**Работу выполнил:**

студенты 1 курса магистратуры ИТМО

(группа M4130) Меньщиков Михаил

Отчёт:

**Лабораторная №1**

Дисциплина:

**Теория информации**

**Содержание:**

[1 Постановка задания 3](#_heading=h.2et92p0)

[2 Решение 4](#_heading=h.wxmiiibaqiy5)

[2.1 Общая постановка 4](#_heading=h.3ih9q3qmmj2f)

[2.2 Эксперименты 6](#_heading=h.zafcovvpazqd)

[2.2.1 Эксперимент №1. VQ-VAE with MaxPool 6](#_heading=h.h2c84g84q0lo)

[2.2.2 Эксперимент №2. VQ-VAE without MaxPool 8](#_heading=h.90ujeglh0edi)

[2.2.3 Эксперимент №3. AE with MaxPool 10](#_heading=h.uo235qbvhks1)

[2.2.4 Эксперимент №4. AE without MaxPool 12](#_heading=h.v9ey2y6okjq8)

[2.2.5 Эксперимент №5. AE optuna search 14](#_heading=h.rg2hwt5kkzua)

[2.2.6 Эксперимент №6. AE long train. Final 17](#_heading=h.ya9i7fv17rq)

[2.3 Структура кода 19](#_heading=h.qbbqpynw38ro)

[2.4 Выводы 20](#_heading=h.wq838aghm6xs)

[3 Заключение 21](#_heading=h.hsjdattb970g)

[Список литературы 22](#_heading=h.fka71a9vdi6)

# 

# **1 Постановка задания**

Целью данной лабораторной работы является реализация алгоритма сжатия/декомпрессии изображений с использованием методов из области машинного обучения. За основу предлагается взять код из следующего gitlab-репозитория: <https://ctlab.itmo.ru/gitlab/eabelyaev/cnnimagecodec>. В качестве способа сравнения качества модифицированного решения с базовым предлагается использовать пример, представленный на Рисунке 1.



**Рисунок 1.** Пример сжатия изображений с помощью базового решения

Предложенный вариант реализации должен быть таким, чтобы выполнялось одно из следующих условий:

* на том же сжатии достигается лучшее субъективное визуальное качество;
* на том же сжатии обеспечивается выигрыш по PSNR-метрике от 0.1 дБ и выше;
* при том же уровне качества обеспечивал от 3% выигрыша по BPP-метрике.

На реализацию накладываются следующие ограничения:

* Тестирование полученного решения должно быть проведено на изображениях в каталога *test* из вышеуказанного gitlab-репозитория.
* Обучение ML-модели должно быть проведено только на изображениях в каталоге *train* из вышеуказанного gitlab-репозитория.

В итоге, по результатам проведённой работы необходимо сделать описание использованной модификации и представить сравнительные метрики.

# **2 Решение**

## **2.1 Общая постановка**

Реализованный алгоритм сжатия/декомпрессии изображений является комбинацией скалярного квантования [1], арифметического кодирования [2] и использования частей обученной AutoEncoder-модели [3]. AutoEncoder-модель состоит из двух основных частей: Encoder - переводит изображение в скрытый тензор (обычно меньшей размерности); Decoder - выполняет восстановление изображения по тензору признаков. В зависимости от вида AutoEncoder-модели выделяют также промежуточную LatentLayet-часть, которая выполняет дополнительные преобразования для получения более репрезентативного скрытого тензора признаков. В рамках данной лабораторной были использованы два вида моделей: базовый AE (без промежуточных слоёв) и VQ-VAE [4, 5]. В качестве размерности промежуточного тензора признаков использовался следующий вариант: 16x16x16.

Общий вид Compress-алгоритма для сжатия изображений выглядит следующим образом:

1. Исходное изображение переводится в скрытый тензор с помощью Encoder-части AutoEncoder-модели.
2. В случае VQ-VAE-модели скрытый тензор обогащается признаками с помощью LatentLayer-части.
3. Выполняется скалярное квантование.
4. Выполняется арифметическое кодирование.
5. Сжатое изображение сохраняется в бинарном виде.

Общий вид Decompress-алгоритма для восстановления изображений выглядит следующим образом:

1. Выполняется арифметическое декодирование сжатого изображения.
2. Выполняется обратное скалярное квантование.
3. Из восстановленного тензора признаков выполняется восстановление исходного изображения с помощью Decoder-части AutoEncoder-модели.

Для работы Compression- и Decompression-алгоритмов AutoEncoder-модель была предварительно обучена на задачу восстановления изображений [6] с использование только train-датасета, предложенного в рамках данной лабораторной. Во время обучения перед Decoder-частью к промежуточному тензору дополнительно добавляется шум (из равномерного распределения). Это сделано для того, чтобы адаптировать Decoder-часть к потере/искажению информации в тензоре признаков. Такая техника позволяет сделать Decoder-часть более устойчивой к ошибкам, накопленным с прошлых шагов, при использовании в Decompression-алгоритме. На этапе валидации AutoEncoder-модели использовались следующие метрики для оценки качества восстановления изображений: PSNR, SCC, SSIM, SAM [7].

Выбор оптимальной конфигурации гиперпараметров, на основе которой производилось обучение AutoEncoder-модели был выполнен с помощью программной библиотеки Optuna [8]. Гиперпараметры, для которых выполнялся перебор значений представлены в Таблице 1.

| **N** | **Название** | **Описание** | **Диапазон перебираемых значений** |
| --- | --- | --- | --- |
| 1 | batch\_size | Количество элементов из тренировочной/валидационной выборки, на которых рассчитывается значение функции ошибки за один forward-проход модели. | [2,4,8,16,32,64] |
| 2 | lr | Скорость обучения модели | [0.5,0.1,0.05,0.01,0.05,0.001] |
| 3 | weights\_init | Распределение, которое будет использоваться для инициализации весов модели, перед непосредственным обучением [8] | [‘normal’,’uniform’,’xavier\_normal’, ‘xavier\_uniform’, ‘kaiming\_uniform’, ‘kaiming\_normal’] |
| 4 | act\_fn | Функция активации, которая будет использоваться в моделе | [‘relu’,’leaky’,’gelu’,’silu’] |
| 5 | use\_batchnorm | Если True, то в модель будут добавлены BatchNorm2d-слои [9], иначе нет | [True, False] |
| 6 | use\_residuals | Если True. то в модель будут добавлены ResidualBlock-слои [10], иначе нет | [True, False] |
| 7 | use\_maxpool | Если True, то в модель будут добавлены MaxPool2d-слои [11], иначе нет | [True, False] |
| 8 | residuals\_nlayers | Количество Conv2d-слоёв [12] в ResidualBlock-слое | [1,2] |
| 9 | model\_type | Тип AutoEncoder-модели, которая будет обучена | [‘ae’, ‘vqvae’] |
| 10 | add\_noise | Если True, то перед Decoder-частью к промежуточному тензору признаков будет добавлен равномерный шум, иначе нет | [True, False] |
| 8 | reconstruction\_loss\_weight | Коэффициент первого слагаемого в используемой функции ошибки при обучение AE- и VQ-VAE-моделей | вещественные числа из полуинтервала [1;20) c шагом 1 |
| 9 | codebook\_loss\_weight | Коэффициент второго слагаемого в используемой функции ошибки при обучении VQ-VAE-модели | вещественные числа из полуинтервала [0.2;20) c шагом 0.2 |
| 11 | commitment\_loss\_weight | Коэффициент третьего слагаемого в используемой функции ошибки при обучении VQ-VAE-модели | вещественные числа из полуинтервала [0.01;20) c шагом 0.01 |
| 12 | codebook\_size | Размер кодовой книги у VQ-VAE-модели | целые числа из полуинтервала [20;100) c шагом 5 |
| 13 | b\_quantization | Степень сжатия изображения для скалярного квантования | [2,3] |

**Таблица 1.** Сформированный набор гиперпараметров для обучения AutoEncoder-модели

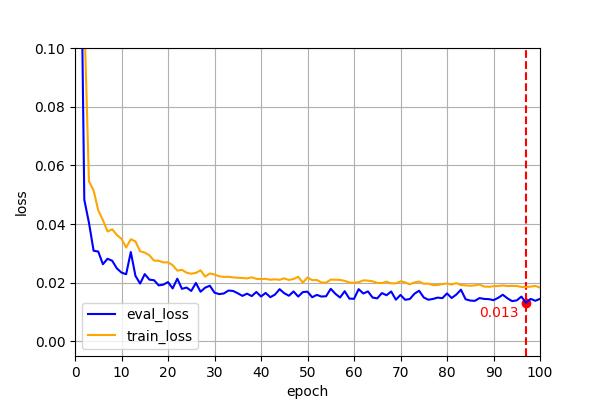
В результате реализации Compression- и Decompression-алгоритмов была выполнена визуальная оценка качества восстановления изображений на нескольких примерах из тестовой выборки.

## **2.2 Эксперименты**

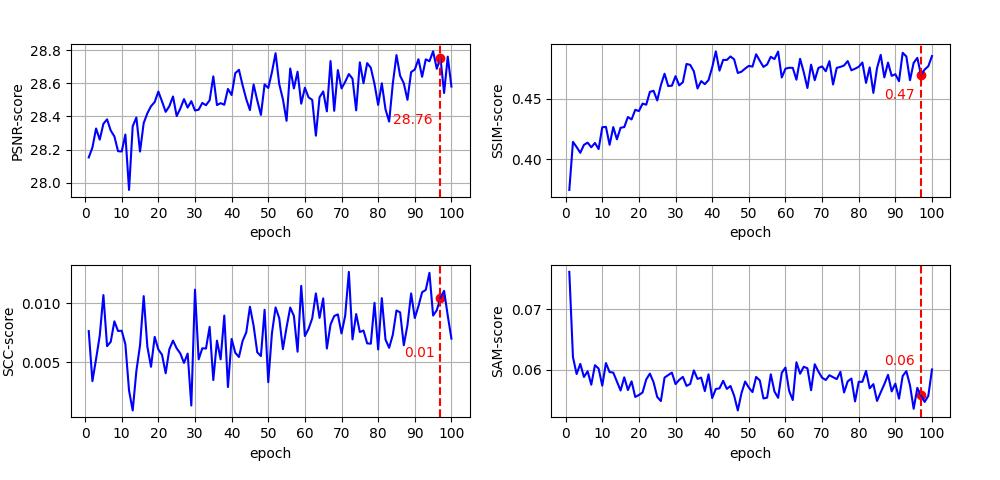
В данном подразделе представлены эксперименты, проведённые в рамках общей постановки решения.

### **2.2.1 Эксперимент №1. VQ-VAE with MaxPool**

В рамках данного эксперимента была обучена VQ-VAE-модель с MaxPool-слоями. Архитектура модели сохранена в github-репозиториии по следующей ссылке: <https://github.com/Dzigen/InfoTheory2024_ITMO_lab1/blob/main/experiments/log1/used_arch.txt>. Для обучения была использована следующая конфигурация гиперпараметров: <https://github.com/Dzigen/InfoTheory2024_ITMO_lab1/blob/main/experiments/log1/used_config.yaml>. На Рисунке 2 представлено изменение значение функции ошибки на тренировочной и валидационной выборках. Лучшая модель по eval\_loss-функции была получена на 97 эпохе обучения: 0.01314. На Рисунке 3 представлено изменение значений метрик качества восстановленных изображений из валидационной выборки. На лучшей эпохе обучения были получены следующие значения: PSNR – 28.76; SSIM – 0.47; SCC – 0.01; SAM – 0.06.

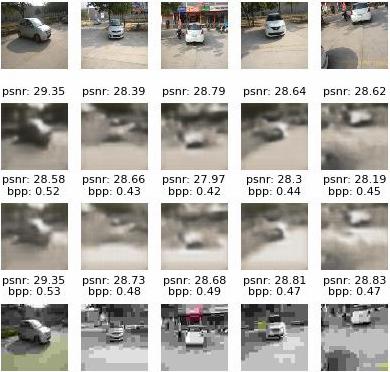


**Рисунок 2.** Изменение функции ошибки при обучении модели в рамках эксперимента №1



**Рисунок 3.** Изменение метрик качества восстановления изображений при обучении модели в рамках эксперимента №1

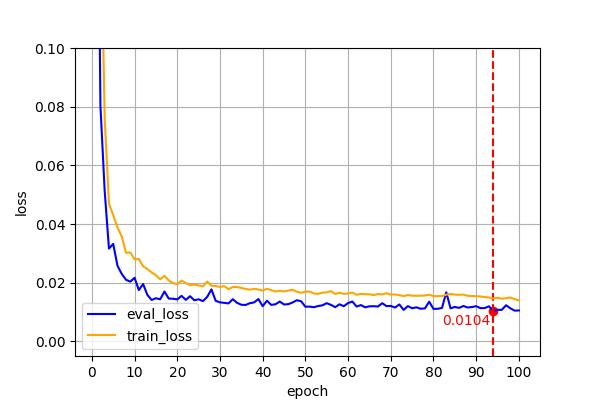
В результате обучения VQ-VAE-модель была интегрирована в алгоритма сжатия/декомпрессии изображений. На Рисунке 4 представлен пример работы алгоритма. В сравнении с JPEG-форматом реализованный алгоритм проигрывает по качеству восстановления (по метрике PSNR) в среднем на 0.55 (~2%), а по качеству сжатия (по метрике BPP) выигрывает на 0.01 (~3%).



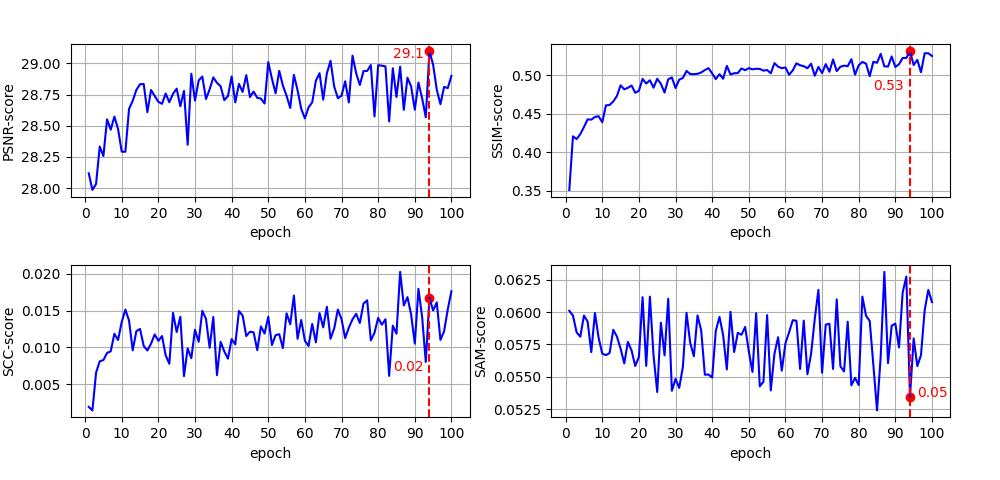
**Рисунок 4.** Пример работы алгоритма сжатия/декомпрессии изображений, полученного в рамках эксперимента №1: (1) первая строка - оригинальные изображения; (2) вторая строка - изображения, восстановленные только с использованием обученной VQ-VAE-модели; (3) третья строка - изображения, восстановленные с использованием полученного алгоритма сжатия/декомпрессии при уровне сжатия b=2 для этапа скалярного квантования; (4) четвёртая строка - изображения, сжатые/восстановленные с помощью JPEG-формата.

### **2.2.2 Эксперимент №2. VQ-VAE without MaxPool**

В рамках данного эксперимента была обучена VQ-VAE-модель без MaxPool-слоёв. Архитектура модели сохранена в github-репозиториии по следующей ссылке: <https://github.com/Dzigen/InfoTheory2024_ITMO_lab1/blob/main/experiments/log2/used_arch.txt>. Для обучения была использована следующая конфигурация гиперпараметров: <https://github.com/Dzigen/InfoTheory2024_ITMO_lab1/blob/main/experiments/log2/used_config.yaml>. На Рисунке 5 представлено изменение значения функции ошибки на тренировочной и валидационной выборках. Лучшая модель по значению eval\_loss-функции была получена на 94 эпохе обучения: 0.0104. На Рисунке 6 представлено изменение значений метрик качества восстановленных изображений из валидационной выборки. На лучшей эпохе обучения были получены следующие значения. На лучшей эпохе обучения были получены следующие значения: PSNR – 29.1 ; SSIM – 0.53 ; SCC – 0.02 ; SAM - 0.05.

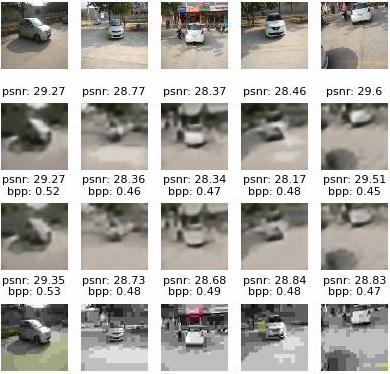


**Рисунок 5.** Изменение функции ошибки при обучении модели в рамках эксперимента №2



**Рисунок 6.** Изменение метрик качества восстановления изображений при обучении модели в рамках эксперимента №2

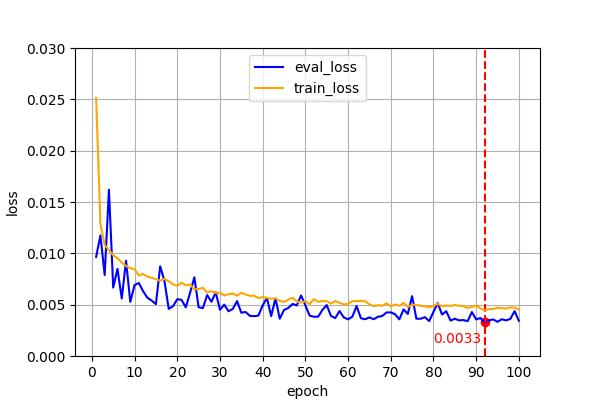
В результате обучения VQ-VAE-модель была интегрирована в алгоритм сжатия/декомпрессии изображений. На Рисунке 7 представлен пример работы алгоритма. В сравнении с JPEG-форматом реализованный алгоритм проигрывает по качеству восстановления (по метрике PSNR) в среднем на 0.06 (~0.2%), а по качеству сжатия (по метрике BPP) выигрывает на 0.01 (~3%).



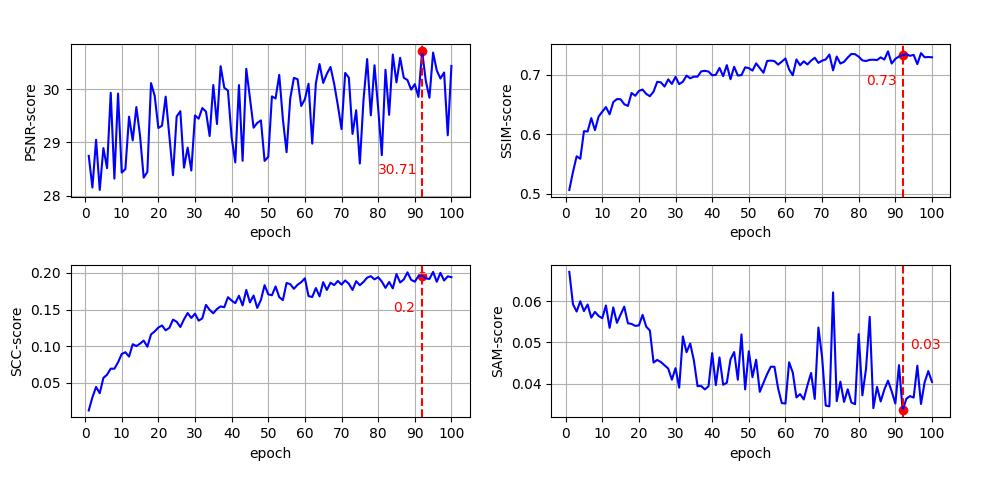
**Рисунок 7.** Пример работы алгоритма сжатия/декомпрессии изображений, полученного в рамках эксперимента №2: (1) первая строка - оригинальные изображения; (2) вторая строка - изображения, восстановленные только с использованием обученной VQ-VAE-модели; (3) третья строка - изображения, восстановленные с использованием полученного алгоритма сжатия/декомпрессии при уровне сжатия b=2 для этапа скалярного квантования; (4) четвёртая строка - изображения, сжатые/восстановленные с помощью JPEG-формата.

### **2.2.3 Эксперимент №3. AE with MaxPool**

В рамках данного эксперимента была обучена AE-модель с MaxPool-слоями. Архитектура модели сохранена в github-репозиториии по следующей ссылке: <https://github.com/Dzigen/InfoTheory2024_ITMO_lab1/blob/main/experiments/log3/used_arch.txt>. Для обучения была использована следующая конфигурация гиперпараметров: <https://github.com/Dzigen/InfoTheory2024_ITMO_lab1/blob/main/experiments/log3/used_config.yaml>. На Рисунке 8 представлено изменение значения функции ошибки на тренировочной и валидационной выборках. Лучшая модель по значению eval\_loss-функции была получена на 92 эпохе обучения: 0.0033. На Рисунке 9 представлено изменение значений метрик качества восстановленных изображений из валидационной выборки. На лучшей эпохе обучения были получены следующие значения: PSNR – 30.71; SSIM – 0.73; SCC – 0.2; SAM – 0.3.



**Рисунок 8.** Изменение функции ошибки при обучении модели в рамках эксперимента №3



**Рисунок 9.** Изменение метрик качества восстановления изображений при обучении модели в рамках эксперимента №3

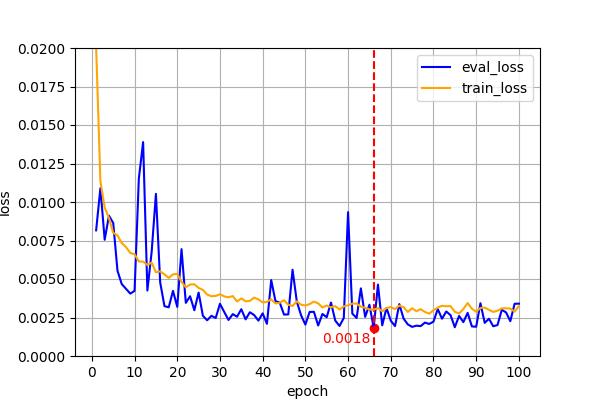
В результате обучения AE-модель была интегрирована в алгоритм сжатия/декомпрессии изображений. На Рисунке 10 представлен пример работы алгоритма. В сравнении с JPEG-форматом реализованный алгоритм проигрывает по качеству восстановления (по метрике PSNR) в среднем на 0.26 (~0.9%), а по качеству сжатия (по метрике BPP) выигрывает на 0.002 (~0.5%).



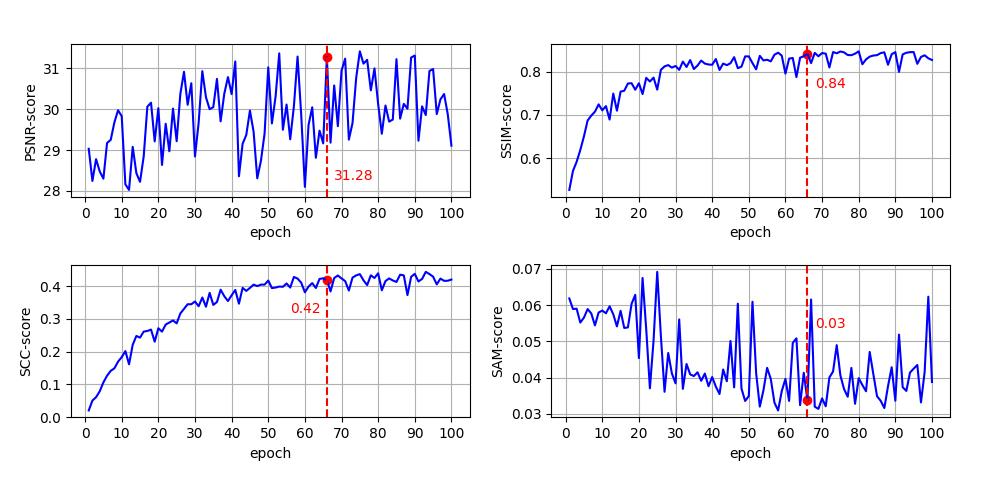
**Рисунок 10.** Пример работы алгоритма сжатия/декомпрессии изображений, полученного в рамках эксперимента №3: (1) первая строка - оригинальные изображения; (2) вторая строка - изображения, восстановленные только с использованием обученной AE-модели; (3) третья строка - изображения, восстановленные с использованием полученного алгоритма сжатия/декомпрессии при уровне сжатия b=2 для этапа скалярного квантования; (4) четвёртая строка - изображения, сжатые/восстановленные с помощью JPEG-формата.

### **2.2.4 Эксперимент №4. AE without MaxPool**

В рамках данного эксперимента была обучена AE-модель без MaxPool-слоёв. Архитектура модели сохранена в github-репозиториии по следующей ссылке: <https://github.com/Dzigen/InfoTheory2024_ITMO_lab1/blob/main/experiments/log4/used_arch.txt>. Для обучения была использована следующая конфигурация гиперпараметров: <https://github.com/Dzigen/InfoTheory2024_ITMO_lab1/blob/main/experiments/log4/used_config.yaml>. На Рисунке 11 представлено изменение значений функции ошибки на тренировочной и валидационной выборках. Лучшая модель по значению eval\_loss-функции была получена на 66 эпохе обучения: 0.0018. На Рисунке 12 представлено изменение значений метрик качества восстановления изображений из валидационной выборки. На лучшей эпохе обучения были получены следующие значения: PSNR – 31.28 ; SSIM – 0.84 ; SCC – 0.42 ; SAM – 0.03 .



**Рисунок 11.** Изменение функции ошибки при обучении модели в рамках эксперимента №4



**Рисунок 12.** Изменение метрик качества восстановления изображений при обучении модели в рамках эксперимента №4

В результате обучения AE-модель была интегрирована в алгоритм сжатия/декомпрессии изображений. На Рисунке 13 представлен пример работы алгоритма. В сравнении с JPEG-форматом реализованный алгоритм проигрывает по качеству восстановления (по метрике PSNR) в среднем на 0.0002 (~0.0007%) и по качеству сжатия (по метрике BPP) на 0.0002 (~0.04%).



**Рисунок 13.** Пример работы алгоритма сжатия/декомпрессии изображений, полученного в рамках эксперимента №4: (1) первая строка - оригинальные изображения; (2) вторая строка - изображения, восстановленные только с использованием обученной AE-модели; (3) третья строки - изображения, восстановленные с использованием полученного алгоритма сжатия/декомпрессии при уровне сжатия b=2 для этапа скалярного квантования; (4) четвёртая строка - изображения, сжатые/восстановленные с помощью JPEG-формата.

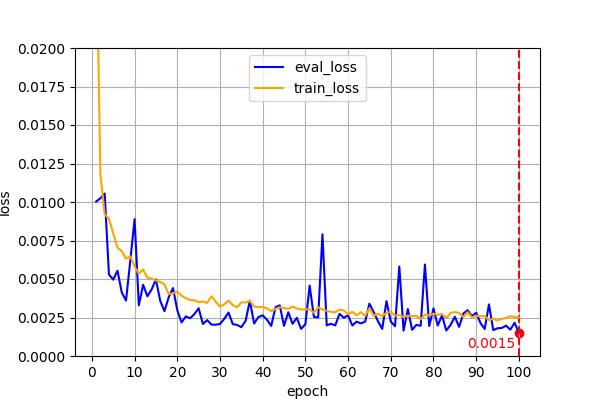
### **2.2.5 Эксперимент №5. AE optuna search**

В рамках данного эксперимента для AE-модели был выполнен перебор гиперпараметров обучения с помощью библиотеки Optuna. При этом были строго заданы два гиперпараметра: в моделе отсутствуют MaxPool-слои (“use\_maxpool”=False); в промежуточный тензор признаков перед подачей на вход к Decoder-части добавляется вектор шума (“add\_noise”=True). Было задано следующее количество попыток для алгоритма поиска: 50. В рамках перебора оценка конфигурации проводилась на 100 эпохах обучения AE-модели. Алгоритм минимизировал лучшее значение eval\_loss-функции при обучении AE-модели на сформированной конфигурации, а также эпоху обучения, на которой это значение ошибки было получено. В результате перебора в Таблице 2 представлена оптимальная конфигурация гиперпараметров для обучения AE-модели со следующими значениями полученных оптимизируемых величин: eval\_loss – 0.00162; best\_epoch – 92.

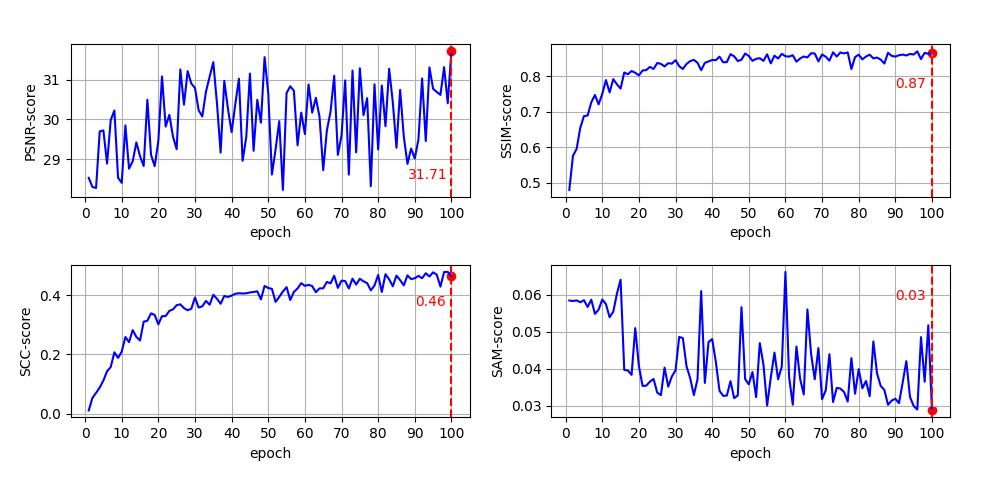
| **N** | **Название** | **Значение** |
| --- | --- | --- |
| 1 | batch\_size | 4 |
| 2 | lr | 0.01 |
| 3 | weights\_init | ‘xavier\_uniform’ |
| 4 | act\_fn | ‘gelu’ |
| 5 | use\_batchnorm | True |
| 6 | use\_residuals | True |
| 7 | residual\_nlayers | 2 |
| 8 | b\_quantization | 3 |
| 9 | reconstructions\_loss\_weight | 1.0 |

**Таблица 2.** Оптимальная конфигурация гиперпараметров, полученная в рамках эксперимента №5

Таким образом, на полученной конфигурации была обучена AE-модель (на 100 эпохах). Архитектура модели сохранена в github-репозиториии по следующей ссылке: <https://github.com/Dzigen/InfoTheory2024_ITMO_lab1/blob/main/experiments/log6/used_arch.txt>. На Рисунке 14 представлено изменение значения функции ошибки на тренировочной и валидационной выборках. Лучшая модель по значению eval\_loss-функции была получена на 100 эпохе обучения: 0.0015. На Рисунке 15 представлено изменение значений метрик качества восстановления изображений из валидационной выборки. На лучшей эпохе обучения были получены следующие значения: PSNR – 31.71; SSIM – 0.87; SCC – 0.46; SAM – 0.03.



**Рисунок 14.** Изменение функции ошибки при обучении модели в рамках эксперимента №5



**Рисунок 15.** Изменение метрик качества восстановления изображений при обучении модели в рамках эксперимента №5

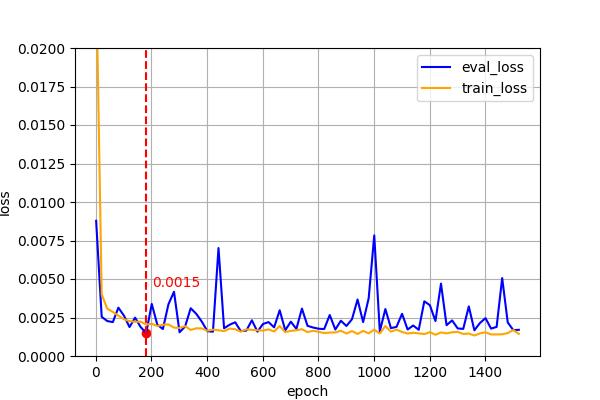
В результате обучения AE-модель была интегрирована в алгоритма сжатия/декомпрессии изображений. На Рисунке 16 представлен сравнительный пример работы алгоритма. В сравнении с JPEG-форматом реализованный алгоритм проигрывает по качеству восстановления (по метрике PSNR) в среднем на 1.4 (~4.5%), а по качеству сжатия (по метрике BPP) выигрывает на 0.001 (~0.24%).



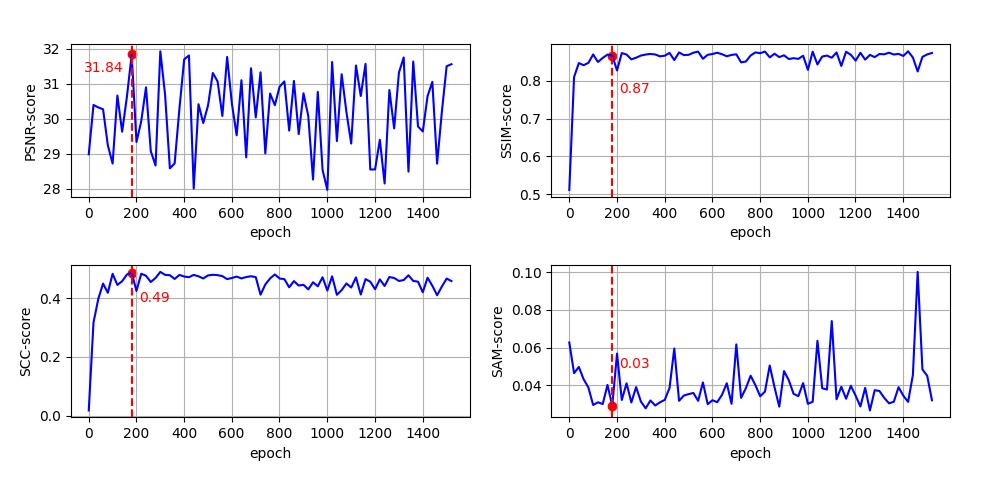
**Рисунок 16.** Пример работы алгоритма сжатия/декомпрессии изображений, полученного в рамках эксперимента №5: (1) первая строка - оригинальные изображения; (2) вторая строка - изображения, восстановленные только с использованием обученной AE-модели; (3) третья строки - изображения, восстановленные с использованием полученного алгоритма сжатия/декомпрессии при уровне сжатия b=3 для этапа скалярного квантования; (4) четвёртая строка - изображения, сжатые/восстановленные с помощью JPEG-формата.

### **2.2.6 Эксперимент №6. AE long train. Final**

В рамках данного эксперимента была использована конфигурация гиперпараметров, полученная в рамках эксперимента №5 для обучения AE-модели на большем количестве эпох: ~1500 эпох. Архитектура модели сохранена в github-репозиториии по следующей ссылке: <https://github.com/Dzigen/InfoTheory2024_ITMO_lab1/blob/main/experiments/log7/used_arch.txt>. На Рисунке 16 представлено изменение значений функции ошибки на тренировочной и валидационной выборках. Лучшая модель по значению eval\_loss-функции была получена на 188 эпохе обучения: 0.00145. На Рисунке 17 представлено изменение значений метрик качества восстановления изображений из валидационной выборки. На лучшей эпохе обучения были получены следующие значения: PSNR – 31.84 ; SSIM – 0.87 ; SCC – 0.49 ; SAM – 0.03 .

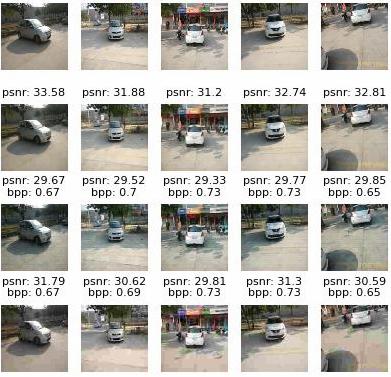


**Рисунок 16.** Изменение функции ошибки при обучении модели в рамках эксперимента №6



**Рисунок 17.** Изменение метрик качества восстановления изображений при обучении модели в рамках эксперимента №6

В результате обучения AE-модель была интегрирована в алгоритм сжатия/декомпрессии изображений. На Рисунке 13 представлен пример работы алгоритма. В сравнении с JPEG-форматом реализованный алгоритм проигрывает по качеству восстановления (по метрике PSNR) в среднем на 0.88 (~2.8%) и по качеству сжатия (по метрике BPP) на 0.0003 (~0.05%).



**Рисунок 18.** Пример работы алгоритма сжатия/декомпрессии изображений, полученного в рамках эксперимента №6: (1) первая строка - оригинальные изображения; (2) вторая строка - изображения, восстановленные только с использованием обученной AE-модели; (3) третья строки - изображения, восстановленные с использованием полученного алгоритма сжатия/декомпрессии при уровне сжатия b=3 для этапа скалярного квантования; (4) четвёртая строка - изображения, сжатые/восстановленные с помощью JPEG-формата.

## **2.3 Структура кода**

Исходный код полученного решения и логи проведённых экспериментов находятся в следующем github-репозитории: <https://github.com/Dzigen/InfoTheory2024_ITMO_lab1>. Структура корневой директории выглядит следующим образом:

* *data/* – каталог с данным для обучения и тестирования/валидации алгоритма сжатия/восстановления изображения.
* *experiments/* – каталог с логами/результатами экспериментов по поиску оптимальной конфигурации алгоритма сжатия/декомпрессии изображений.
* *notebooks/* – каталог с jupyter-блокнотами для интерактивной отладки исходного кода и визуализации полученных результатов.
* *notebooks/inference.ipynb* – jupyter-блокнот для запуска/тестирования реализованного алгоритма сжатия/декомпрессии изображений.
* *src/* – каталог с исходным кодом цикла обучения/валидации VQ-VAE-модели и предложенного алгоритма сжатия/декомпрессии.
* *main.py* – python-скрипт для запуск процесса обучения VQ-VAE модели на конкретной конфигурации гиперпараметров из файла params.yaml.
* *optuna\_search.py* – python-скрипт для запуска процесса поиска оптимальной конфигурации гиперпараметров (для обучения VQ-VAE-модели) с помощью библиотеки Optuna.
* *params.yaml* – конфигурационный файл для обучения VQ-VAE-модели.

Структура директории *src/* выглядит следующим образом:

* *architectures/VQVAE.py* – реализация VQ-VAE-архитектуры.
* *architectures/AE.py* – реализация AE-архитектуры.
* *codec/* – каталог с реализацией арифметического кодера/декодера
* *modules/* – каталог с компонентами реализованных нейросетевых архитектур.
* *compression.py* – реализация алгоритма сжатия/декомпресии изображений.
* *metrics.py* – реализация метрик для оценки качества восстановления изображений.
* *settings.py* – класс для загрузки конфигурационного файла для обучения AutoEncoder-моделей.
* *trainloop.py* – реализация циклас обучения/валидации AutoEncoder-моделей.
* *utils.py* - всопомогательный функционал для цикла обучения и оценки алгоритма сжатия/декомпрессии.

Структура директории experiments/ выглядит следующим образом:

* *logs1/* – каталог с логами и результатами, полученные в рамках эксперимента №1.
* logs2/ – каталог с логами и результатами, полученные в рамках эксперимента №2.
* logs3/ – каталог с логами и результатами, полученные в рамках эксперимента №3.
* logs4/ – каталог с логами и результатами, полученные в рамках эксперимента №4.
* logs6/ – каталог с логами и результатами, полученные в рамках эксперимента №5.
* logs7/ – каталог с логами и результатами, полученные в рамках эксперимента №6.

## **2.4 Выводы**

По результатам проведённых экспериментов была построена сравнительная Таблица 3.

| **N** | **Степень сжатия (b)** | **Тип AutoEncoder-модели** | **Архитектурные особенности** | **Количество эпох обучения** | **Полученный алгоритма** | | **JPEG-формат** | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **PSNR** | **BPP** | **PSNR** | **BPP** |
| 1 | 2 | VQVAE | with MaxPool | 100 | 28.36 | 0.46 | 28.91 | 0.49 |
| 2 | 2 | VQVAE | w/o MaxPool | 100 | 28.90 | 0.48 | 28.96 | 0.49 |
| 3 | 2 | AE | with MaxPool | 100 | 28.73 | 0.50 | 28.99 | 0.50 |
| 4 | 2 | AE | w/o MaxPool | 100 | 28.95 | 0.50 | 28.95 | 0.50 |
| 5 | 3 | AE | w/o MaxPool | 100 | 29.06 | 0.68 | 30.47 | 0.68 |
| 6 | 3 | AE | w/o MaxPool | ~1500 | 29.59 | 0.68 | 30.47 | 0.68 |

**Таблица 3**. Сравнение результатов проведённых экспериментов в рамках данной лабораторной работы

Из Таблицы 3 видно, что VQ-VAE-модель показывает себя лучше на сжатии (по метрике BPP) по сравнению с AE-моделью. В свою очередь AE-модель показывает похожее качество сжатия изображений по сравнению с JPEG-форматом и при этом имеет менее хорошее качество восстановления (по метрике PSNR). Увеличение эпох обучения AE-модели не дало значительного прироста по метрикам качества восстановления изображений.

В рамках дальнейших экспериментов предлагается увеличить количество эпох обучения для VQ-VAE модели и выполнить оценку качества сжатия/восстановления изображений.

# **3 Заключение**

В результате выполнения данной лабораторной работы был реализован алгоритм сжатия/декомпрессии изображений с использованием методов из области машинного обучения. За основу был взять код из следующего gitlab-репозитория: <https://ctlab.itmo.ru/gitlab/eabelyaev/cnnimagecodec>. В качестве способа сравнения качества модифицированного решения с базовым был использован пример, представленный на Рисунке 1.Предложенный вариант реализации удовлетворяет следующим условиям:

* при том же уровне качества обеспечивал от 3% выигрыша по BPP-метрике.

В процессе реализации/сравнения алгоритма были соблюдены следующие требования:

* Тестирование полученного решения должно быть проведено на изображениях в каталога *test* из вышеуказанного gitlab-репозитория.
* Обучение ML-модели должно быть проведено только на изображениях в каталоге *train* из вышеуказанного gitlab-репозитория.

В итоге, по результатам проведённой работы в рамках данного отчёта было выполнено описание использованной модификации и представлены сравнительные метрики.

# **Список литературы**

1. Квантование (обработка сигналов) [Электронный ресурс]. – 2024. – URL: <https://en.wikipedia.org/wiki/Quantization_(signal_processing)> (дата обращения 10.05.2024)
2. Witten I. H., Neal R. M., Cleary J. G. Arithmetic coding for data compression //Communications of the ACM. – 1987. – Т. 30. – №. 6. – С. 520-540.
3. Van Den Oord A. et al. Neural discrete representation learning //Advances in neural information processing systems. – 2017. – Т. 30.
4. Understanding Vector Quantized Variational Autoencoders (VQ-VAE) [Электронный ресурс]. – 2024. – URL: <https://shashank7-iitd.medium.com/understanding-vector-quantized-variational-autoencoders-vq-vae-323d710a888a> (дата обращения 10.05.2024)
5. VQ-VAE | Everything you need to know about it | Explanation and Implementation [Электронный ресурс]. – 2024. – URL: <https://www.youtube.com/watch?v=1ZHzAOutcnw> (дата обращения 10.05.2024)
6. How to Evaluate Image Quality in Python: A Comprehensive Guide [Электронный ресурс]. – 2024. – URL: <https://medium.com/@jaikochhar06/how-to-evaluate-image-quality-in-python-a-comprehensive-guide-e486a0aa1f60> (дата обращения 10.05.2024)
7. Optuna: A hyperparameter optimization framework [Электронный ресурс]. – 2024. – URL: <https://optuna.readthedocs.io/en/stable/index.html> (дата обращения 10.05.2024)
8. torch.nn.init — PyTorch 2.3 documentation [Электронный ресурс]. – 2024. – URL: <https://pytorch.org/docs/stable/nn.init.html> (дата обращения 10.05.2024)
9. BatchNorm2d — PyTorch 2.3 documentation [Электронный ресурс]. – 2024. – URL: <https://pytorch.org/docs/stable/generated/torch.nn.BatchNorm2d.html> (дата обращения 10.05.2024)
10. Building a Customized Residual CNN with PyTorch [Электронный ресурс]. – 2024. – URL: <https://medium.com/@chen-yu/building-a-customized-residual-cnn-with-pytorch-471810e894ed> (дата обращения 10.05.2024)
11. MaxPool2d — PyTorch 2.3 documentation [Электронный ресурс]. – 2024. – URL: <https://pytorch.org/docs/stable/generated/torch.nn.MaxPool2d.html> (дата обращения 10.05.2024)
12. Conv2d — PyTorch 2.3 documentation [Электронный ресурс]. – 2024. – URL: <https://pytorch.org/docs/stable/generated/torch.nn.Conv2d.html> (дата обращения 10.05.2024)