

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки
Факультет Інфокомунікацій
(повна назва)
Кафедра Інформаційно-мережної інженерії
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА

Пояснювальна записка

рівень вищої освіти другий (магістерський)
(рівень вищої освіти)

Дослідження математичної моделі зображень на основі
дискретно-косинусного перетворення
(тема)

Виконав:

студент 2 курсу, групи ІМІм-22-1

Резнік В. М.

(прізвище, ініціали)

Спеціальність 172 Телекомунікації та
Радіотехніка

(код і назва спеціальності)

Тип програми освітньо-професійна

(освітньо-професійна або освітньо-наукова)

Освітня програма «Інформаційно-
мережна інженерія» (повна назва освітньої

програми)

Керівник Професор Безрук В.М.

(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри

(підпис)

(прізвище, ініціали)

2024р

Не містить відомостей, заборонених до відкритого публікування

Студент	_____	<u>Резнік В.М.</u>
	(підпис)	(прізвище та ініціали)
Керівник	_____	<u>Професор Безрук В.М.</u>

Харківський національний університет радіоелектроніки
Факультет Інфокомунікацій
(повна назва)
Кафедра Інформаційно-мережної інженерії
(повна назва)
Рівень вищої освіти другий (магістерський)
(рівень вищої освіти)
Спеціальність 172 «Телекомунікації та радіотехніка»
(код і назва спеціальності)
Тип програми освітньо-професійна
(освітньо-професійна або освітньо-наукова)
Освітня програма «Інформаційно-мережна інженерія»
(повна назва освітньої програми)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____
(підпис)

« _____ » _____ 20 _____ р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові Резніку Володимирі Михайловичу
(прізвище, ім'я, по батькові)

1. Тема роботи «Дослідження математичної моделі зображень на основі дискретно-косинусного перетворення»

затверджена наказом університету від 23 10 2023 р. № 1233 Ст _____

2. Термін подання студентом роботи до екзаменаційної комісії 24 01 2024 р.

3. Вихідні дані до роботи: Дослідити математичну модель зображень на основі дискретного косинусного перетворення, а саме математичну модель ДКП та вплив таблиць квантування коефіцієнтів на якість кінцевого зображення та його розмір. Продемонструвати результати досліджень.

4. Перелік питань, що потрібно опрацювати в роботі:

Огляд алгоритму JPEG;

Алгоритм стиснення JPEG;

Експерименти та результати

5. Перелік графічного матеріалу із зазначенням креслень, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням випускної кафедри) слайди у форматі PowerPoint

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Ознайомлення із завданням.	21.10.2023	виконано
2	Підбір літератури за темою роботи.	22.10 – 30.10.2023	виконано
3	Виконання розділу 1	31.10 – 06.11.2023	виконано
4	Виконання розділу 2	06.11 – 18.11.2023	виконано
5	Виконання розділу 3	18.11 – 05.01.2023	виконано
6	Оформлення презентаційного матеріалу, підготовка до захисту у ЕК	05.01 – 24.01.2024	виконано

Дата видачі завдання 21 жовтня 2023 р.

Студент _____
(підпис)

Керівник роботи _____
(підпис)

Професор Безрук В.М.
(посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка: 90 с., 53 рис., 3 табл., 16 джерел.

DCT, RD-OPT, JPEG

Об'єкт дослідження – Набір тестових зображень з бібліотеки STI.

Мета роботи – дослідження впливу таблиць квантування на якість та розмір кінцевого файлу, порівняння таблиць квантування з алгоритмом визначення оптимальної таблиці квантування RD-OPT.

Результати – в роботі було спочатку порівняно таблиці квантування між собою, потім алгоритм RD-OPT проти найкращих таблиць квантування з вище згаданих, які були вбудовані в інструмент для кодування та декодування зображень на алгоритмі JPEG. В результаті експерименту встановлено, що алгоритм породження таблиць має право на існування та має невелику перевагу над визначеними таблицями квантування.

THE ABSTRACT

Explanatory note: 90 p., 53 fig., 3 tab., 16 sources.

DCT, RD-OPT, JPEG

Object of research – set of test images from the STI library.

The purpose of qualification work is to study the influence of quantization tables on the quality and size of the final file, to compare quantization tables with the algorithm for determining the optimal quantization table RD-OPT.

Result – in the work first compared the quantization tables among themselves, then the RD-OPT algorithm against the best quantization tables from the above, which were built into the tool for encoding and decoding images using the JPEG algorithm. As a result of the experiment, it was established that the table generation algorithm ' the right to exist and has a slight advantage over the defined quantization tables.

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ	9
ВСТУП.....	10
Визначення теми і актуальність дослідження.	10
Мета та завдання роботи.....	11
1 ОГЛЯД АЛГОРИТМУ JPEG	13
1.1 Опис загальних принципів роботи алгоритму JPEG.	15
1.2 Основні компоненти JPEG файлів	17
2 АЛГОРИТМ СТИСНЕННЯ JPEG	21
2.1 Підготовчі етапи	21
2.1.1 Конвертація кольорового простору (з RGB у YCbCr)	22
2.1.1.1 Артефакти хроматичної субдискретизації	29
2.1.2 Розділення зображення на 8x8 блоки	33
2.1.2.1 Артефакти блокування	35
2.2 DCT (Дискретне косинусне перетворення).....	37
2.2.1 Застосування DCT до кожного 8x8 блоку	39
2.2.2 Квантування результатів DCT	46
2.3 Кодування	50
2.3.1 Використання кодування диференціалів DC-коефіцієнтів	51
2.3.2 Використання кодування амплітуд AC-коефіцієнтів за допомогою Zig-Zag сканування	52
2.3.3 Використання RLE (Run-Length Encoding) для стиснення послідовності нульових коефіцієнтів	54
2.4 Декодування зображення у форматі JPEG	55
2.4.1 Порівняння оригіналу з розпакованим зображенням	58

3 ЕКСПЕРИМЕНТИ ТА РЕЗУЛЬТАТИ	61
3.1 Масштабування базової таблиці квантування	61
3.2 Основні розрахункові співвідношення	63
3.3 Порівняння базових таблиць квантування	64
3.4 Порівняння базових таблиць квантування з таблицями RD-OPT.....	72
ВИСНОВКИ	77
ПЕРЕЛІК ПОСИЛАНЬ	78
ДОДАТОК А СЛАЙДИ ПРЕЗЕНТАЦІЇ	80

ПЕРЕЛІК СКОРОЧЕНЬ

JPEG – (Joint Photographic Experts Group) (Об'єднана група експертів із фотографії)

RD-OPT – (Rate distortion optimization) оптимізація швидкість-спотворення

ISO – (International Standart Organisation) Міжнародна організація зі стандартизації

DCT – (Discrete Cosine Transform) Дискретне косинусе перетворення

ДКП - Дискретне Косинусе Перетворення

RLE – (Run-Length Encoding) (Кодування довжини бігу)

MCU – (Minimal Code Unit) Мінімальний Кодований Блок

RGB - (Red Green Blue) Кольорова адитивна модель

JFIF - (JPEG File Interchange Format) Формат обміну файлами JPEG

GIF - (Graphics Interchange Format) формат обміну зображеннями

CMYK - (Cyan, Magenta, Yellow, Key color) субтрактивна колірна модель

YCbCr - сімейство колірних просторів, що використовуються для передачі кольорових зображень

ВСТУП

Визначення теми і актуальність дослідження.

У сучасному світі зображення відіграють надзвичайно важливу роль у багатьох сферах, які постійно розвиваються та розширюються. Вони є носіями оптичної інформації і знаходять застосування в таких сферах, як:

- Медицина: Зображення використовуються для діагностики та моніторингу захворювань, для вивчення анатомії та структури органів і тканин. Медичні зображення допомагають лікарям здійснювати точні діагнози і обирати методи лікування.
- Мультимедіа: Зображення та відео грають важливу роль у розвагах і розважальних заходах. Вони створюють візуальну і аудіовізуальну атмосферу в фільмах, відеоіграх, мультфільмах і в інших розважальних форматах.
- Наука: Зображення використовуються для вивчення мікросвіту, астрономії, біології, фізики та інших наук. Вони допомагають вченим спостерігати і досліджувати різні явища і об'єкти.
- Технології: Зображення грають роль в розпізнаванні обличчя, автоматичному управлінні та робототехніці, допомагаючи системам розуміти оточуючий світ.

З ростом якості зображень, зазвичай, зростає і їх розмір. Зображення, отримані в матриці камери, містять у собі дуже багато інформації, особливо при високому розширенні. Для прикладу, розглянемо два зображення: одне розміру 640x480 пікселів, а інше 1920x1080 пікселів (Full HD):

Зображення 640x480 містить 307,200 пікселів та 921,600 байт (або 900 КБ) інформації в кольоровому просторі RGB.

Зображення 1920x1080 містить 2,073,600 пікселів та 6,220,800 байт (або 6,072 КБ) інформації в тому ж кольоровому просторі.

Як бачимо, при збільшенні розширення зображення обсяг інформації зростає значно. Це призводить до проблеми великого обсягу даних, особливо при зберіганні та передачі таких зображень. У зв'язку з цим, стиснення зображень стає надзвичайно актуальним завданням, оскільки дозволяє зменшити обсяг даних, зберігаючи важливу інформацію та якість зображення.

Мета та завдання роботи.

Мета роботи полягає в дослідженні математичної моделі зображень на основі дискретно-косинусного перетворення (DCT) з метою стиснення цифрових зображень і зменшення їх обсягу з мінімальними втратами якості.

Завдання роботи включають в себе:

1. Аналіз та вивчення теоретичних аспектів DCT: Розгляд теоретичних основ DCT, включаючи математичні принципи та властивості перетворення. Дослідження ролі DCT у стисненні зображень та його застосувань.
2. Аналіз популярного алгоритму стиснення зображень JPEG: Опис алгоритму, який застосовує DCT для стиснення цифрових зображень. Аналіз використання алгоритмів DCT і оберненого DCT для обробки зображень.
3. Експерименти та аналіз результатів: Виконання стиснення зображень з використанням алгоритму JPEG та проведення аналізу якості стиснених зображень. Оцінка рівня стиснення та залежності якості від таблиць квантування коефіцієнтів DCT.
4. Висновки: Формулювання висновків щодо результатів дослідження, підсумки роботи, а також визначення можливих напрямків подальших досліджень у галузі стиснення зображень на основі DCT.

Мета роботи полягає в описі та поясненні ефективності математичної моделі для стиснення зображень з використанням DCT, що знаходить практичне застосування в областях, де обмежені ресурси чи обсяги даних є критичними.

1 ОГЛЯД АЛГОРИТМУ JPEG

JPEG – це скорочення від Joint Photographic Experts Group, що означає «Об'єднана група експертів із фотографії», яка стандартизувала формат наприкінці 1980-х – початку 1990-х років. Це основний формат цифрових зображень, який використовується відтоді, як фотографи почали знімати та зберігати зображення на цифрових пристроях.

JPEG-файл підтримує глибину кольору до 24 біт і використовує метод стиснення із втратами для зручнішого зберігання та надсилання зображень. Це робить JPEG-файли кращими для повсякденного використання, але також означає, що якість початкового зображення буде втрачена [1].

У 1986 році монітори та їхні технології не здатні були створювати екранну графіку. Саме тоді спільнота спеціалістів з усього світу під назвою Міжнародна організація зі стандартизації (ISO) почала працювати над способами відображення фотореалістичних зображень на маленьких екранах.

У той самий час Об'єднана група експертів з фотографії (JPEG) створила власний стандарт для стиснення графічних файлів, який підходив для роботи на звичайному ПК. Вона розробила концепцію стиснення із втратами, яка видаляє візуальні дані, невидимі для людського ока, і встановлює усереднені значення кольорів [1].

JPEG-файли виникли з появою цифрових камер і розвитком та поширенням всесвітньої мережі Інтернет. Наразі кожен користувач має можливість робити знімки та зберігати їх у файлах, які достатньо чітко відображаються на екрані, що є зручним за використання обмеженого сховища камери. Справжня цінність формату JPEG полягає в його здатності зберігати метадані, зокрема дані про геолокацію й час зйомки та навіть про параметри камери.

JPEG-файл може відображати 16,8 мільйонів кольорів, при цьому його розмір є відносно невеликим. Саме це робить його популярним форматом файлу.

Фотографи, як правило, знімають у форматі RAW, щоб отримати максимально чіткий знімок, але часто вони публікують свої роботи в Інтернеті у форматі JPEG. Менший розмір файлів забезпечує швидше відкриття в більшості без суттєвої втрати якості.

Користувачі отримують переваги від використання JPEG-файлів на сайтах. Сторінка завантажується швидше, тож читачі одразу бачать вміст – також JPEG-файли не займають занадто багато місця в сховищі.

Співвідношення розміру файлу та якості зображення – незмінна тема дебатів щодо JPEG-файлів

JPEG є найбільш поширеним форматом графічних файлів, який сумісний з більшістю браузерів, програмного забезпечення та програм.

Завдяки невеликим розмірам ці файли легко передавати та переглядати в Інтернеті. JPEG має алгоритм стиснення із втратами, який видаляє невидимі для людського ока деталі, максимально зменшуючи розмір файлу. У порівнянні з форматами без втрат, зокрема GIF, JPEG-файли є набагато меншими за розміром.

Алгоритм стиснення із втратами може допомогти заощадити простір у сховищі, але занадто високий рівень стиснення призведе до втрати інформації в зображення. Зображення з чіткими краями та лініями втратять різкість після стиснення.

Втрата великої кількості даних може призвести до артефактів – втрати плавного переходу між кольорами, через що зображення виглядатиме погано. Також можуть з'явитися спотворення – згладжування по краях, вуалювання або шум, які суттєво погіршують якість зображення. Щоб уникнути появи спотворень, краще зберігати фотографії у форматах без втрат.

1.1 Опис загальних принципів роботи алгоритму JPEG.

Стандарт JPEG передбачає два основні способи подання кодованих даних. Найбільш поширеним, підтримуваним більшістю доступних кодеків, є послідовне (sequential JPEG) подання даних, яке передбачає послідовний обхід кодованого зображення розрядністю 8 біт на компоненту (або 8 біт на піксель для чорно-білих зображень у відтінках сірого) блоками зліва направо та зверху донизу. Над кожним кодованим блоком зображення виконуються описані вище операції, а результати кодування розміщуються в вихідному потоці у вигляді єдиного «скану», тобто масиву закодованих даних, який відповідає послідовно обхідному («просканованому») зображенню. Основний або «базовий» (baseline) режим кодування допускає тільки таке подання (і кодування за допомогою кодування Хаффмана квантованих коефіцієнтів ДКП). Розширений (extended) режим, крім послідовного, допускає також прогресивне (progressive JPEG) подання даних, кодування зображень розрядністю 12 біт на компоненту/піксель (специфікація JFIF не підтримує стиснення таких зображень) та арифметичне кодування квантованих коефіцієнтів ДКП.

У випадку progressive JPEG стиснені дані записуються в вихідний потік у вигляді набору сканів, кожен з яких описує зображення повністю з все більшою деталізацією. Цього досягається або шляхом запису в кожен скан не повного набору коефіцієнтів ДКП, а лише якоїсь їх частини: спочатку - низькочастотних, в наступних сканах - високочастотних (метод «спектрального вибору»), або шляхом послідовного, від скану до скану, уточнення коефіцієнтів ДКП (метод «послідовних наближень»). Таке прогресивне подання даних є особливо корисним при передачі стиснених зображень за допомогою повільних каналів зв'язку, оскільки дозволяє отримати представлення про всю картину вже після передачі незначної частини файлу JPEG.

Обидві описані схеми (як послідовний, так і прогресивний JPEG) базуються на ДКП і принципово не дозволяють отримати відновлене зображення

абсолютно ідентичним оригіналу. Проте стандарт також дозволяє стиснення, яке не використовує ДКП, а ґрунтується на лінійному передбачувані (без втрат, тобто «без втрат», JPEG), яке гарантує повне, біт-в-біт, збігання оригінального та відновленого зображень. При цьому коефіцієнт стиснення для фотографічних зображень рідко досягає 2, але гарантована відсутність спотворень в деяких випадках є дуже важливою. Значно вищі ступені стиснення можуть бути отримані за допомогою іншого методу стиснення, який не має безпосереднього відношення до стандарту JPEG, а саме методу стиснення JPEG-LS, який описано в стандарті ISO/IEC 14495-1 (ITU T.87 Recommendation) [2].

Кодування стиснення JPEG може бути розкрито на кілька процесів, як показано нижче:

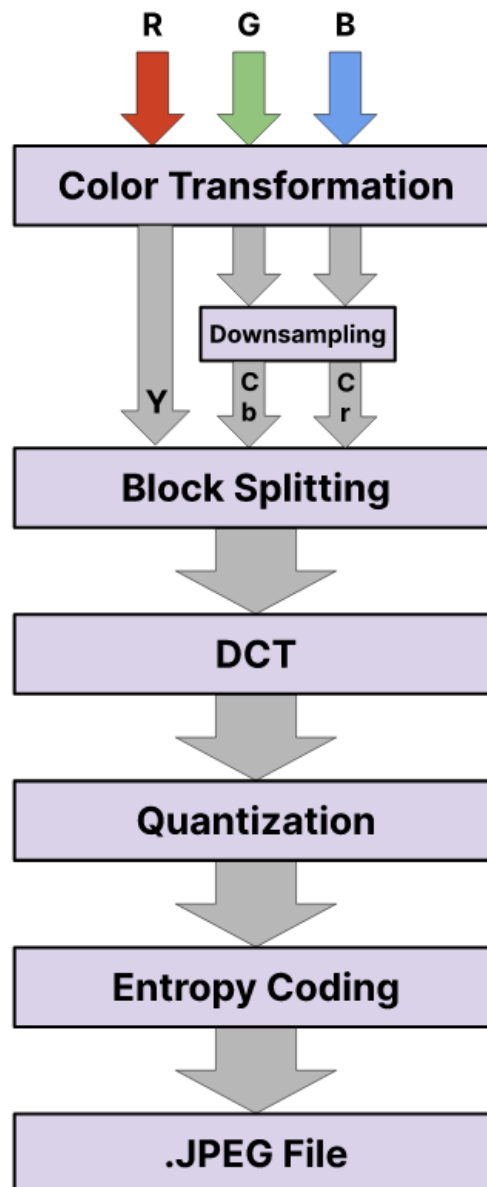


Рисунок 1.1 – Загальний алгоритм JPEG

1.2 Основні компоненти JPEG файлів

Файл JPEG містить послідовність маркерів, кожен з яких починається з байта 0xFF, що свідчить про початок маркера, та байта-ідентифікатора. Деякі маркери складаються тільки з цієї пари байтів, інші містять додаткові дані, що складаються з двобайтового поля з довжиною інформаційної частини маркера (включаючи довжину цього поля, але за вирахуванням двох байтів початку маркера, тобто 0xFF і ідентифікатора) і власне даних. Така структура файлу

дозволяє швидко знайти маркер з необхідними даними (наприклад, з довжиною рядка, числом рядків і числом компонентів кольору стисненого зображення).

Таблиця 2.1 – Основні маркери JPEG.

Основні маркери JPEG [3]				
Маркер	Байти	Довжина	Призначення	Коментарі
SOI	0xFFD8	нема	Початок зображення	
SOF0	0xFFC0	змінний розмір	Початок кадру (базовий, ДКП)	Показує, що зображення кодувалося у базовому режимі з використанням ДКП та коду Хаффмана. Маркер містить число рядків і довжину рядка зображення (двобайтові поля зі зміщенням відповідно 5 і 7 щодо початку маркера), кількість компонентів (байтове поле зі зміщенням 9 щодо початку маркера), число біт на компонент строго 8 (байтове поле зі зміщенням 4 щодо початку маркера), а також співвідношення компонентів (наприклад, 4:2:0).
SOF1	0xFFC1	змінний розмір	Початок кадру (розширений, ДКП, код Хаффмана)	Показує, що зображення кодувалося розширеному (extended) режимі з використанням ДКП і коду Хаффмана. Маркер містить число рядків та довжину рядка зображення, кількість компонентів, число біт на компонент (8 або 12), а також співвідношення компонентів (наприклад, 4:2:0).

Продовження таблиці 2.1

Основні маркери JPEG [3]				
Маркер	Байти	Довжина	Призначення	Коментарі
SOF2	0xFFC2	змінний розмір	Початок кадру (прогресивний, ДКП, код Хаффмана)	Показує, що зображення кодувалося у прогресивному режимі з використанням ДКП та коду Хаффмана. Маркер містить число рядків та довжину рядка зображення, кількість компонентів, число біт на компонент (8 або 12), а також співвідношення компонентів (наприклад, 4:2:0).
DHT	0xFFC4	змінний розмір	Містить таблиці Хаффмана	Задає одну чи більше таблиць квантування.

DQT	0xFFDB	змінний розмір	Містить таблиці Хаффмана	Задає одну чи більше таблиць квантування.
DRI	0xFFDD	4 байти	Вказує довжину рестарт-інтервалу	Задає інтервал між маркерами RST n макроблоках. За відсутності DRI поява в потоці кодованих даних маркерів RSTn є неприпустимим і вважається помилкою. Якщо кодування маркери RST n не застосовуються, маркер DRI або не використовується зовсім, або інтервал повторень в ньому вказується рівним 0.
SOS	0xFFDA	змінний розмір	Початок сканування	Початок першого або чергового скана зображення з напрямом обходу зліва направо зверху донизу. Якщо використовується базовий режим кодування, використовується один скан. У разі використання прогресивних режимів використовується кілька сканів. Маркер SOS є розділяючим між інформативною (заголовком) та закодованою (власне стислими даними) частинами зображення.

Продовження таблиці 2.1

Основні маркери JPEG [3]				
Маркер	Байти	Довжина	Призначення	Коментарі
RSTn	0xFFDn	нема	Перезапуск	Маркери перезапуску використовуються для сегментування кодованих ентропійним кодером даних. У кожному сегменті дані декодуються незалежно, що дозволяє розпаралелити процедуру декодування. При пошкодженні кодованих даних у процесі передачі або зберігання файлу JPEG використання маркерів перезапуску дозволяє обмежити втрати (макроблоки з непошкоджених

				сегментів будуть відновлені правильно). Вставляється в кожному г-му макроблоці, де г інтервал перезапуску DRI маркера. Не використовується за відсутності DRI маркера. n, молодші 3 біти маркера коду, цикли від 0 до 7.
APPn	0xFFEn	змінний розмір	Задається додатком	Наприклад, у файлі EXIF JPEG використовується маркер APP1 для зберігання метаданих, розташованих у структурі, заснованій на TIFF.
COM	0xFFFE	змінний розмір	Коментарій	Містить текст коментарію.
EOI	0xFFD9	нема	Кінець закодованої частини зображення.	

2 АЛГОРИТМ СТИСНЕННЯ JPEG

Алгоритм стиснення JPEG (Joint Photographic Experts Group) передбачає послідовні етапи обробки зображень. На підготовчому етапі виконується конвертація кольорового простору з RGB у YCbCr, що дозволяє розділити інформацію про яскравість та колір. Зображення поділяється на 8x8 блоки для подальшого аналізу.

Далі, застосовується Дискретне косинусне перетворення (DCT) до кожного 8x8 блоку. DCT вимірює наявність частот у блоках і отримує коефіцієнти, які характеризують ці частоти.

Після DCT, коефіцієнти піддаються квантуванню. Це процес, під час якого кожний коефіцієнт ділиться на квантувальний крок, що призводить до втрати точності.

DC-коефіцієнти (перший коефіцієнт DCT в кожному блоку) кодуються як різниця між поточним та попереднім значенням та застосовується Huffman-кодування. AC-коефіцієнти (решта коефіцієнтів DCT) кодуються за допомогою кодування довжин серій (Run-Length Encoding, RLE) та Huffman-кодування.

На завершальному етапі стиснене зображення зберігається у форматі JPEG, забезпечуючи значний обсяг стиснення при мінімальних втратах якості.

2.1 Підготовчі етапи

Підготовчі етапи в алгоритмі стиснення JPEG відіграють важливу роль у забезпеченні ефективності та якості стиснення зображень.

Кольоровий простір RGB використовує три кольорові компоненти (червоний, зелений, синій) для представлення кольорів. У порівнянні із цим, YCbCr має компоненту Y (яскравість) та дві компоненти Cb та Cr (колір). Перетворення до YCbCr дозволяє зменшити обсяг даних, так як компонента Y

зберігає значущу частину інформації, а компоненти Cb та Cr представляють колір, який може бути більш стиснутим без втрати якості.

Розділення зображення на 8x8 блоки дозволяє обробляти його частинами. Кожен блок містить локальну інформацію, і це допомагає зменшити обсяг даних, які потрібно обробляти за один раз та дозволяє використовувати паралельні обчислення.

Подальші етапи, включаючи конвертацію кольорового простору та розділення на блоки, мають важливе значення в алгоритмі стиснення JPEG. Вони допомагають оптимізувати обробку зображень та забезпечити високу якість стиснення при мінімізації втрат.

2.1.1 Конвертація кольорового простору (з RGB у YCbCr)

Кольоровий простір представляє конкретну організацію кольорів, які можуть бути відображені за допомогою кольорової моделі. Кольорова модель - це математична формула, яка використовується для обчислення цих кольорів і може бути представлена різними способами [4]:

- RGB створює білий колір, коли всі кольори додаються разом (адитивна модель) і використовується на екранах, таких як телевізори та монітори комп'ютерів;
- СМΥК створює чорний колір, коли всі кольори додаються разом (віднімальна модель) і використовується в друкарській індустрії.

Людське око більш чутливе до незначних змін яскравості (люмінесценція) ніж до змін кольору (хроматичність). JPEG може використовувати це, перетворюючи у кольорову модель YCbCr, яка розділяє люмінесценцію від хроматичності. Потім ми можемо застосовувати більше стиснення до хроматичних каналів з меншим впливом на якість зображення.

Кольорова модель YCbCr конвертується з кольорової моделі RGB і складається з трьох окремих категорій:

- Люмінесценція (Y) - Дані щодо люмінесценції з кожного каналу червоного, зеленого та синього RGB виділяються та відокремлюються від даних про хроматичність та об'єднуються, щоб створити канал Люмінесценції (Y). Канал Y, сам по собі, містить достатньо даних для створення повного чорно-білого зображення.
- Хроматичний синій (Cb) - Канал Хроматичного синього - це канал синього кольору RGB, віднятий від люмінесценції.
- Хроматичний червоний (Cr) - Канал Хроматичного червоного - це канал червоного кольору RGB, віднятий від люмінесценції.

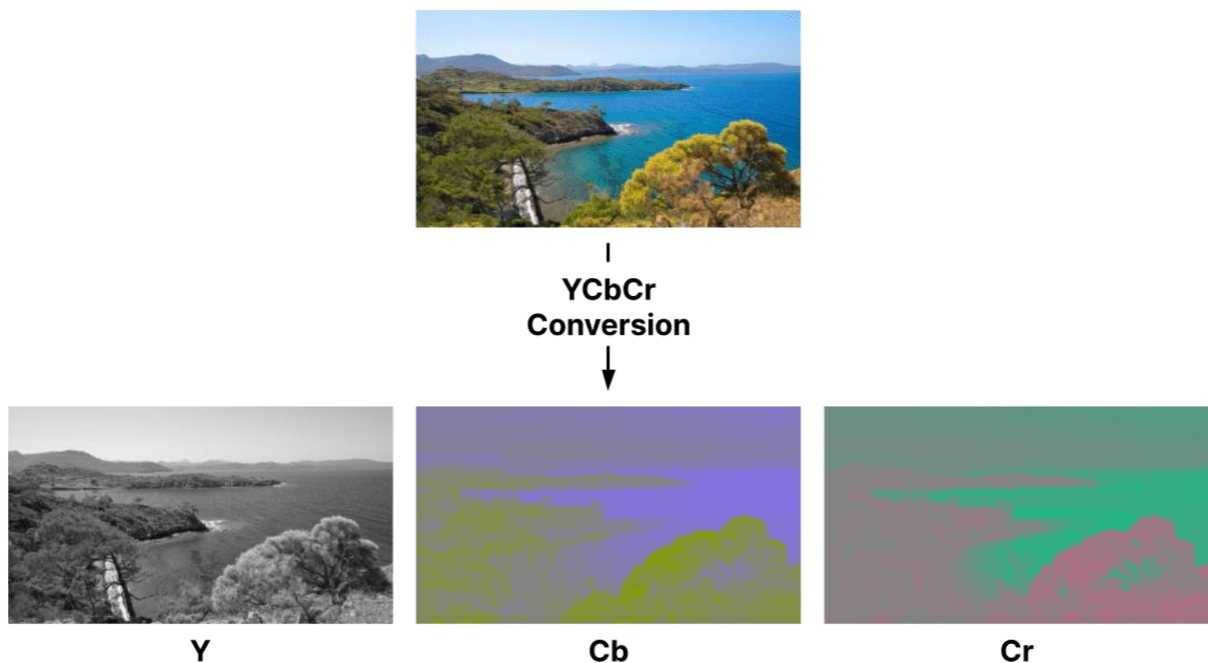


Рисунок 2.1 – Складові кольорового простору YCbCr у зображенні

Після віднімання каналів Y, Cb і Cr від зображення, залишок надає інформацію про хроматичність зеленого кольору. Це може бути математично виділено замість кодування, що дозволяє заощадити пропускну здатність.

Хрома-субдискретизація або зменшення роздільної здатності - це процес, при якому інформація про кольори в каналах C_b та C_r зображення відбирається на меншій роздільній здатності, ніж в оригіналі.

Існує кілька загальновживаних співвідношень субдискретизації для формату JPEG, включаючи:

Структура дискретизації сигналу позначається як співвідношення між трьома частинами X : a : b (наприклад, 4:2:2), що описують число вибірок яскравості і колірних сигналів. Також іноді використовується позначення з чотирма частинами (4:2:2:4), де четверта цифра, якщо вона включена, повинна бути ідентична першій цифрі, яка вказує на наявність сигналу четвертого каналу, що містить інформацію прозорості (альфа-канал). Цими частинами є:

- X - частота дискретизації каналу яскравості, виражена коефіцієнтом базової частоти (ширина макропікселя)
- a - число вибірок колірних сигналів (C_r , C_b) у горизонтальному напрямку в першому рядку
- b - число (додаткових) вибірок колірних сигналів (C_r , C_b) у другому рядку

Позначення субдискретизації має наступний вигляд:

- 4:4:4 - відсутність субдискретизації.
- 4:2:2 - зменшення вдвічі у горизонтальному напрямку (зменшення площі на 50%).
- 4:2:0 - зменшення вдвічі як у горизонтальному, так і у вертикальному напрямку (зменшення площі на 75%).
- 4:1:1 - зменшення у чверть у горизонтальному напрямку (зменшення площі на 75%).

Як видно, співвідношення субдискретизації виражається трьома частинами (J : a : b), де:

- J - це горизонтальний зразок відбору, який зазвичай складається з чотирьох пікселів.

- а - це кількість вибірок хроматичності (Cr, Cb) у першому рядку J пікселів.
- b - зміна між першим і другим рядками пікселів хроматичності. Це може бути або нульовим, або дорівнювати а.

Можна візуалізувати, як це працює, на наступному рисунку.

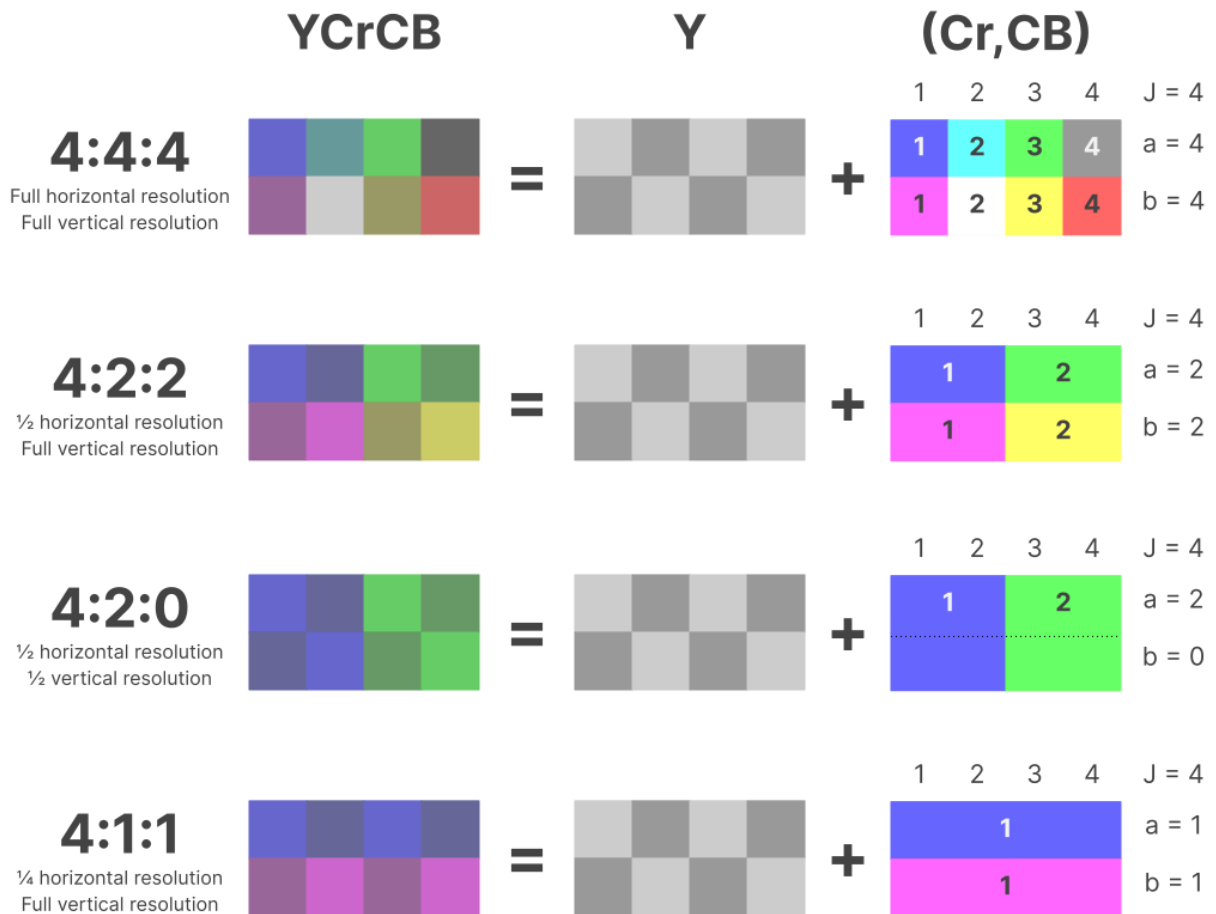


Рисунок 2.2 – Приклади субдискретизації

Контрастні кольори та люмінесценція на діаграмі використовуються для демонстраційних цілей.

Вигляд субдискретизації на практиці:

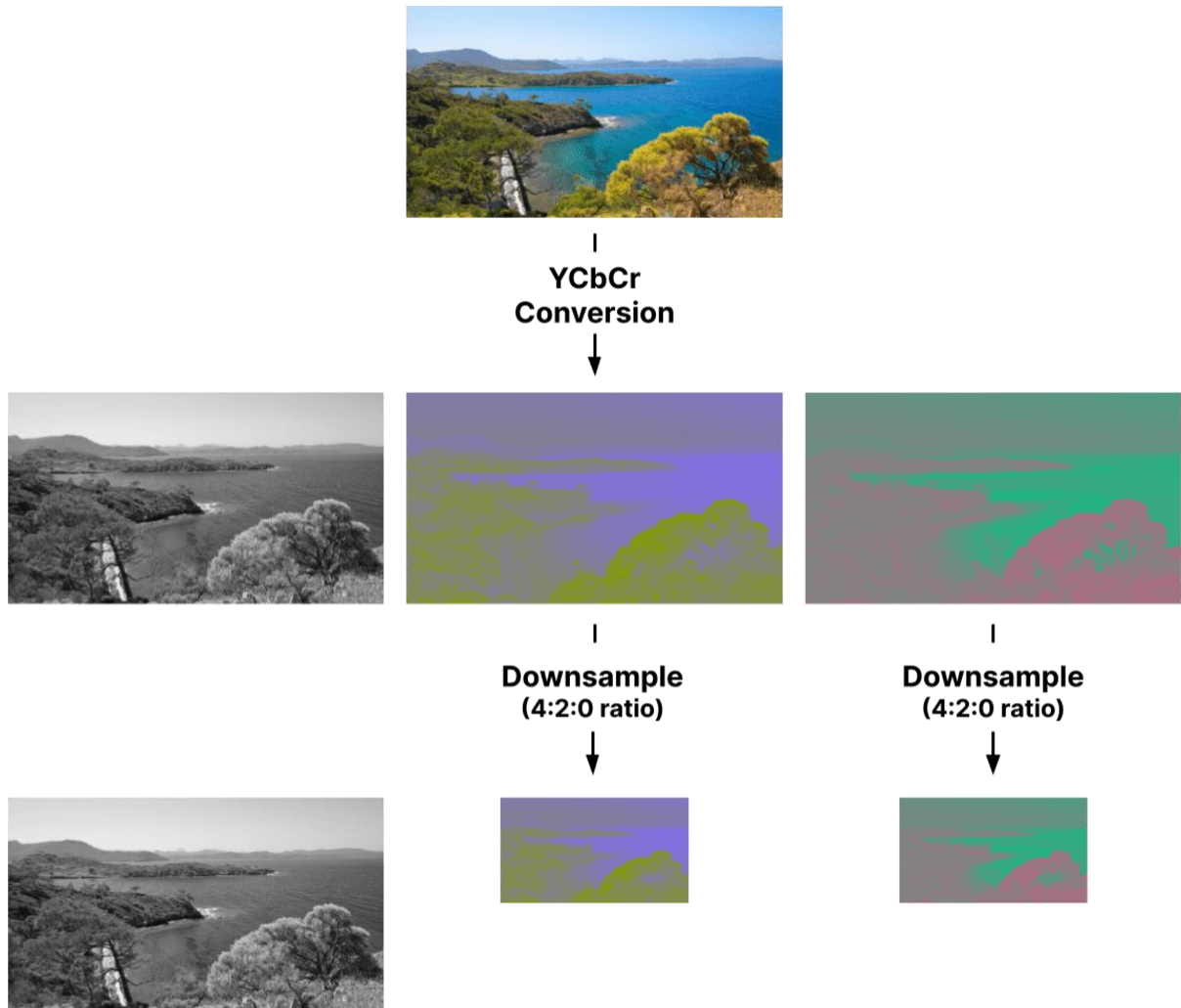


Рисунок 2.3 – Результат субдискретизації на зображенні

Рівень субдискретизації, або чи використовується він взагалі, може залежати від програмного забезпечення та налаштувань, використовуваних для створення файлу JPEG.

Наприклад, деяке графічне програмне забезпечення може вибирати не використовувати субдискретизацію на вищих рівнях якості. Процес субдискретизації є втратним. Іншими словами, пікселі втрачаються і не можуть бути відновлені.

Хроматична субдискретизація може значно скоротити розмір файлу з мінімальним впливом на вигляд зображення.

Конвертація YCbCr.



Рисунок 2.4 – Вплив субдескритизації на розмір файлу та якість зображення

Порівнюючи розмір файлу між оригінальним та після зменшення розміру економія обсягу файлу складає 17,9%. Більш реалістичне порівняння буде полягати в порівнянні розмірів файлів двох зображень після завершення всього процесу кодування JPEG.

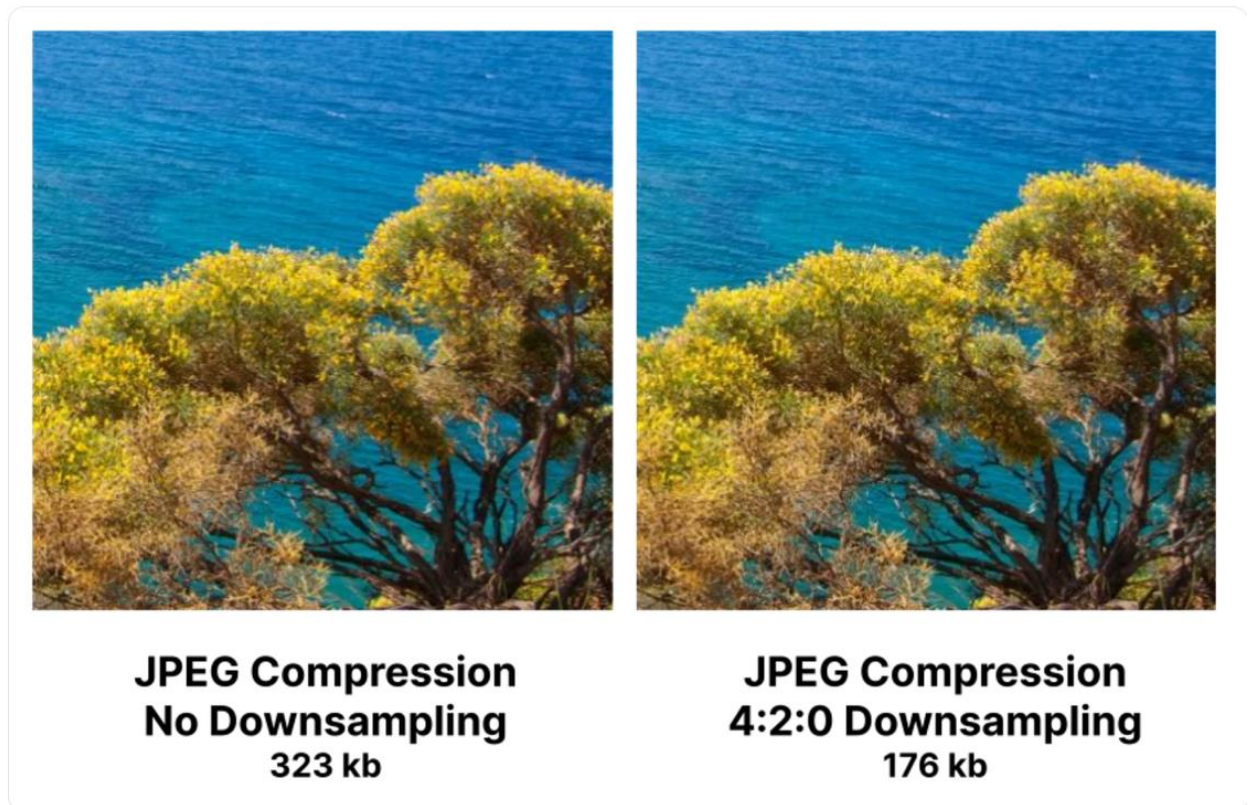


Рисунок 2.5 – Вплив субдискретизації на розмір файлу та якість зображення після JPEG алгоритму

Різниця в розмірі файлу між JPEG-зображенням без субдискретизації та JPEG-зображенням зі субдискретизацією значно помітніша, з економією обсягу файлу в розмірі 45,5%.

Хроматична субдискретизація є втратною, що означає, що в процесі втрачаються деякі дані, які не можна відновити. Проте в окремих ситуаціях можна дуже помітити втрату якості та інші артефакти.

Саме тому, в 2015 році, коли Колін Бенделл з Акатаї провів тест на 1 мільйоні JPEG-зображень, виявилось, що майже 60% зображень не використовували субдискретизацію взагалі. Крім того, лише приблизно 39% зображень використовували 4:2:0.

2.1.1.1 Артефакти хроматичної субдискретизації

Після зменшення роздільності зображення наступним кроком є збільшення розмірів каналів хроматичності до їхнього оригінального розміру. Під час субдискретизації не лише втрачаються дані пікселів, але також змінюються піксельні дані, коли зображення знову масштабується до оригінальних розмірів.

Процес збільшення зображення шляхом створення нових пікселів називається ресемплінгом.

На цьому етапі починають з'являтися перші артефакти, включаючи аліасинг (скалкування), розмиття та обрамлення країв.

Артефакти ресемплінгу.

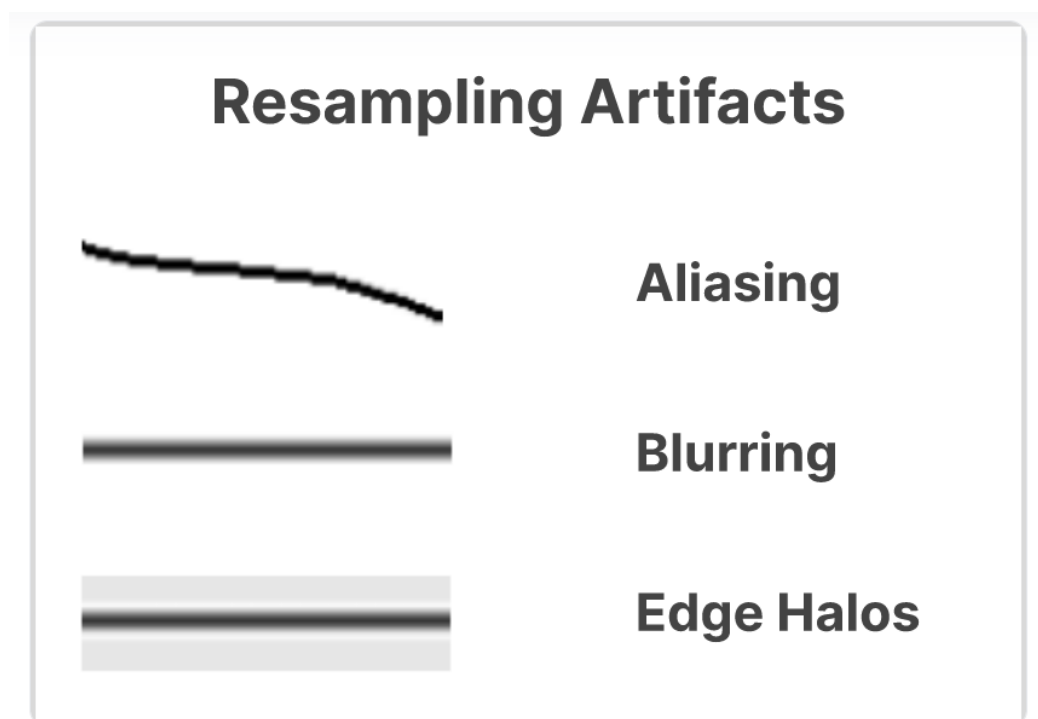


Рисунок 2.6 – Артефакти ресемплінгу

Ресемплінг (зі співвідношенням 4:2:0) бере оригінальний піксель і замінює його чотирма кольоровими пікселями з середньозваженими кольорами.

Кольори змішуються і можуть стати менш яскравими.

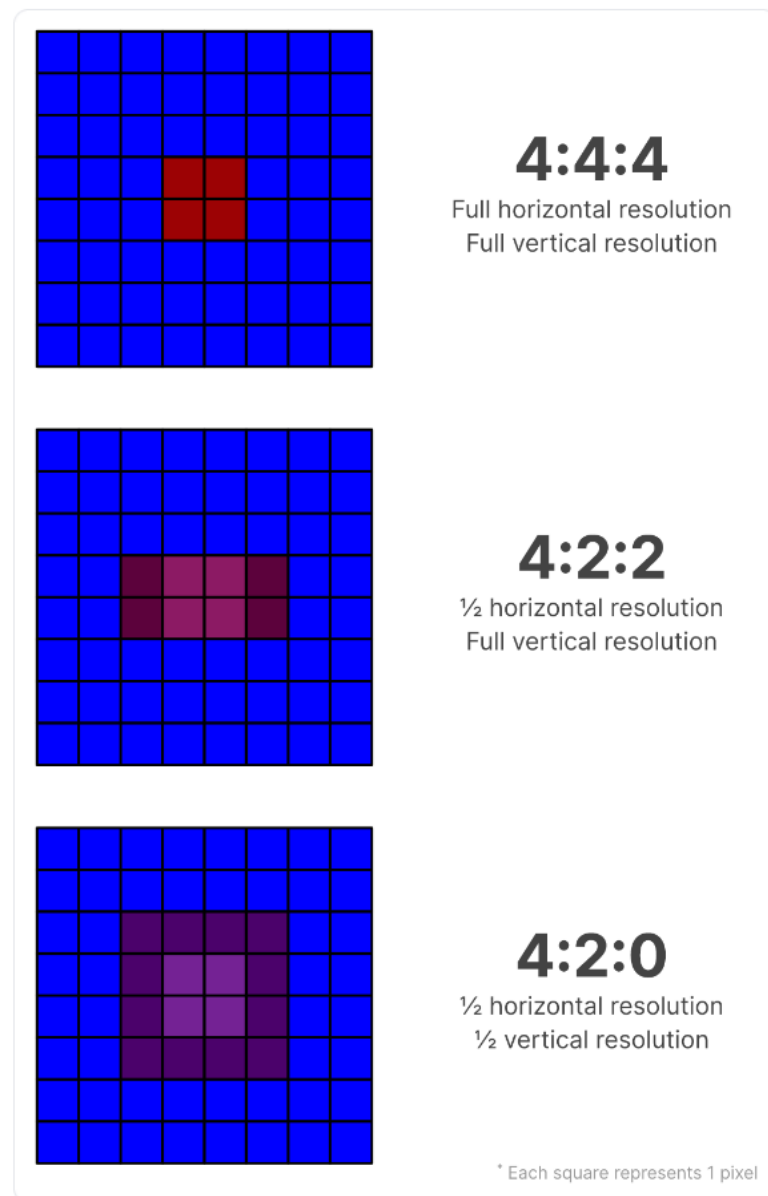


Рисунок 2.7 – Артефакти ресемплінгу

Різні методи ресемплінгу можуть допомогти зменшити вплив цих артефактів, включаючи найближчого сусіда, бікубічний, білінійний, Ланцош та інші.

Хроматична субдискретизація може викликати артефакти в певних ситуаціях, таких як текст на зображенні або інші місця, де кольори різко змінюються.

Артефакти різкої зміни кольору

Хроматичну субдискретизацію слід уникати, коли на зображенні є області, де кольори швидко змінюються. Це може включати логотипи, діаграми або предмети з різкими краями на контрастних фонів.

Це можна побачити на прикладі між зеленим та рожевим обрамленням на зображенні нижче. Перше зображення не має субдискретизації.



Рисунок 2.8 – Тестове зображення гамми без субдискретизації

Наступне зображення показує перше зображення після субдискретизації 4:2:0 та збільшення масштабу для подальшого розгляду.

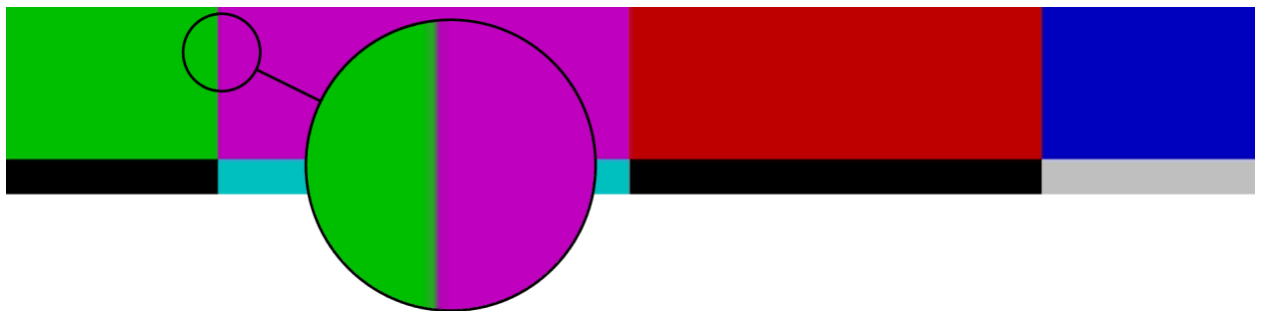


Рисунок 2.9– Тестове зображення гамми з субдискретизацією 4:2:0 та збільшення масштабу

Хроматичну субдискретизацію слід уникати при використанні кольорового тексту або тексту на кольорових фонах.

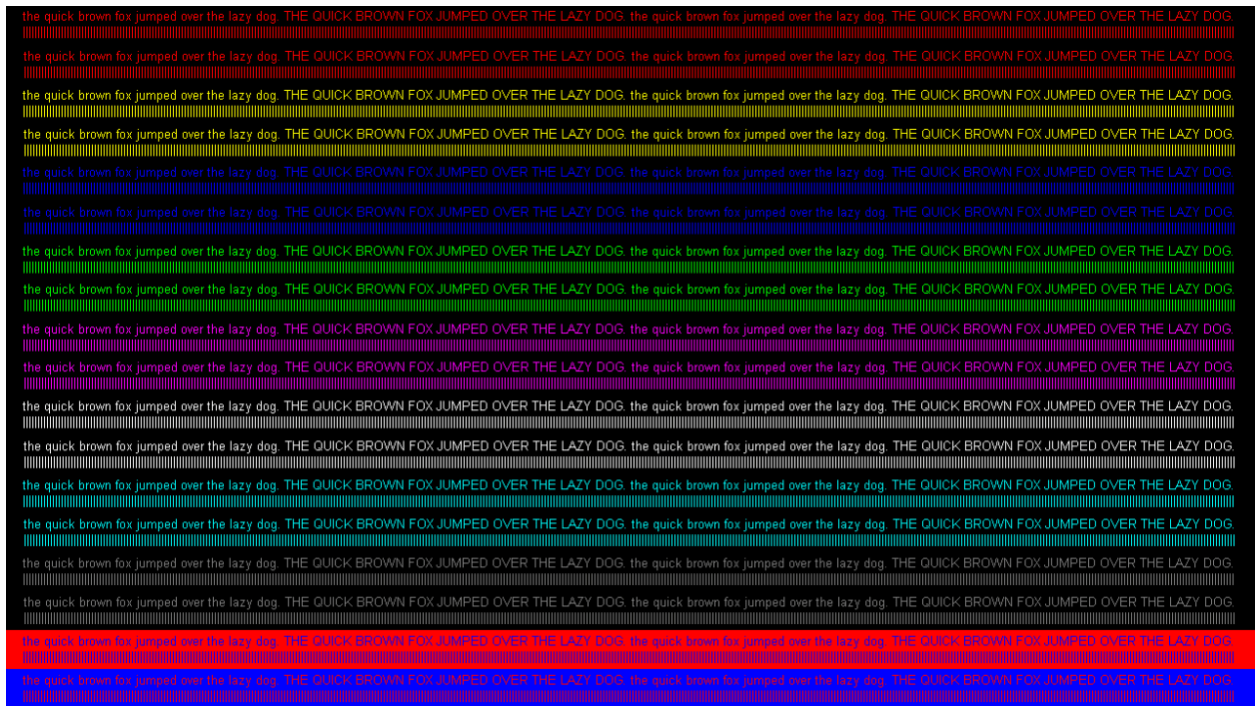


Рисунок 2.10 – Тестове зображення гамми без субдискретизації

Текст на цьому зображенні повністю читається, і кольори яскраві. Після субдискретизації 4:2:0 (високий роздільний здатність) він матиме помітні артефакти.

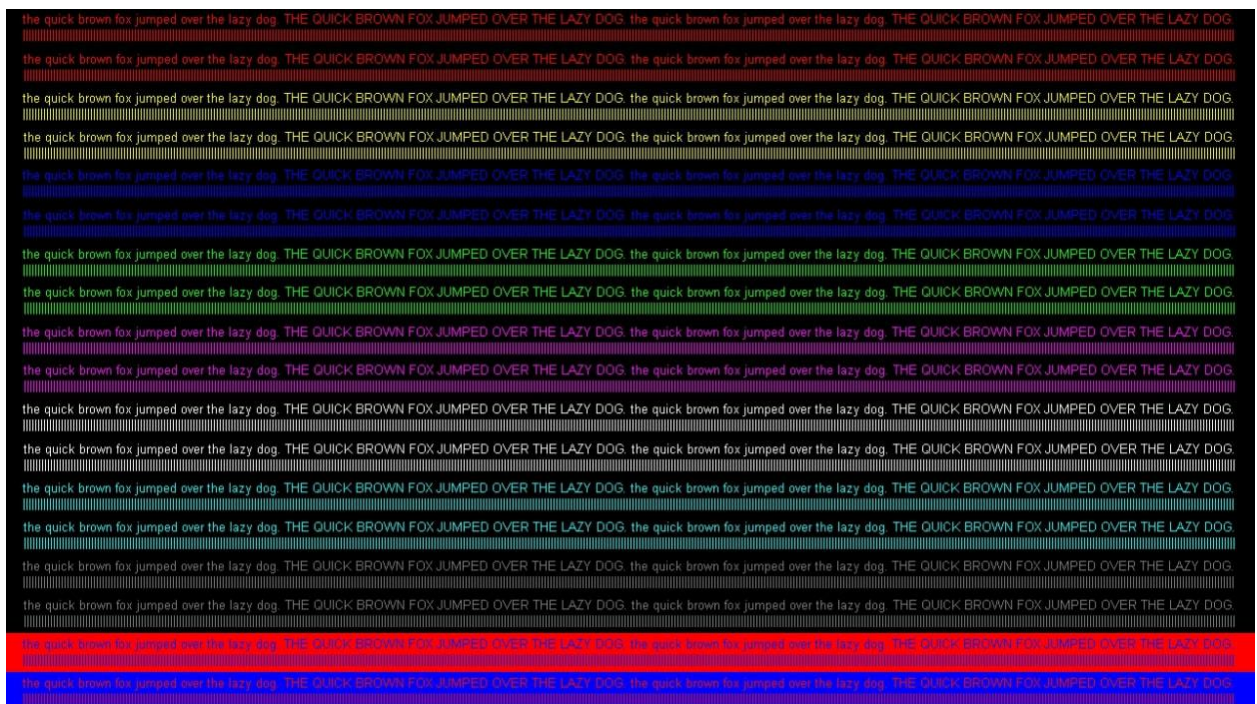


Рисунок 2.11– Тестове зображення гамми з субдискретизацією 4:2:

Деякий з кольорових текстів буде важко прочитати.

Протягом всього процесу кодування JPEG кольорові канали Y, Cb та Cr обробляються окремо.

2.1.2 Розділення зображення на 8x8 блоки

Розділення блоків - це процес поділу кожного каналу в джерелі зображення на піксельні блоки, які називаються Мінімальними Кодованими Блоками (MCU). Кожний канал може мати різний розмір MCU, залежно від того, чи був цей канал субдискретизований [4].

Типові розміри MCU для наступних співвідношень субдискретизації:

- 4:4:4 - 8 x 8 пікселів
- 4:2:2 - 16 * 8 пікселів
- 4:2:0 - 16 * 16 пікселів

Збільшення розміру MCU при субдискретизації зображення має сенс. Наприклад, при субдискретизації хроматичності 4:2:0 кожен блок з 4 пікселів усереднюється до одного кольору.

Розглянемо MCU розміром 8 x 8 пікселів для каналу яскравості.

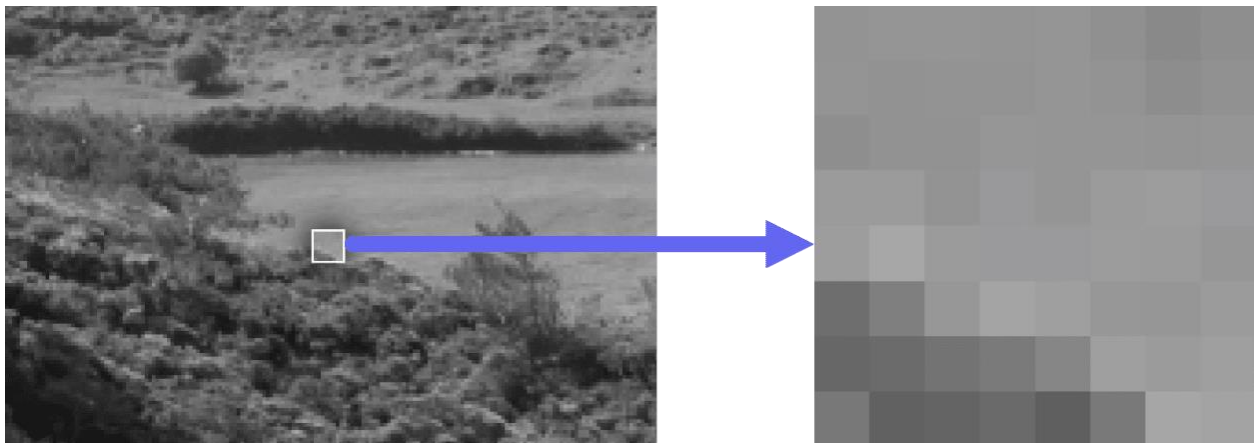


Рисунок 2.12 – Блок 8 на 8 з тестового зображення

Зображення ліворуч показує MCU в контексті збільшеної частини зображення. Зображення праворуч показує збільшену частину. Для роботи алгоритмів стиснення JPEG зображення повинні містити лише повні блоки MCU. Розміри зображення повинні бути кратними MCU, наприклад, 8 пікселів для співвідношення субдискретизації 4:4:4. Наприклад, зображення розміром 501 x 375 пікселів спершу має бути змінено на зображення розміром 504 x 376 пікселів.

Якщо в зображенні є часткові блоки MCU, то програмне забезпечення, яке кодує JPEG-зображення, повинно додати свої власні фіктивні дані до цих блоків.

Дані (або доповнення), додані до зображення, залежать від програмного забезпечення, але можуть використовуватися наступні техніки:

- Використання фіксованого кольору, наприклад, чорного.
- Повторення пікселів на краю зображення (найпоширеніший спосіб).
- Більш вдосконалені техніки.

Ці техніки можна побачити на наступному зображенні.

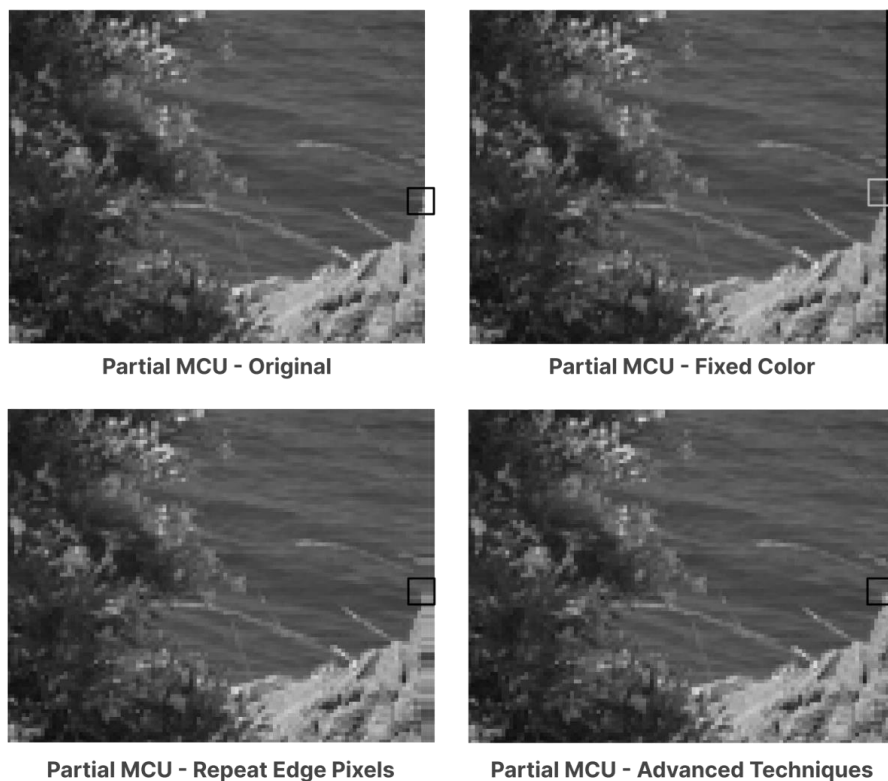


Рисунок 2.13 – Приклади технік доповнення інформації в зображенні

Як тільки фіктивні дані додаються, блоки проходять решту процесу кодування JPEG. Завершивши процес, додаткові пікселі видаляються.

2.1.2.1 Артефакти блокування

У певних обставинах поділ зображення на MCU може викликати деякі візуальні розриви.

Наприклад, робочі границі MCU можуть бути помітними, якщо JPEG було стиснуто дуже низькою якістю.



Original



Low Quality

Рисунок 2.14 – Приклад артефактів блокування

Наведемо межі для візуального розділенні блоків.



Original



Low Quality

Рисунок 2.15 – Приклад артефактів блокування з накладенням MCU

По границях MCU видно раптові зміни кольору та інтенсивності.

MCU Boundaries Clearly Visible



Рисунок 2.16 – Приклад помітних артефактів блокування

Це результат того, що кожен MCU стискується окремо, і з кожного з них видаляється різна кількість інформації. Це справедливо лише для найнижчих налаштувань якості.

2.2 DCT (Дискретне косинусне перетворення)

Дискретні сигнали представляються у вигляді послідовностей (як правило) обмеженої довжини, тоді як перетворення виконуються у вигляді простих множень на оборотні матриці. Перетворена версія дискретної послідовності є альтернативним проявом у векторному просторі дискретних послідовностей. Це альтернативне уявлення може краще описати деякі особливості сигналу, тому деякі операції обробки сигналу можуть мати меншу складність. Дискретні перетворення сигналу є блочно-орієнтованими операціями, тобто повна вхідна послідовність матиме альтернативний вигляд того ж розміру [5]. Тому хорошим варіантом може бути розкладання вхідних даних на частини меншого розміру та застосування послідовностей перетворень до цих блоків. Ця блокова орієнтація обмежує «роздільну здатність» представлення області перетворення, з одного боку, з іншого, вона краще підходить для характеристики області перетворення нестационарних сигналів [5].

Характеристика чи моделювання сигналу можливе за допомогою відомих сигналів. Цей вид опису представляє собою апроксимацію і часто називається репрезентацією сигналу. З використанням прийомів під час апроксимації існують кілька загальних аспектів цих різних концепцій. Цікаве тлумачення можна надати, якщо цю ідею втілити в так званих просторах сигналів. Сигнальні простори можна розглядати як узагальнення евклідового простору, де кожна точка у просторі представляє сигнал. Зберігається концепція відстані, яка може служити для оцінки взаємозв'язку різних сигналів. У типових просторах сигналів важливу роль відіграють поняття метричних, алгебраїчних операцій, лінійності, норми та скаляра, або внутрішнього добутку. Використання останнього також дозволяє визначати ортогональність двох сигналів. Усі ці концепції можна також розширити на стохастичні сигнали [5]. У випадку дискретних сигналів, простір сигналів є просто векторним простором з елементарними векторними алгебраїчними операціями.

Розглянемо простори векторів розмірності N . Це означає, що всі вектори, які складаються з реальних або комплексних чисел, мають цю розмірність, і всі операції у цьому просторі можна представити у вигляді обернених $N \times N$ матриць. Кожна точка в даному просторі є дискретним сигналом і може бути описана своїми координатами. Якщо застосувати перетворення до конкретного елемента векторного простору, новий вектор виражатиме ту ж інформацію в іншій системі координат. Ця інша система координат визначається за допомогою N лінійно незалежних векторів того самого векторного простору, і нові координати задаються через зазначене перетворення [5].

Для двовимірних квадратних масивів або зображень вирази мають вигляд:

$$T(u,v) = \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x,y)r(x,y,u,v), \quad (2.1)$$

та

$$f(x,y) = \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} T(u,v)s(x,y,u,v), \quad (2.2)$$

де $r(x,y,u,v)$ і $s(x,y,u,v)$ – це ядра прямого та оберненого перетворення. Перетворення $T(u,v)$ і ядро оберненого перетворення $s(x,y,u,v)$ знову можна представити як вагові коефіцієнти та базисні вектори відповідно з рівнянням (2.2) визначення лінійного розкладання $f(x,y)$. Ядро прямого перетворення $r(x,y,u,v)$ є роздільним, якщо

$$r(x,y,u,v) = r_1(x,u)r_2(y,v), \quad (2.3)$$

та симетрична, якщо r_1 функціонально дорівнює r_2 , отже

$$r(x, y, u, v) = r_1(x, u)r_1(y, v). \quad (2.4)$$

Якщо ядра перетворення є дійсними й ортонормованими, і обидва r і s є роздільними та симетричними, матричні еквіваленти рівнянь. (2.1) і (2.2) є

$$T = AFA^T, \quad (2.5)$$

та

$$F = A^T T A. \quad (2.6)$$

де F — матриця $N \times N$, що містить елементи $f(x, y)$, T — її перетворення $N \times N$, а A — об'єднання N базисних векторів перетворення в матрицю перетворення $N \times N$. Попереднє і після множення F на A і A^T в рівнянні (2.5) обчислюють перетворення стовпців і рядків F відповідно. Це розбиває 2-D перетворення на два 1-D перетворення, дзеркально відображаючи процес для 2-D DFT [5].

2.2.1 Застосування DCT до кожного 8x8 блоку

Кожний 8x8 піксельний блок кожного каналу (Y, Cb, Cr) конвертується з просторового представлення в частотний діапазон за допомогою двовимірного дискретного косинусного перетворення типу II.

Косинусна функція, така як $y = \cos(x)$, змінюється між 1 та -1 на вісі Y, від 0 до π , потім 2π і так далі.

Цю косинусну функцію можна візуалізувати на наступній діаграмі у вигляді хвилі:

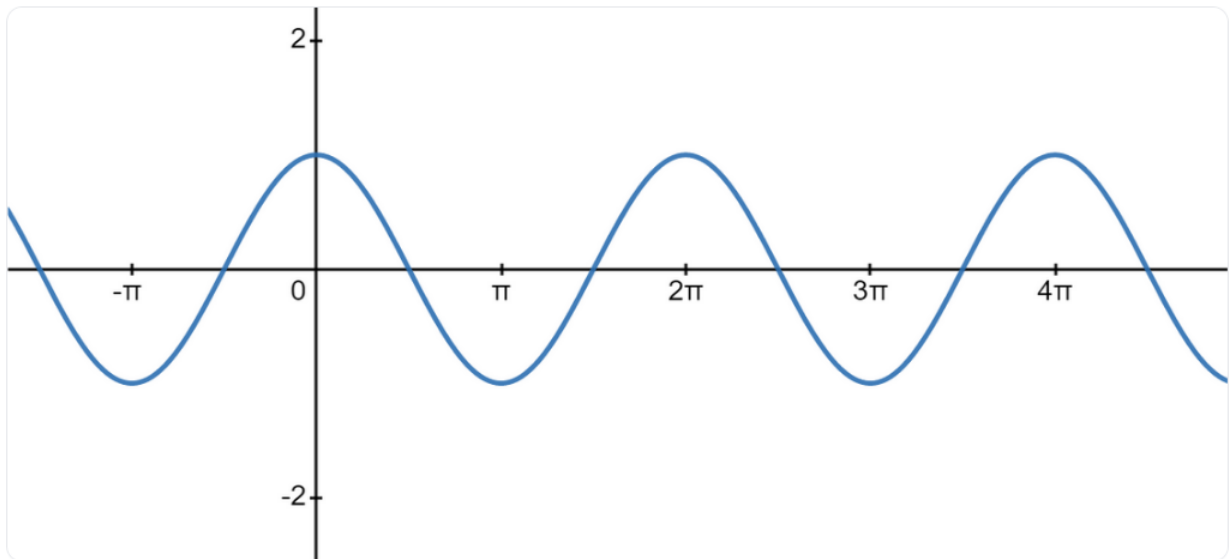


Рисунок 2.17 – Приклад простої косинусної функції

Як можна побачити, коли $x=0$, $y=1$. При $x=\pi$, $y=-1$. Нарешті, при $x=2\pi$, $y=1$ знову. Ця закономірність повторюється.

Суть роботи дискретного косинусного перетворення полягає в тому, що ми беремо дані зображення і намагаємося представити їх як суму декількох косинусних хвиль.

Наприклад є стандартна косинусна хвиля та друга хвиля із значно вищою частотою.

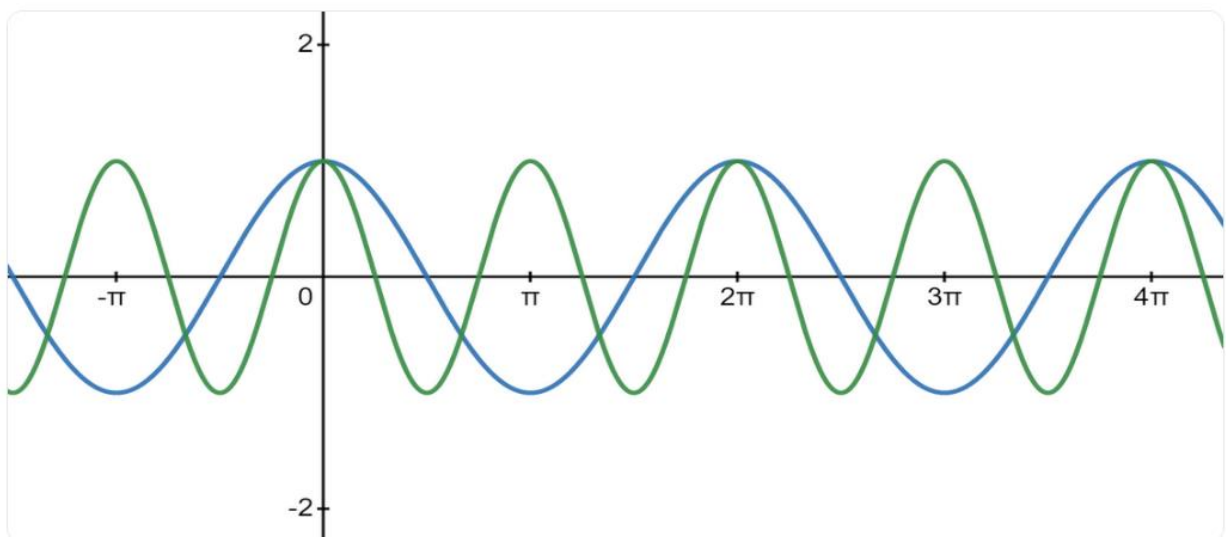


Рисунок 2.18 – Приклад косинусної функції з двома хвилями

Якщо їх додати разом, ми отримаємо зовсім іншу і значно складнішу хвилю.

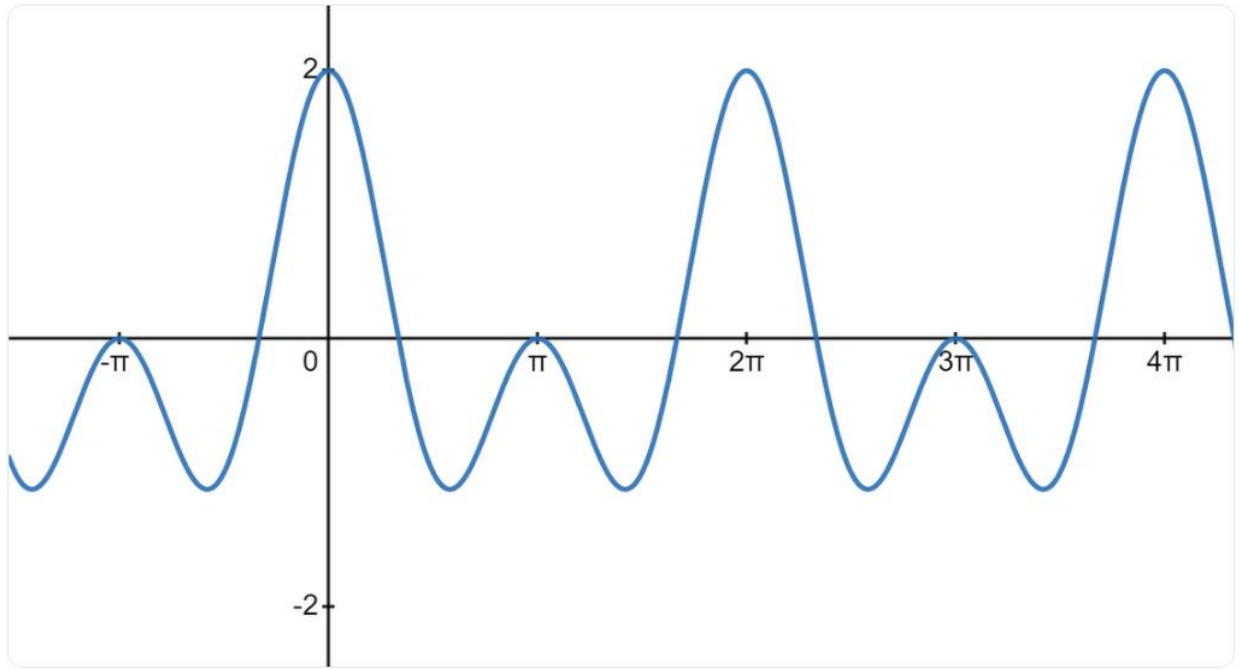


Рисунок 2.19 – Результат обчислення косинусної функції з двома хвилями

Отримана хвиля вдвічі вища, ніж вихідний сигнал, тому ми використовуємо зважену середню.

У цьому випадку було поділено всі значення на два через використання двох косинусних хвиль.

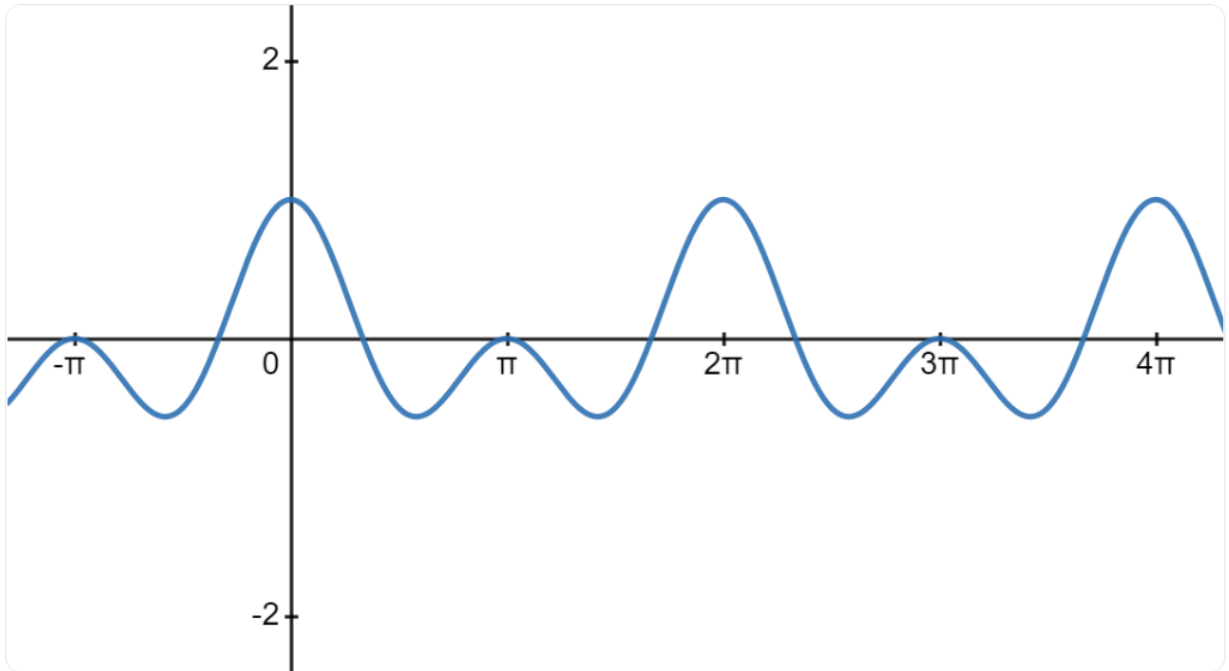


Рисунок 2.20 – Нормований результат обчислення косинусної функції з двома хвилями

На практиці можна комбінувати більше 2 або більше косинусних хвиль, щоб створити набагато складніші зображення. Фактично будь-який 8×8 піксельний блок можна представити у вигляді суми вагованих (позитивних або негативних) косинусних перетворень.

JPEG використовує набір з 64 різних косинусних значень для створення будь-якого 8×8 піксельного блоку. Ці значення візуалізовані нижче:

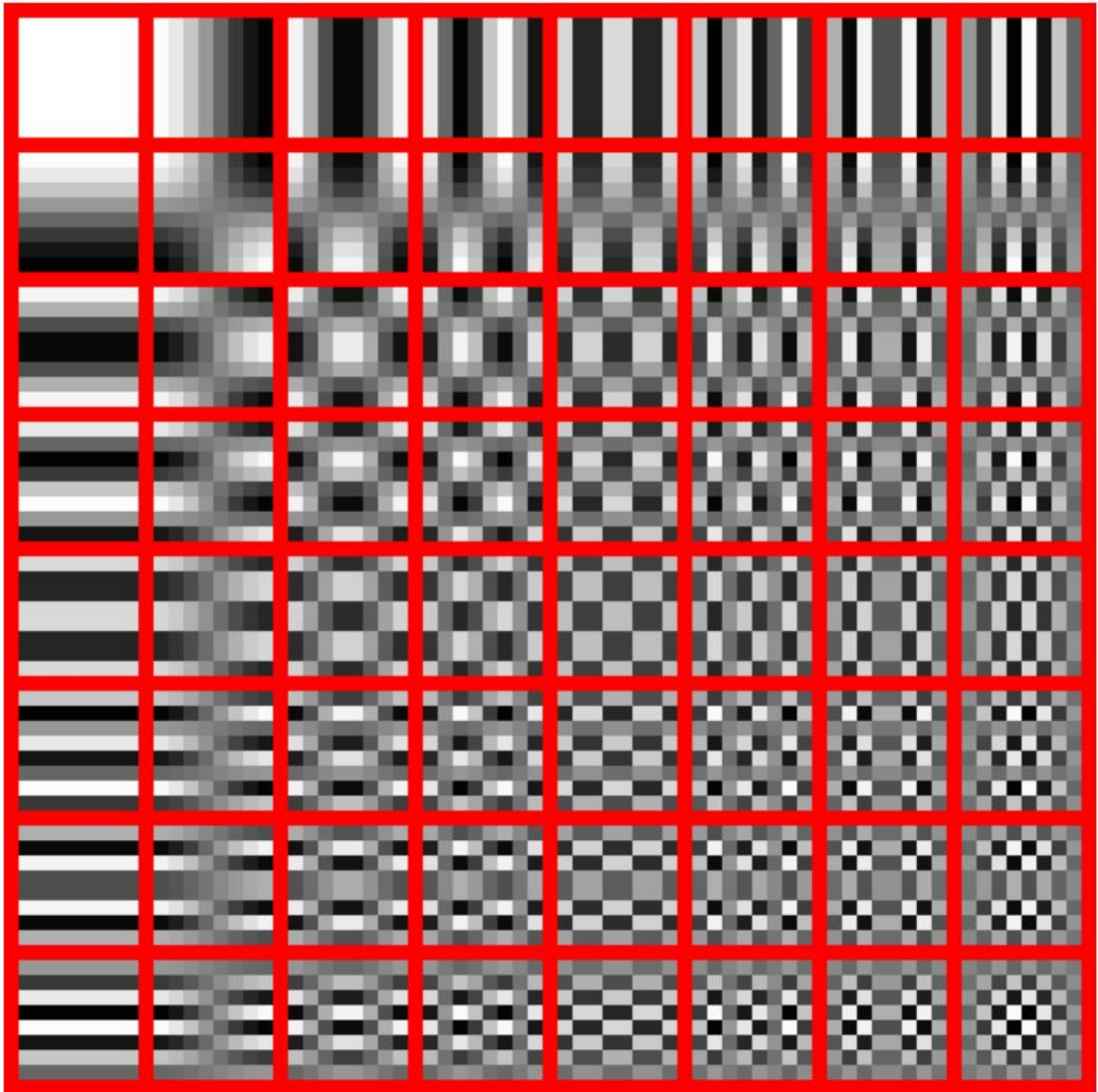


Рисунок 2.21 – Стандартні косинусні значення JPEG

Детальний огляд того, як це працює.

Крок 1. Перетворення в Частотно-Доменне Представлення

Для проведення розрахунків на зображенні спочатку необхідно конвертувати кожен канал Y, Cb і Cr в частотно-доменне представлення.

Та маю наступний вигляд:

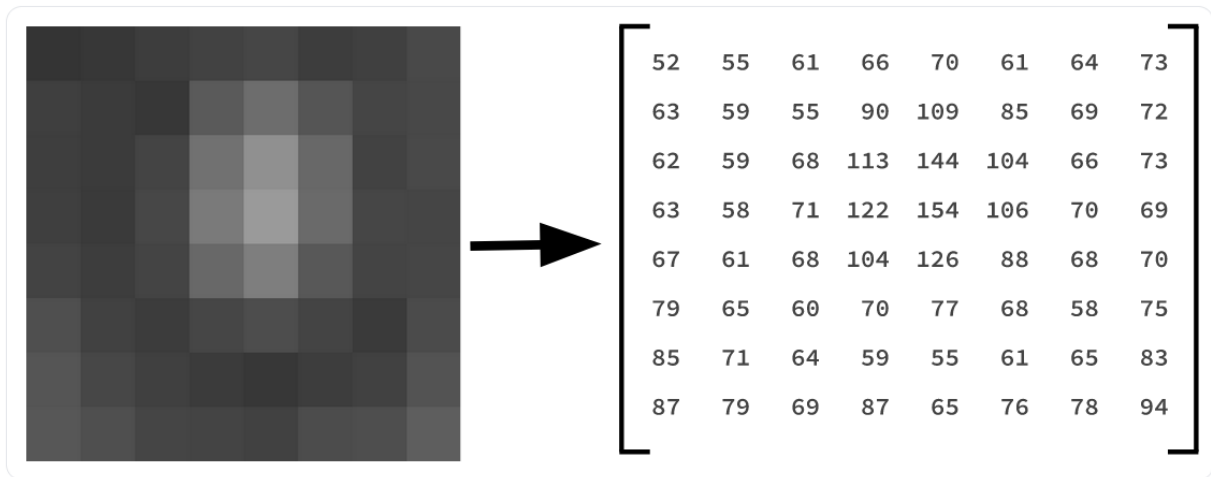


Рисунок 2.22– Представлення блоку зображення 8 на 8 у математичній матриці

Кожна цифра в матриці представляє піксель в 8x8 зображенні. Темні пікселі представлені меншими цифрами, а світлі пікселі мають більше значення. Також вона має значення між 0 та 255, і середина становить 128.

Крок 2. Перецентрування навколо Нуля

Перед обчисленням DCT для кожного 8x8 блоку, кожне значення в матриці повинно бути перецентроване навколо нуля. Іншими словами, середина повинна бути 0.

Тому кожна цифра повинна бути зменшена на 128, щоб діапазон став від -128 до 127.

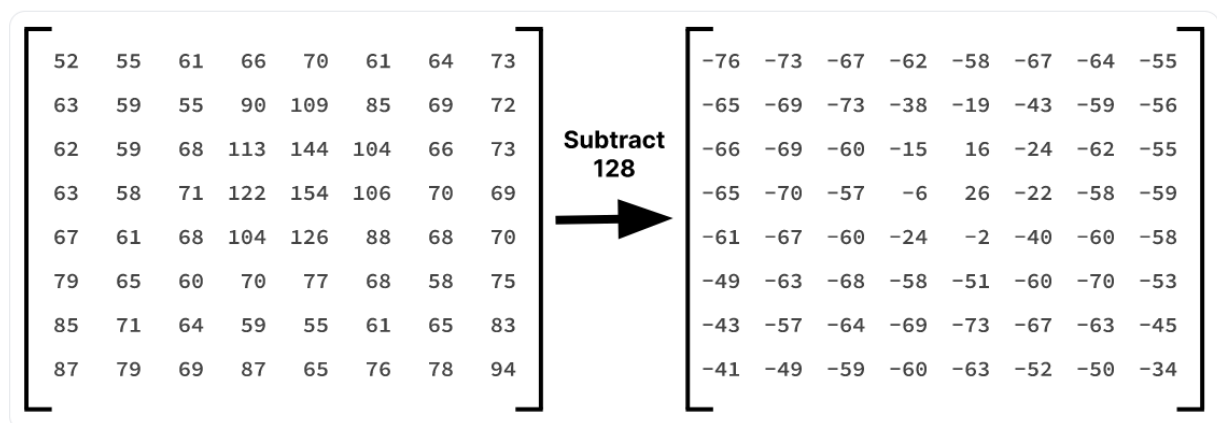


Рисунок 2.23 – Перетворення значень матриці навколо нуля

Причина перецентрування навколо нуля полягає в зменшенні вимог до динамічного діапазону на етапі обчислення DCT, який ми розглянемо далі.

Крок 3. Розрахунок коефіцієнтів ДКП.

Наступний крок - розрахувати двовимірні коефіцієнти Дискретного Косинусного Перетворення (DCT), які можуть бути обчислені за наступною формулою [4]:

$$G_{u,v} = \frac{1}{4} \alpha(u) \alpha(v) \sum_{x=0}^7 \sum_{y=0}^7 g_{x,y} \cos \left[\frac{(2x+1)u\pi}{16} \right] \cos \left[\frac{(2y+1)v\pi}{16} \right] \quad (2.7)$$

Проведемо обчислення на прикладі матриці вище, щоб отримати коефіцієнти DCT.

-415.38	-30.19	-61.20	27.24	56.12	-20.10	-2.39	0.46
4.47	-21.86	-60.76	10.25	13.15	-7.09	-8.54	4.88
-46.83	7.37	77.13	-24.56	-28.91	9.93	5.42	-5.65
-48.53	12.07	34.10	-14.76	-10.24	6.3	1.83	1.95
12.12	-6.55	-13.20	-3.95	-1.87	1.75	-2.79	3.14
-7.73	2.91	2.38	-5.94	-2.38	0.94	4.30	1.85
-1.03	.18	0.42	-2.42	-0.88	-3.02	4.12	-0.66
-0.17	.14	-1.07	-1.07	-1.17	-0.10	0.50	1.68

Рисунок 2.24 – Матриця після виконання ДКП

Існують два типи коефіцієнтів DCT:

- DC коефіцієнт - Це коефіцієнт DCT в лівому верхньому куті, інакше відомий як постійний компонент. Це визначає основний відтінок для всього 8x8 піксельного блоку.
- AC коефіцієнти - Це решта 63 коефіцієнти DCT, інакше відомі як змінні компоненти.

Розрахунок DCT схильний до агрегації більшого сигналу в одному куті матриці, що означає, що ці косинусні хвилі вносять більшу кількість даних в зображення.

2.2.2 Квантування результатів DCT

Процес квантування спрямований на зменшення загального розміру коефіцієнтів DCT, щоб їх можна було більш ефективно стиснути на останньому етапі кодування Ентропії.

Квантування базується на принципі, що люди краще розрізняють невеликі відмінності в яскравості на великій області, але не дуже добре розрізняють високочастотні зміни в яскравості.

Коефіцієнти в нижньому правому куті матриці в частотному домені мають набагато менший вплив на кінцеве зображення, ніж коефіцієнти низької частоти в лівому верхньому куті.

Процес квантування використовує цей факт для значного зменшення кількості інформації у високочастотних компонентах. Це відбувається шляхом ділення коефіцієнта DCT на матрицю квантування. Матриця квантування визначає ступінь стиснення, відомий як налаштування якості (зазвичай від 0 до 100).

Для розрахунку квантованих коефіцієнтів DCT діляться коефіцієнти DCT на матрицю квантування. Результат потім округлюється до найближчого цілого числа. Процес квантування кожного каналу Y (яскравість), Cb (колір синій) і Cr (колір червоний) роблять окремо. В JPEG-зображенні є дві таблиці квантування: одна для каналу яскравості та одна для каналів кольору.

Типова матриця квантування для каналу яскравості при якості 50% (як вказано в стандарті JPEG) має наступний вигляд:

16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	55
14	13	16	24	40	57	69	56
14	17	22	29	51	87	80	62
18	22	37	56	68	109	103	77
24	35	55	64	81	104	113	92
49	64	78	87	103	121	120	101
72	92	95	98	112	100	103	99

Рисунок 2.25– Матриця Квантування Яскравості - 50% якості

Для каналів кольору використовується наступна матриця квантування:

17	18	24	47	99	99	99	99
18	21	26	66	99	99	99	99
24	26	56	99	99	99	99	99
47	66	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99

Рисунок 2.26 – Матриця Квантування Кольору - 50% якості

Чим більше значення в матриці, тим більше стиснення та нижча якість. Матриці квантування, використовувані при кодуванні JPEG, зберігаються в зображенні. Це тому, що вони потрібні для процесу декодування, про який я розповім пізніше в цьому розділі. Наступним кроком є використання цих матриць для квантування коефіцієнтів DCT [4].

Для розрахунку квантованих коефіцієнтів DCT ви повинні поділити коефіцієнти DCT на таблицю квантування і округлити їх до найближчого цілого числа.

$$B_{j,k} = \text{round} \left(\frac{G_{j,k}}{Q_{j,k}} \right) \text{ for } j = 0,1,2,\dots,7; k = 0,1,2,\dots,7 \quad (2.8)$$

Де:

- B = Квантовані коефіцієнти DCT
- G = Коефіцієнти DCT
- Q = Таблиця квантування

Застосовуючи до всієї матриці коефіцієнтів DCT, дає такий результат:

-26	-3	-6	2	2	-1	0	0
0	-2	-4	1	1	0	0	0
-3	1	5	-1	-1	0	0	0
-3	1	2	-1	0	0	0	0
1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Рисунок 2.27– Квантовані Коефіцієнти DCT

На зображення впливає лише 20 косинусних функцій. Багато високочастотних компонентів було округлено до нуля, а інші зменшено до одиничних цифр. Через це кількість бітів, необхідних для збереження зображення, значно зменшилася, що дозволило зменшити розмір файлу JPEG.

Окрім хроматичного субсемплінгу, цей крок квантування є єдиним втратним процесом даних у всьому процесі стиснення JPEG. Кількість втрат

даних на зображенні залежить від таблиці квантування , яку використовують, або, кажучи простіше, від рівня якості.

Таблиця 3.1 – Вплив квантування на якість зображення.

Зображення	Якість	Розмір	Ступінь стиснення
	Вища якість (100)	81.4 кб	2.7:1
	Висока якість (50)	14.7 кб	15:1
	Середня якість (25)	9.4 кб	23:1

Продовження таблиці 3.1

Зображення	Якість	Розмір	Ступінь стиснення
	Низька якість (10)	4.8 кб	46:1
	Найнижча якість (0)	1.5 кб	144:1

2.3 Кодування

Кодування в JPEG використовується для стиснення зображень, щоб зменшити їхній розмір, зберегти місце на сховищі та покращити швидкість передачі через мережу. Основна мета кодування JPEG - зменшити розмір файлу зображення, зберігаючи при цьому якість зображення на прийнятному рівні [4].

Також воно допомагає досягти високого ступеня стиснення без втрати якості, що робить його популярним для збереження та передачі зображень в онлайн-середовищі, на фотокамерах і при зберіганні зображень на сховищах.

2.3.1 Використання кодування диференціалів DC-коефіцієнтів

DC-коефіцієнт (перший елемент матриці квантованих DCT-коефіцієнтів) не враховується при застосуванні кодування ентропії. Він обробляється за допомогою Диференційної Пульсової Кодової Модуляції, або DPCM.

Замість зберігання повного значення DC-коефіцієнта для кожного 8x8 піксельного блоку, зберігається різниця між поточним і попереднім блоком. Зміни між DC-коефіцієнтами зазвичай невеликі, це часто призводить до збереження менших чисел, які займають менше місця. DPCM продемонстровано на зображенні нижче:

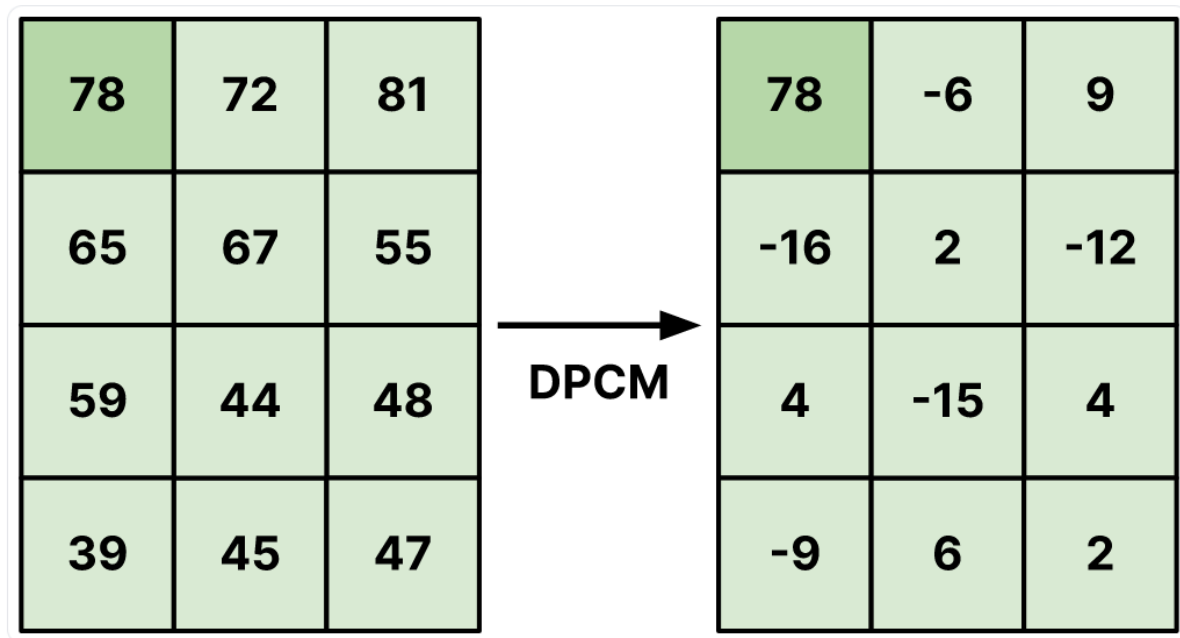


Рисунок 2.28 – Приклад кодування DC-коефіцієнтів

Кодування ентропії на DC-коефіцієнтах за допомогою Диференційної Пульсової Кодової Модуляції. Зображення показує приклад, що показує відмінності.

2.3.2 Використання кодування амплітуд АС-коефіцієнтів за допомогою Zig-Zag сканування

Схеми кодування за довжиною послідовності (RLE) використовувалися для передачі аналогових телевізійних сигналів ще у 1967 році [6]. У 1983 році компанією Hitachi був отриманий патент на кодування за довжиною послідовності [7][8][9]. RLE особливо підходить для зображень на основі палітри, таких як іконки на комп'ютері, і використовувався як популярний метод стиснення зображень на ранніх онлайн-сервісах, таких як CompuServe, перед появою більш вдосконалених форматів, таких як GIF [10]. Він не працює добре для зображень з неперервними відтінками, таких як фотографії, хоча JPEG використовує його для коефіцієнтів, які залишаються після перетворення та квантування блоків зображення.

Порядок обробки коефіцієнтів під час кодування довжиною послідовності має значення. Якщо зчитувати послідовності горизонтально, ми отримаємо наступне:

-26	-3	-6	2	2	-1	0	0
0	-2	-4	1	1	0	0	0
-3	1	5	-1	-1	0	0	0
-3	1	2	-1	0	0	0	0
1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Рисунок 2.29 – зчитування DCT-коефіцієнтів в горизонтальному порядку для порівняння

Отримаємо послідовність:

-26 -3 -6 2 2 -1 0 0 0 0 1 1 -4 -2 0 -3 1 5 -1 -1 0 0 0 0 0 0 0 -1 2 1 -3 1 0 0 0 0
0 0

Якщо ми розмістимо коефіцієнти у зигзаговому вигляді, ми отримаємо наступне:

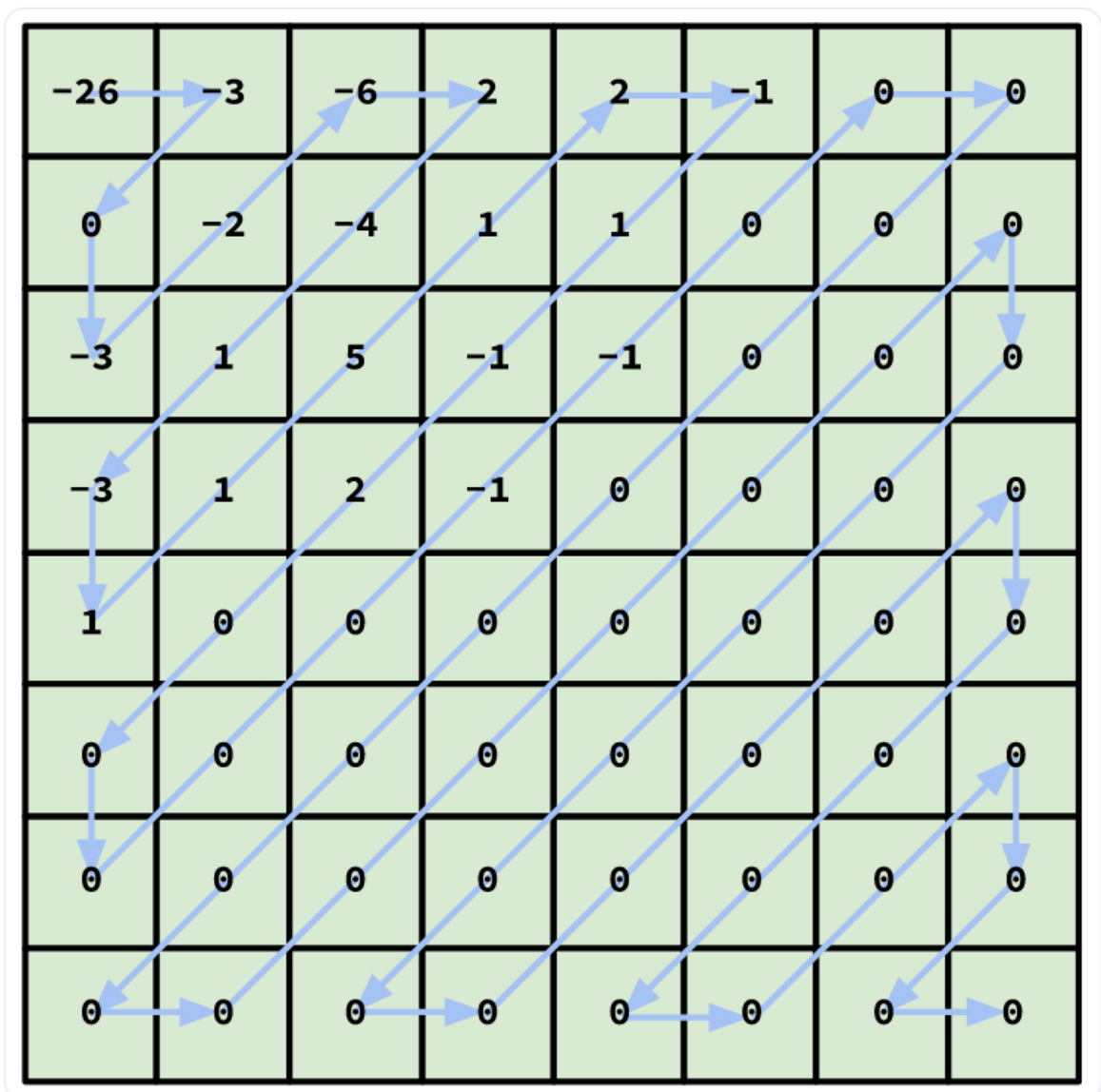


Рисунок 2.30 – зчитування DCT-коефіцієнтів у зигзаговому порядку

$(0, 2)(-3); (1, 2)(-3); (0, 1)(-2); (0, 2)(-6); (0, 1)(2); (0, 1)(-4); (0, 1)(1); (0, 2)(-3); (0, 1)(1); (0, 1)(1); (0, 2)(5); (0, 1)(1); (0, 1)(2); (0, 1)(-1); (0, 1)(1); (0, 1)(-1); (0, 1)(2); (5, 1)(-1); (0, 1)(-1); (0, 0);$

Якщо слідувати амплітудам у вищевказаному, то вони йдуть у вигляді "-3, -3, -2 ...", а не "-3, 0, -3, -2". Перший нуль опущено з послідовності. Тому що він представлений RUNLENGTH у другому елементі послідовності "(1, 2)(-3)". У цьому випадку довжина кодування дорівнює 1, що вказує на те, що перед амплітудою "-3" йде нуль.

У попередньому прикладі.

$(0, 2)(-3); (1, 2)(-3); (0, 1)(-2); (0, 2)(-6); (0, 1)(2); (0, 1)(-4); (0, 1)(1); (0, 2)(-3); (0, 1)(1); (0, 1)(1); (0, 2)(5); (0, 1)(1); (0, 1)(2); (0, 1)(-1); (0, 1)(1); (0, 1)(-1); (0, 1)(2); (5, 1)(-1); (0, 1)(-1); (0, 0);$

Послідовність містить два особливі кодові слова:

- $(0, 0);$ - використовується для позначення закінчення послідовності там, де залишилися нульові коефіцієнти. Це відомо як "Кінець блоку" або "EOB" (End-of-block).
- $(15, 0)(0)$ - використовується, коли кількість нулів перевищує 15, перш ніж з'явиться наступний ненульовий коефіцієнт AC.

2.4 Декодування зображення у форматі JPEG

Усі кроки повинні бути виконані в зворотному порядку для декодування зображення. Після зворотнього етапу кодування ентропії (включаючи розкодування DC-коефіцієнта), ми починаємо з наступної матриці:

-26	-3	-6	2	2	-1	0	0
0	-2	-4	1	1	0	0	0
-3	1	5	-1	-1	0	0	0
-3	1	2	-1	0	0	0	0
1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Рисунок 2.31 – Квантовані DCT коефіцієнти

Існують дві матриці квантування: одна для яскравості та одна для хроматичних каналів. Розглянемо канал яскравості.

Має використовуватись та сама матриця квантування, з якою зображення кодувалось. У цьому випадку матриця, зазначена в стандарті JPEG для якості 50%. Вона показана нижче.

16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	55
14	13	16	24	40	57	69	56
14	17	22	29	51	87	80	62
18	22	37	56	68	109	103	77
24	35	55	64	81	104	113	92
49	64	78	87	103	121	120	101
72	92	95	98	112	100	103	99

Рисунок 2.32 – Матриця квантування яскравості з якістю 50%

Результатом буде наступна матриця:

-416	-33	-60	32	48	-40	0	0
0	24	-56	19	26	0	0	0
-42	13	80	-24	-40	0	0	0
-42	17	44	-29	0	0	0	0
18	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Рисунок 2.33 – Результат квантованих DCT коефіцієнтів, помножених на матрицю квантування

Під час декодування зображення розраховується Дискретне косинусне перетворення (DCT) за допомогою DCT типу II.

Тепер обчислюється обернене до попереднього розрахунку, інакше відоме як DCT типу III. Це відбувається допомогою наступної формули:

$$f_{x,y} = \frac{1}{4} \sum_{u=0}^7 \sum_{v=0}^7 \alpha(u) \alpha(v) F_{u,v} \cos \left[\frac{(2x+1)u\pi}{16} \right] \cos \left[\frac{(2y+1)v\pi}{16} \right] \quad (2.9)$$

Якщо застосувати цю формулу до матриці, отримаємо наступне:

-66	-63	-71	-68	-56	-65	-68	-46
-71	-73	-72	-46	-20	-41	-66	-57
-70	-78	-68	-17	20	-14	-61	-63
-63	-73	-62	-8	27	-14	-60	-58
-58	-65	-61	-27	-6	-40	-68	-50
-57	-57	-64	-58	-48	-66	-72	-47
-53	-46	-61	-74	-65	-63	-62	-45
-47	-34	-53	-74	-60	-47	-47	-41

Рисунок 2.34 – Результати розрахунку двовимірного оберненого DCT

Під час кодування відцентровувалися коефіцієнти в матриці навколо нуля, віднімаючи 128 від кожного значення.

Цей останній крок відмічається додаванням 128 до кожного запису.

62	65	57	60	72	63	60	82
57	55	56	82	108	87	62	71
58	50	60	111	148	114	67	65
65	55	66	120	155	114	68	70
70	63	67	101	122	88	60	78
71	71	64	70	80	62	56	81
75	82	67	54	63	65	66	83
81	94	75	54	68	81	81	87

Рисунок 2.35– Результати розрахунку двовимірного оберненого DCT

Якщо процес розпакування нижче нуля або більше 255, числа стискаються до цього діапазону.

2.4.1 Порівняння оригіналу з розпакованим зображенням

Навіть при налаштуваннях якості 50%, різниця між оригінальним та стислим зображенням все ж мінімальна.

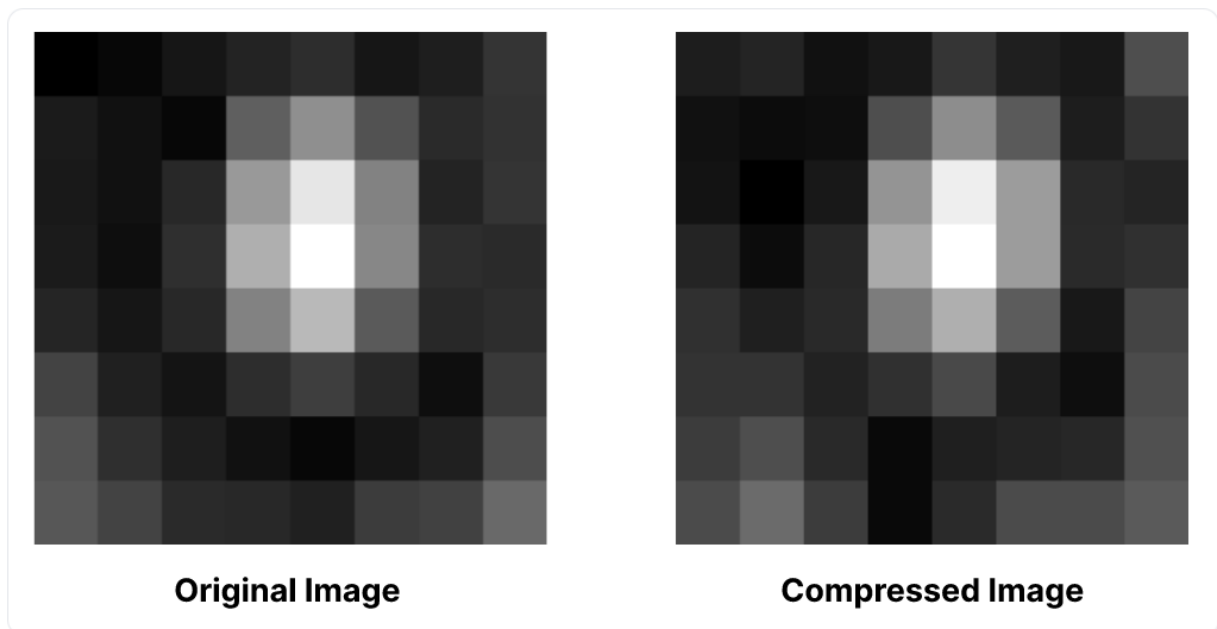
52	55	61	66	70	61	64	73	62	65	57	60	72	63	60	82
63	59	55	90	109	85	69	72	57	55	56	82	108	87	62	71
62	59	68	113	144	104	66	73	58	50	60	111	148	114	67	65
63	58	71	122	154	106	70	69	65	55	66	120	155	114	68	70
67	61	68	104	126	88	68	70	70	63	67	101	122	88	60	78
79	65	60	70	77	68	58	75	71	71	64	70	80	62	56	81
85	71	64	59	55	61	65	83	75	82	67	54	63	65	66	83
87	79	69	87	65	76	78	94	81	94	75	54	68	81	81	87

Original Image

Compressed Image

Рисунок 2.36 – Порівняння зображення до та після стиснення JPEG у частотному діапазоні

Так виглядають пікселі цього зображення в рамках одного блоку:



Original Image

Compressed Image

Рисунок 2.37 – Візуальне порівняння зображення до та після стиснення JPEG

Беручи до уваги, що це зображення 8x8 пікселів, різниця між двома зображеннями є мінімальною.

Зображення у форматі JPEG працюють добре з усіма сучасними браузерами, включаючи Chrome, Firefox.

3 ЕКСПЕРИМЕНТИ ТА РЕЗУЛЬТАТИ

3.1 Масштабування базової таблиці квантування

В якості вибірки використовувались 33 зображення у форматі.pgm з ресурсу <https://www.cl72.org/090imagePLib.html> та <https://www.hlevkin.com/hlevkin/06testimages.htm> здебільшого розміру 512x512.

Засобами libjpeg зображення були стиснені з показником якості в діапазоні від 5 до 100 з кроком 5. З отриманих зображень були витягнуті таблиці квантування, що використовуються libjpeg за замовченням. Нижче наведені приклади цих таблиць (рис. 3.1-3.3).

160	110	100	160	240	255	255	255
120	120	140	190	255	255	255	255
140	130	160	240	255	255	255	255
140	170	220	255	255	255	255	255
180	220	255	255	255	255	255	255
240	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255

Рисунок 3.1 – Значення таблиці квантування при якості $QF = 5$

16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	55
14	13	16	24	40	57	69	56
14	17	22	29	51	87	80	62
18	22	37	56	68	109	103	77
24	35	55	64	81	104	113	92
49	64	78	87	103	121	120	101
72	92	95	98	112	100	103	99

Рисунок 3.2 – Значення таблиці квантування при якості $QF = 50$

8	6	5	8	12	20	26	31
6	6	7	10	13	29	30	28
7	7	8	12	20	29	35	28
7	9	11	15	26	44	40	31
9	11	19	28	34	55	52	39
12	18	28	32	41	52	57	46
25	32	39	44	52	61	60	51
36	46	48	49	56	50	52	50

Рисунок 3.3 – Значення таблиці квантування при якості $QF = 75$

Ці дії дали нам можливість побачити як працює масштабування таблиці квантування згідно параметру якості QF , який приймає значення від 0 до 100.

В літературі описано декілька способів масштабування таблиць квантування. Перший спосіб масштабування має наступний вигляд [12]:

$$S = \begin{cases} \frac{50}{QF} & \text{якщо } QF \leq 50, \\ 2 - \frac{QF}{50} & \text{якщо } 50 < QF < 100, \end{cases} \quad (3.1)$$

де QF - бажаний показник якості, S – масштабний множник. Другий спосіб був описаний в статті [13]:

$$S = \begin{cases} \frac{5000}{QF} & \text{якщо } QF \leq 50, \\ 200 - 2QF & \text{якщо } 50 < QF < 100. \end{cases} \quad (3.2)$$

Аби отримати значення таблиці квантування необхідно скористатися формулою $T_s[i] = (S * T_b[i] + 50/100)$, де $T_b[i]$ – відповідне значення базової таблиці. Варіант, описаний у статті [13], дає таблиці, що співпадають з таблицям libjreg для показників якості $QF > 50$. Для показників якості $QF < 50$ трапляються таблиці квантування із значеннями, які відрізняються на 1 від відповідних таблиць libjreg. Отже генератор, запозичений з роботи [12], не такий точний. Тому в подальшому будемо використовувати формулу (3.2).

3.2 Основні розрахункові співвідношення

Позначимо через $D(I, T_s)$ спотворення зображення I викликані його квантуванням за таблицею T_s , внаслідок чого отримується зображення I' . Будемо визначати $D(I, T_s)$ як середньоквадратичну помилку між відповідними значеннями пікселів зображень I та I' . Отже

$$D(I, T_s) = \sum_{n=0}^{63} D_n(I, T_s[n]), \quad (3.3)$$

$$D_n(I, T_s[n]) = \frac{1}{64} \text{Mean} \{ (\hat{f}[n] - \hat{f}'[n])^2 \}, \quad (3.4)$$

де $\hat{f}'[n] = \text{round}(\hat{f}[n]/T_s[n]) \cdot T_s[n]$, $\hat{f}[n]$ – значення відповідного ДКП коефіцієнта, а операція усереднення $\text{Mean}\{\}$ виконується по відношенню до всіх 8x8 блоків зображення.

Оскільки ми в основному стурбовані впливом різних таблиць квантування на задане зображення, в подальшому замість $D(I, T_s)$ будемо писати $D(T_s)$. Аналогічно позначимо спотворення $D_n(I, T_s[n])$ в n -му коефіцієнті через $D_n(T_s[n])$, тоді

$$D(T_s) = \sum_{n=0}^{63} D_n(T_s[n]). \quad (3.5)$$

Досягнутий ступінь стиснення зазвичай виражається через швидкість (*rate*) стисненого зображення, яка є кількістю бітів, що припадає на піксель:

$$\text{rate} = \frac{\text{size of compressed image in bits}}{\text{number of pixels in the image}}. \quad (3.6)$$

Слідуючі [11] замість швидкості будемо використовувати ентропію $R(T_s)$ квантованих ДКП коефіцієнтів

$$R(T_s) \approx \sum_{n=0}^{63} R_n(T_s[n]), \quad (3.7)$$

$$R_n(T_s[n]) = \frac{1}{64} \text{Entropy}\{\text{round}(\hat{f}[n]/q)\}, \quad (3.8)$$

В останньому співвідношенні ентропія обчислюється по всіх 8x8 блоках ДКП коефіцієнтів зображення.

Для оцінки якості стиснених зображень використовувалось пікове співвідношення сигнал-шум, що обчислюється за формулою:

$$PSNR = 10 \log_{10} \left(\frac{256^2}{D(T_s)} \right) \quad (3.9)$$

3.3 Порівняння базових таблиць квантування

На даний час компресор `sjreg`, який є частиною бібліотеки `libjreg`, передбачає можливість вибору 9 базових таблиць квантування. Їх можна обрати за допомогою параметру `n`, що приймає значення `n=0-8`. Ступінь стиснення зображення регулюється шляхом вибору бажаної якості зображення, що визначається через показник якості $QF=0..100$. Якщо значення QF відоме, тоді базова таблиця масштабується за формулою (3.2).

Порівняємо між собою базові таблиці квантування. В таблиці 3.1 наведені значення базових таблиць квантування, що використовувались в дослідженні.

Таблиця 3.1 – Перелік базових таблиць, що були використані в дослідженні

Номер та назва	Значення
JPEG Annex K (n=0)	16, 11, 10, 16, 24, 40, 51, 61, 12, 12, 14, 19, 26, 58, 60, 55, 14, 13, 16, 24, 40, 57, 69, 56, 14, 17, 22, 29, 51, 87, 80, 62, 18, 22, 37, 56, 68, 109, 103, 77, 24, 35, 55, 64, 81, 104, 113, 92, 49, 64, 78, 87, 103, 121, 120, 101, 72, 92, 95, 98, 112, 100, 103, 99
flat (n=1)	16, 16
n=2	12, 17, 20, 21, 30, 34, 56, 63, 18, 20, 20, 26, 28, 51, 61, 55, 19, 20, 21, 26, 33, 58, 69, 55, 26, 26, 26, 30, 46, 87, 86, 66, 31, 33, 36, 40, 46, 96, 100, 73, 40, 35, 46, 62, 81, 100, 111, 91, 46, 66, 76, 86, 102, 121, 120, 101, 68, 90, 90, 96, 113, 102, 105, 103

Продовження табл. 4.1

Номер та назва	Значення
n=3, ImageMagic	16, 16, 16, 18, 25, 37, 56, 85, 16, 17, 20, 27, 34, 40, 53, 75, 16, 20, 24, 31, 43, 62, 91, 135, 18, 27, 31, 40, 53, 74, 106, 156, 25, 34, 43, 53, 69, 94, 131, 189, 37, 40, 62, 74, 94, 124, 169, 238, 56, 53, 91, 106, 131, 169, 226, 311, 85, 75, 135, 156, 189, 238, 311, 418
n=4, ImageMagic	9, 10, 12, 14, 27, 32, 51, 62, 11, 12, 14, 19, 27, 44, 59, 73, 12, 14, 18, 25, 42, 59, 79, 78, 17, 18, 25, 42, 61, 92, 87, 92, 23, 28, 42, 75, 79, 112, 112, 99, 40, 42, 59, 84, 88, 124, 132, 111, 42, 64, 78, 95, 105, 126, 125, 99, 70, 75, 100, 102, 116, 100, 107, 98
Relevance of human vision to JPEG-DCT compression (1992) Klein, Silverstein and Carney. n=5	10, 12, 14, 19, 26, 38, 57, 86, 12, 18, 21, 28, 35, 41, 54, 76, 14, 21, 25, 32, 44, 63, 92, 136, 19, 28, 32, 41, 54, 75, 107, 157, 26, 35, 44, 54, 70, 95, 132, 190, 38, 41, 63, 75, 95, 125, 170, 239, 57, 54, 92, 107, 132, 170, 227, 312, 86, 76, 136, 157, 190, 239, 312, 419

Продовження табл. 4.1

Номер та назва	Значення
DCTune perceptual optimization of compressed dental X-Rays (1997) Watson, Taylor, Borthwick n=6	7, 8, 10, 14, 23, 44, 95, 241, 8, 8, 11, 15, 25, 47, 102, 255, 10, 11, 13, 19, 31, 58, 127, 255, 14, 15, 19, 27, 44, 83, 181, 255, 23, 25, 31, 44, 72, 136, 255, 255, 44, 47, 58, 83, 136, 255, 255, 255, 95, 102, 127, 181, 255, 255, 255, 255, 241, 255, 255, 255, 255, 255, 255, 255
A visual detection model for DCT coefficient quantization (12/9/93) Ahumada, Watson, Peterson n=7	15, 11, 11, 12, 15, 19, 25, 32, 11, 13, 10, 10, 12, 15, 19, 24, 11, 10, 14, 14, 16, 18, 22, 27, 12, 10, 14, 18, 21, 24, 28, 33, 15, 12, 16, 21, 26, 31, 36, 42, 19, 15, 18, 24, 31, 38, 45, 53, 25, 19, 22, 28, 36, 45, 55, 65, 32, 24, 27, 33, 42, 53, 65, 77
An improved detection model for DCT coefficient quantization (1993) Peterson, Ahumada and Watson n=8	14, 10, 11, 14, 19, 25, 34, 45, 10, 11, 11, 12, 15, 20, 26, 33, 11, 11, 15, 18, 21, 25, 31, 38, 14, 12, 18, 24, 28, 33, 39, 47, 19, 15, 21, 28, 36, 43, 51, 59, 25, 20, 25, 33, 43, 54, 64, 74, 34, 26, 31, 39, 51, 64, 77, 91, 45, 33, 38, 47, 59, 74, 91, 108

Використовуючи формули (3.8) та (3.9) можна побудувати графік залежності пікового відношення сигнал-шум PSNR від кількості бітів на піксель

bpp, для тестових зображень квантованих з використанням однієї і тієї самої базової таблиці квантування.

Нижче (рис. 3.4-3.12) наведено графіки розкиду для різних базових таблиць квантування. Окремі точки на графіках позначають кожне з тестових зображень, а пряма відповідає лінійній регресії.

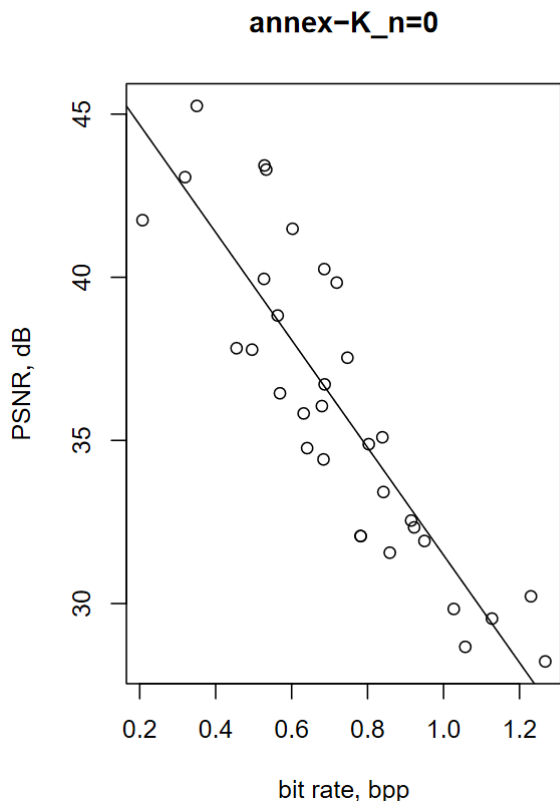


Рисунок 3.4 – Відношення PSNR до
bpp для таблиці n=0

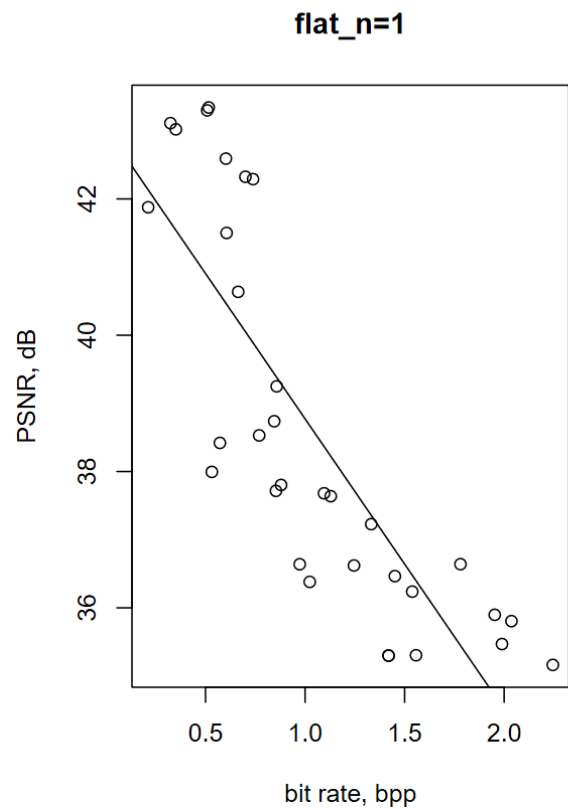


Рисунок 3.5 – Відношення PSNR до
bpp для таблиці n=1

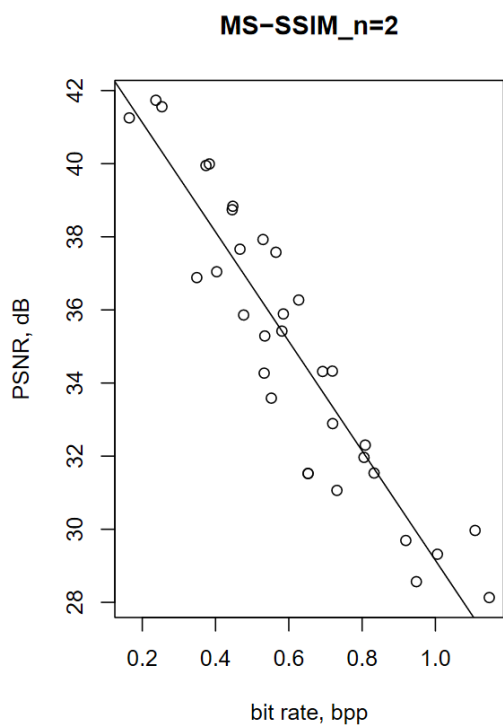


Рисунок 3.6 – Відношення PSNR до
bpp для таблиці n=2.

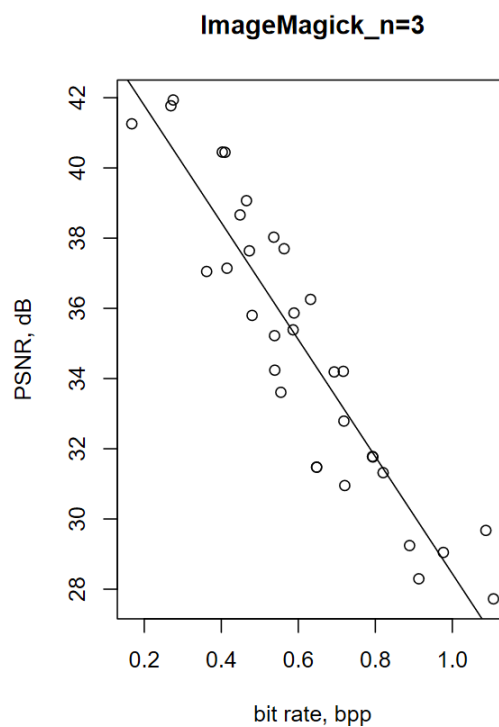


Рисунок 3.7 – Відношення PSNR до
bpp для таблиці n=3.

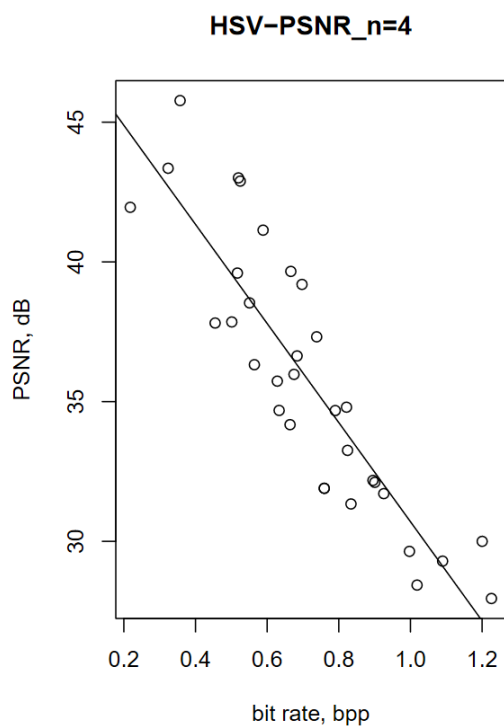


Рисунок 3.8 – Відношення PSNR до
bpp для таблиці n=4.

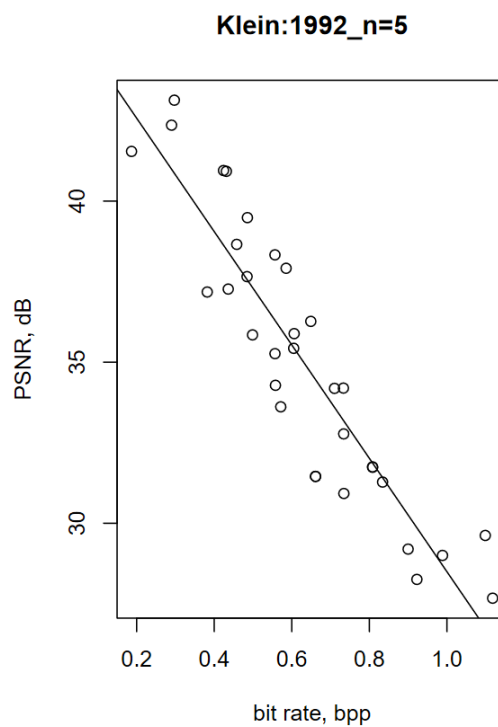


Рисунок 3.9 – Відношення PSNR до
bpp для таблиці n=5.

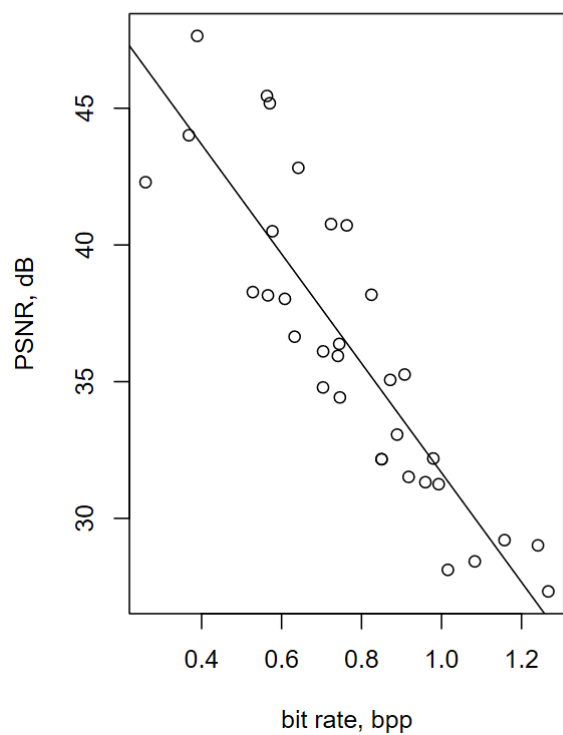
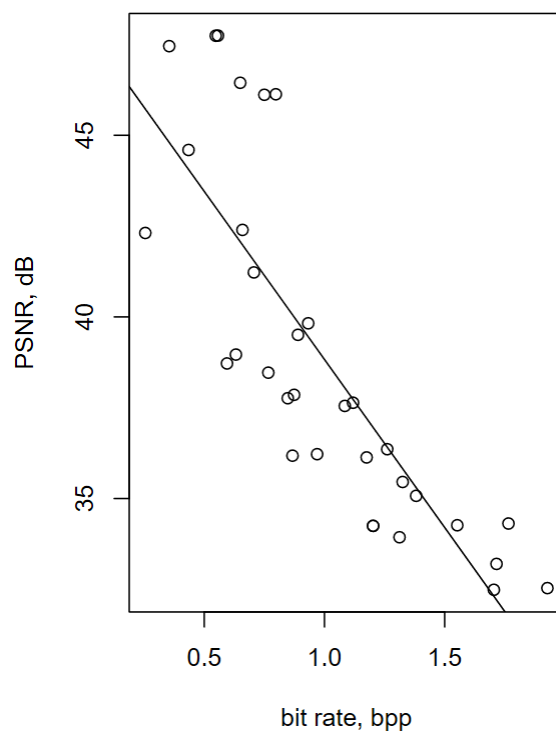
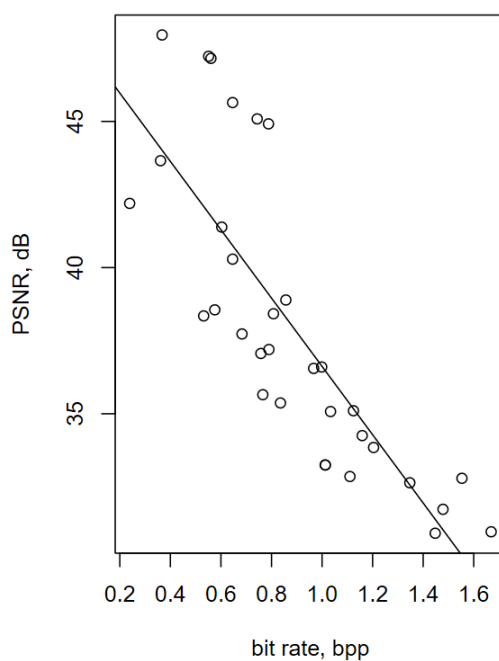
Watson:1997-DCTune-X-Ray_n=6Рисунок 3.10 – Відношення PSNR до
bpp для таблиці n=6.**Ahumada:1993_n=7**Рисунок 3.11 – Відношення PSNR до
bpp для таблиці n=7.**Peterson:1993_n=8**

Рисунок 3.12 – Відношення PSNR до bpp для таблиці n=8.

Рисунок 3.13 містить прямі регресії всіх порівнюваних базових таблиць з рис. 3.4-3.12. Базові таблиці забезпечують якість зображення, що відповідає показнику якості $QF \approx 50$, тобто $PSNR \approx 30$ дБ. Оскільки тестові зображення відрізняються за своєю природою: наприклад, медичні рентгенівські зображення не мають різких переходів яскравості, а також не містять багато деталей, в порівнянні з зображеннями, наближеними до текстурних зображень як, скажімо, baboon. Отже, медичні зображення матимуть низький бітрейт і відносно високий PSNR. І навпаки, подібні до текстурних зображення, наприклад baboon, типово матимуть відносно високу швидкість передачі даних і низький PSNR (через втрату деталей внаслідок квантування), що пояснює від'ємний нахил прямої регресії. Зазначимо, що базова таблиця квантування $n=6$ була оптимізована для рентгенівських зображень, а відтак добре працює для низьких бітрейтів (швидкостей передачі).

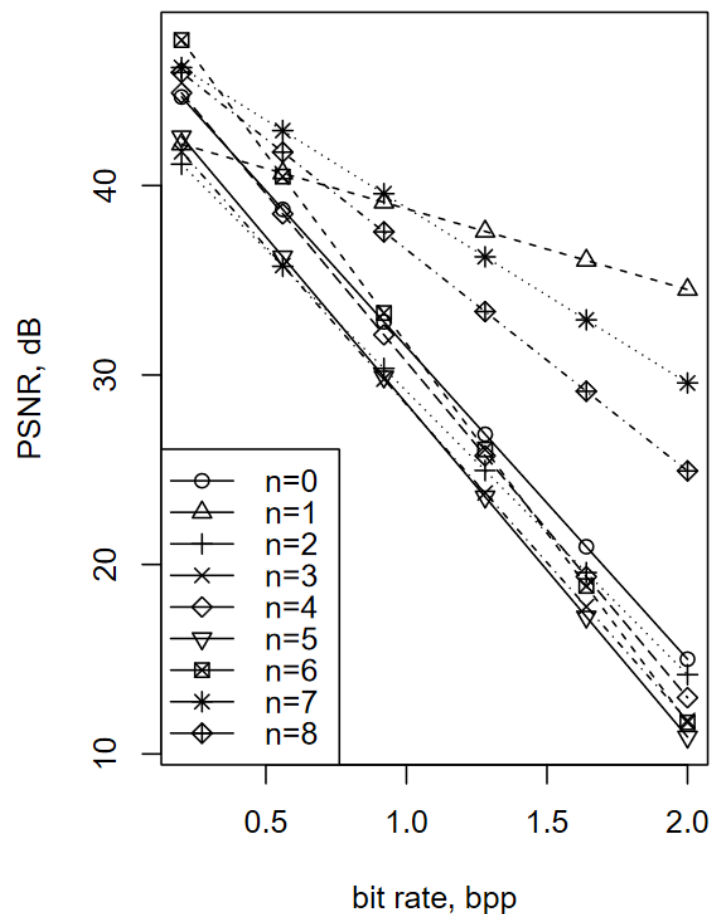


Рисунок 3.13 – Відношення PSNR до bpr для таблиць $n=0-8$.

Спостерігаючи на рис. 3.13 апроксимовані прямі, які описують залежність PSNR від bpp для базових таблиць квантування, слід відзначити 3 базові таблиці, а саме $n=7$, $n=8$ і $n=0$. Також окремо можна зауважити таблицю $n=6$, що є оптимальною для компресії рентгенівських медичних зображень (зображень з низьким бітрейтом). Порівняємо зазначені таблиці за критерієм мінімуму результуючого PSNR з таблицями, що породжуються алгоритмом RD-OPT [11]. Для цього будемо масштабувати відібрані базові таблиці за формулою (3.2) при $QF=20(5)100$ і порівнювати з таблицями квантування, породжуваними алгоритмом RD-OPT (при обмеженні на PSNR або bpp).

Практично, великі значення PSNR означають кращу деталізацію зображення, чому відповідають більші bpp. Це, наприклад, забезпечується таблицями квантування $n=7$ та $n=8$ запропонованими групою вчених Ahumada, Watson та Peterson [14, 15]. Базова таблиця, що використовується libjpeg за замовченням, $n=0$, є найкращою з гірших, див. рис. (3.13).

3.4 Порівняння базових таблиць квантування з таблицями RD-OPT

Особливістю роботи алгоритма RD-OPT є те, що він породжує окремі таблиці квантування для кожного зображення і бажаного значення PSNR/bpp. Порівняємо між собою відібрані базові таблиці квантування ($n=0,7,8$) з таблицями, що породжуються алгоритмом RD-OPT за критерієм максимуму результуючого PSNR при заданому bpp. В якості реалізації алгоритму RD-OPT використовувалась утіліта, створена його авторами [16]. У випадку з базовими таблицями квантування було використано вираз (3.2) для генерації таблиць, що відповідають певному показнику якості QF , а вже після цього кроку побудовано графік залежності PSNR(bpp).

Всі 33 тестові зображення можна умовно поділити на дві групи: зображення, що містять в своєму складі істотні області монотонності (без різких

змін яскравості) та зображення з областями подібними до текстурних. До першої групи відносяться, наприклад, медичні зображення рентгенівських знімків (див. рис. 3.14). До другої групи відносяться звичайні зображення такі як clown (див. рис. 3.15).

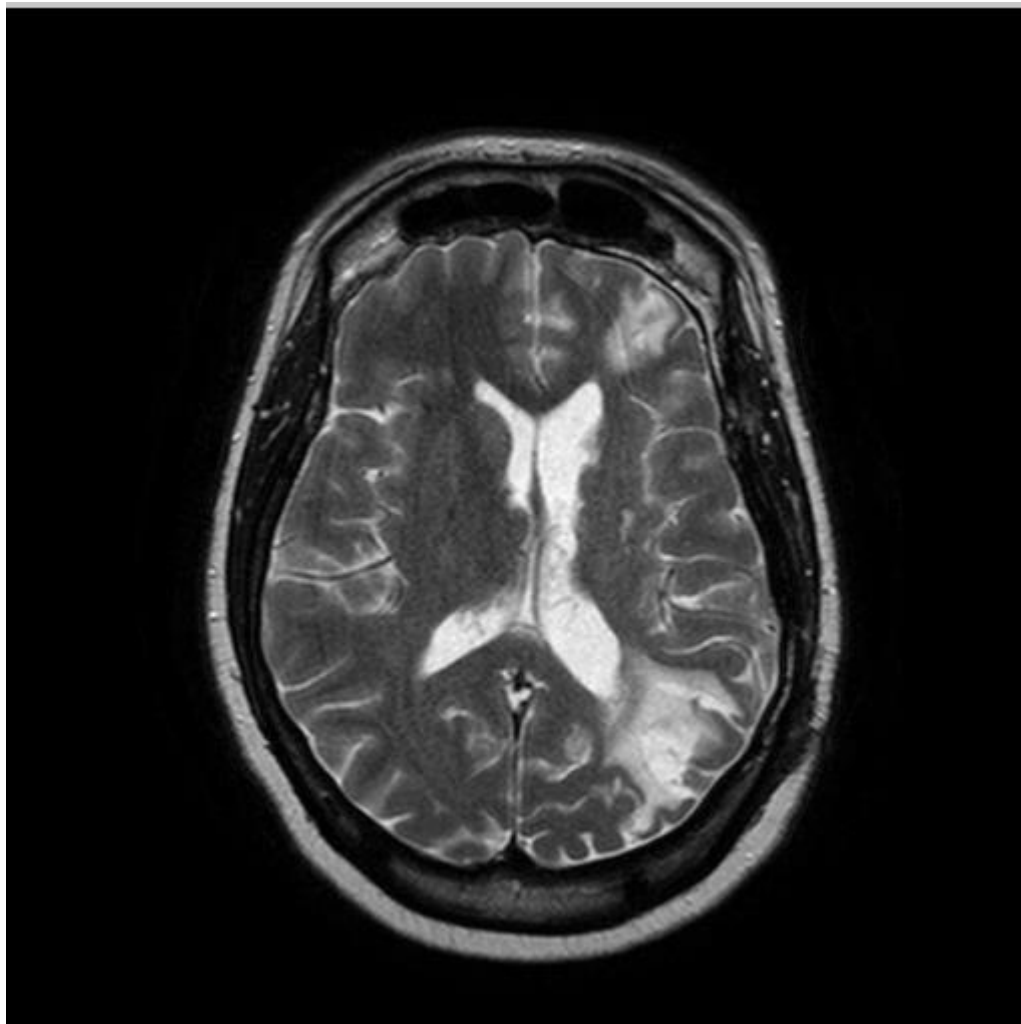


Рисунок 3.14 – Тестове зображення brainCT.pgm



Рисунок 3.15 – Тестове зображення clown.pgm

На рисунках 3.16-3.17 показані залежності PSNR(bpp), що забезпечуються різними таблицями квантування для двох зображень brainCT та clown відповідно. Порівняльний аналіз отриманих залежностей свідчить про те, що для монотонних зображень таблиці RD-OPT не дають суттєвого виграшу, тобто базові таблиці квантування породжують зображення з PSNR близькими до оптимальних. Варто також зазначити, що у випадку зображення brainCT, спеціалізована таблиця з номером $n=6$ програє за PSNR таблиці $n=0$, що використовується в libjpeg за замовченням. Крім того синтезовані в роботах [14, 15] таблиці квантування дають значення PSNR близькі до оптимальних, тобто тих, що отримуються за допомогою RD-OPT. У випадку подібних до текстурних зображень (рис. 3.17), таблиці породжувані алгоритмом RD-OPT, дають суттєвий

виграш за PSNR тільки за умови $\text{bpp} > 1$, що практично знецінює отриманий результат, оскільки компресія за таких значень bpp майже не відбувається. Таблиця $n=6$ прогнозовано забезпечує найгірший результат, оскільки вона оптимізувалась під монотонні зображення. В свою чергу таблиці з номерами $n=0,7,8$ поведуть себе майже однаково.

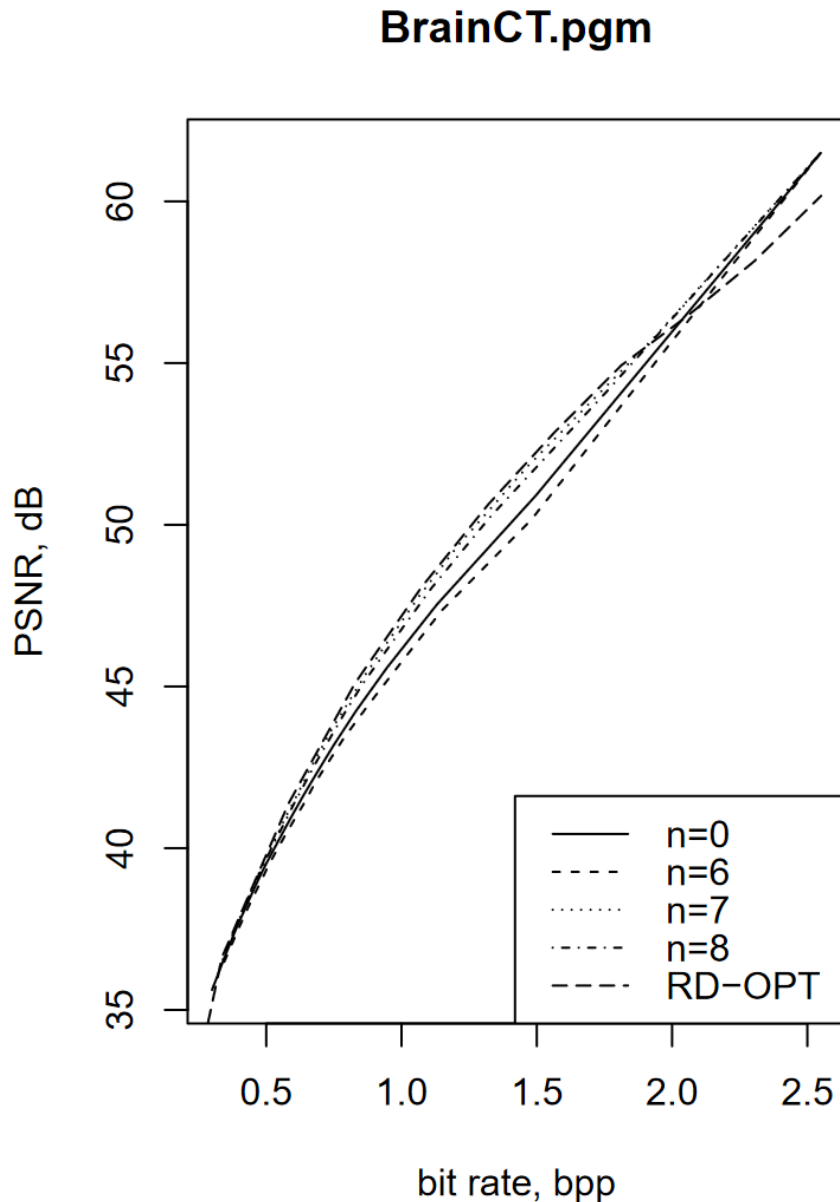


Рисунок 3.16 – Порівняння таблиць квантування з алгоритмом на прикладі зображення brainCT.pgm

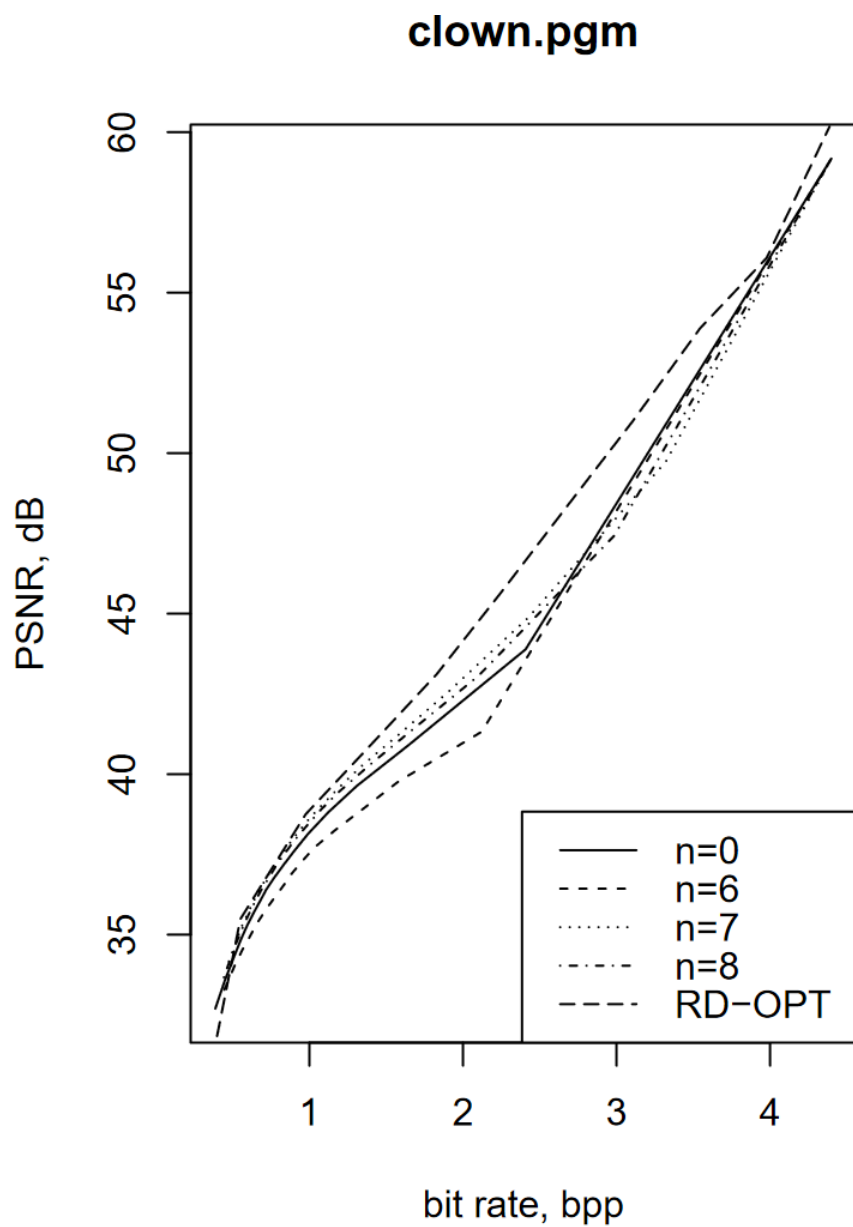


Рисунок 3.17 – Порівняння таблиць квантування з алгоритмом на прикладі зображення clown.pgm

ВИСНОВКИ

В роботі було проведено дослідження математичної моделі зображень на основі дискретно-косинусного перетворення. Детально проаналізовано популярний алгоритм JPEG, який використовується для стиснення з втратами цифрових зображень. На мові програмування R створено програмну модель частини конвеєру (pipe line) алгоритма JPEG, а саме відтворений процес розбиття зображення на блоки 8x8 пікселів; обчислення прямого (зворотного) двовірного дискретного косинусного (ДКП) перетворення та подальшого квантування отриманих ДКП коефіцієнтів. Проведені експериментальні дослідження по стисненню зображень з використанням розробленої програмної моделі при різних базових таблицях квантування та подальший аналіз якості стиснених зображень дозволив обрати найкращі за критерієм максимуму PSNR базові таблиці квантування. Ними виявились таблиці запропоновані в роботах [14, 15], а також таблиця, що використовується бібліотекою libjpeg за замовченням. Крім того виконано порівняння відібраних базових таблиць квантування з таблицями, породжуваними алгоритмом RD-OPT, який було запропоновано В. Ратнакармом та М. Лівні [11].

За результатами порівняльного аналізу можна зробити висновок про незначну перевагу таблиць квантування, породжуваних алгоритмом RD-OPT, над базовими таблицями квантування, відібраними на першому етапі експерименту. Це дає підґрунтя для інтеграції алгоритму RD-OPT у сучасні кодери JPEG для покращення якості їх роботи.

Проведені дослідження дали нам краще уявлення про вплив таблиць квантування на якість зображення, а отже і про роботу алгоритму стиснення JPEG загалом. Подальші роздуми можуть продовжуватись в напрямку узгодження альтернативних таблиць квантування з таблицями коду Хаффмана, який використовується в алгоритмі JPEG для економного кодування квантованих ДКП коефіцієнтів.

ПЕРЕЛІК ПОСІЛАНЬ

1. Файли JPEG. [Електронний ресурс] Режим доступу: <https://www.adobe.com/ua/creativecloud/file-types/image/raster/jpeg-file.html>
2. JPEG - Wikipedia [Електронний ресурс] Режим доступу: <https://en.wikipedia.org/wiki/JPEG>
3. ISO/IEC 10918-1 : 1993(E) p.28 [Електронний ресурс] Режим доступу: <https://www.webcitation.org/6183BhkYF?url=http://www.digicamsoft.com/itu/itu-t81-28.html>
4. The Ultimate Guide to JPEG including JPEG Compression | Encoding (2023) [Електронний ресурс] Режим доступу: <https://www.thewebmaster.com/jpeg-definitive-guide/>
5. Rafael C. Gonzalez, Richard E. Woods, "Digital Image Processing", pp. 250-315, 2018.
6. Robinson, A. H.; Cherry, C. (1967). "Results of a prototype television bandwidth compression scheme". *Proceedings of the IEEE*. IEEE. 55 (3): 356–364. doi:10.1109/PROC.1967.5493.
7. "Run Length Encoding Patents". Internet FAQ Consortium. 21 March 1996. Retrieved 14 July 2019. [Електронний ресурс] Режим доступу: http://www.ross.net/compression/patents_notes_from_ccfaq.html
8. "Method and system for data compression and restoration". Google Patents. 7 August 1984. Retrieved 14 July 2019. [Електронний ресурс] Режим доступу: <https://patents.google.com/patent/US4586027A>
9. "Data recording method". Google Patents. 8 August 1983 отримано 20 жовтня 2023р [Електронний ресурс] Режим доступу: <https://patents.google.com/patent/JPH0828053B2/en>
10. Dunn, Christopher (1987). "Smile! You're on RLE!" (PDF). The Transactor. Transactor Publishing. 7 (6): 16–18. Retrieved 2015-12-06. [Електронний ресурс] Режим доступу: http://csbruce.com/cbm/transactor/pdfs/trans_v7_i06.pdf
11. Ratnakar, V. and Livny, M., "An efficient algorithm for optimizing DCT quantization," in *IEEE Transactions on Image Processing*, vol. 9, no. 2, pp. 267-270, Feb. 2000.

12. Wong, P. H. W. and Au, O. C. "A capacity estimation technique for JPEG-to-JPEG image watermarking," in IEEE Transactions on Circuits and Systems for Video Technology, vol. 13, no. 8, pp. 746-752, Aug. 2003.
13. Kornblum, Jesse D. Using JPEG quantization tables to identify imagery processed by software. Digit. Investig. 5 (September, 2008), S21–S25.
14. Peterson, Heidi A., Ahumada, Albert J. Jr., and Watson, Andrew B. "Improved detection model for DCT coefficient quantization", Proc. SPIE 1913, Human Vision, Visual Processing, and Digital Display IV, (8 September 1993).
15. Ahumada, Albert J. Jr. and Peterson, Heidi A. "A visual detection model for DCT coefficient quantization," AIAA 1993-4515. 9th Computing in Aerospace Conference. October 1993.
16. " RD-OPT Home Page " отримано 20 листопада 2023р. [Електронний ресурс] Режим доступу: <https://pages.cs.wisc.edu/~ratnakar/rdopt.html>