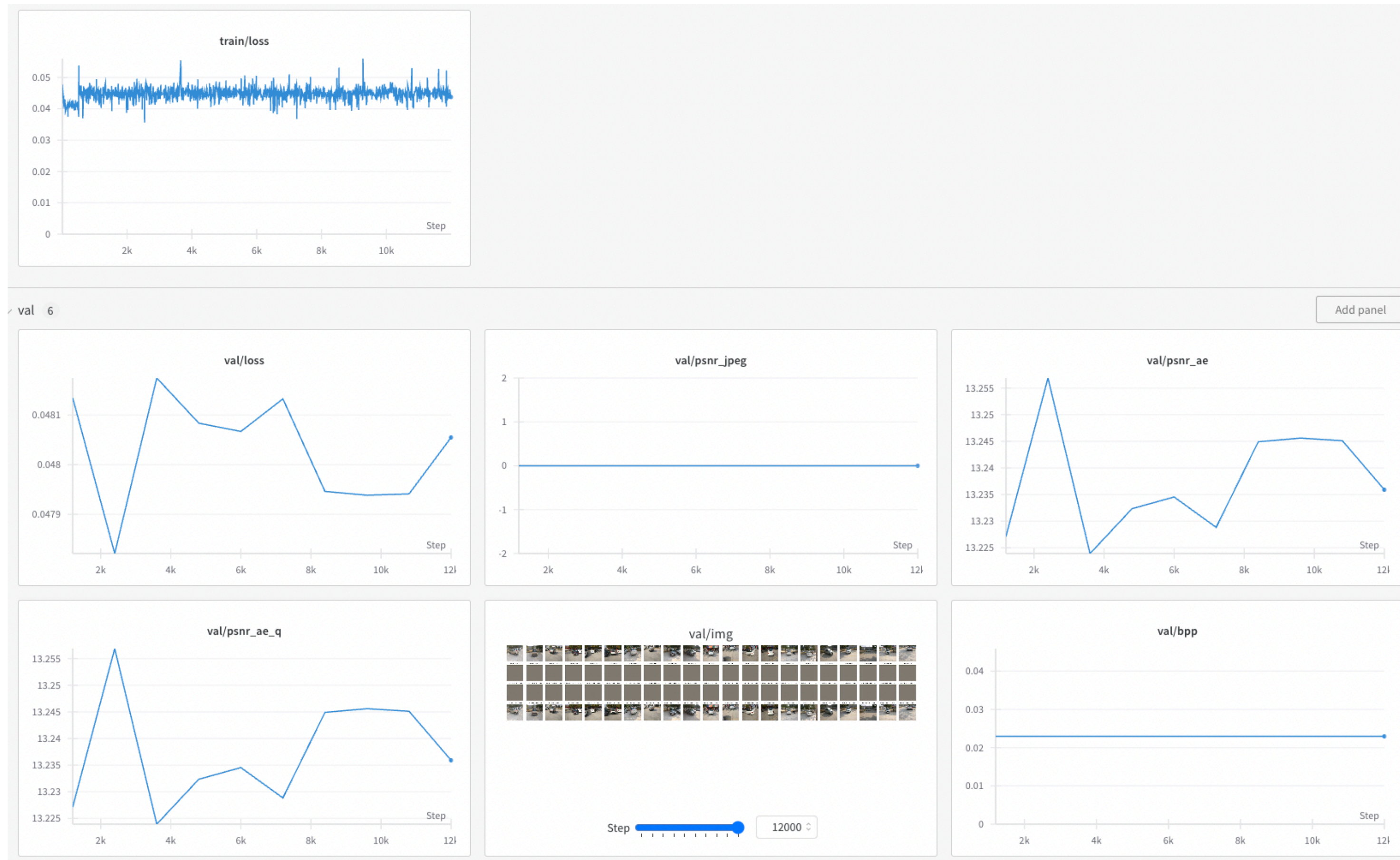


Данные после 3000 эпох

Вариационный автокодировщик (**VAE**) — это генеративная модель, которая обучается кодировать данные в вероятностное латентное пространство и восстанавливать их, минимизируя различия между оригинальными и реконструированными данными через стохастическое кодирование.

Как видно по результату VAE не получилось интегрировать в процесс сжатия изображения

VAE проблемы



Метрики VAE из wandb

- **Loss** не уменьшается, и кажется, что это проблема. Однако у генеративных методов бывает так, что алгоритм работает хорошо, даже если значение loss не снижается.
- Проблема **вероятно** в несовпадении размерностей выходных слоев энкодера и декодера VAE, так как латентный вектор имеет фиксированную, заранее задаваемую длину после энкодирования. Пофиксить их я не успел

ResidualBlock

Оригинал

auto encoder

auto encoder с
квантованием 2

jpeg



Данные после 3000 эпох

Основная идея заключается в том, чтобы добавить прямые связи (skip connections) между слоями.

- 1. Борьба с затуханием градиента:**
Проблема затухания градиента может затруднить обучение нижних слоев, поскольку градиенты становятся слишком малыми. Остаточные блоки помогают избежать этой проблемы
- 2. Облегчение обучения глубоких моделей:**
Остаточные блоки позволяют строить более глубокие сети без существенного ухудшения производительности или трудностей с обучением

Артефакты

Репозиторий: https://github.com/mcdnmd/image_codec