

МИНОБРНАУКИ РОССИИ  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
**«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ  
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
ИМЕНИ Н. Г. ЧЕРНЫШЕВСКОГО»**

Кафедра математической кибернетики и компьютерных наук

**СРАВНЕНИЕ МОДЕЛЕЙ СЖАТИЯ ИЗОБРАЖЕНИЙ**

**АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ**

студента 4 курса 451 группы  
направления 09.03.04 — Программная инженерия  
факультета КНиИТ  
Ланкина Ивана Сергеевича

Научный руководитель  
доцент, к. ф.-м. н.

\_\_\_\_\_

А. С. Иванова

Заведующий кафедрой  
к. ф.-м. н., доцент

\_\_\_\_\_

С. В. Миронов

## СОДЕРЖАНИЕ

ВВЕДЕНИЕ .....	3
1 Теоретическая часть .....	4
1.1 Сжатие с потерями .....	4
1.2 Сжатие без потерь .....	4
1.3 Сжатие почти без потерь .....	5
1.4 Другие методы сжатия .....	5
1.5 Нейронные сети .....	7
2 Программа и результаты .....	9
2.1 Сжатие .....	9
2.2 Разжатие .....	10
2.3 Интерфейс .....	11
2.4 Сравнение результатов по метрикам .....	12
ЗАКЛЮЧЕНИЕ .....	13

## ВВЕДЕНИЕ

Сжатие данных важно для хранения и передачи информации. Для большинства типов данных необходимо сжатие без потерь, то есть из сжатого файла можно получить обратно точно такие же данные, как и в исходном. Для мультимедийных файлов, таких как изображения, можно использовать сжатие с потерями, так как необходимо только визуальное сходство данных. Для человеческого глаза не имеет большого значения, если из сжатого файла будут получены цвета, RGB-значения которых отличаются от исходного всего на несколько единиц.

Модели сжатия изображения находятся в постоянной разработке. С каждым годом создаются новые методы для того, чтобы сохранить качество изображения в наилучшем формате. В данной работе предлагается реализовать приложение, использующее программное обеспечение Stable Diffusion как инструмент для сжатия изображений.

Целью данной работы является изучение различных алгоритмов сжатия изображений и сравнение их друг с другом по точности.

В результате выполнения данной работы должны быть решены следующие задачи:

- изучение различных моделей сжатия изображений и их алгоритмов;
- реализация собственного метода сжатия в приложении с графическим интерфейсом;
- сравнение этих методов по эффективности.

## **1 Теоретическая часть**

### **1.1 Сжатие с потерями**

Методы с потерями очень полезны для сжатия изображений. Они могут обмануть человеческий глаз и уменьшить количество деталей без ущерба для общего впечатления. Сжатие с потерями подразумевает, что некоторая часть информации будет упрощена или изменена. Ключевым моментом является баланс между качеством и размером изображения. Большинство методов преобразуют изображение в дискретный спектр, обрезают и упрощают коэффициенты, затем они сжимают данные без потерь. Яркими представителями сжатия с потерями являются дискретное косинусное преобразование, преобразование Карунена-Лоэва и преобразование на основе вейвлетов.

Ярким представителем дискретного косинусного преобразования является формат сжатия JPEG. Стандарт JPEG-1 (ISO/IEC 10918) был установлен в 1992 году (обновленная версия, 1994) после его разработки, которая началась в 1986 году. Хотя этот стандарт часто рассматривается как единая спецификация, на самом деле он состоит из четырех разных частей и комбинации режимов кодирования.

1. JPEG-1 (ISO/IEC 10918-1) определяет основную технологию кодирования и включает множество вариантов кодирования фотографических изображений;
2. Описывает тестирование на соответствие;
3. Описывает набор расширений технологий кодирования первой части, и посредством поправки был добавлен формат файла SPIFF;
4. Занимается регистрацией профилей JPEG-1, профилей SPIFF, тегов SPIFF, цветовых пространств SPIFF, типов сжатия SPIFF и определяет Регистрационные органы;
5. Определяет формат обмена файлами JPEG (JFIF).

Можно с уверенностью сказать, что JPEG-1 является одним из самых успешных мультимедийных стандартов, определенных до сих пор.

### **1.2 Сжатие без потерь**

Методы сжатия без потерь работают для любого типа данных. Теоретически, можно использовать любой алгоритм сжатия без потерь для изображений. Но на практике это не очень эффективно. Некоторые методы лучше

подходят для изображений, чем другие. Можно разделить их на различные категории, основываясь на том, как они работают. Существует три основных типа — словарные методы, методы прогнозирования и преобразования на основе вейвлетов.

### **1.3 Сжатие почти без потерь**

Методы почти без потерь — это особый тип сжатия. Они допускают небольшую потерю качества, которую трудно заметить. Их PSNR обычно очень высок. Например, если два пикселя рядом друг с другом имеют значения 129, 130, то распакованные значения могут быть 130, 130. Эти методы часто основаны на версиях без потерь. Некоторые методы, основанные на предсказании, хороши для сжатия почти без потерь. Они могут использовать предсказание с некоторой потерей качества. Примерами являются измененные предиктор MED и алгоритм LOCO-I. Были предложены несколько представлений данного метода:

- В 2007 предложили метод, который использует только предсказание и граф ошибок предсказания. Для каждой строки изображения строится граф. На гранях в качестве меток указывается ошибка предсказания. Затем находятся ребра с одинаковыми или похожими значениями ошибок. Для этих значений используется сжатие RLE;
- В 2012 был предложен другой метод. Нужно поделить пиксели на удвоенное количество битов потерь и округляют их. Алгоритм представляет пиксели, используя различные порядки сканирования (зиг-заг, змейка и другие). Здесь хранится и сжимается только разница между значениями с помощью кодирования Хаффмана. Используется перебор для выбора лучшего порядка сканирования. Перебираются все комбинации и выбирается лучшая.

### **1.4 Другие методы сжатия**

Существуют и другие методы, которые были изучены. Один из них основан на кубической сплайн-интерполяции изображения. Он берет только несколько пикселей изображения и интерполирует их кубическим сплайном. Контрольные точки сплайна выбираются так, чтобы минимизировать ошибку аппроксимации. Один из способов сделать это — метод бисекций. Он прекращает деление участка, если ошибка меньше порогового значения. Он изменя-

ет значения неконтрольных точки на пустые и сохраняет изображение. При реконструкции он вычисляет недостающие значения с помощью кубической сплайн-интерполяции. Эта техника протестирована на изображениях RGB, но она работает и для градаций серого.

Структуры основанные на квадродеревьях также могут быть использованы для сжатия изображений. Один из таких алгоритмов делит лист квадродерева на основе его анализа. Он анализирует квантованную гистограмму блока различий. В гистограмме находят пики, превышающие порог. Количество пиков используется для классификации блока. Если блок имеет низкую детализацию, сохраняется его среднее значение, в противном случае он деление блока продолжается. Если блок имеет размер  $4 \times 4$ , он прекращает деление. Вместо этого он сохраняет блок с двоичной маской. Он выбирает лучшую маску из predetermined шаблонов.

Словарные методы распространены в сжатии изображений без потерь. Однако, существуют и сжатие с потерями на основе словарных методов. Основная идея таких методов заключается в том, чтобы сделать более длинные последовательности с одинаковым значением с помощью квантизации. Если есть последовательность данных 159, 160, 159, 157, 170. Она квантизируется и получается последовательность 159, 159, 159, 159, 159, 170. Такие данные легче сжать с помощью словарного метода. Другой вариант — использовать сходство блоков. Если ошибка различия двух блоков меньше порога, можно заменить один блок другим. Это также улучшает сжатие.

Многие сжатия могут быть улучшены с помощью обновления гистограммы на этапе предварительной обработки. Были предложены методы использующие контурное сжатие. Они отслеживают контуры с одинаковым значением яркости пикселей и сохраняют контуры с помощью алгоритма RLE и энтропийного кодирования.

WebP — это новый алгоритм сжатия изображений, который был создан компанией Google и другими участниками с открытым исходным кодом в 2013 году. Его цель — уменьшить размер изображений в Интернете по сравнению с JPG, сохранив при этом тот же уровень качества изображения.

WebP создан на основе алгоритма сжатия видео VP8 (который также используется для видеοформата WebM). Одна из ключевых особенностей VP8 заключается в том, что он может сжимать каждый отдельный кадр видео, а

также разницу между последовательными кадрами. WebP использует ту же основную технологию, что и WebM, но в нем также добавлены некоторые новые функции и обновления, характерные для него как формата файлов изображений.

## 1.5 Нейронные сети

Нейронная сеть — это продукт объединения исследований в области нейронауки и математики. Она отлично справляется с задачами нелинейного преобразования и классификации. Сеть состоит из множества слоев простых единиц, называемых нейронами (перцептронами). Эти единицы взаимодействуют друг с другом посредством взвешенных связей. Нейроны активируются взвешенными связями от предыдущих нейронов. Чтобы сделать сеть нелинейной, для всех промежуточных слоев используются функции активации. Базовая структура нейронной сети имеет один входной слой, один выходной слой и несколько скрытых слоев. Каждый слой имеет разное количество нейронов.

В 1960-х годах был разработан и изучен метод обучения для простого перцептрона. Позже, в 1970-х и 1980-х годах, для обучения многослойного перцептрона (MLP) была введена методика обратного распространения, основанная на цепном правиле производных для целей обучения. Многослойные структуры в основном обучались методом стохастического градиентного спуска с обратным распространением, хотя он был вычислительно дорогим и склонным к плохим локальным минимумам. Однако плотные связи между соседними слоями в нейронных сетях приводили к квадратичному росту числа параметров модели и ограничивали вычислительную эффективность нейронных сетей. Благодаря применению совместного использования параметров для многослойного перцептрона в 1990 году была предложена более легкая версия нейронной сети, названная сверточной нейронной сетью (CNN), которая использовалась для распознавания документов, что позволило обучать нейронные сети большого масштаба.

Среди различных систем кодирования основными методами сжатия изображений являются преобразование и предсказание. JPEG — самый популярный стандарт сжатия изображений, который состоит из основных модулей преобразования и предсказания. В JPEG входное изображение делится на  $8 \times 8$  непрерывных блоков, каждый из которых преобразуется в частотную область с помощью блочного дискретного косинусного преобразования (БДКП). Для

каждого преобразованного блока коэффициенты дискретного косинусного преобразования затем сжимаются в двоичный поток с помощью квантизации и энтропийного кодирования. В таких стандартах кодирования изображений на основе блоков сжатие обычно зависит от блока и должно выполняться последовательно блок за блоком, что ограничивает параллелизм сжатия при использовании платформы параллельных вычислений, например, GPU. Более того, независимая стратегия оптимизации для каждого отдельного инструмента кодирования также ограничивает улучшение производительности сжатия по сравнению со сжатием со сквозной оптимизацией. Существует другая траектория технологического развития, основанная на нейросетевых методах для сжатия изображений.

В данной работе для сжатия изображения будет использоваться Stable Diffusion, который был разработан для создания изображения по текстовым описаниям. Сам Stable Diffusion состоит и использует для генерации изображения 3 обученные нейронные сети:

- Variational Autoencoder;
- U-Net;
- Text Encoder;

## 2 Программа и результаты

Для написания программы использовался язык Python 3. Были использованы библиотеки для взаимодействия с нейронными сетями и для написания интерфейса приложения. Среди использованных библиотек были такие как:

- `diffusers` — для подключения работы Stable Diffusion;
- `torch` — для организации работы с нейронными сетями;
- `PIL` — для работы с изображениями;
- `skimage.metrics` — для подключения метрик PSNR и SSIM;
- `os` — для работы с файлами ОС;
- `shutil` — для создания репозиторий;
- `io` — для работы с потоками ввода-вывода;
- `PyQT5` — для создания интерфейса;
- `numpy` — для создания файла npz и разархивации из него;
- `libimagequant` — для преобразования изображений и дизеринга;
- `mozjpeg_lossless_optimization` — для оптимизации изображений JPEG;
- `matplotlib.pyplot` — для создания графиков.

Программа состоит из нескольких частей:

- `compress.py` — модуль, который реализует сжатие файла(ов) в npz-архивы;
- `open_npz.py` — реализация разжатия изображения из архива;
- `interface.py` — интерфейс, реализация взаимодействия программы с пользователем;
- `model.py` — модуль с методами и загруженной нейронной сетью, которые используются в остальных файлах.

Программа поддерживает ОС Windows, Linux и MacOS. Минимальные требования по использованию проекта: 8 ГБ ОЗУ и GPU NVIDIA с 6 ГБ ОЗУ. Приложение поддерживает открытие файлов JPG, WEBP и PNG для изображений и NPZ и WEBP для сжатых данных. Для реализации использовалась версия Stable Diffusion 1.5, взятая с ресурса HuggingFace. Для подключения работы Stable Diffusion будет использоваться класс `StableDiffusionImg2ImgPipeline`.

### 2.1 Сжатие

Для сжатия использовались 3 метода сжатия: JPEG, WEBP lossy и сжатие через Stable Diffusion.

Для сжатия через Stable Diffusion используется VAE. Далее выполняется

следующий алгоритм:

1. Считывается исходное изображение через PIL.Image;
2. При помощи VAE изображение переводится в скрытое пространство;
3. Данные скрытого пространства квантизируются;
4. Полученный массив проходит дизеринг для уменьшения негативного эффекта от квантования;
5. Далее полученный результат квантования и дизеринга преобразуется в numpy массивы;
6. С помощью библиотеки numpy данные сохраняются в prz-файл.

Для сжатия JPEG и WEBP используется стандартный метод библиотеки PIL.Image save(), в котором подбором определяется до какой степени можно сжать изображение, чтобы оно не превышало размер полученного prz-файла. Для этого в каждом случае подбирается параметр quality, который может принимать значения от 0 до 95 для JPEG и от 0 до 80 для WEBP. Чем меньше значение, тем больше сжатие и меньше размер картинки.

Для JPEG включается оптимизация и настраивается подвыборка. Так как сжатие JPEG нужно только для сравнения, эти значения фиксированы. Для лучшего качества изображения применяется оптимизация через библиотеку mozjpeg\_lossless\_optimization.

Для WEBP параметр method, который может принимать значения от 0 до 6, отвечающий за соотношение качество/время, устанавливается максимально в сторону качества.

## 2.2 Разжатие

Разархивация происходила для файла prz или сохраненных данных скрытого пространства в виде файла WEBP. Так как исходные файлы весом примерно 500-600 КБ, в виде данных скрытого пространства, имеют размер примерно 15 КБ, а prz-файлы 5 КБ, то оба формата можно считать сжатой версией исходного изображения. Разжатие из файлов происходит следующим образом:

1. Если на вход получены данные скрытого пространства, то файл открывается как изображение в формате RGBA;
2. Если на вход получен архив prz, то файл разархивируется с помощью библиотеки numpy, полученные данные преобразуются в изображение в формате RGBA;

3. Полученное изображение преобразуется в массив данных и после де-квантизации снова преобразуется в изображение;
4. Для полученного изображения определённое количество раз выполняется операция удаления шума, после чего оно сохраняется.

В операции избавления от шума используется нейронная сеть U-net с настраиваемыми параметрами.

### 2.3 Интерфейс

Для графического интерфейса использовался язык Python и преимущественно библиотека PyQT5.

Старт программы начинается с начального экрана. Далее можно открыть файл или папку с файлами. Для различных файлов можно сделать сжатие или разжатие файла. Если выбраны форматы PNG, JPG, WEBP, то следует выбрать сжатие файла для последующих действий. Если выбраны файлы, которые являются архивированными данными, такие как NPZ или WEBP (с предварительной настройкой параметров программы), то следует провести разжатие этих файлов.

После сжатия файл или группа файлов по умолчанию сохранится в подпапку «output» в текущей репозитории. Папку сохранения сжатых файлов можно изменить с помощью кнопки «Save». В данную папку сохраняются файлы сжатия в форматах NPZ и WEBP с другими промежуточными изображениями. После разжатия файл или группа файлов сохранится в текущую папку, в которой находились файлы для разархивации. Для сохранения разархивированных изображений используется формат PNG.

При открытии изображения оно появляется в приложении справа от основного окна. После сжатия появляются изображения в формате JPG и WEBP сжатые до того же размера, что и prz-файл. При разжатии prz-файла или webp-файла разархивированное изображение открывается справа от исходного или справа от основного окна, если исходное изображение не открыто.

У приложения существуют 4 настройки:

- Выбор формата файла, из которого будет производиться разжатие. Этот параметр необходим для правильного считывания файлов для разжатия, в частности, для понимания программой, является webp-файл файлом для сжатия или разжатия, или для понимания какие файлы какого формата следует выбирать при разархивации папки со сжатыми файлами, если

таковых форматов там несколько;

- Выбор размера шага для операции удаления шума из изображения;
- Сила с которой будет использоваться удаление шума из изображения;
- Количество этапов удаления шума.

Данные настройки можно комбинировать между собой для получения наилучшего качества изображения. При запуске программы эти параметры уже заданы оптимальными значениями для большинства изображений. Открытые в программе изображения при закрытии можно открыть повторно в тех же окнах при нажатии на соответствующие кнопки интерфейса. При нажатии на крестик программа запросит подтверждение о выходе из приложения.

## **2.4 Сравнение результатов по метрикам**

В данной работе используются метрики PSNR и SSIM и приведены результаты сравнения по ним. Для измерения метрик были написаны отдельные функции `decompress_file_with_metrics` и `decompress_many_with_metrics`. Для проверки моделей на качество чаще всего используют 2 пакета изображений: Set5 и Set14. Первый представляет собой 5 изображений («ребенок», «птица», «бабочка», «голова», «женщина»). Второй — 14 изображений.

## ЗАКЛЮЧЕНИЕ

В данной работе было написано приложение с графическим интерфейсом для отображения результатов работы метода сжатия через Stable Diffusion на основе языка Python 3 и библиотеки PyQt5. В приложении также показаны результаты сжатий JPEG и WEBP. Была использована модель Stable Diffusion 1.5, код которой есть в открытом доступе. Было замечено, что показания метрик на результатах написанной программы сильно зависят от разнообразия данных в изображениях.

Была изучена литература по различным методам сжатия изображений и изучены их возможности. Были описаны и продемонстрированы все функции и работа приложения на тестовых изображениях.

В ходе написания работы были изучены и решены следующие задачи:

- Изучены различные модели сжатия изображения;
- Написано приложение с использованием 3х методов сжатия изображения (jpeg, webp, Stable Diffusion);
- Приведено сравнение всех методов.