

СРАВНИТЕЛЬНЫЙ АНАЛИЗ РАЗЛИЧНЫХ ФОРМАТОВ ИЗОБРАЖЕНИЯ В КОНТЕКСТЕ ВЕБ ПРИЛОЖЕНИЯ

COMPARATIVE ANALYSIS OF DIFFERENT IMAGE FORMATS IN THE CONTEXT OF A WEB APPLICATION

**B. Goryachkin
D. Mozhaev
A. Andrianov**

Summary: The article is devoted to the issues of converting images into the most economical in size and high-quality formats for the human eye. The formats are compared in terms of the compression ratio and the quality of the resulting image for the human eye, which makes it much easier to choose one or another image format for solving applied problems. An experiment was conducted to compress and evaluate the quality of the most popular types of images among users on the Internet. The analysis of situations when it is worth using one or another image format is carried out.

Keywords. Format, image, image optimization, image compression, image quality, image download speed.

Горячкин Борис Сергеевич

кандидат технических наук, доцент,
Московский государственный технический
университет им. Н.Э. Баумана
bsgor@mail.ru

Можаев Дмитрий Владимирович

Московский государственный технический
университет им. Н.Э. Баумана
mdv413@mail.ru

Андрианов Алексей Александрович

Московский государственный технический
университет им. Н.Э. Баумана
alks.andrnv@gmail.com

Аннотация. Статья посвящена вопросам перевода изображения в максимально экономные по размеру и качественные для человеческого глаза форматы. Проведено сравнение форматов по степени сжатия и качеству получившегося изображения для человеческого глаза, что позволяет намного проще выбирать тот или иной формат изображения для решения прикладных задач. Проведен эксперимент по сжатию и оценке качества наиболее популярных типов изображений у пользователей в интернете. Проведен анализ ситуаций, когда стоит использовать тот или иной формат изображения.

Ключевые слова. Формат, изображение, оптимизация изображения, сжатие изображения, качество изображения, скорость загрузки изображения.

Введение

При создании приложений с подгрузкой большого количества графической информации возникает проблема: как сэкономить место на сервере хранения данных и быстро отдавать изображение клиенту. Для решения данной задачи каждый раз проводят ряд экспериментов и стараются подобрать универсальный вариант решения данного вопроса. В данной статье были рассмотрены основные графические форматы и проведено сравнение, которое позволит решить этот вопрос в качестве инструкций к различным форматам — когда их надо применять с/без сжатия и с/без потерь.

Предлагаемая в статье система позволит сэкономить время и деньги команде разработки нагруженных приложений.

Специфические особенности графических форматов

Для начала необходимо обозначить основные особенности сравниваемых форматов: JPEG, WEBP, AVIF.

— JPEG

JPEG — или же Joint Photographic Experts Group это одноименное название организации и формата файлов,

созданном в 1982 и усовершенствующем по его использованию. JPEG подразумевает возможность сжатия с и без потерь.

JPEG эффективно сжимает фото и изображения с реалистичными сценами и плавными переходами цветов. Он широко используется в цифровой фотографии и для передачи изображений через интернет.

Для сжатия чертежей, текстов и графики с резким контрастом JPEG с потерями не подходит из-за артефактов. Вместо этого стоит использовать форматы без потерь, как JPEG-LS, TIFF, GIF, PNG или Lossless JPEG.

JPEG (как и другие форматы сжатия с потерями) не подходит для сжатия изображений при многоэтапной обработке, так как искажения в изображения будут вноситься каждый раз при сохранении промежуточных результатов обработки.

JPEG не должен использоваться и в тех случаях, когда недопустимы даже минимальные потери, например при сжатии астрономических или медицинских изображений. В таких случаях может быть рекомендован предусмотренный стандартом JPEG режим сжатия Lossless JPEG (который, однако, не поддерживается большинством популярных кодеков) или стандарт сжатия JPEG-LS.

Этапы сжатия

Шаги процесса превращения картинки в формат JPEG включают семь основных этапов. В первом этапе происходит цветовая субдискретизация, где формат изображения RGB преобразуется в YCbCr, что позволяет сократить размерность изображения за счет изменения восприятия его цветовой составляющей. Далее следует дискретизация изображения, которая усредняет компоненты яркости Y у соседних пикселей для уменьшения размерности. На третьем этапе применяется дискретное косинусное преобразование, которое находит паттерны в изображении по формуле Фурье, выраженные в виде паттернов, подобных тем, что изображены на рисунке 1.

После этого изображение кодируется данными волновых рисунков, происходит «Зигзар» сканирование для удаления высоких частот и квантование для усреднения результатов. Затем происходит групповое кодирование цельных участков изображения и применение алгоритма Хаффма для сжатия однотипных данных в изображении. Полный путь преобразования изображения представлен на рисунке 2, а обратный путь преобразования — это прохождение всех этапов в обратном направлении.

— WEBP

WebP — сравнительно новый формат от Google. Картинки в этом формате занимают на 30% меньше места на странице благодаря особому сжатию, построенному на кодировании ключевых кадров в видеокодеке VP8.

Как устроено сжатие в WebP

Сжатие с потерями основано на сходстве соседних пикселей по яркости и цвету. Изображение делится на макроблоки, где декодер предсказывает яркость и цвет следующего пикселя на основе предыдущих. Предсказуемые данные исключаются, улучшая сжатие. Затем результат обрабатывается с помощью ДКП, квантования и энтропийного кодирования.

— Lossy WebP

Lossy WebP применяет арифметическое кодирование для энтропийного сжатия. WebP адаптивно распределяет биты между фрагментами изображения, что гибче, чем код Хаффмана в JPEG. WebP умно делит изображение на блоки 4x4 и 16x16 пикселей, в зависимости от деталей и монотонности и далее кодирует именно такие схожие части.

Как происходит прогнозирование

Декодер VP8 имеет 2 класса прогнозирования:

- Intra — внутрикадровое пространственное предсказание блока на основе значений пикселей из соседних, уже закодированных блоков, слева и сверху.
- Inter — межкадровое временное предсказание (оценка векторов движения).

Intra имеет четыре алгоритма прогнозирования для блоков 16x16 и 8 для детализирующих блоков 4 x 4:

- H_PRED горизонтальное прогнозирование. Заливает следующую колонку на основе той, что находится слева от нее.
- V_PRED вертикальное прогнозирование работает аналогично H_PRED, но в вертикальном направлении. Он заполняет следующий ряд блока, используя информацию из верхнего ряда.
- DC_PRED заполняет блок, используя усредненные значения цвета и яркости пикселей соседних строк и колонок.
- TM_PRED учитывает усредненные значения строки A и колонки L, а также пиксель P, который находится сверху и слева от блока. Алгоритм начинает с пикселя в колонке L и заполняет каждую строку блока, учитывая различия между пикселями в колонке, начиная от пикселя P. Это позволяет более точно прогнозировать значения пикселей на основе соседних элементов.

Изображение разбивается на сегменты, которые имеют явно схожие характеристики. Для каждого такого сег-

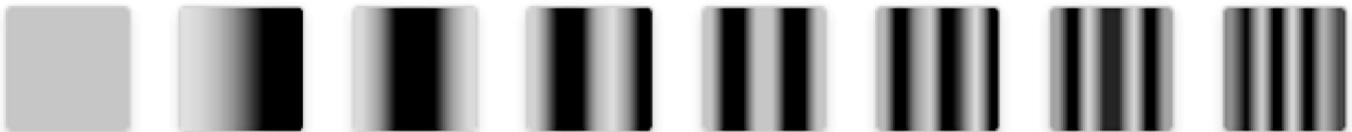


Рис. 1. Паттерны изображения разложенного по формуле Фурье



Рис. 2. Алгоритм превращения изображение в JPEG формат

мента параметры сжатия и способы прогнозирования настраиваются независимо. Таким образом биты перераспределяются туда, где они наиболее полезны.

Сжатие без потерь

При сжатии без потерь используется вариант алгоритма LZ77— кода Хаффмана. А также пространственное прогнозирование и преобразование цветового пространства.

Сжатие с альфа-каналом

Формат WebP позволяет получить сжатую картинку с альфа-каналом без потерь. Раньше, чтобы получить прозрачность все изображение должно было быть lossless. А в WebP можно уменьшить вес картинки с прозрачными областями.

Цветовое преобразование

Также в WebP используется методы адаптивного квантования цветовой составляющей, чтобы предотвратить влияние цветочных каналов друг на друга. Изображение делится на блоки и для каждого блока применяется свой режим трансформации green_to_red, green_to_blue или red_to_blue. Цветовое преобразование сохраняет неизменным значение зеленого канала G, преобразует красный R в зависимости от зеленого, и синий B в зависимости от зеленого, а затем в зависимости от красного.

Цветовое кеширование

Сжатие lossless WebP использует уже обработанные фрагменты изображения для работы с новыми пикселями. В случае если подходящие совпадения не найдены, используется локально созданная палитра. Эта палитра постоянно обновляется цветами, найденными при сканировании картинки.

Индексирование палитры

Если в картинке используется менее 256 цветов, алгоритм создает отдельный массив индексов цветов и сохраняет его отдельно, чтобы подменить значение цвета на индекс для каждого пикселя.

Для картинок с небольшим количеством мелких деталей используется технология апскейлинга. Прежде чем кодироваться — изображение ресайзится.

— **AVIF**

AVIF является новым форматом хранения изображений, основанным на видеокодеке AV1, и считается одним из самых современных форматов. По ранним публикациям и исследованиям, AVIF показал достойные результаты

по сравнению с JPEG и WebP. Однако, несмотря на то, что этот формат уже поддерживается большинством браузеров, кодирование и декодирование изображений в AVIF все еще является передовой технологией, которая имеет некоторые проблемы. Формат кодирует изображение в формате YCbCr или YUV, аналогично форматам WebP и JPEG, что позволяет сократить размер исходного файла. Однако, AVIF использует более современные алгоритмы и динамически переключается между ними в зависимости от типа изображения и его характеристик.

	JPEG	VP8(WEBP)	AV1(AVIF)
Блоки	8x8	гибко от 4x4 до 16x16	гибко от 4x4 до 128x128
Фильтры	-	Deblocking	Deblocking + CDEF + ещё 2
Предсказатель	-	DC/directional (V + H)/TM	DC/directional (56 directions)/Path / Smooth/CfL/IBC
Преобразователь	DCT	DCT/WHT	DCT/ADST/FlipADST/DTX
Сжатие без потерь	Хаффман	Булевский арифметический кодировщик	Многосимвольное энтропийное кодирование
Квантование	Две таблицы	Гибко	Гибко

Рис. 3. Сравнительная таблица различных типов изображений и их особенностей

Как видно, из таблицы на рисунке 3 AVIF по сравнению с WEBP использует чуть больший набор инструментов сокращения размера файла как:

1. Увеличенное динамическое разбиение чанков с 4x4 — 16x16 до 4x4 — 128–128. Что позволяет уменьшать размерность одинаковых кусочков изображений.
2. Улучшенные алгоритмы фильтрации для улучшения видео. Так как три алгоритма идеально сочетаются друг с другом, получится значительное улучшение именно визуального качества без потерь в размерах.
3. Увеличенное количество направлений предсказания пикселей до 56 направлений.
4. Функция трансформации использует по мимо DST, отображающее самый насыщенный свет также и его улучшенные версии, что позволяет сократить объем файла.

Благодаря этим свойствам AVIF уже теоретически должен выдавать картинку меньшего размера с немного лучшим качеством, чем JPEG и WEBP.

Классификация типов изображений

Для сравнения форматов изображений необходимо выбрать изображения, которые с большей вероятно-

стью будут использоваться в пользовательском приложении. Для этого были взят список частотных изображений из списка российской поисковой системы Яндекс. По данным Яндекса¹, из миллиардов запросов, которые они обрабатывают каждый день можно выделить 15 типов частотных запросов их пользователей: фото, картинки, рисунки, обои, карты, схемы, раскраски, фоны, таблицы, открытки, дизайн, картины, татуировки, стиль, образцы.

Каждый из запросов для понимания тенденций сжатия для различных классов изображений можно классифицировать по категориям: Буквенно-цифровой тип изображений, графическое изображение, смешанное изображение. Для дальнейшего анализа изображения можно классифицировать следящим образом (табл. 1).

Таблица 1.

Разбиение типов картинок по классам отображаемой информации

Буквенно-цифровое	Таблицы, Образцы
Графическое	Рисунки, Раскраски, Фоны, Дизайн, Картины, Стиль
Смешанное	Фото, Картинки, Обои, Карты, Схемы, Открытки, Татуировки

Оценка погрешности при сжатии изображений с/без потерь и с/без сжатия

Независимо от использования сжатия или алгоритма потерь можно сравнить исходное изображение с сжатым или отсутствием такового по парам пикселей при условии идентичности размерности изображения. Для данного сравнения были использованы специально разработанные для данных задач общие алгоритмы сравнения: SSIM и UQI.

— SSIM

SSIM представляет собой метод оценки сходства между двумя изображениями. Индекс SSIM — это подход полного сопоставления, то есть он оценивает качество, основываясь на исходном изображении (несжатом или без искажений). Индекс SSIM является улучшением традиционных подходов, таких как PSNR (peak signal-to-noise ratio) и метод среднеквадратической ошибки (MSE), которые оказались несовместимы с физиологией человеческого восприятия.

Основным отличием метода от ранее упомянутых (MSE и PSNR) является то, что метод учитывает «восприятие ошибки» благодаря учету структурных изменений информации. Концепция заключается в том, что пиксели имеют сильную корреляцию, особенно когда они близки в пространстве. Такие зависимости содержат важную информацию об объектах и сцене в целом.

— UQI

Метрика определяет насколько сжатое изображение похоже на исходное, учитывая тип и степень искажения. Оценка универсальна, так как показывает сходство сжатого изображения с оригиналом, учитывая различные виды искажений. Значение UQI не зависит от особенностей изображения и его искажений. Оценка находится в диапазоне $[-1, 1]$, где $UQI = 1$ означает наилучшее качество сигнала (табл. 2).

Таблица 2.

Первичное сравнение форматов

Формат	Процент поддержки клиентскими браузерами	Возможность оптимизации загрузки	Тип сжатия
JPEG 2000	18,4%	Да	С потерями
WEBP	97,41%	Да	Бес/с потерями
AVIF	82,64%	Нет	Бес/с потерями

Данные в таблице 2 позволяют оценить, что не смотря на достаточно старость формата JPEG поддерживают его не все браузеры. Данные для таблицы были взяты с сайта для оценки совместимостей различных функций в браузерах <https://caniuse.com/?search=JPEG>.

Экспериментальное исследование разных алгоритмов сжатия графических форматов

Для преобразования картинок используется сервис <https://squoosh.app/editor>.

Так как оригиналы изображений изначально были взяты в формате JPEG, поэтому для сравнения с оригиналом для JPEG используется потеря 25 % качества (коэффициент 0.75 в сервисе сравнения в формате Browser JPEG). Для WEBP потери так же составляют 25 %, а для AVIF из-за ограничения размерности шкалы сжатия потери измеряются в 25 % с коэффициентом качества в <https://squoosh.app/> 0.5, а также с 0.3 из-за возможностей формата лучше сохранять качество при большем уменьшении качества картинки.

Сравнение изображений

На рисунках 4, 5, 6 представлены данные после проведения эксперимента — преобразования картинок в заданный формат и с определенным сжатием, а также после определения их отклонения от изначальной картинки с помощью методов измерения схожести SSIM и UQI.

Оценка полученных результатов

Для анализа полученных данных можно воспользоваться следующими методами:

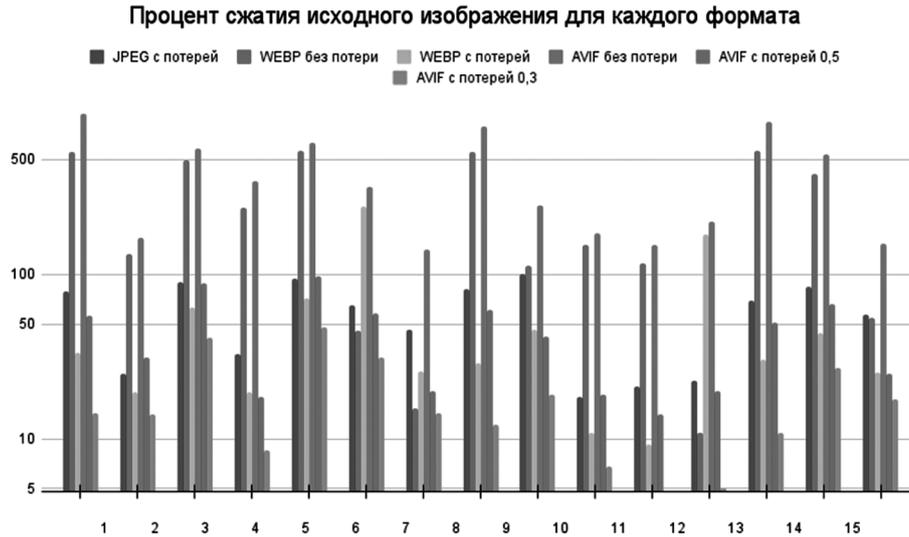


Рис. 4. Процент сжатия исходного изображения для каждого формата

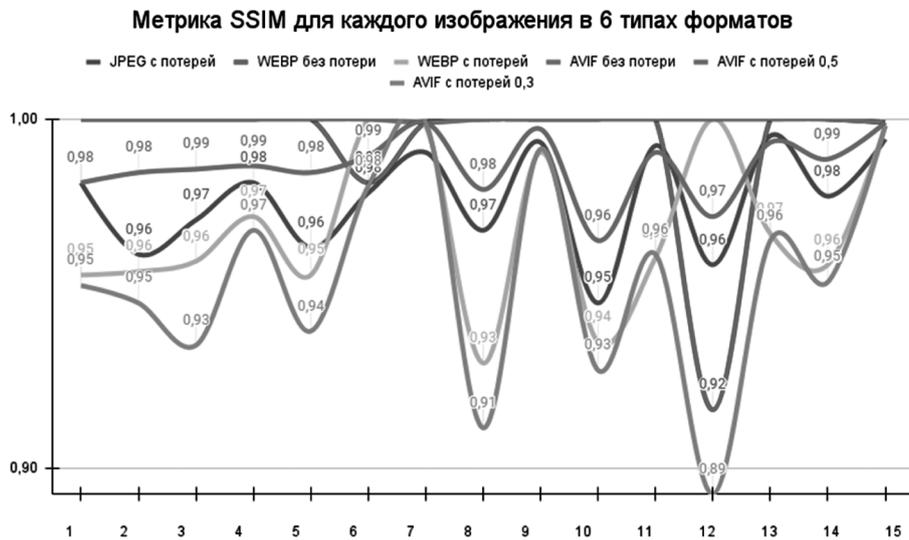


Рис. 5. Метрика SSIM для каждого изображения в 6 типах форматов

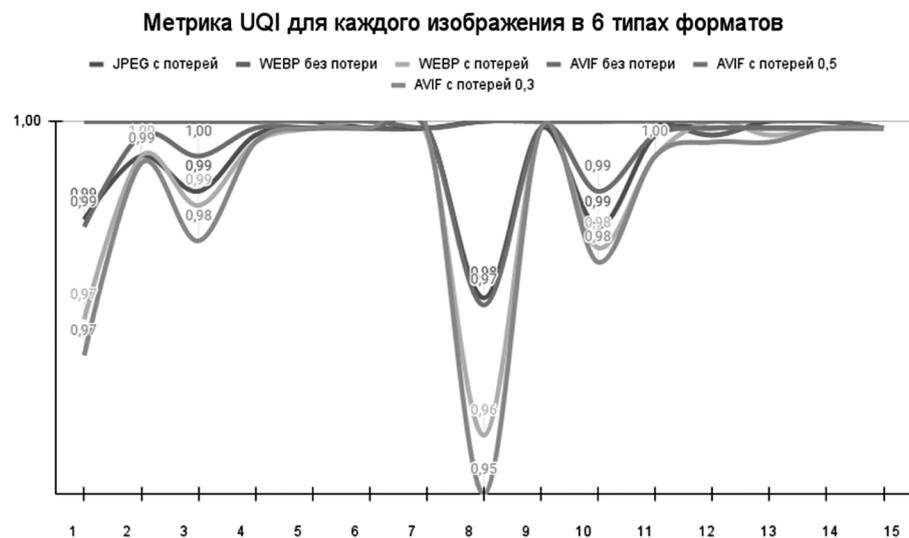


Рис. 6. Метрика UQI для каждого изображения в 6 типах форматов

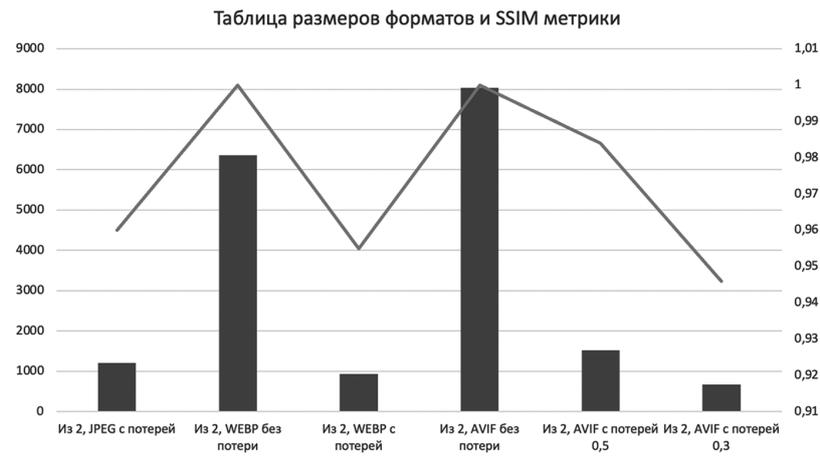


Рис. 7. Таблица размеров форматов и SSIM метрики

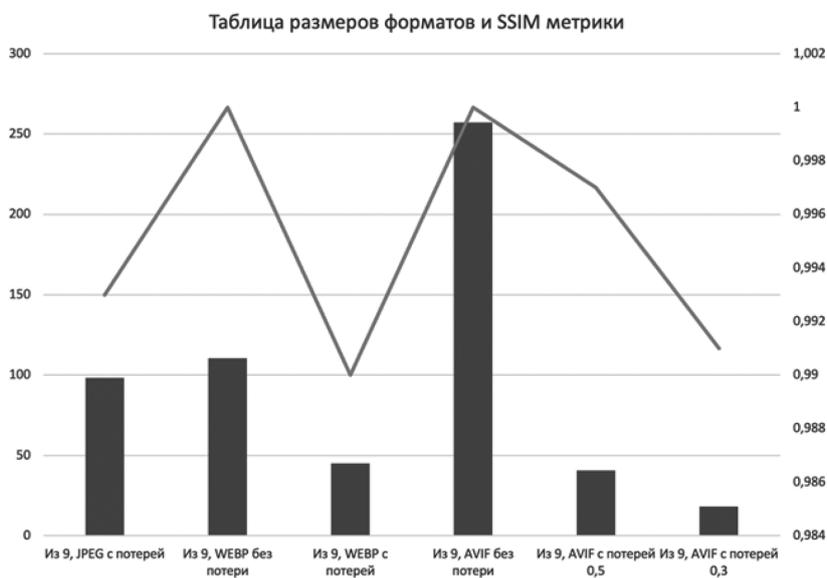


Рис. 8. График сравнения изображения из буквенно-цифрового класса

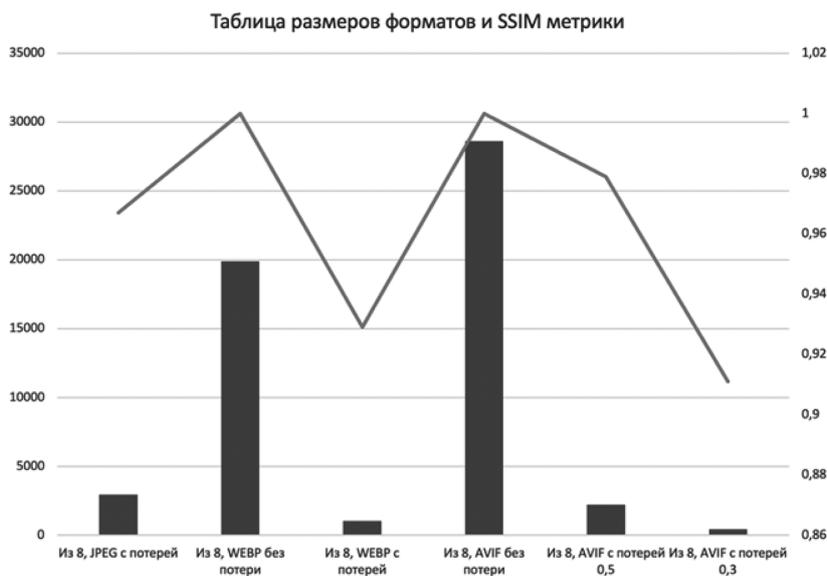


Рис. 9. График сравнения изображения из графического класса

1. Численным с помощью метрик и степени сжатия
2. Сравнение визуальных неточностей

Анализ с помощью метрик и степеней сжатия

Для второго изображения таблицы форматов, относящегося к классу смешанных изображений, выведем график зависимости сжатия и метрики SSIM в зависимости от формата данных (рис. 7).

Как видно из изображения самую лучший уровень сжатия показывает именно «AVIF с потерей 0,3», и качественная метрика у него хотя не очень высокая, но позволяет передать основные элементы изображения.

Для поиска взаимосвязи между классом изображения и его степень сжатия/качеством с исходным изображением также было проведено сравнение для изображения из буквенно-цифрового класса и графического.

Для девятого изображения — таблицы, относящиеся к классу буквенно-цифровых изображений, был получен график представленное на рисунке 8.

Для восьмой картинки — фоны, относящиеся к графическому классу графических изображений, был получен график представленное на рисунке 9.

Как видно из рисунков 7, 8 и 9 корреляции между разными классами изображения и отклонениями в определенном типе изображения метрик схожести не наблюдается. Однако, видна явная взаимосвязь между типом изображения, тем насколько сильно может пережать его определенный формат и метрикой качества.

Так наблюдается тенденция чем больше изображение наполнено графическим разнообразием — ближе к чисто графическому изображению, тем ниже степень качества получается на выходе после изменения формата, а чем ближе к четкой структуре — буквенно-цифровому изображению, тем лучше оно пережимается и уровень сходства с изначальным изображением выше. Лидером по степени сжатия к размеру изображения является формат AVIF с максимальными потерями, что представлено на рисунке 10.

Для построения графиков на рисунке 10 была использована следующая формула позволяющая увидеть визуально взаимосвязь потерь и качества рисунка:

$$\text{Взаимосвязь} = \frac{SSIM}{\% \text{сжатия}}$$

Из графика можно сделать вывод, что буквенно-цифровая и смешанная информация пережимается с учетом сохранения качества, чем чисто графическая информация и для случаев, когда важен именно размер изображения и его внешний вид в меньшей степени лучше использовать формат AVIF с потерями 0,3.

Визуальное сравнение

Для более точного сравнения форматов также было проведено и визуальное сравнение изображений, что представлено ниже (рис. 11–13).

Если сравнивать 3 формата с потерями AVIF 0.3, JPEG, WEBP, то можно заметить, что:

- AVIF: размывает узоры облаков, делая их мутными;
- JPEG: отображает узоры облаков, но теряет точки бликов. Это связано с алгоритмом дискретизации самого формата;



Рис. 10. Степень сжатия в зависимости от качества



Рис. 11. Фрагмент изображения 2 в формате AVIF



Рис. 12. Фрагмент изображения 2 в формате JPEG



Рис. 13. Фрагмент изображения 2 в формате WEBP

— WEBP: максимально качественно передаёт текстуру изображения вместе с бликами.

Также при рассмотрении самого тяжелого изображения — восьмого в разных форматах на предмет визуальных неточностей в формате AVIF опять наблюдается значительное размытие текстур изображения, в JPEG потеря некоторых четких границ текстур. Поэтому считаю, что в данном эксперименте не смотря на степень сжатия лучше всего себя показывает именно формат WEBP.

Заключение

В результате анализа, сделанного в данной статье можно сделать вывод, что для того, чтобы отобразить изображения, где необходимо сохранить качество,

но значительно сжать изображение идеально подойдет формат WEBP с потерями. А для случаев, где допустимы большие потери, но важно именно наличие картинки, например предпросмотр картинки, можно эффективно использовать AVIF. Благодаря функциям браузеров можно комбинировать полезные свойства всех изображений и при, допустим, недоступности AVIF подгрузить WEBP или JPG, что значительно улучшит опыт взаимодействия с целевым интернет ресурсом.

Экономия при данном виде оптимизаций может быть полезна сервисам, завязанным на отображение большого кол-ва изображений или имеющим CDN, чтобы сократить оплату межсерверного трафика и сэкономить на плате за серверные услуги.

ЛИТЕРАТУРА

1. THE INTERNATIONAL TELEGRAPH AND TELEPHONE CONSULTATIVE COMMITTEE «INFORMATION TECHNOLOGY — DIGITAL COMPRESSION AND CODING OF CONTINUOUS-TONE STILL IMAGES — REQUIREMENTS AND GUIDELINES» от 01.09.1992 № Recommendation T.81.
2. Nabajeet Barman, Maria G Martini Kingston An Evaluation of the Next-Generation Image Coding Standard AVIF // IEEE. — 2020.
3. Частотные картинки сети интернет // Яндекс URL: <https://yandex.ru/company/researches/2018/images> (дата обращения: 10.11.2022).
4. Описание работы формата WEBP // Google URL: <https://developers.google.com/speed/webp/docs/compression> (дата обращения: 10.11.2022).
5. Описание работы формата WEBP // Habr URL: <https://habr.com/ru/company/io/blog/261651/> (дата обращения: 10.11.2022).
6. Особенности формата AVIF // Habr URL: <https://habr.com/ru/company/ruvds/blog/534110/> (дата обращения: 10.11.2022).
7. Работа AVIF формата под капотом // hacks.mozilla.org URL: <https://hacks.mozilla.org/2018/06/av1-next-generation-video-the-constrained-directional-enhancement-filter/> (дата обращения: 16.12.2022).

© Горячкин Борис Сергеевич (bsgor@mail.ru); Можяев Дмитрий Владимирович (mdv413@mail.ru);
 Андрианов Алексей Александрович (alks.andrnv@gmail.com).
 Журнал «Современная наука: актуальные проблемы теории и практики»