

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Міністерство освіти і науки України

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Міністерство освіти і науки України

Кваліфікаційна наукова
праця на правах рукопису
УДК 621.382

Мазін Максим Юрійович

ДИСЕРТАЦІЯ
МЕТОДИ СТИСНЕННЯ ЗОБРАЖЕНЬ В СИСТЕМАХ ІНТЕРНЕТУ
РЕЧЕЙ НА МІКРОКОНТРОЛЕРАХ

17 – Електроніка та телекомунікації

171 – Електроніка

Подається на здобуття наукового ступеня доктора філософії.

Дисертація містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело.

_____/ Мазін Максим Юрійович

Науковий керівник: Оникієнко Юрій Олексійович
кандидат технічних наук, доцент
Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»

Київ - 2025

АНОТАЦІЯ

Мазін М.Ю. Методи стиснення зображень в системах інтернету речей на мікроконтролерах. - Кваліфікаційна наукова робота на правах рукопису. Дисертація на здобуття наукового ступеня кандидата технічних наук за спеціальністю 171 – Акустичні та мультимедійні електронні системи. - Національний технічний університет України "Київський політехнічний інститут імені Ігоря Сікорського", МОН України, Київ, 2025

Дисертація присвячена вирішенню актуального завдання – дослідженню ефективності використання вейвлет-перетворень для стиснення зображень у вбудованих системах з обмеженими ресурсами та визначенню ефективних параметрів компресії. Для цього були проаналізовані існуючі методи стиснення, проведено їх порівняння, виявлено основні недоліки та запропоновано шляхи їх усунення.

У роботі вперше поставлена та розв'язана задача знаходження впливу рівня декомпозиції вейвлет-перетворення на якість відновленого зображення у системах на мікроконтролерах, досліджено особливості роботи різних типів вейвлетів (Хаара, Добеші, Коіфлет) у поєднанні з JPEG-стисненням. Вперше проведено порівняльний аналіз використання алгоритмів ентропійного стиснення, зокрема методів Хаффмана та RLE, у контексті їх застосування для обмежених обчислювальних ресурсів. Запропоновано та реалізовано новий метод адаптивної корекції артефактів, що виникають при багаторазовому застосуванні вейвлет-перетворень у поєднанні з JPEG-стисненням, який дозволяє покращити якість відновленого зображення без значного збільшення обчислювальних витрат.

Для вирішення поставлених задач використано експериментальні методи досліджень, чисельне моделювання процесів компресії та обробки зображень, а також тестування розроблених алгоритмів на апаратних платформах з використанням мікроконтролерів. Проведено натурні та лабораторні дослідження, які підтвердили високу точність отриманих математичних моделей. Визначено кращі параметри стиснення для ефективного використання

у вбудованих системах, а також досліджено вплив об'єму оперативної пам'яті на продуктивність компресії у реальних апаратних умовах.

Практичне значення отриманих результатів полягає у можливості застосування розроблених алгоритмів для оптимізації стиснення зображень у системах відеоспостереження, розпізнавання образів, бездротової передачі даних. Виконані дослідження дозволяють розробникам вбудованих систем обирати ефективні методи компресії для конкретних задач, мінімізуючи використання пам'яті та процесорних ресурсів при збереженні високої якості оброблених зображень.

Для дослідження ефективності стиснення зображень із використанням вейвлет-перетворень та ентропійних методів компресії були застосовані чисельні методи моделювання та експериментальні дослідження. Використано алгоритмічне моделювання процесів стиснення, а також тестування запропонованих методів на реальному апаратному забезпеченні, що дозволило отримати кількісні залежності між параметрами перетворень і якістю відновленого зображення.

Запропоновано новий підхід до оптимізації компресії, що базується на комбінованому використанні вейвлет-перетворення та ентропійного кодування, зокрема методів Хаффмана та RLE. У роботі проведено порівняння різних рівнів декомпозиції вейвлетів Хаара, Добеші та Коіфлета, що дозволило оцінити їхній вплив на якість зображення та рівень стиснення. Отримані результати показали, що ефективність методів значною мірою залежить від вибору рівня декомпозиції: із збільшенням рівня деталізація зображення втрачається, але підвищується коефіцієнт стиснення. Це дозволяє визначити кращі параметри компресії для конкретних застосувань у вбудованих системах.

Виявлено, що поєднання вейвлет-перетворень із ентропійним кодуванням дозволяє значно зменшити розмір зображення при мінімальних втратах якості. Зокрема, використання вейвлет-перетворень перед застосуванням алгоритму Хаффмана дозволяє досягти компресії до 30% ефективніше порівняно з традиційними методами. Водночас метод RLE продемонстрував високу

ефективність для зображень з великою кількістю повторюваних пікселів, однак поступався методу Хаффмана у випадку складних структурованих зображень.

Проведено чисельне моделювання впливу об'єму оперативної пам'яті мікроконтролера на швидкість виконання алгоритмів. Дослідження показали, що використання зовнішньої пам'яті дозволяє обробляти більш складні зображення, але значно знижує продуктивність стиснення. Це свідчить про необхідність балансування між якістю компресії та швидкістю алгоритму залежно від можливостей апаратної платформи.

Розроблено та проведено тестування адаптивного алгоритму корекції артефактів, які виникають під час багаторазового застосування вейвлет-перетворень у поєднанні з JPEG-стисненням. Виявлено, що артефакти мають періодичний характер і залежать від рівня декомпозиції. Запропонований метод корекції дозволяє усунути ці дефекти за рахунок використання медіанного фільтрування, що дає змогу підвищити якість відновленого зображення без значного збільшення обчислювальних витрат.

Удосконалено програмне забезпечення для оцінки ефективності стиснення у вбудованих системах, яке враховує параметри компресії, можливості апаратної платформи та потреби конкретного застосування. Отримані результати можуть бути використані для вибору ефективніших алгоритмів обробки зображень у системах відеоспостереження, розпізнавання образів, бездротової передачі даних та інших задач, де необхідно зменшити розмір переданих файлів при збереженні їх високої якості.

Також у роботі була поставлена та розв'язана задача впливу змінних частотних характеристик на ефективність вейвлет-компресії зображень у системах з обмеженими ресурсами. Актуальність цієї задачі зумовлена широким використанням бездротових технологій передавання даних, де стиснення зображень є критично важливим для зменшення затримок і енергоспоживання. Одним із основних факторів, що впливає на ефективність алгоритмів стиснення, є частотний розподіл сигналу, оскільки зміни спектрального складу можуть суттєво позначатися на коефіцієнтах вейвлет-декомпозиції.

Запропонована математична модель дозволяє визначити часову залежність втрат інформації при передаванні зображень, що зазнають частотних змін у процесі компресії та декомпресії. Проаналізовано ефект частотної дисперсії, який виникає при накладанні вейвлет-перетворення на сигнал із динамічною змінною спектральною характеристикою. Проведене дослідження показує, що зі збільшенням кількості рівнів декомпозиції зображення втрачає дрібні деталі, що є критичним для застосувань у розпізнаванні образів та відеоспостереженні.

Практичні експерименти з передачею зображень через бездротові канали зв'язку показали, що навіть при високих рівнях спектральних змін ефективність компресії залишається стабільною, що підтверджує надійність використання вейвлет-перетворень у складних умовах. Дослідження доводить, що запропонований метод може бути успішно використаний у реальних системах бездротової передачі даних, де необхідно досягати балансу між якістю відновленого зображення та швидкістю його передавання.

Під час роботи над теоретичною та експериментальною частинами були проведені натурні та лабораторні дослідження ефективності застосування вейвлет-компресії для бездротової передачі зображень у мережах LPWAN на основі LoRa. Для оцінки якості відновлених зображень використовувались об'єктивні метрики, такі як PSNR (Peak Signal-to-Noise Ratio) та SSIM (Structural Similarity Index), а також суб'єктивний аналіз візуальної сприйнятливості артефактів, що виникають при багаторазовій компресії.

У ході експериментів проводилося порівняння зображень, переданих без компресії та з використанням різних методів вейвлет-стиснення. Виявлено, що застосування Daubechies-4, Coiflet та Haar-вейвлетів дає різну ступінь втрат інформації. Для ефективного поєднання високої якості відновленого зображення та мінімального використання пропускної здатності мережі найбільш ефективним виявилось застосування Daubechies-4 із модифікованими коефіцієнтами порогової обробки. Результати показали, що для коефіцієнтів порогової компресії нижче 20% рівень втрат залишається в межах прийнятного значення ($\pm 1,5$ дБ у порівнянні з оригінальним зображенням).

Процедура проведення експериментів включала тестування передавання зображень у реальних умовах LPWAN-мереж із використанням LoRa-модулів на різних відстанях і за різних рівнів перешкод. Досліджено вплив частотного діапазону, рівня сигналу та ширини смуги LoRa на якість переданого зображення. Отримані результати свідчать, що за умов низької пропускну здатності мережі застосування багаторівневої вейвлет-компресії дозволяє зменшити обсяг переданих даних до 60% без суттєвих втрат якості.

Практичні результати експериментів підтвердили високу кореляцію між розрахунковими значеннями та фактичними вимірюваннями. Виявлено, що застосування адаптивної компресії, яка враховує особливості сигналу та мережеских обмежень, дозволяє досягти кращого співвідношення між швидкістю передачі та якістю відновленого зображення. Отримані результати можуть бути використані для подальшої оптимізації алгоритмів стиснення в умовах обмежених ресурсів, зокрема в системах відеоспостереження, розпізнавання образів та передавання зображень у сенсорних мережах.

Таким чином, отримані експериментальні результати та чисельні моделювання можуть бути інтегровані у процес розробки оптимізованих алгоритмів компресії для мереж з обмеженими ресурсами, що підвищить ефективність передавання зображень у реальних застосуваннях.

Ключові слова: стиснення зображень; вейвлет-перетворення; JPG; RLE; код Хаффмана; Haar; Daubechies; Coiflet; PSNR; SSIM; 32-х бітні мікроконтролери; LoRa; завадостійкість.

ABSTRACT

Mazin M.Yu. Image compression methods in Internet of Things systems on microcontrollers. - Qualification scientific work in the form of a manuscript. Dissertation for the degree of Candidate of Technical Sciences in specialty 171 - Acoustic and multimedia electronic systems. - National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", Ministry of Education and Science of Ukraine, Kyiv, 2025

The dissertation is devoted to solving a topical problem - studying the effectiveness of using wavelet transforms for image compression in embedded systems with limited resources and determining the optimal compression parameters. For this purpose, existing compression methods were analysed, their comparison was carried out, the main shortcomings were identified and ways to eliminate them were proposed. The work is the first to set and solve the problem of finding the influence of the decomposition level of the wavelet transform on the quality of the restored image in microcontroller systems, and to study the features of the operation of different types of wavelets (Haar, Daubéchy, Coiflet) in combination with JPEG compression. For the first time, a comparative analysis of the use of entropy compression algorithms, in particular the Huffman and RLE methods, in the context of their application for limited computing resources was carried out. A new method of adaptive correction of artifacts that arise during repeated use of wavelet transforms in combination with JPEG compression was proposed and implemented, which allows improving the quality of the restored image without a significant increase in computational costs.

To solve the problems, experimental research methods were used, numerical modelling of image compression and processing processes, as well as testing of the developed algorithms on hardware microcontroller platforms. Field and laboratory studies were conducted, which confirmed the high accuracy of the obtained mathematical models. The optimal compression parameters for effective use in embedded systems were determined, and the influence of the amount of RAM on compression performance in real hardware conditions was also investigated.

The practical significance of the results obtained lies in the possibility of using the developed algorithms to optimize image compression in video surveillance systems, pattern recognition, and wireless data transmission. The research performed allows developers of embedded systems to choose optimal compression methods for specific tasks, minimizing the use of memory and processor resources while maintaining high quality of processed images.

Numerical modelling methods and experimental studies were used to study the efficiency of image compression using wavelet transforms and entropy compression methods. Algorithmic modelling of compression processes was used, as well as testing of the proposed methods on real hardware, which allowed obtaining quantitative relationships between the transformation parameters and the quality of the restored image.

A new approach to compression optimization is proposed, based on the combined use of wavelet transform and entropy coding, in particular the Huffman and RLE methods. The paper compares different levels of decomposition of Haar, Daubech and Coiflet wavelets, which made it possible to assess their impact on image quality and compression level. The results obtained showed that the effectiveness of the methods largely depends on the choice of the decomposition level: with increasing level, image detail is lost, but the compression ratio increases. This allows determining the optimal compression parameters for specific applications in embedded systems.

It was found that the combination of wavelet transforms with entropy coding allows to significantly reduce the image size with minimal loss of quality. In particular, the use of wavelet transforms before applying the Huffman algorithm allows to achieve compression up to 30% more efficiently compared to traditional methods. At the same time, the RLE method demonstrated high efficiency for images with a large number of repeating pixels, but was inferior to the Huffman method in the case of complex structured images.

Numerical modelling of the influence of the microcontroller's RAM size on the speed of algorithm execution was carried out. Studies have shown that the use of external memory allows processing more complex images, but significantly reduces

compression performance. This indicates the need to balance between compression quality and algorithm speed depending on the capabilities of the hardware platform. An adaptive algorithm for correcting artifacts that arise during multiple applications of wavelet transforms in combination with JPEG compression was developed and tested. It was found that the artifacts are periodic in nature and depend on the level of decomposition. The proposed correction method allows eliminating these defects by using median filtering, which allows improving the quality of the restored image without significantly increasing computational costs.

The software for evaluating and compression efficiency in embedded systems, which takes into account compression parameters, hardware platform capabilities and the needs of a specific application. The results obtained can be used to select optimal image processing algorithms in video surveillance systems, pattern recognition, wireless data transmission and other tasks where it is necessary to reduce the size of the transmitted files while maintaining their high quality.

The work also posed and solved the problem of the influence of variable frequency characteristics on the efficiency of wavelet compression of images in systems with limited resources. The relevance of this problem is due to the widespread use of wireless data transmission technologies, where image compression is critically important for reducing delays and energy consumption. One of the main factors affecting the efficiency of compression algorithms is the frequency distribution of the signal, since changes in the spectral composition can significantly affect the wavelet decomposition coefficients.

The proposed mathematical model allows you to determine the time dependence of information loss when transmitting images that undergo frequency changes in the compression and decompression process. The effect of frequency dispersion that occurs when applying a wavelet transform to a signal with dynamically changing spectral characteristics is analysed. The study shows that with an increase in the number of decomposition levels, the image loses fine details, which is critical for applications in pattern recognition and video surveillance.

Practical experiments with image transmission via wireless communication channels have shown that even at high levels of spectral changes, the compression efficiency remains stable, which confirms the reliability of using wavelet transforms in difficult conditions. The study proves that the proposed method can be successfully used in real wireless data transmission systems, where it is necessary to achieve a balance between the quality of the restored image and its transmission speed.

During the work on the theoretical and experimental parts, field and laboratory studies of the effectiveness of using wavelet compression for wireless image transmission in LoRa-based LPWAN networks were conducted. To assess the quality of the restored images, objective metrics such as PSNR (Peak Signal-to-Noise Ratio) and SSIM (Structural Similarity Index) were used, as well as a subjective analysis of the visual susceptibility of artifacts that arise during multiple compression.

During the experiments, a comparison was made of images transmitted without compression and using different wavelet compression methods. It was found that the use of Daubechies-4, Coiflet and Haar wavelets gives different degrees of information loss. For the optimal combination of high quality of the restored image and minimal use of network bandwidth, the most effective was the use of Daubechies-4 with modified threshold processing coefficients. The results showed that for threshold compression coefficients below 20%, the level of loss remains within acceptable values (± 1.5 dB compared to the original image).

The experimental procedure included testing image transmission in real conditions of LPWAN networks using LoRa modules at different distances and with different levels of interference. The influence of the frequency range, signal level and LoRa bandwidth on the quality of the transmitted image was studied. The results obtained show that under conditions of low network bandwidth, the use of multi-level wavelet compression allows reducing the amount of transmitted data by up to 60% without significant loss of quality.

Practical experimental results confirmed a high correlation between calculated values and actual measurements. It was found that the use of adaptive compression, which takes into account the characteristics of the signal and network limitations,

allows achieving the optimal ratio between the transmission speed and the quality of the restored image. The results obtained can be used for further optimization of compression algorithms in conditions of limited resources, in particular in video surveillance systems, pattern recognition and image transmission in sensor networks.

Thus, the obtained experimental results and numerical simulations can be integrated into the process of developing optimized compression algorithms for networks with limited resources, which will increase the efficiency of image transmission in real applications.

Keywords: image compression; wavelet transform; JPG; RLE; Huffman coding; Haar; Daubechies; Coiflet; PSNR; SSIM; 32-bit microcontroller; LoRa; noise immunity.

Список публікацій здобувача

1. Мазін М. Ю., Оникієнко Ю. О. Застосування вейвлет-перетворення для обробки зображень у мікроконтролерних системах Інтернету речей // Технології та інжиніринг. – 2023. – №3(14). – С. 15–22. DOI: <https://doi.org/10.30857/2786-5371.2023.3.2>
2. Мазін М. Ю., Оникієнко Ю. О. Особливості використання методів Хаффмана та RLE для стиснення зображень в системах на мікроконтролерах // Технології та інжиніринг. – 2023. – №6(17). – С. 21–28. DOI: <https://doi.org/10.30857/2786-5371.2023.6.2>
3. Мазін, М., Оникієнко, Ю. (2024). Аналіз методів обробки зображень в системах Інтернету речей на основі вейвлет перетворень. *Technologies and Engineering*, 25(6), 53-60. DOI: <https://doi.org/10.30857/2786-5371.2024.6.5>.

ВСТУП.....	15
1 ОГЛЯД БЕЗДРОТОВИХ СЕНСОРІВ ЗОБРАЖЕНЬ	21
1.1. Сенсорні системи як складова Інтернету речей (IoT)	21
1.2. Типи існуючих бездротових сенсорних систем	23
1.3. Алгоритми обробки зображень у бездротових сенсорних системах ..	24
1.4. Проблеми існуючих систем та перспективи запропонованої системи	25
Висновки	34
2 ПОРІВНЯННЯ ВЕЙВЛЕТ-ПЕРЕТВОРЕНЬ ДЛЯ РОБОТИ В СИСТЕМАХ ІНТЕРНЕТУ РЕЧЕЙ	36
2.1 Аналіз використання вейвлет-перетворень в сенсорних системах	36
2.2 Теоретичні засади вейвлет-перетворень	42
2.4 Аналіз результатів експерименту для порівняння вейвлет-перетворень Хаара, Добеші та Коіфлет	65
Висновок.....	81
3 ПОРІВНЯННЯ МЕТОДІВ СТИСЕННЯ ДЛЯ РОБОТИ В СИСТЕМАХ ІНТЕРНЕТУ РЕЧЕЙ	83
3.1 Аналіз використання методів стиснення в сенсорних системах	83
3.2 Теоретичні відомості методів стиснення	84
3.3 Аналіз наукових праць з використанням методів Хаффмана та RLE	97
3.4 Аналіз результатів експерименту з використанням методів Хаффмана та RLE у системах інтернету речей	100
Висновки	106
4 РЕАЛІЗАЦІЯ КОМБІНОВАНОГО МЕТОДУ СТИСЕННЯ В СИСТЕМІ ІНТЕРНЕТ РЕЧЕЙ	109
4.1 Опис роботи JPEG для реалізації комбінованого методу	109
4.2 Опис комбінованої роботи вейвлет-перетворень з JPEG	112
4.3 Порівняння запропонованої системи з існуючими системами інтернет речей.....	117
4.4 Аналіз результатів реалізації системи стиснення зображень комбінованим методом	119
Висновки	129

5 РЕАЛІЗАЦІЯ АЛГОРИТМУ ВИПРАВЛЕННЯ АРТЕФАКТІВ

ЗОБРАЖЕННЯ	132
5.1 Опис появи артефактів під час роботи комбінованого методу.....	132
5.2 Аналіз математичного та програмного рішення	134
5.3 Аналіз та порівняння результатів експерименту	137
Висновки	141
ЗАГАЛЬНІ ВИСНОВКИ.....	144
ПЕРЕЛІК ЛІТЕРАТУРИ	146

ВСТУП

Обґрунтування вибору теми дослідження. Сучасний розвиток Інтернету речей (IoT) та його інтеграція в різні галузі стимулюють впровадження передових методів обробки даних, особливо зображень. Потреба у вирішенні задач, пов'язаних з фільтрацією, стисненням і передачею зображень, зростає разом зі збільшенням кількості розумних пристроїв у таких сферах, як охорона здоров'я, промисловість, транспорт, сільське господарство тощо. Висока ефективність обробки даних в системах IoT обмежується апаратними ресурсами пристроїв, таких як мікроконтролери. Виникає необхідність адаптації сучасних алгоритмів для роботи в умовах низької енергоспоживання, обмеженої пам'яті та невисокої обчислювальної потужності.

Методи вейвлет-перетворень показали свою ефективність у вирішенні задач обробки зображень завдяки можливості багаторівневого аналізу даних в частотно-часовій області. Зокрема, вейвлети Daubechies, Haar, Coiflet широко застосовуються для стиснення, фільтрації та відновлення зображень. Вейвлети забезпечують гнучкість у виборі компромісу між швидкістю обробки, обсягом пам'яті та якістю відновлених даних, що робить їх перспективними для використання на мікроконтролерах.

Аналіз сучасних досліджень показує, що вейвлет-перетворення мають значний потенціал для використання в IoT-системах, проте існує недолік розробок, спрямованих на оптимізацію цих методів для мікроконтролерів. Особливо це стосується адаптації вейвлет-алгоритмів для обробки зображень у реальному часі в умовах обмежених ресурсів.

Значний внесок у дослідження вейвлет-перетворень та їх адаптації для IoT зробили такі науковці, як М. Шалунов, А. Садовський, Х. Ван, Л. Джан, Т. Браун, П. Ізраель, В. Чжан, І. Беяков, А. Джеймс та інші. Водночас, проблематика використання вейвлетів у контексті обробки зображень на мікроконтролерах залишається недостатньо вивченою.

Дослідження в цьому напрямі є актуальними, оскільки дозволяють підвищити ефективність роботи IoT-пристроїв, що сприятиме подальшому розвитку розумних систем, таких як бездротові сенсорні мережі, системи моніторингу та аналізу даних у реальному часі.

Мета і задачі дослідження. Метою роботи є розробка системи стиснення зображень на мікроконтролері з метою зменшення обсягу збережених даних у вбудованих системах з обмеженими ресурсами, аналіз методів стиснення зображень RLE та Хаффмана, дослідження вейвлет-перетворень, таких як вейвлет Хаара, Добеші та Коіфлет, для обробки зображень, підвищення ефективності стиснення за допомогою комбінованого підходу вейвлет-перетворення з JPEG на мікроконтролерах, а також виправлення артефактів, що виникає під час перетворень.

Об'єкт дослідження – методи стиснення та вейвлет-перетворень для обробки зображень на мікроконтролері з подальшим виправленням артефактів.

Предмет дослідження – системи на мікроконтролерах в яких виконується отримання і обробка зображень.

Для досягнення поставленої мети необхідно вирішити такі завдання:

1. Дослідити методи стиснення зображень, такі як RLE та метод Хаффмана, із використанням вейвлет-перетворення;
2. Дослідити вплив використання вейвлет-перетворення з JPEG стисненням;
3. Дослідити використання евклідової відстані як додаткової метрики оцінки якості перетвореного зображення;
4. Провести експериментальне дослідження для виявлення та виправлення похибок після використання вейвлет-перетворення.

Методи дослідження. Для досягнення поставленої мети використано методи математичного моделювання для побудови адаптивного алгоритму, методи порівняльного аналізу для оцінки ефективності різних модифікацій

алгоритму дельта-кодування, а також експериментальний метод для тестування розробленого алгоритму на реальних графічних даних.

Наукова новизна одержаних результатів полягає у наступному:

1. Запропоновано удосконалений метод, що об'єднує вейвлет-перетворення з традиційними методами стиснення, такими як RLE та метод Хафмана, які дозволяють досягти зменшення розміру файлів зображень без втрат якості в умовах обмежених ресурсів мікроконтролера.

2. Вперше використано евклідової відстані як метрики, яка враховує розбіжність між піксельними значеннями оригінального та перетвореного зображення, що дозволить оцінити перетворене зображення на наявність великих розбіжностей спричинені накопиченням математичної похибки.

3. Проведено адаптацію алгоритмів вейвлет перетворення Daubechies, Haar та Coiflet та проведено їх порівняльний аналіз ефективності для задач обробки зображень у реальному часі в умовах обмежених ресурсів мікроконтролера. Досліджено залежність часу виконання та якості перетвореного зображення від розміру зображення та кількості ітерацій вейвлет перетворення. Досліджено вплив додаткових перетворень на якість вихідного зображення, що виражено значеннями PSNR, SSIM та MSE.

4. Запропоновано метод виправлення похибки, що виникає під час математичних обчислень вейвлет перетворення та спричиняє виникненню артефактів на перетвореному зображенні. Експериментально підтверджено ефективність алгоритму для виявлення та виправлення похибки, що призводить до значного підвищення якості перетвореного зображення.

Особистий внесок здобувача. Результати досліджень, що наведені у дисертаційній роботі та винесені на захист, були отримані особисто автором або ж за його активної участі та були опубліковані у спеціалізованих фахових виданнях України.

В рамках наукової роботи [1], що була опублікована в співавторстві у фаховому виданні України, здобувачем особисто виконано наступне: запропоновано використання вейвлет-перетворення Хаара як додаткове перетворення в JPEG для стиснення зображення, враховуючи обмежені ресурси мікроконтролера. Методологія передбачала експериментальну оцінку якості зображення обробленою за допомогою вейвлет-перетворення Хаара та стиснення з використанням перетворення JPEG на модулі мікроконтролера ESP32-CAM, який знімає зображення за допомогою вбудованої камери..

В рамках наукової роботи [2], що була опублікована в співавторстві у фаховому виданні України, здобувачем особисто створено програмний модуль для реалізації алгоритмів стиснення Хаффмана та RLE на мікроконтролері ESP32-CAM. Досліджено ефективність використання алгоритмів Хаффмана та RLE, для стиснення зображень, використовуючи обмежені ресурси мікроконтролера, порівняно ефективність стиснення та розміри стиснених файлів.

В рамках наукової роботи [3], що була опублікована в співавторстві у фаховому виданні України, здобувачем особисто проведено дослідження ефективності використання вейвлет-перетворень для стиснення зображень на мікроконтролері ESP32. Реалізовано програмний модуль, що здійснює вейвлет-декомпозицію зображень з використанням перетворень Хаара, Добеші та Коїфлета, а також їх подальшу компресію у форматі JPEG. Проведено експериментальний аналіз рівня компресії, швидкості обробки та впливу рівня декомпозиції на якість зображень. Оцінено вплив накопичення похибок на появу артефактів та визначено кращі параметри компресії для мінімізації втрат якості при обмежених обчислювальних ресурсах.

Практичне значення одержаних результатів:

Використання вейвлет-перетворення разом із JPEG дозволяє зменшити розмір зображень до мінімуму, зберігаючи при цьому візуальну якість. Це особливо корисно для зберігання та передачі зображень у системах з обмеженою

пропускною здатністю, таких як LPWAN-мережі, IoT-пристрої або супутникові канали зв'язку.

Реалізований алгоритм виявлення та виправлення похибок дозволяє мінімізувати вплив артефактів, що виникають внаслідок багаторазових операцій обробки, таких як перетворення, стиснення та декомпресія. Це забезпечує високу точність передачі зображень для критично важливих застосувань, наприклад, у медичній діагностиці, супутниковому моніторингу та безпекових системах.

Алгоритм виправлення артефактів сприяє значному покращенню візуального сприйняття стиснених зображень кінцевими користувачами. Це особливо важливо у випадках, коли якість є критичним параметром, наприклад, у рекламній графіці або цифровій фотографії.

Зменшення розміру файлів без втрати якості дозволяє значно оптимізувати використання дискового простору та знизити вартість передачі даних через мережі, особливо в умовах обмеженої пропускної здатності.

Гнучкість методу дозволяє адаптувати алгоритми до специфічних типів зображень (фотографії, рентгенівські знімки, картографічні зображення), що підвищує універсальність застосування запропонованого підходу.

Апробація результатів дисертації. Матеріали дисертаційних досліджень обговорювалися на міжнародних конференціях:

- Міжнародна мультидисциплінарна наукова інтернет-конференція " Світ наукових досліджень. Випуск 31", м. Ополе, 20-21 червня 2024 року.

Публікації. Основні результати дисертаційної роботи представлені у 3 публікаціях наукових робіт, у тому числі в 2 статтях у наукових фахових виданнях України, які включені до міжнародних науко метричних баз; 1 тези доповіді у збірниках матеріалів конференцій.

Структура та обсяг дисертації. Робота складається зі вступу, чотирьох розділів, висновків, списку використаних джерел із 97 найменувань. Робота

містить 28 рисунків та 14 таблиць. Загальний обсяг дисертаційної роботи становить 144 сторінок.

1 ОГЛЯД БЕЗДРотовИХ СЕНСОРІВ ЗОБРАЖЕНЬ

1.1. Сенсорні системи як складова Інтернету речей (IoT)

Інтернет речей (IoT) є однією з ключових технологій сучасного світу, що забезпечує збір, обробку та передачу даних від сенсорів до кінцевих користувачів або систем аналізу. Основною складовою IoT є бездротові сенсорні системи (Wireless Sensor Systems, WSS), які використовуються у різних сферах, таких як розумне місто, медицина, аграрний сектор, промисловість та екологічний моніторинг [4]. Головна мета таких систем — автоматизований збір даних та їх подальший аналіз для прийняття рішень або оптимізації процесів.

Бездротові сенсорні системи складаються з великої кількості вузлів, які можуть бути розташовані у важкодоступних місцях для контролю фізичних параметрів середовища. Вони використовують різні протоколи зв'язку, серед яких LPWAN (LoRa, Sigfox, NB-IoT), Wi-Fi, Bluetooth та Zigbee [5]. Кожен із цих стандартів має свої переваги та обмеження, які визначають сферу їх застосування.

Бездротові сенсорні мережі також використовують різні архітектури, включаючи однорангові мережі (peer-to-peer), ієрархічні мережі та мережі на основі кластеризації. Вибір архітектури впливає на загальну продуктивність, витрати енергії та ефективність зв'язку між вузлами. Наприклад, однорангові мережі мають перевагу в гнучкості, але можуть страждати від перевантаження при збільшенні кількості вузлів.

Крім того, бездротові сенсорні системи часто застосовують методи енергоефективного управління для продовження терміну служби сенсорних вузлів. Серед таких методів можна виділити адаптивне планування передачі даних, використання сплячого режиму та алгоритми збору даних з урахуванням стану мережі. Наприклад, у системах, що базуються на LoRa, використовується механізм адаптивної швидкості передачі (ADR), що дозволяє балансувати між дальністю зв'язку та витратами енергії вузла.

Сучасні дослідження також зосереджуються на вдосконаленні бездротових сенсорних мереж шляхом інтеграції з технологіями машинного навчання та

штучного інтелекту. Це дозволяє підвищити точність аналізу зібраних даних та автоматизувати процеси прийняття рішень на основі прогнозних моделей. Наприклад, у сільському господарстві застосування алгоритмів машинного навчання для обробки даних з сенсорів допомагає передбачати розвиток шкідників або хвороби рослин, що значно покращує врожайність та ефективність управління ресурсами.

Використання сенсорних систем у різних галузях

1. Розумні міста: Сенсорні мережі використовуються для моніторингу дорожнього руху, контролю якості повітря, управління вуличним освітленням та оптимізації використання енергетичних ресурсів. Наприклад, у роботі [6] розглядається застосування IoT у розумних містах, зокрема в управлінні транспортом і зменшенні заторів шляхом аналізу трафіку в реальному часі.

2. Медицина: Носимі пристрої, такі як розумні годинники та імплантовані сенсори, дозволяють проводити безперервний моніторинг життєвих показників пацієнтів. У дослідженні [7] розглядаються переваги застосування IoT у медицині, включаючи діагностику хронічних захворювань та екстрений моніторинг пацієнтів у віддалених регіонах.

3. Агропромисловість: Використання сенсорів дозволяє автоматизувати процеси в сільському господарстві, наприклад, визначення вологості ґрунту, рівня добрив і температури повітря. [8] досліджують, як технології IoT сприяють сталому розвитку аграрного сектору, оптимізуючи витрати води та підвищуючи врожайність.

4. Промисловість: Сенсорні мережі використовуються для контролю роботи обладнання, виявлення несправностей та прогнозного обслуговування. [9] аналізують, як промисловий IoT може допомогти зменшити простої та аварійні зупинки виробництва завдяки безперервному моніторингу стану обладнання.

5. Екологічний моніторинг: Сенсори дозволяють виявляти забруднення повітря, води та ґрунту в реальному часі, що є критично важливим для зменшення впливу людини на навколишнє середовище. [10] розглядають роль

бездротових сенсорних мереж у виявленні та запобіганні екологічним катастрофам, таким як витoki небезпечних речовин або лісові пожежі.

Розробка нових методів обробки даних та оптимізація алгоритмів стиснення зображень є важливими напрямками досліджень у цій сфері. Запропоновані в цій роботі методи спрямовані на покращення ефективності передачі даних у бездротових сенсорних системах.

1.2. Типи існуючих бездротових сенсорних систем

Бездротові сенсорні системи можна класифікувати за різними критеріями, що визначають їх ефективність у конкретних сценаріях застосування. Основні типи включають:

1. Системи на основі низькоенергетичних мереж LPWAN (Low Power Wide Area Network): LoRa (Long Range) – використовується для довготривалої передачі даних на великі відстані з мінімальним енергоспоживанням. Застосовується в екологічному моніторингу та розумному сільському господарстві [11]. Sigfox – працює у вузько смуговому режимі та підходить для передачі малих обсягів даних у розумних містах та індустріальних додатках [12]. NB-IoT (Narrowband IoT) – розроблена для зв'язку в умовах низької пропускної здатності та використовується у віддаленому моніторингу пристроїв [13].

2. Системи на основі бездротових персональних мереж (WPAN - Wireless Personal Area Networks): Bluetooth Low Energy (BLE) – широко використовується у носимих пристроях та IoT-інтеграції в побутових умовах [14]. Zigbee – підходить для розумного будинку та промислових сенсорних систем завдяки низькому енергоспоживанню [15].

3. Системи на основі бездротових локальних мереж (WLAN - Wireless Local Area Networks): Wi-Fi – забезпечує високу швидкість передачі даних, що корисно для моніторингу відео та складних сенсорних додатків у розумних будівлях [16].

4. Супутникові сенсорні системи: Використовуються для глобального моніторингу, наприклад, у віддалених районах або океанографічних дослідженнях, де інші технології недоступні [17].

Різні типи бездротових сенсорних систем мають унікальні переваги та обмеження, що робить їх придатними для різних застосувань у сфері IoT.

1.3. Алгоритми обробки зображень у бездротових сенсорних системах

Обробка зображень у бездротових сенсорних системах є важливим напрямом досліджень, особливо з огляду на обмежені ресурси сенсорних вузлів. Основні методи включають стиснення даних, покращення якості зображень та застосування машинного навчання для автоматизованого аналізу.

Одним із перспективних напрямків є використання адаптивних методів обробки зображень, які дозволяють регулювати рівень стиснення відповідно до умов мережі. Адаптивні алгоритми використовують машинне навчання для визначення кращих параметрів стиснення в реальному часі, що дозволяє зменшити втрати при передачі та покращити якість відновлених зображень. Серед таких методів:

- Адаптивне квантування – техніка, що змінює рівень деталізації зображення залежно від поточного рівня сигналу в мережі [18].
- Методи субдискретизації – зменшення просторової роздільної здатності в нерелевантних областях, що дозволяє передавати критично важливі деталі із вищою точністю [19].
- Алгоритми фільтрації артефактів – нейромережеві підходи для покращення візуальної якості відновлених зображень, зменшення шумів та артефактів, спричинених втратами пакетів [20].

Розподілені обчислення є ще одним напрямком оптимізації бездротових сенсорних мереж. У таких системах обробка зображень відбувається безпосередньо на вузлах мережі, що дозволяє зменшити обсяг переданих даних і покращити швидкість реакції системи. Основні переваги цього підходу включають:

- Зменшення навантаження на канал передачі – попередня обробка та виділення ключових особливостей безпосередньо на сенсорних вузлах значно скорочує обсяг даних, що передаються до центрального сервера [21].

- Зниження затримок – локальна обробка дозволяє отримувати результати швидше, що критично для систем реального часу, таких як відеоспостереження та охоронні системи [22].

- Підвищена надійність мережі – можливість відновлення або повторного аналізу зображень у разі втрати даних під час передачі [23].

1.4. Проблеми існуючих систем та перспективи запропонованої системи

Незважаючи на широкий спектр застосувань, сучасні бездротові сенсорні мережі мають низку обмежень, які ускладнюють їх використання для високопродуктивних та енергоефективних обчислень:

- Обмежені ресурси енергоспоживання.
- Висока вартість передачі даних.
- Низька ефективність стиснення зображень.
- Відсутність адаптивних алгоритмів для управління якістю передачі.

Більшість сенсорних вузлів живляться від батареї, що обмежує їхню тривалість роботи. Методи енергозбереження, такі як використання енергоефективних протоколів та технологій збору енергії, ще не є достатньо ефективними для широкомасштабного впровадження [24].

Передача великих обсягів даних через бездротові мережі є енерговитратною. Наприклад, у бездротових візуальних сенсорних мережах (WVSN) передача зображень вимагає значних обчислювальних ресурсів і потужності зв'язку, що обмежує застосування таких систем у реальному часі [26].

Використання традиційних алгоритмів стиснення, таких як JPEG або H.264, не завжди є ефективним для сенсорних мереж, оскільки вони можуть спричиняти значні затримки та споживати багато енергії.

У більшості існуючих рішень відсутні методи, які б дозволяли динамічно змінювати рівень деталізації або ступінь стиснення відповідно до умов мережі та наявних ресурсів енергії [24].

Для подолання зазначених проблем пропонується система, яка використовує:

- Вейвлетне перетворення для ефективного стиснення зображень. Метод дозволяє зменшити обсяг переданих даних без значних втрат якості. Використання швидкого вейвлетного перетворення (FWT) та дискретного вейвлетного перетворення (DWT) дозволяє знизити енергоспоживання при обробці [24].

- Адаптивне управління рівнем стиснення. На основі поточного рівня заряду сенсорного вузла система може змінювати параметри компресії зображень, що дозволяє балансувати між якістю переданої інформації та витратами енергії [26].

- Передачу даних через LoRa. Вибір LoRa як каналу передачі обумовлений його низьким енергоспоживанням, великою зоною покриття та можливістю роботи у важкодоступних регіонах. Завдяки цьому система може передавати стислий відеопотік або зображення навіть у середовищах з обмеженою пропускнуою здатністю [24].

- Розподілену обробку зображень на вузлах. Часткова обробка виконується безпосередньо на мікроконтролерах вузлів (наприклад, ESP32-CAM), що зменшує навантаження на центральний сервер та підвищує ефективність системи.

Запропоноване рішення у роботі [24] починається із захоплення зображення модулем ESP32-CAM. Оскільки сенсорні пристрої мають обмежені ресурси, перед передачею виконується попередня обробка, яка включає конвертацію кольорового зображення у відтінки сірого та фільтрацію шумів для зменшення обсягу даних. Далі використовується вейвлетне перетворення (DWT), яке ділить зображення на низькочастотні та високочастотні компоненти. Основна інформація зберігається в низькочастотних компонентах, тоді як менш значущі деталі можуть бути частково відфільтровані для зменшення розміру файлу.

Отримані дані додатково стискаються за допомогою JPEG або RLE, що дозволяє ще більше скоротити обсяг інформації для передачі. Особливістю цього

підходу є адаптивне управління рівнем стиснення: система автоматично змінює параметри компресії відповідно до поточної пропускну здатності каналу LoRa. Це дозволяє ефективно використовувати мережу, мінімізуючи витрати енергії на передачу.

LoRa (Long Range) — це технологія бездротового зв'язку з низьким енергоспоживанням, призначена для використання в мережах Інтернету речей (IoT). Вона базується на модуляції з розширенням спектра за допомогою чирп-сигналів (Chirp Spread Spectrum, CSS) і забезпечує зв'язок на значних відстанях (до кількох десятків кілометрів) при низькому рівні споживання енергії [25]. LoRa використовується в таких сферах, як розумні міста, промисловий IoT, сільське господарство, моніторинг навколишнього середовища та бездротові сенсорні мережі.

Однією з найважливіших характеристик LoRa є її стійкість до завад та імпульсного шуму, що є критичним у промислових і міських середовищах. У дослідженні [11] аналізується вплив імпульсного шуму на LoRa-зв'язок. Було виявлено, що за присутності імпульсного шуму, змодельованого за допомогою розподілу Міддлтона класу А (Middleton Class-A Noise, MCAN), ефективність передачі суттєво знижується, а втрати відношення сигнал/шум (SNR) можуть сягати 10 дБ. Однак, збільшення коефіцієнта розширення спектра (SF) дозволяє знизити вплив такого шуму.

Результати дослідження показують, що в середовищах з сильним імпульсним шумом вибір вищого SF (наприклад, $SF = 12$ замість $SF = 7$) зменшує вплив шуму та покращує надійність зв'язку. Це пояснюється тим, що алгоритм декодування LoRa використовує операцію перетворення Фур'є (FFT), яка зменшує вплив імпульсного шуму після демодуляції [11].

Технологія LoRa має низку переваг, серед яких низьке енергоспоживання, значна дальність зв'язку та стійкість до перешкод. Завдяки використанню модуляції CSS та широкому вибору параметрів конфігурації, LoRa здатна працювати в умовах значного рівня шуму, що робить її оптимальним рішенням для промислового IoT та інших застосувань. Проте, як показано у дослідженні

[11], на ефективність LoRa можуть впливати імпульсні завади, що вимагає правильного налаштування параметрів модуляції для забезпечення надійності зв'язку.

Оскільки LoRa підтримує лише обмежений розмір пакета, зображення розбивається на фрагменти, кожен з яких містить номер кадру для правильного складання зображення після отримання. Пакети передаються через LoRa-модуль до базової станції, яка отримує їх та пересилає на сервер. На сервері відбувається зворотне вейвлетне перетворення, реконструкція стисненого зображення та його відновлення у вихідний формат. Після цього зображення передається в користувацький інтерфейс, наприклад, у веб-додаток або систему моніторингу, де користувач може переглянути отриману інформацію. На рис. Рисунок 1 представлено основну концепцію запропонованої в статті системи

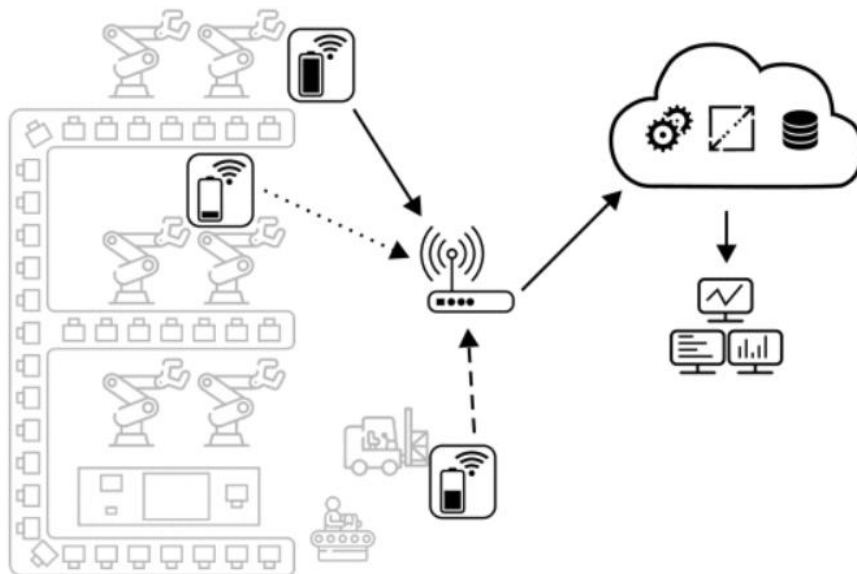


Рисунок 1 – Датчики IoT, що відображають різні стани зберігання енергії, передають зібрані дані в хмару з різними рівнями стиснення [24].

Такий підхід дозволяє значно зменшити обсяг переданих даних, мінімізувати енергоспоживання сенсорного пристрою та забезпечити передачу зображень навіть у мережах з низькою пропускнуою здатністю, таких як LoRa.

У роботі [26] досліджується застосування вейвлетного перетворення для стиснення даних у бездротових візуальних сенсорних мережах (WVSN). Основна

мета полягає в підвищенні енергоефективності та забезпеченні передачі зображень у реальному часі для таких застосувань, як моніторинг навколишнього середовища.

У WVSN сенсорні вузли оснащені камерами для збору візуальної інформації. Однак передача необроблених зображень споживає значну кількість енергії та вимагає широкої пропускну здатності, що є проблематичним для сенсорних мереж з обмеженими ресурсами. Для вирішення цієї проблеми автори пропонують використовувати дискретне вейвлетне перетворення (DWT) для стиснення зображень перед їх передачею.

Процес роботи системи включає кілька етапів:

1. Захоплення зображення: Сенсорний вузол з камерою захоплює зображення навколишнього середовища.

2. Вейвлетне перетворення: Отримане зображення піддається DWT, яке розкладає його на піддіапазони з різними частотними компонентами. Це дозволяє виділити основні особливості зображення та зменшити обсяг даних.

3. Квантування та стиснення: Після перетворення коефіцієнти квантуються та стискаються для подальшого зменшення розміру даних.

4. Передача даних: Стиснені дані передаються через бездротову мережу до центрального вузла або сервера для подальшої обробки та аналізу.

Автори проводять аналіз продуктивності різних вейвлетів для забезпечення енергоефективної та реальної передачі зображень у WVSN. Вони досліджують, як вибір конкретного вейвлета впливає на якість стиснення та енергоспоживання, що є критичним для додатків, таких як моніторинг навколишнього середовища. Схема роботи наведена на рис. Рисунок 2.

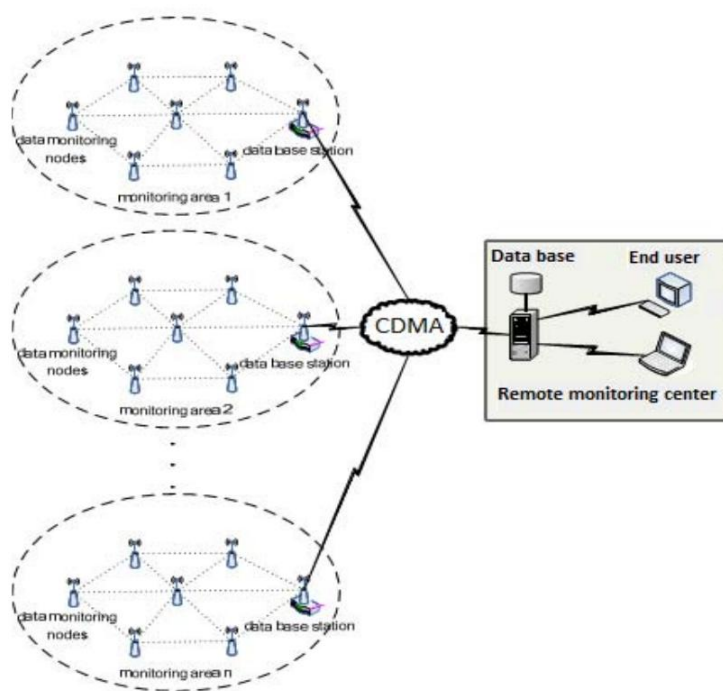


Рисунок 2 – Огляд системи WVSN для моніторингу середовища

Використання вейвлетного стиснення дозволяє значно зменшити обсяг переданих даних, що, у свою чергу, знижує енергоспоживання сенсорних вузлів та продовжує їхній термін служби. Це робить можливим застосування WVSN у віддалених або важкодоступних районах, де заміна або підзарядка батареї є складною.

На рис. Рисунок 3 наведено перелік використаних елементів та їх взаємодія у запропонованій в роботі системі.



Рисунок 3 – Схема роботи системи на мікроконтролері

Система складається з наступних основних компонентів:

1. Сенсор ESP32-CAM: збирає та обробляє зображення, стискає їх за допомогою методу комбінованого стиснення та передає через модуль LoRa.
2. Передавач LoRa: приймає дані від сенсорних вузлів, буферизує та передає від сенсора на сервер.
3. Сервер обробки та зберігання: отримує стислий потік даних, виконує реконструкцію зображень та аналіз отриманої інформації.
4. Користувачський інтерфейс: надає доступ до збережених даних та аналітичних результатів у реальному часі.

Запропонований підхід дозволяє значно зменшити навантаження на мережу, підвищити тривалість роботи сенсорних вузлів та забезпечити ефективне передавання зображень у бездротових сенсорних системах.

На рис. Рисунок 4 представлено покроковий процес обробки зображення, що включає комбіноване стиснення з використанням вейвлет-перетворення та JPEG, а також передачу через LoRa.

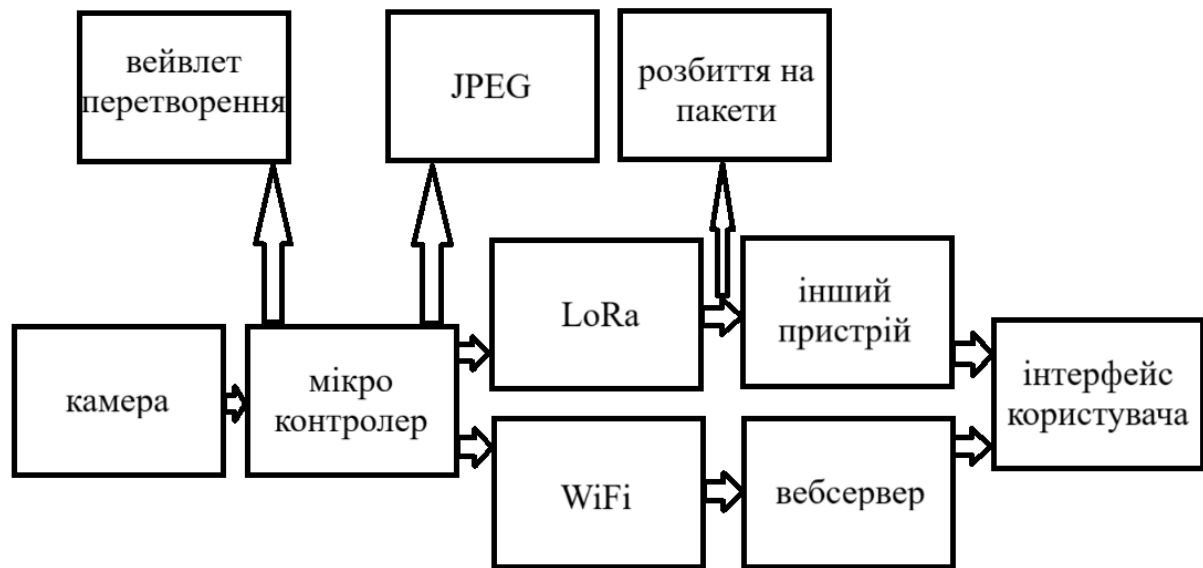


Рисунок 4 – Покроковий процес обробки зображень

1. Захоплення зображення

Сенсорний модуль ESP32-CAM захоплює зображення за допомогою вбудованої камери. Отримане зображення зазвичай має формат RGB або YUV та певну роздільну здатність (наприклад, 320×240 пікселів). При необхідності, проводиться конвертація у відтінки сірого (Grayscale) для зменшення обсягу даних.

2. Вейвлетне перетворення

Виконується дискретне вейвлетне перетворення (DWT), наприклад, з використанням вейвлетів Daubechies, Haar або Coiflet. Зображення ділиться на чотири підзображення: LL – низькочастотні коефіцієнти (головна інформація зображення), LH – горизонтальні деталі, HL – вертикальні деталі, HH – діагональні деталі. Видаляються незначущі високочастотні компоненти для подальшого стиснення. Може застосовуватись порогове значення (thresholding) для подальшої оптимізації.

3. Кодування JPEG

До LL-компоненти застосовується стандартний JPEG-кодер. Основні етапи JPEG-стиснення: перетворення кольору в простір $YCbCr$ (якщо використовується кольорове зображення), поділ на блоки 8×8 пікселів,

дискретне косинусне перетворення (DCT) – перетворення зображення у частотну область, квантування – видалення малозначущих коефіцієнтів, ентропійне кодування (Huffman або RLE) – стискання залишкових даних.

4. Розбиття даних на пакети для передачі через LoRa

JPEG-стиснене зображення поділяється на фрагменти (наприклад, по 256 байтів, відповідно до обмежень LoRa). До кожного пакету додається ідентифікатор пакета (щоб уникнути втрати даних і можливості повторної передачі), контрольна сума для перевірки коректності отримання. Використовується модифікований LoRa-протокол (наприклад, LoRaWAN або власна реалізація з повторними запитами).

5. Передача через LoRa

Сенсорний вузол передає пакети через LoRa-модуль. LoRa забезпечує низьку швидкість передачі (~300 біт/с - 50 кбіт/с), але велику дальність зв'язку (до 10 км). Передача може бути реалізована у трьох режимах: передача з підтвердженням (ACK) – кожен пакет повинен бути підтверджений перед тим, як передавати наступний, без підтвердження (Unacknowledged Mode) – мінімізує затримки, але є ризик втрати пакетів, повторна передача при помилці (Retransmission on Error) – відправка лише втрачених або пошкоджених пакетів.

6. Прийом даних на LoRa-базовій станції

Базова станція LoRa приймає пакети та передає їх на сервер обробки через Wi-Fi, Ethernet або інший зв'язок. Відбувається перевірка цілісності переданих даних. Якщо деякі пакети загублені, може бути ініційована повторна передача.

7. Відновлення зображення на сервері

Сервер збирає всі пакети в єдиний файл JPEG. Виконується декодування JPEG, відновлюючи стиснене зображення. Виконується зворотне вейвлетне перетворення (IDWT) для реконструкції високочастотних компонентів. Об'єднання отриманих частотних компонентів для формування остаточного зображення.

8. Відображення на користувацькому інтерфейсі

Відновлене зображення передається на веб-інтерфейс або мобільний додаток. Користувач може переглядати отримане зображення у реальному часі або з затримкою залежно від налаштувань системи.

Переваги запропонованого підходу

1. Значне зменшення обсягу переданих даних за рахунок комбінованого стиснення (вейвлети + JPEG).
2. Зменшене енергоспоживання завдяки локальній обробці на ESP32-CAM.
3. Можливість передачі через LoRa навіть у віддалених регіонах з низькою швидкістю зв'язку.
4. Гнучкість адаптивного стиснення відповідно до умов мережі та залишкового заряду сенсора.
5. Реконструкція високочастотних компонентів для покращення якості відновленого зображення.

Запропонований метод поєднує ефективне вейвлетне стиснення для усунення надлишкової інформації з JPEG-компресією для досягнення високого ступеня зменшення розміру файлу. Передача зображень через LoRa стає можливою навіть за умов низької пропускної здатності завдяки ефективному розбиттю даних на пакети. Впровадження цього підходу дозволяє застосовувати бездротові візуальні сенсорні системи у середовищах з обмеженими ресурсами, наприклад, у розумному сільському господарстві, екологічному моніторингу та системах безпеки.

Висновки

Сенсорні системи відіграють ключову роль у розвитку Інтернету речей (IoT), забезпечуючи автоматизований збір, обробку та передачу даних у різних сферах, включаючи розумні міста, медицину, агропромисловість, промисловість та екологічний моніторинг. Використання бездротових сенсорних мереж значно розширює можливості моніторингу та аналізу, проте існуючі технології мають певні обмеження, такі як високе енергоспоживання, вузька пропускна здатність каналів зв'язку та необхідність ефективного стиснення переданих даних.

Для розв'язання цих проблем сучасні дослідження спрямовані на вдосконалення бездротових сенсорних систем за допомогою нових алгоритмів обробки зображень, адаптивного управління ресурсами та використання штучного інтелекту. Зокрема, комбіновані методи стиснення, такі як вейвлетне перетворення у поєднанні з JPEG, дозволяють ефективно зменшити обсяг переданих даних без значних втрат якості. Використання LoRa для бездротової передачі забезпечує високу енергоефективність і можливість роботи в умовах обмежених ресурсів.

Таким чином, розробка та впровадження інноваційних методів обробки даних і стиснення зображень є критично важливими для підвищення продуктивності та надійності бездротових сенсорних систем. Подальші дослідження в цій галузі можуть сприяти створенню більш ефективних IoT-рішень, що будуть застосовуватися у віддаленому моніторингу, автоматизованих системах управління та прогнозуванні змін у навколишньому середовищі.

2 ПОРІВНЯННЯ ВЕЙВЛЕТ-ПЕРЕТВОРЕНЬ ДЛЯ РОБОТИ В СИСТЕМАХ ІНТЕРНЕТУ РЕЧЕЙ

2.1 Аналіз використання вейвлет-перетворень в сенсорних системах

Сучасні сенсорні мережі складаються з вузлів, які зазвичай оснащені датчиками, обчислювальними блоками, модулями бездротового зв'язку з низькою швидкістю передачі даних і джерелами живлення, такими як акумулятори. Основна вимога до таких систем полягає у їхній енергоефективності та невисокій вартості. Тому для вузлів зазвичай використовуються недорогі мікроконтролери з обмеженими обчислювальними ресурсами та невеликою кількістю оперативної пам'яті [27]. Окрім традиційних сенсорних вузлів, можливе створення камерних сенсорних мереж, де вузли додатково оснащуються невеликими камерами для моніторингу навколишнього середовища чи відстеження об'єктів [28], [29]. Проте такі системи мають підвищені вимоги до ресурсів, зокрема пам'яті та швидкості обробки даних.

Навіть якщо зображення можуть бути збережені на вузлі у зовнішній флеш-пам'яті [30], їхня передача через мережу потребує значних енергетичних ресурсів та високої пропускної здатності каналу. У зв'язку з цим виникає необхідність виконання обробки або стиснення зображень перед передачею, що дозволяє зменшити обсяг даних і, відповідно, споживання енергії.

Одним із широко використовуваних методів попередньої обробки зображень є вейвлет-перетворення, яке забезпечує ефективне стиснення та зменшення обсягу даних. Раніше методи обробки зображень потребували значних обсягів оперативної пам'яті, що перевищувало можливості стандартних мікроконтролерів. Проте сучасні розробки, такі як фракційні хвильові фільтри, дозволяють значно скоротити вимоги до пам'яті. Наприклад, у роботі [31] запропоновано фільтр, який використовує лише 1,5 кБ оперативної пам'яті для обробки зображення розміром 256×256 пікселів із 8-бітовою глибиною кольору, що є ефективним рішенням для недорогих сенсорних вузлів. Однак недоліком таких фільтрів є тривалий час обробки.

Сучасні 32-бітні мікроконтролери значно покращують ситуацію. Висока продуктивність і можливість підключення зовнішньої пам'яті дозволяють зменшити час виконання операцій і реалізовувати складніші алгоритми обробки. Наприклад, у роботі [32] запропоновано адаптивне вейвлет-перетворення Хаара для зменшення енергоспоживання та забезпечення високої якості вихідного зображення. Цей підхід також подовжує термін служби акумуляторів, оскільки зменшує кількість втрачених даних під час стиснення.

Для мереж із великою кількістю вузлів, розташованих поблизу, надмірна інформація у вигляді зображень може суттєво збільшити споживання енергії. У роботі [33] запропоновано підхід, що використовує вейвлет-перетворення в комбінації з використанням пріоритетів даних та врахуванням стану каналу зв'язку. Цей метод забезпечує ефективну передачу зображень і збільшує термін служби батареї вузлів. Зображення розкладається на чотири підзображення за допомогою однорівневого вейвлет-перетворення, після чого підзображення передаються через різні канали під контролем кластерного вузла.

Не менш важливим є питання оптимізації використання пам'яті, яка необхідна для виконання вейвлет-перетворення. У роботі [34] описано архітектуру, яка використовує фракційний хвильовий фільтр (FrWF) для зменшення обсягу пам'яті під час обробки зображень. Водночас lifting-based реалізація перетворення, запропонована в роботі [35], демонструє зменшення обчислювальної складності на 40% порівняно з FrWF, що є значною перевагою для високороздільних зображень.

Дослідження, проведене у роботі [36], показало, що використання модернізованого алгоритму стиснення зображень на основі вейвлет-перетворення дозволяє суттєво зменшити розмір вихідного файлу для медичних зображень розміром 256×256 пікселів. При цьому досягнуто значення пікового співвідношення сигнал-шум (PSNR) понад 42 дБ, що свідчить про збереження високої якості зображення після обробки. Однак варто зазначити, що це дослідження обмежене використанням зображень менших розмірів, ніж стандартний кварталний відеографічний масив (QVGA). Це може обмежити

застосування методу для більших зображень або відеоданих, які є поширеними у багатьох сучасних застосунках.

У роботі [37] вивчалася ефективність вейвлету Коіфлет для стиснення зображень такого ж розміру (256×256 пікселів). Дослідники змогли стиснути зображення до 75% від їхнього початкового розміру, досягнувши значення PSNR у 56 дБ, що свідчить про високу якість. Однак алгоритм реалізований без використання мікроконтролерів, що зняло обмеження щодо обчислювальних ресурсів. Це означає, що при застосуванні в системах із низькою продуктивністю ефективність методу може знижуватися.

Дослідження [38] продемонструвало успішне використання двостороннього фільтра разом із вейвлет-перетворенням на основі двоядерних мікроконтролерів Blackfin (Analog Devices, США) для видалення шуму у відеокліпах середнього розміру в реальному часі. Це підкреслює перспективність застосування високопродуктивних мікроконтролерів для обробки відеоданих у реальному часі, але також вказує на високе енергоспоживання та складність реалізації таких алгоритмів у сенсорних мережах.

У роботі [39] досліджено алгоритми стиснення зображень на основі вейвлет-перетворення для мікрокомп'ютера Raspberry Pi 3 Model B. Запропонований підхід дозволив досягти стиснення зображення у 18 разів порівняно з оригіналом. Однак Raspberry Pi належить до класу мікрокомп'ютерів із більшою обчислювальною потужністю, ніж стандартні мікроконтролери, що обмежує можливість прямого перенесення цих результатів на платформи з низькими ресурсами.

У дослідженні, проведеному [40], застосовано вейвлет-перетворення Добеші, Хаару та ортогонального базису для обробки й стиснення відеоданих. Автори дійшли висновку, що використання вейвлет-перетворень значно підвищує ефективність стиснення, дозволяючи зберегти високу якість зображення та суттєво зменшити кількість артефактів. Проте важливо зазначити, що обробка виконувалась без використання мікроконтролерів, що забезпечило більші обчислювальні ресурси та зняло обмеження щодо апаратної

продуктивності. Це робить результати дослідження менш адаптованими до умов систем із низькими обчислювальними можливостями, такими як мікроконтролери в IoT-системах.

У роботі [41] автори створили алгоритм на основі вейвлет-перетворення, який використовувався для обробки зображень з метою аналізу й розпізнавання осіб. Реалізація була виконана за допомогою мови програмування Python, що значно розширило обчислювальні можливості та забезпечило ефективність обробки. Однак відсутність обмежень щодо апаратного забезпечення не дозволяє оцінити ефективність методу у вбудованих системах із низькими ресурсами, зокрема мікроконтролерах. Незважаючи на це, запропонований алгоритм демонструє потенціал для розробки більш адаптивних рішень для систем розпізнавання образів.

Дослідження [42] пропонує новий підхід до алгоритму вейвлет-перетворення. Проведено оцінку ефективності застосування методу, а також порівняння з іншими методами за критерієм середньоквадратичної похибки (MSE). Однак використання лише однієї метрики MSE має свої обмеження, оскільки цей показник не враховує інших важливих факторів, таких як візуальна якість зображення або кореляція між пікселями. Для отримання більш точних результатів доцільно використовувати кілька метрик, таких як PSNR, SSIM або евклідові відстані.

У роботі [43] досліджено застосування вейвлет-перетворення для обробки звукових сигналів із метою фільтрації та стиснення. Автори зазначили, що запропонований підхід є адаптивним і значно підвищує ефективність обробки. Проте дослідження зосереджено лише на мовних сигналах, які мають простішу структуру порівняно із зображеннями. Це вказує на те, що для впровадження таких алгоритмів для обробки зображень потрібні додаткові дослідження та модифікації.

Дослідження [44] пропонує метод сегментарної обробки відео за допомогою вейвлет-перетворення. Зазначено, що групи значень представляються кодами, які займають менший обсяг у бітах, що свідчить про

високу ефективність стиснення. Однак недоліком підходу є високі вимоги до обчислювальних ресурсів, що значно обмежує його застосування в умовах мікроконтролерів. Для зменшення вимог до апаратного забезпечення доцільно використовувати спрощені моделі вейвлет-перетворення або комбіновані алгоритми.

У дослідженні [45] розглянуто використання дворівневого вейвлет-перетворення для обробки даних у режимі реального часу. Результати показали, що зображення стиснене в чотири рази в порівнянні з його оригінальним розміром. Це свідчить про високу ефективність методу для роботи з великими обсягами даних, зокрема хвильових форм, які постійно записуються під час динамічного моніторингу енергосистем. Такий підхід є перспективним для багатьох застосувань у реальному часі. Проте основним обмеженням цього методу є його погана адаптивність до апаратних ресурсів мікроконтролерів, що може обмежувати його використання в енергоефективних системах з обмеженими обчислювальними можливостями.

Робота [46] показала, що значення пікового співвідношення сигнал-шум (PSNR) для оброблених зображень варіюється в межах 29–32 залежно від типу зображення. У дослідженні [47] значення PSNR досягають 38 для вейвлет-перетворення Хаара та 39,2 для перетворення Добеші. Це свідчить про те, що вибір вейвлету значно впливає на якість стиснення, залежно від розміру та типу зображення.

В іншому дослідженні [1] висунуто припущення, що поява артефактів під час обробки зображень за допомогою вейвлет-перетворення Хаара пов'язана з накопиченням похибок на кожному рівні декомпозиції. Це обумовлено малою кількістю коефіцієнтів і їх простими значеннями. Хоча вейвлет Хаара є ефективним для швидкої обробки даних, його недоліком є втрата частини інформації на кожному рівні, що призводить до поступового погіршення якості зображення.

У дослідженні [48] встановлено, що вейвлет-перетворення Добеші забезпечує вищу якість відео порівняно з перетворенням Хаара. Цей висновок

обгрунтовано тим, що перетворення Добеші використовує більше коефіцієнтів і забезпечує кращу деталізацію, тоді як Хаар, хоч і швидший, може втрачати деякі деталі через свою простішу структуру.

У роботі [49] значення пікового співвідношення сигнал-шум (PSNR) для другого рівня декомпозиції вейвлет-перетворення Хаара варіювались у межах від 32,8 до 35,4 залежно від типу зображення. Це свідчить про те, що навіть у межах одного типу перетворення якість стиснення суттєво залежить від структури зображення. Дослідження [50] зосередилось на аналізі якості стиснення для четвертого рівня декомпозиції вейвлет-перетворень Хаара та Добеші. Значення PSNR змінювались у межах від 22,2 до 35,9 залежно від характеристик зображень, причому більші значення PSNR спостерігались для зображень із меншою кількістю деталей. Ці результати підкреслюють, що рівень декомпозиції та тип зображення мають значний вплив на ефективність стиснення та якість відновленого зображення.

В роботі [51] проведено порівняння різних метрик для оцінки схожості зображень і виявлення артефактів. Найефективнішою виявилась метрика, заснована на евклідових відстанях. У роботі [52] показано, що представлення зображень у вигляді евклідових відстаней дозволяє точно визначати відмінності між віддаленими точками, локалізувати ці області та ідентифікувати аномалії. Це підтверджує ефективність використання метрики евклідових відстаней для аналізу оброблених зображень, які часто містять складні артефакти.

На основі цих даних і дослідження [53], у якому показано, що евклідова відстань демонструє високу кореляцію з реальними візуальними артефактами, запропоновано використовувати максимальну евклідову відстань між кольорами пікселів двох зображень однакової розмірності для розрахунку подібності та оцінки артефактів. Такий підхід враховує зміни в кожному кольоровому каналі, що є критичним для оброблених зображень зі складною структурою.

2.2 Теоретичні засади вейвлет-перетворень

Для двовимірного вейвлет-перетворення необхідна двовимірна функція масштабування $\varphi(x, y)$, а також три двовимірні вейвлет-функції $\psi^H(x, y)$, $\psi^V(x, y)$ та $\psi^D(x, y)$. Кожна з них є добутком двох одновимірних функцій. Якщо виключити одновимірні результати, такі як $\varphi(x)$ та $\psi(x)$, то можна записати роздільну функцію як [54],[55]:

$$\varphi(x, y) = \varphi(x)\varphi(y) \quad (1)$$

і роздільні, чутливі до спрямованості вейвлет-функції як:

$$\psi^H(x, y) = \psi(x)\varphi(y) \quad (2)$$

$$\psi^V(x, y) = \varphi(x)\psi(y) \quad (3)$$

$$\psi^D(x, y) = \psi(x)\psi(y) \quad (4)$$

Ці вейвлети описують варіації інтенсивності для зображень у різних напрямках: ψ^H вимірює варіації вздовж стовпців (наприклад, горизонтальних країв), ψ^V реагує на варіації вздовж рядків (наприклад, вертикальних країв) і ψ^D відповідає варіаціям вздовж діагоналей. Чутливість до напрямку наслідком роздільності в рівняннях $\psi^H(x, y) = \psi(x)\varphi(y)$

$$(2) \text{—} \psi^D(x, y) = \psi(x)\psi(y) \quad (4).$$

Враховуючи роздільне двовимірне масштабування та вейвлет-функції, розширення одновимірного дискретного перетворення DWT до двовимірного є простим. Спочатку визначають масштабовані та трансльовані базові функції [54],[55]:

$$\varphi_{j,m,n}(x, y) = 2^{j/2} \varphi(2^j x - m, 2^j y - n) = \frac{1}{\sqrt{2}} [\varphi_{j,2m,n}(x, y) + \varphi_{j,2m+1,n}(x, y)] \quad (5)$$

$$\psi_{j,m,n}^i(x, y) = 2^{j/2} \varphi(2^j x - m, 2^j y - n) = \frac{1}{\sqrt{2}} [\varphi_{j,m,2n}(x, y) - \varphi_{j,m,2n+1}(x, y)] \quad (6)$$

де верхній індекс i визначає спрямовані вейвлети в рівняннях в формулах

$$\psi^H(x, y) = \psi(x)\varphi(y) \quad (2) \quad \psi^D(x, y) = \psi(x)\psi(y)$$

(4) і приймає значення H , V та D . Тоді дискретне вейвлет-перетворення функції зображення $f(x, y)$ з розміром $M \times N$ наступне [54],[55]:

$$W_\varphi(j_0, m, n) = \frac{1}{\sqrt{MN}} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) \varphi_{j_0, m, n}(x, y) \quad (7)$$

$$W_\psi^i(j, m, n) = \frac{1}{\sqrt{MN}} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) \psi_{j, m, n}^i(x, y) \quad (8)$$

де j_0 – довільний початковий масштаб, $W_\varphi(j_0, m, n)$ – коефіцієнти апроксимації $f(x, y)$ з масштабом j_0 . $W_\psi^i(j, m, n)$ – коефіцієнти деталізації з масштабом j для напрямку i , де i може бути горизонтальним (H), вертикальним (V) або діагональним (D); M і N – розміри зображення по осях x та y відповідно. $\varphi_{j_0, m, n}(x, y)$ – функція масштабування з початковим значенням j_0 ; $\psi_{j, m, n}^i(x, y)$ – вейвлет-функція на рівні ітерації j ;

Таким чином, перша формула визначає коефіцієнти апроксимації сигналу на певному масштабі, а друга формула визначає коефіцієнти деталізації сигналу на різних напрямках (горизонтальному, вертикальному і діагональному) на різних масштабах.

Подібно до одновимірного дискретного вейвлет-перетворення, 2-D DWT можна реалізувати за допомогою цифрових фільтрів і понижувальних пристроїв. За допомогою роздільних двовимірних функцій масштабування та вейвлет-функцій розглядається 1-D FWT отриманих рядків $f(x, y)$, за якими слідує 1-D FWT отриманих стовпців. На рис. Рисунок 5 – Алгоритм покрокового вейвлет перетворення Хаара [54],[55]. показано процес у вигляді блок-схеми. Зверніть увагу, що 2-D FWT відфільтровує коефіцієнти $j + 1$ масштабної апроксимації для

побудови масштабної апроксимації j та коефіцієнтів деталізації. Однак у двовимірному випадку отримано три набори коефіцієнтів деталізації — горизонтальні, вертикальні та діагональні деталі.

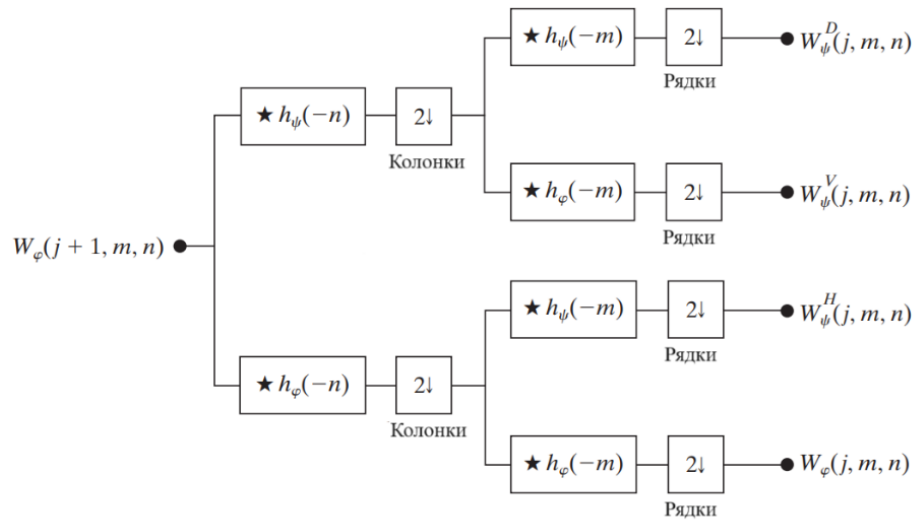


Рисунок 5 – Алгоритм покрокового вейвлет перетворення Хаара [54],[55].

Одновимірний блок фільтрів на рис. 1 можна повторити (пов'язавши результат наближення до входу іншого блоку фільтрів), щоб створити P масштабне перетворення, у якому масштаб j дорівнює $J-1, J-2, \dots, J-P$. Як і в одновимірному випадку, зображення $f(x,y)$ використовується як $W_\phi(J,m,n)$ вхід. Згортаючи його рядки за допомогою $h_\phi(-n)$ і $h_\psi()$ і зменшуючи його стовпці, отримано два підзображення, горизонтальна роздільна здатність яких зменшується в 2 рази.

Компонент високої частоти або деталізації характеризують високочастотну інформацію зображення з вертикальною орієнтацією; низькочастотна компонента наближення містить свою низькочастотну вертикальну інформацію. Потім обидва підзображення фільтруються по стовпцях і зі зниженою дискретизацією, щоб отримати чотири вихідні допоміжні зображення розміром у чверть — W_ϕ , W_ψ^H , W_ψ^V і W_ψ^D . Ці підзображення, які показано на середині рис. 1, є внутрішніми продуктами $f(x, y)$ і двовимірного масштабування та вейвлет-функцій у рівняннях (1)–(4), після чого відбувається зменшення дискретизації на два в кожному вимірі. Дві ітерації процесу

фільтрації дають дворівневе розкладання в крайньому правому куті рис. Рисунок 6 – Рівні декомпозиції зображення [54],[55] [57].

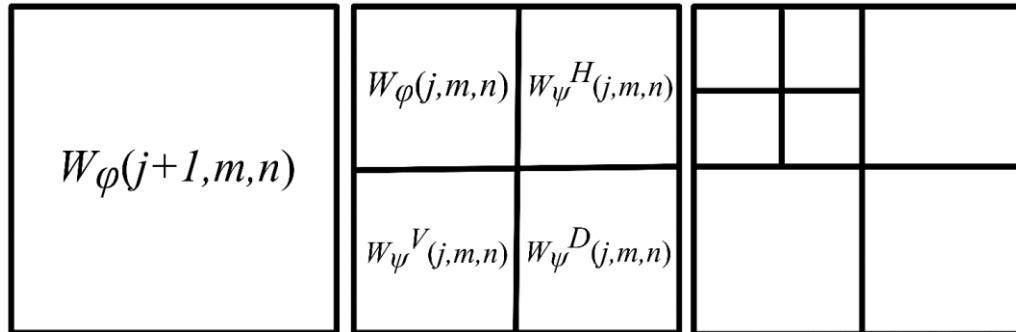


Рисунок 6 – Рівні декомпозиції зображення [54],[55]

Враховуючи W_φ і W_ψ^i з рівнянь

$$W_\varphi(j_0, m, n) = \frac{1}{\sqrt{MN}} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) \varphi_{j_0, m, n}(x, y)$$

$$(7) \text{ і } W_\psi^i(j, m, n) = \frac{1}{\sqrt{MN}} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) \psi_{j, m, n}^i(x, y) \quad (8),$$

$f(x, y)$ можна отримати за допомогою зворотного вейвлет-перетворення [54],[55]:

$$f(x, y) = \frac{1}{\sqrt{MN}} \sum_m \sum_n W_\varphi(j_0, m, n) \varphi_{j_0, m, n}(x, y) + \frac{1}{\sqrt{MN}} \sum_{i=H, V, D} \sum_{j=j_0}^{\infty} \sum_m \sum_n W_\psi^i(j, m, n) \psi_{j, m, n}^i(x, y) \quad (9)$$

Для перетворення обрано вейвлет Хаара, оскільки його алгоритм простий у реалізації, складається з простих арифметичних операцій та має низькі вимоги до використання пам'яті. Це є важливим під час його застосування для мікроконтролерів, які мають обмежені ресурси.

На рис. Рисунок 7 – Алгоритм покрокового інверсного вейвлет перетворення Хаара [54],[55]. показано блок фільтрів синтезу, який змінює щойно описаний процес. Під час кожної ітерації чотири підзображення j і деталізовані підзображення підвищують свою якість та згортаються за допомогою двох одновимірних фільтрів — один працює зі стовпцями підзображень, а інший — із його рядками. Додавання результатів дає $j + 1$

збільшення масштабу, і процес повторюється, доки вихідне зображення не буде реконструйовано [56].

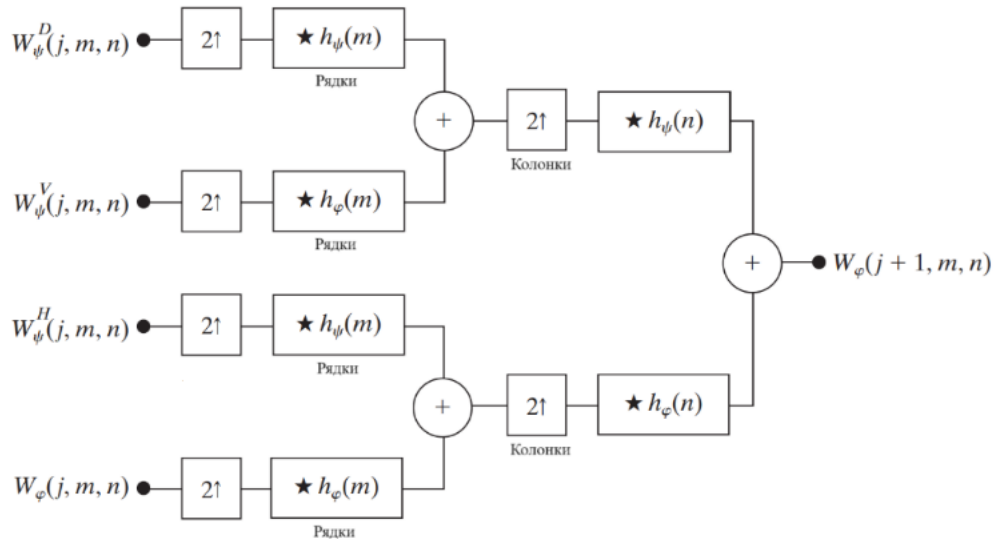


Рисунок 7 – Алгоритм покрокового інверсного вейвлет перетворення Хаара [54],[55].

Рисунок Рисунок 8 – Обчислення двовимірного трирівневого швидкого вейвлет-перетворення (FWT), (а) початкове зображення; (b) одношкальне FWT; (c) двошкальне FWT; (d) трирівневе FWT [54],[55].(а) демонструє комп'ютерно-генероване зображення, яке складається з двовимірних синусоподібних імпульсів на чорному фоні. Метою цього прикладу є пояснення механізму обчислення двовимірного швидкого вейвлет-перетворення (2D FWT) для заданого зображення. На рисунках Рисунок 8 – Обчислення двовимірного трирівневого швидкого вейвлет-перетворення (FWT), (а) початкове зображення; (b) одношкальне FWT; (c) двошкальне FWT; (d) трирівневе FWT [54],[55].(b)– Рисунок 8 – Обчислення двовимірного трирівневого швидкого вейвлет-перетворення (FWT), (а) початкове зображення; (b) одношкальне FWT; (c) двошкальне FWT; (d) трирівневе FWT [54],[55].(d) представлені результати трьох етапів 2D FWT для зображення з рисунка Рисунок 8 – Обчислення двовимірного трирівневого швидкого вейвлет-перетворення (FWT), (а) початкове зображення; (b) одношкальне FWT; (c) двошкальне FWT; (d)

трирівневе FWT [54],[55].(a). Для їх отримання використовувалася дворівнева фільтрувальна схема та фільтри декомпозиції, наведені на рисунках Рисунок 9 – Симлети четвертого порядку: (a)–(b) фільтри декомпозиції; (c)–(d) фільтри реконструкції; (e) одномірний вейвлет; (f) одномірна функція масштабування; (g) один із трьох двовимірних вейвлетів, $\psi^V(x, y)$ [54],[55](a) і Рисунок 9 – Симлети четвертого порядку: (a)–(b) фільтри декомпозиції; (c)–(d) фільтри реконструкції; (e) одномірний вейвлет; (f) одномірна функція масштабування; (g) один із трьох двовимірних вейвлетів, $\psi^V(x, y)$ [54],[55](b).

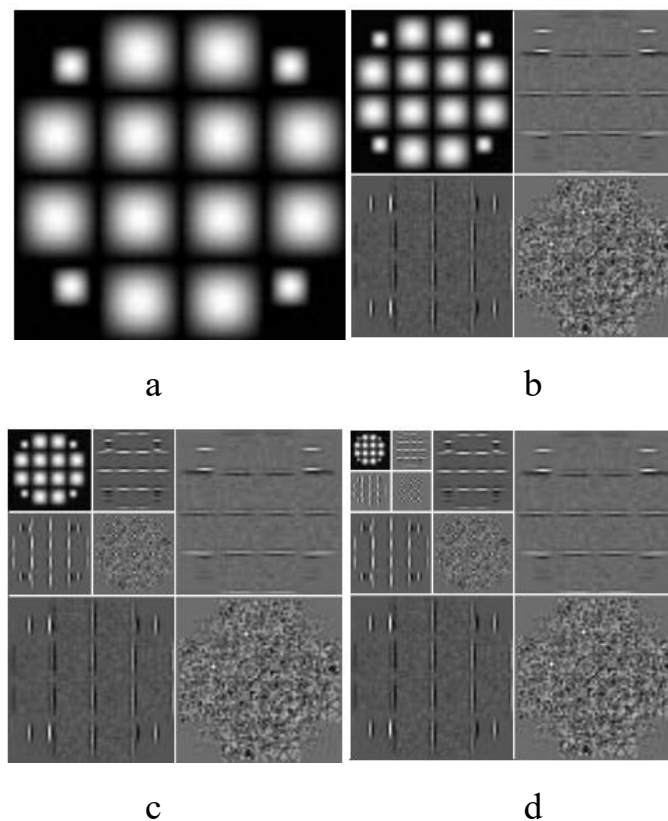
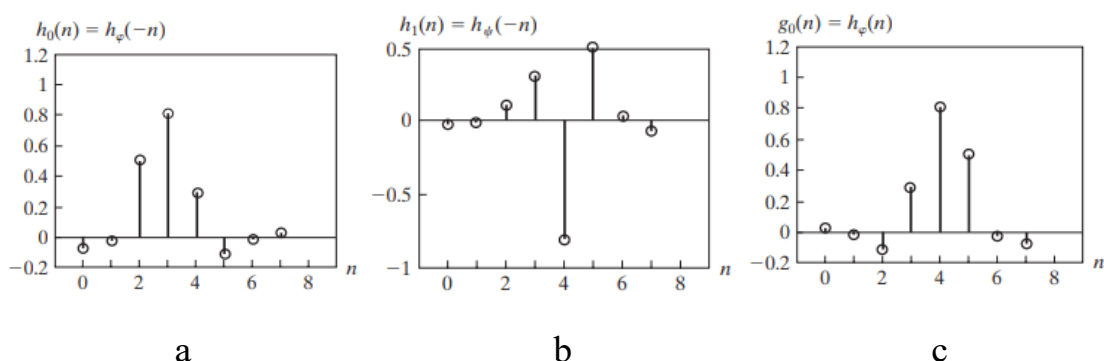


Рисунок 8 – Обчислення двовимірного трирівневого швидкого вейвлет-перетворення (FWT), (a) початкове зображення; (b) одношкальне FWT; (c) двошкальне FWT; (d) трирівневе FWT [54],[55].

Рисунок Рисунок 8 – Обчислення двовимірного трирівневого швидкого вейвлет-перетворення (FWT), (a) початкове зображення; (b) одношкальне FWT; (c) двошкальне FWT; (d) трирівневе FWT [54],[55].(b) показує результат одноетапного 2D FWT для зображення з рисунка Рисунок 8 – Обчислення

двовимірного трирівневого швидкого вейвлет-перетворення (FWT), (a) початкове зображення; (b) одношкільне FWT; (c) двошкільне FWT; (d) трирівневе FWT [54],[55].(a). На першому етапі вихідне зображення подавалося на вхід фільтрувальної схеми. Чотири вихідні зображення, зменшені у чверть розміру (відповідно до апроксимації та горизонтальних, вертикальних і діагональних деталей), були зібрані в єдину композицію, утворюючи результат, представлений на рисунку Рисунок 8 – Обчислення двовимірного трирівневого швидкого вейвлет-перетворення (FWT), (a) початкове зображення; (b) одношкільне FWT; (c) двошкільне FWT; (d) трирівневе FWT [54],[55].(b).

На другому етапі (рисунку Рисунок 8 – Обчислення двовимірного трирівневого швидкого вейвлет-перетворення (FWT), (a) початкове зображення; (b) одношкільне FWT; (c) двошкільне FWT; (d) трирівневе FWT [54],[55].(c)) використовувалася підзона з верхнього лівого кута рисунку Рисунок 8 – Обчислення двовимірного трирівневого швидкого вейвлет-перетворення (FWT), (a) початкове зображення; (b) одношкільне FWT; (c) двошкільне FWT; (d) трирівневе FWT [54],[55].(b), яка є результатом апроксимації. Ця підзона, яка мала чверть початкового розміру, була замінена новими зображеннями, отриманими в результаті другого проходу фільтрації. В результаті розмір кожного зображення становив уже 1/16 від розміру початкового зображення.



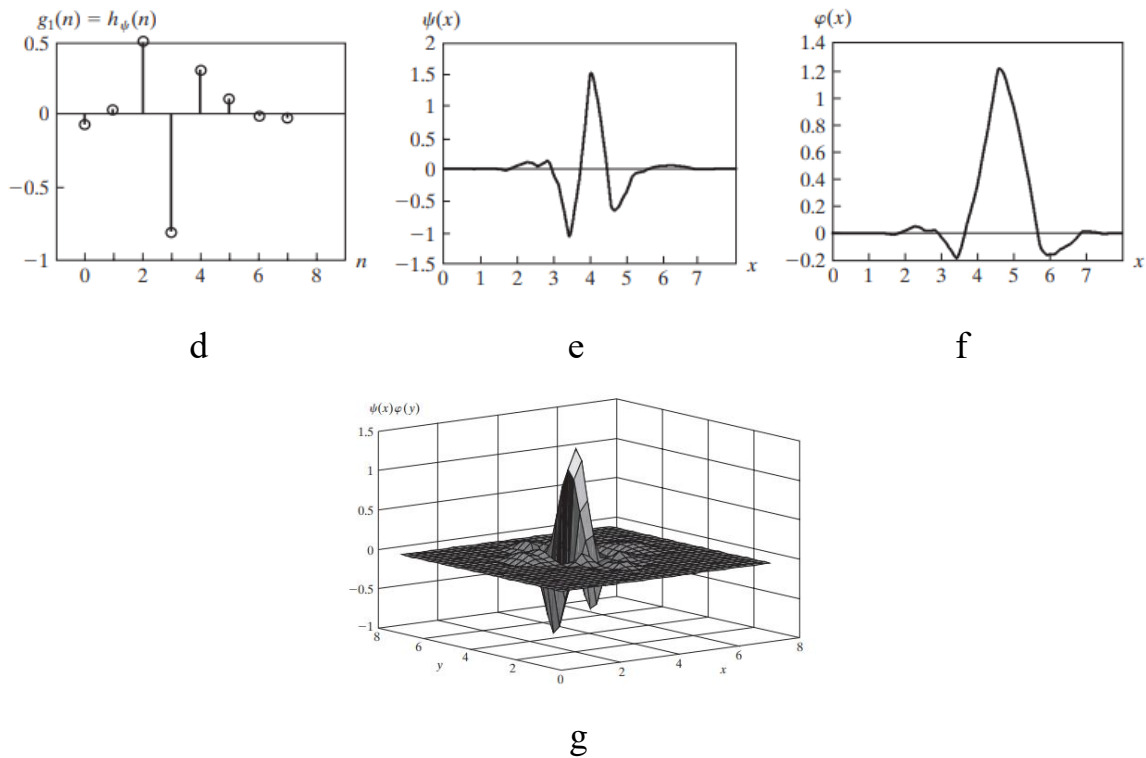


Рисунок 9 – Симлети четвертого порядку: (а)–(б) фільтри декомпозиції; (с)–(д) фільтри реконструкції; (е) одномірний вейвлет; (ф) одномірна функція масштабування; (г) один із трьох двовимірних вейвлетів, $\psi^V(x, y)$ [54],[55]

Рисунок 8 – Обчислення двовимірного трирівневого швидкого вейвлет-перетворення (FWT), (а) початкове зображення; (б) одношкальне FWT; (с) двошкальне FWT; (д) трирівневе FWT [54],[55].(д) демонструє три рівня 2D FWT, де підзона з верхнього лівого кута рисунка Рисунок 8 – Обчислення двовимірного трирівневого швидкого вейвлет-перетворення (FWT), (а) початкове зображення; (б) одношкальне FWT; (с) двошкальне FWT; (д) трирівневе FWT [54],[55].(с) була подана на вхід фільтрувальної схеми. Кожен наступний етап фільтрації генерував чотири вихідні зображення розміром у чверть від попереднього, які заміняли вхідне зображення. Слід звернути увагу на напрямкові властивості підзображень, які на кожному рівні детально показують горизонтальні, вертикальні та діагональні компоненти.

Для проведення цієї декомпозиції використовувалися симлети — клас вейвлетів, що мають високу кількість нульових моментів при мінімальній асиметрії. Їх основні властивості були описані Добеші у 1992 році. На рисунках

Рисунок 9 – Симлети четвертого порядку: (a)–(b) фільтри декомпозиції; (c)–(d) фільтри реконструкції; (e) одномірний вейвлет; (f) одномірна функція масштабування; (g) один із трьох двовимірних вейвлетів, $\psi^V(x, y)$ [54],[55](e) та Рисунок 9 – Симлети четвертого порядку: (a)–(b) фільтри декомпозиції; (c)–(d) фільтри реконструкції; (e) одномірний вейвлет; (f) одномірна функція масштабування; (g) один із трьох двовимірних вейвлетів, $\psi^V(x, y)$ [54],[55](f) представлені функції масштабування та вейвлети для симлетів четвертого порядку. Відповідні фільтри декомпозиції та реконструкції зображені на рисунках Рисунок 9 – Симлети четвертого порядку: (a)–(b) фільтри декомпозиції; (c)–(d) фільтри реконструкції; (e) одномірний вейвлет; (f) одномірна функція масштабування; (g) один із трьох двовимірних вейвлетів, $\psi^V(x, y)$ [54],[55](a)– Рисунок 9 – Симлети четвертого порядку: (a)–(b) фільтри декомпозиції; (c)–(d) фільтри реконструкції; (e) одномірний вейвлет; (f) одномірна функція масштабування; (g) один із трьох двовимірних вейвлетів, $\psi^V(x, y)$ [54],[55](d), а їх коефіцієнти для низькочастотного фільтра реконструкції наведені на рисунку 6.

На рисунку Рисунок 9 – Симлети четвертого порядку: (a)–(b) фільтри декомпозиції; (c)–(d) фільтри реконструкції; (e) одномірний вейвлет; (f) одномірна функція масштабування; (g) один із трьох двовимірних вейвлетів, $\psi^V(x, y)$ [54],[55](g) схематично показано, як одновимірні функція масштабування та вейвлет можуть бути об'єднані для створення роздільного двовимірного вейвлета.

n	$h_{\phi}(n)$
0	0.0322
1	-0.0126
2	-0.0992
3	0.2979
4	0.8037
5	0.4976
6	-0.0296
7	-0.0758

Рисунок 10 – Ортонормовані коефіцієнти фільтра симлетів четвертого порядку для $h_{\phi}(n)$ [54],[55]

Завершуючи цей приклад, розглянуті значення, які демонструють практичну цінність вейвлетів в обробці зображень. Основний підхід включає три кроки:

1. Обчислення 2D вейвлет-перетворення зображення.
2. Модифікація отриманого перетворення.
3. Застосування зворотного перетворення.

Оскільки вейвлет-вектори для масштабування та вейвлет-функцій використовуються як низько- та високочастотні фільтри, більшість фільтраційних технік, базованих на перетворенні Фур'є, мають еквівалентні підходи у вейвлет-доміні [56].

Можливість змінювати роздільність сигналу дає змогу обробляти лише ті деталі, що дійсно важливі для конкретної задачі. У галузі комп'ютерного зору Берт і Адельсон [57] запропонували багаторівневу піраміду, яка спершу опрацьовує зображення з низькою роздільністю, а потім за потреби поступово підвищує її. Цей розділ систематизує підхід до багатороздільних апроксимацій, що закладає основу для побудови ортогональних вейвлетів.

Щоб апроксимувати функцію f на рівні роздільності 2^j , використовують дискретну сітку вибірок, які визначають локальні середні значення f у ділянках із розміром, пропорційним 2^j . Таким чином, багатороздільна апроксимація вибудовується як вкладені одна в одну сітки зростаючої точності. Формально, апроксимацію функції на рівні 2^j можна розуміти як ортогональну проекцію на

простір $V_j \subset L^2(R)$. Простір V_j об'єднує всі можливі апроксимації при роздільності 2^j . Ортогональною проекцією $f \in V_j$ функція $f_j \in V_j$, яка мінімізує норму $\|f - f_j\|$. Визначення, введені Мала та Месром [58],[59], чітко формулюють математичні властивості таких багатороздільних просторів. Звернімо увагу, що параметр масштабу 2^j є оберненим до роздільності 2^{-j} , аби уникнути плутанини. Послідовність замкнених підпросторів $\{V_j\}_{j \in \mathbb{Z}}$ простору $L^2(R)$ називається багатороздільною апроксимацією, якщо виконуються такі шість умов:

$$\forall (j,k) \in \mathbb{Z}^2, f(t) \in V_j \Leftrightarrow f(t - 2^j k) \in V_j \quad (10)$$

$$\forall j \in \mathbb{Z}, V_{j+1} \subset V_j \quad (11)$$

$$\forall j \in \mathbb{Z}, (t) \in V_j \Leftrightarrow f\left(\frac{t}{2}\right) \in V_{j+1} \quad (12)$$

$$\lim_{j \rightarrow +\infty} V_j = \bigcap_{j=-\infty}^{+\infty} V_j = \{0\} \quad (13)$$

$$\lim_{j \rightarrow +\infty} V_j = \overline{\bigcup_{j=-\infty}^{+\infty} V_j} = L^2(R) \quad (14)$$

Існує така функція θ , що множина $\{\theta(t - n)\}_{n \in \mathbb{Z}}$ утворює базис Ріса для простору V_0 . Щоб дати інтуїтивне пояснення цих математичних властивостей, розглянемо їх по черзі.

$$\text{Властивість } \forall (j,k) \in \mathbb{Z}^2, f(t) \in V_j \Leftrightarrow f(t - 2^j k) \in V_j \quad (10)$$

означає, що простір V_j інваріантний відносно будь-якого зсуву на величину, пропорційну масштабу 2^j . Згодом побачимо, що цей простір можна інтерпретувати як рівномірну сітку із кроком 2^j , яка відповідає апроксимації сигналу з роздільністю $2^{\{-j\}}$.

$$\text{Включення } \forall j \in \mathbb{Z}, V_{j+1} \subset V_j \quad (11) \quad \text{відображає}$$

причинно-наслідкову структуру: апроксимація на рівні роздільності $2^{\{-j\}}$ містить усю необхідну інформацію для отримання апроксимації на грубішому рівні 2^{-j-1} .

Розтяг функцій у V_j у 2 рази (взяття аргументу $t/2$) збільшує деталізацію

$$\forall j \in \mathbb{Z}, (t) \in V_j \Leftrightarrow f\left(\frac{t}{2}\right) \in V_{j+1} \quad (12)$$

вдвічі. Властивість

гарантує, що такий розтяг визначає апроксимацію на грубішому рівні $2^{\{-j-1\}}$.

Коли роздільність $2^{\{-j\}}$ прямує до 0, з $\lim_{j \rightarrow +\infty} V_j = \bigcap_{j=-\infty}^{+\infty} V_j = \{0\}$

(13) впливає, що втрачено всі деталі функції f , і:

$$\lim_{j \rightarrow +\infty} \|P_{V_j} f\| = 0 \quad (15)$$

З іншого боку, коли роздільність $2^{\{-j\}}$ прямує до нескінченності, властивість $\lim_{j \rightarrow +\infty} V_j = \overline{\bigcup_{j=-\infty}^{+\infty} V_j} = L^2(R)$ (14) вимагає, щоб апроксимація сигналу збігалася до самого сигналу f :

$$\lim_{j \rightarrow -\infty} \|f - P_{V_j} f\| = 0 \quad (16)$$

При збільшенні роздільності $2^{\{-j\}}$ (тобто при зменшенні j) швидкість спадання похибки апроксимації $\|f - P_{V_j} f\|$ залежить від регулярності функції f . Цю похибку можна пов'язати з рівномірною ліпшицевою регулярністю f .

Вейвлети Добеші, Хаара та Коїфлет мають різну структуру та властивості, однак їх об'єднує загальний підхід до обчислення коефіцієнтів, який базується на розкладанні сигналу на низькочастотні (апроксимаційні) та високочастотні (детальні) компоненти. Нижче описано алгоритм обчислення коефіцієнтів для кожного вейвлета та їх застосування для стиснення й відновлення зображень.

Вейвлет Хаара є найпростішим із вейвлетів, що використовує базову математичну операцію згортки сигналу із фільтрами. Основна властивість вейвлета Хаара полягає у його симетричності та використанні тільки двох коефіцієнтів для фільтрації. Базис Хаара отримують за допомогою

багатороздільної апроксимації функцій, сталих на часткових інтервалах. Масштабна функція визначається як $\phi = 1_{[0,1]}$. Фільтр $h[n]$ має дві ненульові вагові коефіцієнти, рівні $2^{-1/2}$, при $n=0$ та $n=1$. Отже:

$$\frac{1}{\sqrt{2}}\psi\left(\frac{t}{2}\right) = \sum_{n=-\infty}^{+\infty} (-1)^{1-n} h[1-n]\phi(t-n) = \frac{1}{\sqrt{2}}(\phi(t-1) - \phi(t)) \quad (17)$$

Отже низькочастотний фільтр $h[k] = \begin{cases} \frac{1}{\sqrt{2}}, k = 0, 1 \\ 0, \text{ в іншому випадку} \end{cases}$

Високочастотний фільтр $g[k] = \begin{cases} \frac{1}{\sqrt{2}}, k = 0 \\ -\frac{1}{\sqrt{2}}, k = 1 \\ 0, \text{ в іншому випадку} \end{cases}$

Обчислення коефіцієнтів здійснюється для низькочастотних та високочастотних компонентів. Формули для обчислення наведено нижче:

$$a[k] = \sum_n f[n]h[2k-n] \quad (18)$$

$$d[k] = \sum_n f[n]g[2k-n] \quad (19)$$

де $f[n]$ — вхідний сигнал.

Вейвлет Хаара дозволяє швидко розкласти зображення на основні складові:

1. Кодування: Зображення розділяється на матриці низькочастотних і високочастотних коефіцієнтів. Низькочастотна частина містить загальну структуру зображення, а високочастотна — деталі (контури, текстури).

2. Декодування: Зворотна процедура виконується через згортку з фільтрами відновлення.

Вейвлет Хаара має найкоротший носій серед усіх ортогональних вейвлетів. Однак він не надто придатний для апроксимації гладких функцій, оскільки має лише один зникаючий момент.

Вейвлети Добеші є складнішими, оскільки вони забезпечують більшу гладкість і кількість нульових моментів. Наприклад, для Daubechies-4 ($M=4$):

$$\text{Низькочастотний фільтр } (h[k]): \quad h[0] = \frac{1+\sqrt{3}}{4\sqrt{2}}, h[1] = \frac{3+\sqrt{3}}{4\sqrt{2}}, h[2] = \frac{3-\sqrt{3}}{4\sqrt{2}}, \\ h[3] = \frac{1-\sqrt{3}}{4\sqrt{2}}$$

$$\text{Високочастотний фільтр } (g[k]): \quad g[k] = (-1)^k h[M-1-k]$$

Вейвлети Добеші досягають мінімального розміру носія для заданої кількості зникаючих моментів p . Вейвлети з компактним носієм можна одержати, використовуючи спряжені дзеркальні фільтри h із кінцевою імпульсною характеристикою. Розглянемо дійсні причинно-зумовлені фільтри $h[n]$, що робить $\hat{h}(\omega)$ тригонометричним поліномом:

$$\hat{h}(\omega) = \sum_{n=0}^{N-1} h[n] e^{-in\omega} \quad (20)$$

Щоби вейвлет ψ мав p зникаючих моментів, вимагається наявність нуля порядку p у \hat{h} при $\omega=\pi$. Для побудови тригонометричного полінома мінімального розміру, що має p нулів у точці $\omega=\pi$, факторизуємо $(1+e^{-i\omega})$, оскільки $(1+e^{-i\omega})^p$ – це найпростіший варіант полінома з p нулями. Таким чином:

$$\hat{h}(\omega) = \sqrt{2} \left(\frac{1+e^{-i\omega}}{2} \right)^p R(e^{-i\omega}) \quad (21)$$

Основна складність полягає в побудові полінома $R(e^{-i\omega})$ мінімального ступеня m , щоб \hat{h} задовольняв умову:

$$|\hat{h}(\omega)|^2 + |\hat{h}(\omega + \pi)|^2 = 2 \quad (22)$$

Як наслідок, h має $N = m + p + 1$ ненульових коефіцієнтів. Теорема Добеші стверджує, що мінімальний степінь R дорівнює $m=p-1$.

Конструктивний доказ цієї теореми дозволяє побудувати причинні спряжені дзеркальні фільтри довжини $2p$. У Таблиці Таблиця 1 –Значення фільтрів для $p = 2,3$ зникаючих моментів та Таблиця 2–Значення фільтрів для $p = 4,5$ зникаючих моментів наведено коефіцієнти фільтрів Добеші для $2 \leq p \leq 5$. Наступна пропозиція показує, що вейвлети Добеші, отримані з цих спряжених дзеркальних фільтрів, мають носій мінімального розміру.

Нижче подано таблицю фільтрів Добеші для вейвлетів із p зникаючими моментами.

Таблиця 1 –Значення фільтрів для $p = 2,3$ зникаючих моментів

n	h₂[n]	n	h₃[n]
0	0.482962913145	0	0.332670552950
1	0.836516303738	1	0.806891509311
2	0.224143868042	2	0.459877502218
3	-0.129409522551	3	-0.135011020010
		4	-0.085441273882
		5	0.035226291882

Таблиця 2–Значення фільтрів для $p = 4,5$ зникаючих моментів

n	h₄[n]	n	h₅[n]
0	0.230377813309	0	0.160102397974
1	0.714846570553	1	0.603829269797
2	0.630880767930	2	0.724308528438
3	-0.027983769417	3	0.138428145901
4	-0.187034811719	4	-0.242294887066
5	0.030841381836	5	-0.032244869585

6	0.032883011667	6	0.077571493840
7	-0.010597401785	7	-0.006241490213
		8	-0.012580751999
		9	0.003335725285

При $p=1$ отримуємо вейвлет Хаара. На Рисунку Рисунок 11 – Функція масштабування Добеші ϕ та вейвлет ψ з p зникаючими моментами наведено графіки ϕ та ψ для $p=2,3,4$.

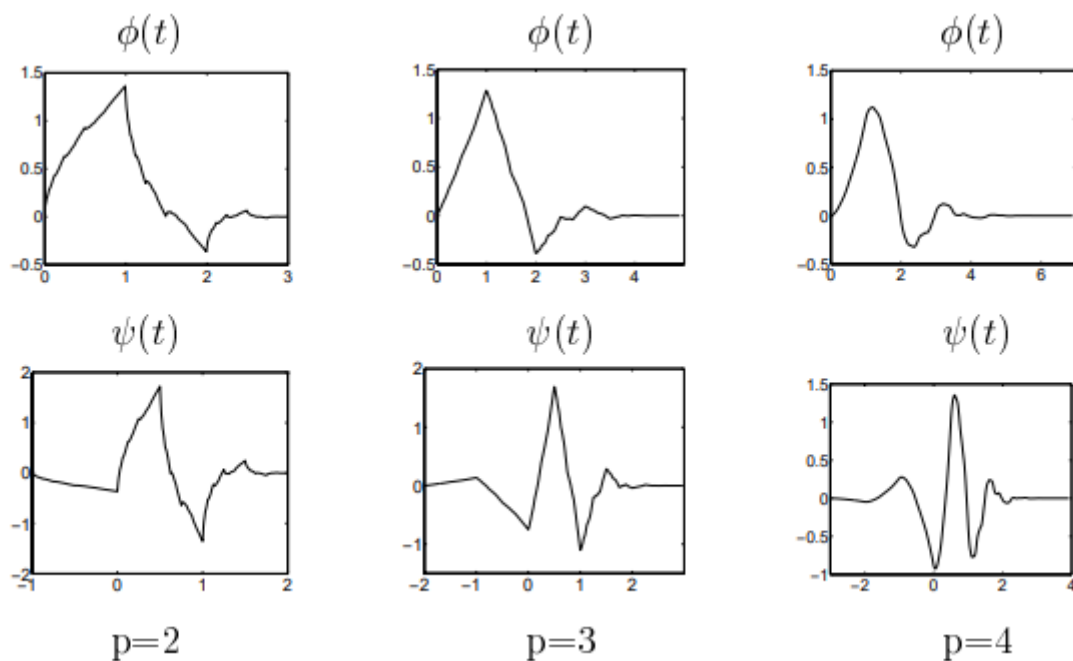


Рисунок 11 – Функція масштабування Добеші ϕ та вейвлет ψ з p зникаючими моментами

Використання для зображень

1. Кодування: Завдяки більшим нульовим моментам, вейвлети Добеші краще стискають зображення, зберігаючи його загальну структуру.

2. Декодування: Використання багаторівневої декомпозиції дозволяє точно відновити зображення з мінімальними спотвореннями.

Коїфлети забезпечують більшу симетрію та точність для аналізу сигналів і зображень.

Даубеші [61] побудовано сімейство вейвлетів ψ із p зникаючими моментами та мінімально можливим розміром носія, при цьому їх масштабна функція ϕ додатково задовольняє умови:

$$\int_{-\infty}^{+\infty} \phi(t) dt = 1 \text{ та } \int_{-\infty}^{+\infty} t^k \phi(t) dt = 0 \text{ для } 1 \leq k < p \quad (23)$$

Такі масштабні функції стають у пригоді для побудови точних квадратурних формул. Якщо функція $f \in C^k$ -гладкою у деякому околі точки $2^J n$ для деякого $k < p$, тоді розклад за масштабними коефіцієнтами $\langle f, \phi_{j,n} \rangle$ приблизно відтворює $f(2^J n)$ з точністю до членів порядку $O(2^{(k+1)J})$. Формально, якщо розглянути дрібномасштабний рівень 2^J , то:

$$2^{-J/2} \langle f, \phi_{j,n} \rangle \approx f(2^J n) + O(2^{(k+1)J}) \quad (24)$$

Зі зменшенням масштабу 2^J (збільшенням J) масштабні коефіцієнти дедалі точніше відтворюють локальні значення сигналу. Зі зростанням p підвищується і порядок апроксимації.

$$\text{Додаткова умова } \int_{-\infty}^{+\infty} \phi(t) dt = 1 \text{ та } \int_{-\infty}^{+\infty} t^k \phi(t) dt = 0 \text{ для } 1 \leq k < p$$

(23) змушує збільшити розмір носія вейвлету ψ . У результаті отримуємо коіфлет, носій якого має довжину $3p-1$, тоді як аналогічний вейвлет Добеші при тих самих p мав би носій довжини $2p-1$.

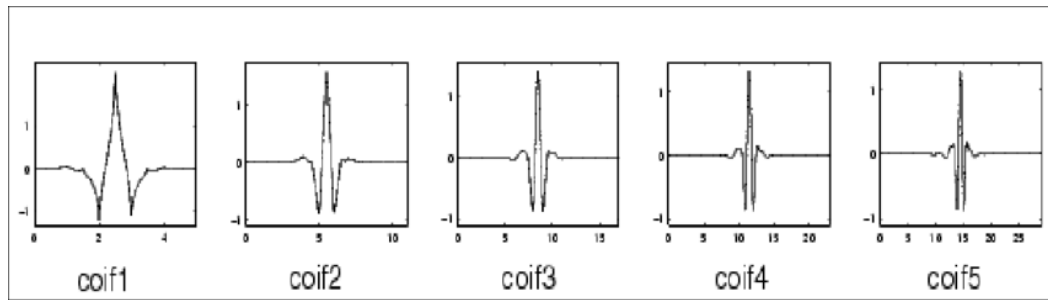


Рисунок 12 – Функції вейвлету Коїфлет порядку 1-5

Наприклад, для Коїфлет-2 ($M=6$) низькочастотний фільтр ($h[k]$):
 $h[0] = -0.015655728135$, $h[1] = -0.072732619512$, $h[2] = 0.384864846864$,
 $h[3] = 0.852572020212$, $h[4] = 0.337897662458$, $h[5] = -0.072732619512$

Високочастотний фільтр ($g[k]$): $g[k] = (-1)^k h[M - 1 - k]$

Таблиця 3 – Значення коефіцієнтів вейвлету Коїфлет [62]

k	C6	C12
-4		0.0231751934774337
-3		-0.0586402759669371
-2	-0.1028594569415370	-0.0952791806220162
-1	0.4778594569415370	0.5460420930695330
0	1.2057189138830700	1.1493647877137300
1	0.5442810861169260	0.5897343873912380
2	-0.1028594569415370	-0.1081712141834230
3	-0.0221405430584631	-0.0840529609215432
4		0.0334888203265590
5		0.0079357672259240
6		-0.0025784067122813
7		-0.0010190107982153

1. Кодування: Коіфлети ідеально підходять для аналізу медичних зображень, де важлива точність і мінімізація спотворень.

2. Декодування: Симетрія вейвлетів Коіфлет дозволяє зменшити артефакти під час реконструкції зображення.

Детальний опис застосування коефіцієнтів для кодування та декодування зображень складається з кодування зображення, кодування значущих коефіцієнтів, багаторівнева декомпозиція. Кодування зображень за допомогою вейвлетів включає такі етапи, як аналіз зображення на основі вейвлетів та стиснення даних.

Розкладання зображення на низькочастотні та високочастотні компоненти відбувається за допомогою низькочастотних коефіцієнтів (апроксимація): Ці коефіцієнти містять основну інформацію про яскравість і загальну структуру зображення. Наприклад, вони описують розмитий вигляд зображення без чітких деталей. Також задіюються високочастотні коефіцієнти (деталі): Вони відповідають за збереження деталей зображення, таких як краї, текстури та тонкі особливості. Зазвичай розкладаються на горизонтальні, вертикальні та діагональні компоненти.

Стиснення даних відбувається шляхом обнуління незначущих високочастотних коефіцієнтів, що має такі особливості, що мають багато високочастотних коефіцієнтів, що мають невеликі значення, що незначно впливають на сприйняття зображення людиною. Їх можна відкинути без суттєвої втрати якості. Також для цього використовується порогове значення, нижче якого всі коефіцієнти обнуляються.

Кодування значущих коефіцієнтів відбувається шляхом відкидання коефіцієнтів, що стискаються за допомогою стандартних методів ентропійного кодування, таких як алгоритм Хаффмана або арифметичне кодування.

Багаторівнева декомпозиція на кожному рівні декомпозиції зображення зменшується за розміром, а його деталі зберігаються у високочастотних коефіцієнтах. Це дозволяє ефективно стискати зображення.

Значення вейвлетів для кодування:

- Вейвлети Добеші добре підходять для компресії завдяки високій кількості нульових моментів, що дозволяє точно передавати основні деталі зображення.
- Вейвлети Хаара забезпечують швидке обчислення, що підходить для задач із обмеженими ресурсами.
- Коїфлети надають високу точність, що критично для аналізу складних зображень, наприклад, у медичній діагностиці.

Декодування зображення виконується через зворотне дискретне вейвлет-перетворення (IDWT). Основні етапи:

1. Відновлення низькочастотної компоненти: низькочастотні коефіцієнти апроксимації комбінуються з деталями з високочастотних компонентів. Для цього використовується згортка з фільтрами відновлення ($h_r[k]$ і $g_r[k]$).
2. Об'єднання деталей, що виражається застосування високочастотних коефіцієнтів з горизонтальних, вертикальних і діагональних компонентів додаються до низькочастотної частини, створюючи повне зображення.
3. Ітеративна реконструкція для багаторівневої декомпозиції повторюється, починаючи з найвищого рівня та поступово переходячи до початкового розміру зображення.

Приклади застосування:

1. JPEG2000: у форматі JPEG2000 використовується багаторівневе вейвлет-перетворення (зазвичай вейвлети Добеші). Низькочастотні коефіцієнти зберігають основну інформацію, а високочастотні стискаються за допомогою порогового обнуління. Це дозволяє досягти високого ступеня стиснення зображень без суттєвої втрати якості.
2. Фільтрація шумів: Застосовується обнуління високочастотних коефіцієнтів, які відповідають за дрібні текстури, що можуть бути спричинені шумами. Відновлення зображення після фільтрації забезпечує чіткість основних деталей.
3. Медичні зображення: Коїфлети забезпечують точне відображення складних структур, таких як кровоносні судини чи пухлини на рентгенівських

або МРТ-зображеннях. Застосовується для компресії медичних даних, що зменшує вимоги до зберігання, зберігаючи діагностичну якість.

Коефіцієнти вейвлетів дозволяють ефективно розділити зображення на частини, що мають різний внесок в його візуальну структуру. Це є ключем до успішного застосування вейвлетів у задачах компресії, фільтрації шумів і аналізу високоточних даних.

2.3 Аналіз наукових праць з використанням вейвлет перетворень Хаара, Добеші та Коіфлет

Евклідова відстань є фундаментальною метрикою, широко застосовуваною в різних галузях для вимірювання подібності або відмінності між об'єктами.

У статті [63] розглянуто реалізацію системи розпізнавання мови для керування рухом інвалідного візка. Для розпізнавання мовних сигналів використовувалися два підходи: лінійне прогнозне кодування в поєднанні з евклідовою квадратною відстанню та метод динамічного програмування. Результати показали, що обидва методи можуть бути реалізовані на мікроконтролері MCS51, проте метод динамічного програмування забезпечує кращу точність розпізнавання.

У дослідженні [64] представлено розробку недорогої вбудованої системи для класифікації швидкозмінних імпульсів на основі електроокулограми (EOG). Система складається з сигнального кондиціонера на основі операційного підсилювача та мікроконтролера Arduino. Для класифікації швидкозмінних імпульсів використовувалася евклідова відстань, що дозволило досягти високої точності розпізнавання.

У роботі [65] розглянуто реалізацію нейронної мережі Кохонена типу "переможець забирає все" на мікроконтролерах з ядрами AVR та ARM CortexM3. Для обчислення відстаней між вхідними векторами та вагами нейронів використовувалася евклідова відстань. Результати лабораторних тестів показали ефективність реалізації нейронної мережі на обох типах мікроконтролерів.

У статті [66] описано систему моніторингу дотримання соціальної дистанції за допомогою алгоритму YOLO для виявлення людей. Система використовує евклідову відстань для обчислення відстаней між людьми в кадрі, що дозволяє виявляти порушення соціальної дистанції. Розроблена система може бути реалізована на базі Raspberry Pi, що робить її доступною для широкого використання.

У дослідженні [67] запропоновано підхід для оцінки багатокрокової відстані в бездротових сенсорних мережах за допомогою жадібного алгоритму. Евклідова відстань використовується для визначення відстані між вузлами мережі, що дозволяє покращити точність оцінки та ефективність маршрутизації.

У роботі [68] розроблено рекомендаційну систему на основі машинного навчання для допомоги вбудованим програмістам у швидкому пошуку та запиті пов'язаних кодових сегментів та інструкцій. Система використовує евклідову відстань для оцінки схожості між кодовими сегментами, що сприяє підвищенню ефективності розробки програмного забезпечення.

У статті [69] розглядаються методи виявлення та запобігання атакам на основі побічних каналів у криптографічних системах. Автори використовують евклідову відстань для аналізу кореляцій між сигналами, що дозволяє ефективно ідентифікувати потенційні вразливості у криптографічних алгоритмах

У дослідженні [70] представлено алгоритми класифікації сигналів у бездротових сенсорних мережах. Евклідова відстань застосовується для вимірювання подібності між векторами ознак, що сприяє підвищенню точності класифікації даних та оптимізації їх передачі у мережах.

У роботі [71] розглянуто методи зниження енергоспоживання в бездротових сенсорних мережах шляхом кластеризації вузлів. Евклідова відстань використовується для формування кластерів та визначення ефективних маршрутів передачі даних, що значно покращує енергоефективність роботи мережі.

У статті [72] описується система моніторингу стану здоров'я пацієнтів за допомогою носимих пристроїв. Евклідова відстань використовується для аналізу

біометричних даних та виявлення аномалій, що дозволяє швидко реагувати на можливі проблеми зі здоров'ям.

У дослідженні [73] запропоновано алгоритм маршрутизації в мобільних мережах, який базується на евклідовій відстані. Метод дозволяє покращити якість зв'язку між вузлами, зменшити затримки та підвищити надійність передачі даних.

У статті [74] аналізується використання евклідової відстані та косинусної подібності у завданнях вибору ознак для моделювання виявлення вторгнень. Результати показали, що косинусна подібність демонструє перевагу у виявленні подібності між векторами в умовах високовимірного простору, оскільки враховує кут між векторами, а не лише їхню довжину. Це робить її корисною для обробки текстових даних чи аналізу мережевого трафіку, де напрямок ознак має велике значення.

Дослідження [75] порівнює евклідову відстань та відстань Манхеттена для класифікації образів у задачах машинного навчання. Відстань Манхеттена показала кращі результати для задач із високовимірними даними та великим рівнем шуму, оскільки враховує лише ортогональні проекції. Евклідова ж відстань, за рахунок обчислення прямих ліній між точками, виявилась менш стійкою до викидів.

У статті [76] досліджується відстань Хаверсина у задачах географічного позиціонування. На відміну від евклідової відстані, яка ігнорує сферичність Землі, відстань Хаверсина враховує кривизну планети, забезпечуючи значно точніші результати для обчислень на глобальних масштабах. Для прикладних завдань GPS та інших систем геолокації евклідова відстань може давати похибки, особливо при великих відстанях.

Робота [77] акцентує увагу на використанні відстані Чебишева, яка визначає найбільшу різницю між координатами векторів. Цей підхід виявився корисним у задачах планування виробництва та логістики, де максимальні відхилення можуть бути критичними для оцінки ризиків чи оптимізації процесів.

У порівнянні з евклідовою відстанню, Чебишевська метрика дозволяє швидше виявляти потенційно небезпечні ситуації.

У статті [78] розглянуто використання евклідової, манхеттенської та чебишевської відстаней для онлайн-верифікації підписів. Автори застосовують методи екстракції ознак у вигляді векторів, щоб порівняти підписи. Результати показали, що манхеттенська відстань забезпечує більш точну верифікацію в умовах, де важлива сума абсолютних різниць між координатами. Евклідова відстань виявилася ефективною для ідеалізованих даних, тоді як чебишевська краще працює з даними, що містять критичні відхилення.

У роботі [79] представлено використання евклідової та манхеттенської відстаней для вирівнювання форм у задачах комп'ютерного зору. Евклідова відстань виявилася ефективною обчислювальною, але менш стійкою до шуму. Манхеттенська метрика показала перевагу у випадках із нерівномірно розподіленими аномаліями, що робить її кориснішою у застосуваннях із сильним шумом або великими даними.

Дослідження [80] зосереджується на порівнянні евклідової та Махаланобісової відстаней для класифікації даних у високовимірному просторі. Махаланобісова відстань враховує кореляцію між ознаками, що дозволяє краще враховувати геометрію простору та знижує похибки класифікації для даних із сильними кореляційними зв'язками. Евклідова ж відстань, хоч і простіша у реалізації, виявилася менш точною в таких умовах.

2.4 Аналіз результатів експерименту для порівняння вейвлет-перетворень Хаара, Добеші та Коіфлет

Для усіх подальших експериментів використовувався однаковий набір апаратного та програмного забезпечення для дослідження ефективності вейвлет перетворення при стисненні зображень.

Для проведення експерименту використано 32-бітні мікроконтролери, що є потужними пристроями, які забезпечують високу продуктивність при роботі з сигналами різного типу. Однак обробка зображень накладає серйозні вимоги до

обсягу оперативної пам'яті, яку мають більшість мікроконтролерів загального призначення. У таких пристроях оперативної пам'яті зазвичай недостатньо для роботи з великими обсягами даних, характерними для обробки зображень. Цю проблему можна вирішити шляхом підключення зовнішньої оперативної пам'яті через послідовний інтерфейс. Проте такий підхід має свої обмеження, оскільки використання зовнішньої пам'яті знижує швидкість обробки відеопотоку, що є важливим аспектом у реальних задачах, таких як передача та обробка відео в режимі реального часу.

Для проведення досліджень обрано модуль ESP32-CAM, який складається з двох основних частин: плати мікроконтролера з вбудованою камерою та пам'яттю і плати з USB-інтерфейсом та колами живлення. Основним елементом цієї системи є мікроконтролер ESP32, який оснащений двоядерним процесором Tensilica Xtensa LX6, працює на тактовій частоті до 240 МГц, і має вбудований модуль Wi-Fi. Завдяки цьому модуль здатен передавати зображення безпосередньо на веб-сторінку, відкриту в браузері, що значно спрощує інтеграцію пристрою у мережеві системи та його використання для передачі даних. Вбудована PSRAM пам'ять об'ємом 4 МБ дозволяє зберігати та обробляти більші обсяги даних, що є критично важливим для роботи з високоякісними зображеннями. Камера модуля підтримує роздільну здатність 2 мегапікселі, що забезпечує достатню якість для багатьох прикладних завдань.

Мікроконтролер ESP32-CAM має ряд переваг у порівнянні з іншими мікроконтролерами, що робить його популярним вибором для багатьох завдань, пов'язаних із обробкою зображень, відеопотоку та бездротової передачі даних. Основними перевагами цього модуля є його висока продуктивність, інтеграція важливих компонентів, а також зручність у використанні для розробників.

По-перше, ESP32-CAM має потужний двоядерний мікроконтролер Tensilica Xtensa LX6 із максимальною тактовою частотою 240 МГц. Це дозволяє йому обробляти складні алгоритми, включаючи обробку зображень, виконання вейвлет-перетворень або інших математично інтенсивних задач. Багато інших

мікроконтролерів, таких як STM32F103 або AVR-мікроконтролери, не мають такої високої обчислювальної потужності.

Другою перевагою ESP32-CAM є вбудована підтримка Wi-Fi та Bluetooth, що дозволяє передавати оброблені дані без використання додаткових модулів. Інші мікроконтролери, наприклад, серії STM32 або AT Mega, зазвичай вимагають підключення зовнішніх модулів, таких як ESP8266 або HC-05, для реалізації бездротового зв'язку. Це ускладнює дизайн пристроїв і збільшує їхню вартість.

Третьою важливою перевагою ESP32-CAM є його інтегрована камера з роздільною здатністю 2 мегапікселі та підтримка PSRAM (4 МБ оперативної пам'яті). Це значно полегшує обробку великих обсягів даних із камери без додаткового обладнання. У той час як інші мікроконтролери потребують зовнішніх модулів камер і оперативної пам'яті, ESP32-CAM пропонує готове рішення "все в одному". Наприклад, для роботи з камерами на STM32 або Arduino доводиться використовувати додаткові компоненти, такі як модулі OV7670 або OV2640, а також зовнішні SD-карти для зберігання даних.

Ще однією перевагою ESP32-CAM є підтримка високоякісного JPG-стиснення завдяки наявності спеціальних бібліотек для обробки зображень. Це дозволяє не тільки передавати зображення ефективно, але й задавати рівень якості стиснення для зменшення розміру переданих файлів. Інші мікроконтролери часто не мають такої гнучкості, вимагаючи складнішої реалізації алгоритмів стиснення.

Додатковою перевагою ESP32-CAM є наявність компактного форм-фактор та низька вартість у порівнянні з аналогами. Завдяки цьому модуль є економічно вигідним рішенням для розробки IoT-пристроїв, які потребують відео- або фото функціональності. Інші мікроконтролери, такі як STM32 з підтримкою камер, зазвичай дорожчі, особливо у конфігураціях із додатковими модулями пам'яті або бездротового зв'язку.

Нарешті, ESP32-CAM має підтримку багатьох інструментів розробки, таких як Arduino IDE, ESP-IDF, та інші програмні платформи. Це забезпечує

простоту інтеграції та розробки навіть для початківців. У той час як для більшості інших мікроконтролерів може знадобитися спеціалізоване програмне забезпечення або складніша настройка, ESP32-CAM є надзвичайно зручним у використанні.

Загалом, основні переваги ESP32-CAM у порівнянні з іншими мікроконтролерами включають високу продуктивність, інтеграцію Wi-Fi і камери, підтримку PSRAM, економічність і зручність використання. Ці характеристики роблять його ідеальним вибором для проектів, пов'язаних із обробкою зображень, передачею даних і побудовою IoT-рішень.

Зовнішній вигляд модулю ESP-CAM з підключеною камерою наведено на рис. Рисунок 13 – Зовнішній вигляд модулю ESP32-CAM з камерою та платою USB інтерфейсу.



Рисунок 13 – Зовнішній вигляд модулю ESP32-CAM з камерою та платою USB інтерфейсу

Обраний модуль ESP32-CAM продемонстрував свою ефективність у реалізації задач із обробки зображень. На рис. Рисунок 13 – Зовнішній вигляд модулю ESP32-CAM з камерою та платою USB інтерфейсу показано зовнішній вигляд модуля з підключеною камерою та USB-інтерфейсом. Завдяки своїм характеристикам, мікроконтролер на базі ESP32 може використовуватись у задачах, які вимагають обробки зображень і передачі результатів через бездротові мережі. Наявність Wi-Fi відкриває широкі можливості для

застосування, наприклад, у системах відеоспостереження, моніторингу або аналізу зображень в умовах обмежених ресурсів.

Для оцінки можливостей вейвлет-перетворення в обробці зображень застосовано дискретне вейвлет-перетворення Хаара. Цей метод відомий своєю простотою та ефективністю, що робить його ідеальним вибором для мікроконтролерів із обмеженими ресурсами. Основною метою експериментів визначити, наскільки ефективно вейвлет-перетворення Хаара може зменшити розмір зображення, збереженого у форматі JPG, без значної втрати якості. В ході експериментів порівняно два варіанти: зображення, стиснене лише за допомогою JPEG, і зображення, до якого перед стисненням застосовувалося вейвлет-перетворення. Цей підхід дозволяє зменшити розмір зображення, що, у свою чергу, скорочує час передачі зображення від мікроконтролера до інших пристроїв.

Для спрощення та автоматизації процесу порівняння створено веб-інтерфейс у вигляді HTML-сторінки, яка зберігалася у пам'яті мікроконтролера. Через цей інтерфейс користувач міг переглядати оброблене зображення, отримувати інформацію про його розмір, а також задавати кількість ітерацій вейвлет-перетворення і рівень якості зображення у форматі JPG. Конвертація зображення у формат JPG виконувалася за допомогою бібліотеки обробки зображень, яка підтримує різноманітні формати вхідних даних і дозволяє налаштовувати рівень якості вихідного файлу.

Перевагою бібліотеки для ESP32 є її універсальність, яка забезпечує сумісність із багатьма форматами вхідних даних, і можливість роботи з різними параметрами якості зображень. Для оцінки якості обробки зображення з використанням методів JPEG + WDT виконувалося інверсне перетворення отриманих даних на комп'ютері. Відновлене зображення виводилося на екран, що дозволяло провести детальний аналіз впливу вейвлет-перетворення на якість зображення після стиснення. Такий підхід продемонстрував ефективність комбінування вейвлет-перетворення Хаара з JPEG-стисненням для зменшення розміру зображень без значної втрати їхньої якості.

У процесі розробки системи стиснення зображень на мікроконтролері ESP32 були використані вейвлет-перетворення Хаара, Добеші та Коіфлета для попередньої обробки зображень, отриманих із вбудованої камери. Основним завданням мінімізувати розмір зображень після їхнього стиснення у форматі JPEG, забезпечуючи при цьому прийнятну якість.

Для зручності налаштувань та тестування системи була реалізована веб-сторінка, яка зберігалася у пам'яті ESP32. Вона надавала можливість динамічного регулювання кількості ітерацій вейвлет-перетворення та рівня якості JPEG-стиснення. Для реалізації компресії JPEG використовувалася спеціалізована бібліотека, оптимізована для роботи на ресурсозалежних пристроях.

Детальний опис процесу експерименту:

1. Для експерименту підготовлено тестові зображення. Для експерименту обрано зображення розміром 320×240 пікселів, отримане з камери. Кожне зображення підлягало обробці трьома типами вейвлет-перетворень: Добеші, Хаара та Коіфлета. Цей етап дозволив створити уніфіковану базу зображень для подальшої обробки та аналізу.

2. Розробка алгоритму процесу вейвлет-перетворення. Кожне зображення декомпоноване на кілька рівнів із використанням обраних вейвлетів. Під час обробки враховувалися коефіцієнти високих і низьких частот для кожного типу вейвлет-перетворення, що дозволяло розподілити деталі зображення на різні частотні компоненти. Результати цього етапу представлені у вигляді таблиці, де вказані параметри кожного перетворення.

3. Проведено оцінку якості зображення. Для кожного вейвлет-перетворення були розраховані метрики якості, зокрема:

- Середньоквадратична похибка (MSE): Визначала рівень спотворення пікселів після обробки.
- Пікове співвідношення сигнал-шум (PSNR): Оцінювало збереження якості зображення після стиснення.

- Індекс структурної схожості (SSIM): Використовувався для аналізу схожості структур відновленого зображення з оригіналом.

- Евклідова відстань: Визначала відмінності між кольорними характеристиками пікселів до і після обробки.

4. Оцінено час обробки. Вимірювався час, необхідний для обробки кожного зображення на мікроконтролері ESP32. Були враховані кількість математичних операцій, виконаних на кожному етапі, та їх вплив на швидкість обробки.

5. Оцінено рівень стиснення. Для кожного методу вейвлет-перетворення розраховувалося відношення розміру оригінального зображення до розміру обробленого та стисненого JPEG-файлу. Цей показник демонстрував ефективність зменшення обсягу даних після застосування обробки.

6. Оцінено різницю між пікселями. Вимірювалася різниця між пікселями для всіх трьох типів вейвлетів (Хаара, Добеші та Коіфлета) на 2-му та 4-му рівнях декомпозиції. Значення різниці виражалися у шкалі від 0 до 1, де 0 – повна схожість пікселя між обома зображеннями, 1 – повна відмінність пікселя. У рамках алгоритму значення кольору пікселів були представлені в діапазоні від 0 (чорний колір) до 255 (білий колір).

Проведено додатковий аналіз, де розглянуто:

1. Проведено порівняння вейвлетів. Вейвлет-перетворення Добеші забезпечувало більш точну деталізацію зображення, але вимагало більше обчислювальних ресурсів. Вейвлет Хаара виявився швидшим і менш ресурсомістким, але іноді втрачав важливі деталі зображення. Вейвлет Коіфлета продемонстрував високу ефективність для зображень із більш складними структурними елементами.

2. Досліджено залежність якості від рівня декомпозиції. Під час аналізу встановлено, що вищі рівні декомпозиції зменшують розмір файлу, але можуть впливати на якість зображення. На другому рівні декомпозиції якість залишалася на прийнятному рівні, тоді як на четвертому рівні починали проявлятися помітні спотворення.

3. Розглянуто переваги веб-інтерфейсу. Розробка веб-сторінки дозволила зменшити час налаштувань і зробила процес експерименту більш гнучким. Користувач міг легко змінювати параметри, що дозволило проводити більше тестів за короткий проміжок часу.

4. Досліджено результати оцінки різних параметрів якості зображення. Використано метрики MSE, PSNR і SSIM, що дало змогу отримати комплексну оцінку якості відновленого зображення. Евклідова відстань показала себе як корисний інструмент для аналізу локальних змін пікселів, особливо для оцінки структурних змін.

У результаті проведеного експерименту були отримані зображення, що відповідають різним типам вейвлет-перетворень, застосованих на другому та четвертому рівнях декомпозиції. Ці зображення дали можливість проаналізувати якість обробки, виявити видимі артефакти та оцінити ефективність кожного методу. На рисунку Рисунок 14 – Оригінальне зображення, отримане з камери модулю ESP-32CAM показано оригінальне зображення розміром 320×240 пікселів, отримане за допомогою вбудованої камери модуля ESP32-CAM. Це зображення стало базовим для подальшого застосування вейвлет-перетворень та аналізу результатів.



Рисунок 14 – Оригінальне зображення, отримане з камери модулю ESP-32CAM

Спершу оригінальне зображення оброблено за допомогою вейвлет-перетворень Хаара, Добеші та Коіфлета на другому рівні декомпозиції. Результати перетворення були збережені у форматі JPEG та представлені на рисунку Рисунок 15 – Зображення після вейвлет перетворення другого ступеня (а) – Хаара, (б) – Добеші, (в) – Коіфлет.



а



б



в

Рисунок 15 – Зображення після вейвлет перетворення другого ступеня (а) – Хаара, (б) – Добеші, (в) – Коіфлет

На наступному етапі те саме зображення піддалося обробці з використанням четвертого рівня декомпозиції з подальшим збереженням у форматі JPEG. Отримані результати наведено на рисунку Рисунок 16 – Зображення після вейвлет перетворення четвертого ступеня (а)– Хаара, (б)– Добеші, (в)– Коіфлет, що дозволяє візуально порівняти ефективність різних рівнів декомпозиції та помітити зміни у деталях і структурі зображення.



а



б



в

Рисунок 16 – Зображення після вейвлет перетворення четвертого ступеня (а)– Хаара, (б)– Добеші, (в)– Коіфлет

У таблиці Таблиця 4 – Порівняння результатів перетворення JPEG та перетворення JPEG + вейвлет подано детальні результати експериментів, що містять інформацію про кожен тип вейвлет-перетворення, рівень декомпозиції, якість зображення, час виконання алгоритму, рівень стиснення та кількість арифметичних операцій, виконаних мікроконтролером. Основні метрики, що використовувалися для оцінки якості, включають MSE (середньоквадратична похибка), що демонструє ступінь спотворення пікселів після обробки, PSNR (пікове співвідношення сигнал-шум), що визначає збереження якості зображення після стиснення, а також SSIM (індекс структурної схожості) для оцінки схожості структур відновленого зображення з оригіналом. Для розрахунку кількості арифметичних дій використана наступна формула:

$$NoAO = (h \cdot w) \cdot l \cdot c \quad (25)$$

де h – висота зображення (240 пікселів), w – ширина зображення (320 пікселів), l – рівень декомпозиції, c – кількість коефіцієнтів що використовуються для вейвлет-перетворення. Кількість коефіцієнтів для вейвлет перетворення Хаара дорівнює двом, для Добеші – чотирьом, для Коіфлет – шести.

Таблиця 4 – Порівняння результатів перетворення JPEG та перетворення JPEG + вейвлет

DWT	Рівень декомпозиції	MSE	PSNR	SSIM	Час перетворення, мс	Рівень стиснення, разів	Кількість арифметичних операцій, $(h * w) * l * c$
Haar	2	25,153	34,124	0,965	40	5,03	166 400
	4	73,449	29,470	0,921	84	11,11	332 800
Daubeshies4	2	25,274	34,103	0,965	96	5,31	665 600
	4	78,241	28,974	0,894	178	11,27	1 331 200
Coiflets1	2	25.144	34.126	0.965	105	5,21	998 400
	4	89.630	29.103	0.901	212	10,52	1 996 800

Для кожного методу також розраховано рівень стиснення, який визначається як відношення обсягу оригінального JPEG-файлу до файлу,

обробленого вейвлет-перетворенням і стисненого за допомогою JPEG. Це дозволяє оцінити ефективність алгоритму з точки зору зменшення обсягу даних. Рівень декомпозиції відображає кількість ітерацій або етапів, протягом яких зображення розбивається на низькочастотні та високочастотні компоненти. Другий рівень декомпозиції дозволяє зберегти більше деталей зображення, тоді як четвертий рівень значно зменшує розмір файлу, але може спричиняти втрату деталей.

Для оцінки обчислювальної складності розраховано кількість арифметичних операцій, виконаних мікроконтролером під час обробки. Використана формула враховує операції, необхідні для обчислення коефіцієнтів високих і низьких частот, та дозволяє оцінити ресурсомісткість алгоритму.

Результати показали, що рівень стиснення залежить не лише від типу вейвлет-перетворення, а й від рівня декомпозиції. Збільшення рівня декомпозиції дозволяє значно зменшити розмір файлу, але може негативно впливати на якість зображення, особливо для структурно складних даних.

Візуальний аналіз зображень, збережених після обробки, дозволив визначити видимі артефакти, зокрема втрату різкості та спотворення кольорів. Це особливо помітно на четвертому рівні декомпозиції для вейвлетів Хаара та Коїфлета, тоді як Добеші зберігає кращу деталізацію.

Вейвлет Хаара продемонстрував високу швидкість обробки, що робить його ефективним для систем із низькими обчислювальними ресурсами. Добеші забезпечив кращу якість зображення за рахунок використання більшої кількості коефіцієнтів. Вейвлет Коїфлета показав гарну ефективність для структурно складних зображень, але вимагав більше часу на обробку.

Застосування вейвлет-перетворення Хаара для обробки зображень показало добрі результати на другому рівні декомпозиції. Значення MSE склало 25.153, PSNR — 34.124, а SSIM — 0.965, що свідчить про високу якість відновленого зображення. Однак на четвертому рівні якість суттєво погіршилася: MSE збільшилося до 73.840, PSNR знизилося до 29.470. Це свідчить про втрату деталей зображення через обмеження простої структури

вейвлету Хаара на вищих рівнях декомпозиції. Час обробки на другому рівні складав лише 40 мс, що є найменшим серед усіх досліджених вейвлетів, однак на четвертому рівні він зріс до 84 мс. Коефіцієнт стиснення на четвертому рівні становив 11.22, демонструючи гарну здатність до зменшення обсягу даних.

Вейвлет-перетворення Добеші продемонструвало дещо нижчі значення MSE і PSNR на другому рівні декомпозиції порівняно з Хааром, але якість зображення залишалася на високому рівні, про що свідчить SSIM — 0.965. На четвертому рівні декомпозиції якість зображення значно погіршилась (MSE — 78.241, PSNR — 28.974), але залишалася прийнятною. Час обробки став більшим через складнішу структуру цього вейвлету: 96 мс на другому рівні та 178 мс на четвертому рівні. Це обумовлено тим, що кількість арифметичних операцій для Добеші у чотири рази перевищує відповідний показник для Хаара. Однак вейвлет Добеші забезпечив вищий коефіцієнт стиснення на четвертому рівні (11.27), що свідчить про його здатність ефективно зменшувати розмір файлу.

Вейвлет-перетворення Коїфлет показало кращі результати MSE та PSNR на другому рівні декомпозиції порівняно з іншими вейвлетами. Проте на четвертому рівні якість зображення виявилася нижчою, ніж у Хаара, але все ж кращою, ніж у Добеші. Значення MSE для четвертого рівня становило 89.630, що вказує на значну втрату якості, а PSNR, хоч і є вище за Добеші, поступався Хаару. На другому рівні значення SSIM залишалося високим — 0.901, що демонструє збереження структурної інформації зображення. Час обробки для Коїфлета є найдовшим серед усіх досліджених вейвлетів: 212 мс на четвертому рівні. Це пояснюється більшою кількістю коефіцієнтів цього вейвлету, через що кількість арифметичних операцій була у 1.5 рази більшою, ніж для Добеші. Хоча коефіцієнт стиснення на четвертому рівні залишався високим (10.52), він виявився найнижчим серед усіх досліджених методів.

Отримані значення PSNR для другого рівня декомпозиції практично однакові (34,1). Для різних типів зображень ці показники можуть варіюватися в межах 29–32, залежно від особливостей оброблюваних даних. У деяких випадках

PSNR для перетворення Хаара може досягати 38, а для перетворення Добеші — 39,2, що пов'язано з розміром і типом вихідного зображення.

На другому рівні декомпозиції артефакти були майже непомітними для всіх типів вейвлет-перетворень. Спостерігалася певна блоковість артефактів, однакова для всіх методів, що вказує на стабільність роботи алгоритмів на цьому рівні.

На четвертому рівні декомпозиції артефакти стали більш вираженими і проявлялися по-різному для кожного типу вейвлет-перетворення. Для перетворення Хаара чітко помітно регулярні темні крапки. Ці артефакти можна пояснити накопиченням похибок під час обчислень через просту структуру вейвлету. Така структура ефективна для швидкої обробки, але втрата певної кількості інформації на кожному рівні декомпозиції призводить до поступового погіршення якості.

Для перетворення Добеші також спостерігалися артефакти на четвертому рівні декомпозиції, проте вони мали інший характер і були менш помітними. Це свідчить про здатність цього вейвлету краще зберігати як високочастотні, так і низькочастотні компоненти зображення. Така особливість робить перетворення Добеші більш придатним для збереження деталей і текстур порівняно з Хааром. Дослідження підтверджують, що перетворення Добеші забезпечує вищу якість обробки зображень і відео порівняно з Хааром. Це робить його більш перспективним для застосувань, де важлива не лише швидкість, а й висока якість відновлення зображень після стиснення.

Артефакти, які спостерігалися на четвертому рівні декомпозиції під час вейвлет-перетворення Коіфлет, були помітнішими, ніж при використанні вейвлет-перетворення Добеші. Відновлене зображення мало схожу якість із тим, що отримане за допомогою Хаара, хоча артефакти виявилися менш вираженими. Значна кількість коефіцієнтів, які використовує вейвлет Коіфлет, призвела до значної кількості арифметичних операцій: 998 400 для другого рівня декомпозиції та 1 996 800 для четвертого рівня. Це робить його ресурсозатратним у порівнянні з іншими методами.

На другому рівні декомпозиції значення PSNR залишалися в межах від 32,8 до 35,4, залежно від типу обраного зображення. На четвертому рівні цей показник змінювався від 22,2 до 35,9, що вказує на сильну залежність якості від характеру вхідного зображення. У порівнянні з другим рівнем декомпозиції, артефакти ставали більш помітними на четвертому рівні, що підтверджує втрату деталей при більшій кількості ітерацій обробки.

Результати експерименту показали, що показники MSE, PSNR і SSIM мають не завжди коректне відображення якості відновленого зображення. Наприклад, високі значення MSE можуть свідчити про суттєві відмінності від оригінального зображення, тоді як високі показники PSNR і SSIM вказують на збереження прийнятної якості. Ця розбіжність особливо проявилася при використанні вейвлет-перетворення Коіфлет, де MSE демонструвало втрату точності, однак PSNR і SSIM показували достатню візуальну якість відновленого зображення. При цьому зазначені метрики не враховують вплив артефактів, які можуть значно впливати на сприйняття обробленого зображення.

Для кращої оцінки якості зображення важливо враховувати артефакти, що виникають у результаті обробки. Одним із методів, який може забезпечити точнішу оцінку, є використання евклідової відстані. Такий підхід дозволяє виявляти відмінності між пікселями, аналізуючи зміни у кожному кольоровому каналі. Це дає змогу врахувати складні артефакти, які можуть бути присутніми в оброблених зображеннях, і більш об'єктивно оцінити їхню якість. Розрахунок максимальної евклідової відстані між кольорами пікселів двох зображень однакової розмірності може бути використаний для оцінки якості обробки та визначення впливу артефактів на кінцевий результат.

$$d \max(\sqrt{(R'_{ij} - R_{ij})^2 + (G'_{ij} - G_{ij})^2 + (B'_{ij} - B_{ij})^2})_{\max} \quad (26)$$

де i та j це позиція пікселя по ширині та довжині, R_{ij} , G_{ij} , B_{ij} – значення кольорів червоного, зеленого та синього для оригінального зображення, R'_{ij} , G'_{ij} , B'_{ij} – значення кольорів червоного, зеленого та синього для відновленого зображення.

У Таблиці Таблица 5 – Порівняння результатів евклідової відстані для різних вейвлет-перетворень наведено відносну різницю між пікселями для трьох вейвлетів: Хаара, Добеші та Коіфлет на 2-му та 4-му рівнях декомпозиції. Відносна різниця оцінюється від 0 до 1, де 0 означає повну схожість пікселів, а 1 — максимальну різницю між ними.

Таблица 5 – Порівняння результатів евклідової відстані для різних вейвлет-перетворень

Метод	Рівень декомпозиції	Відносна різниця
Хаар	2	0.041
	4	0.537
Добеші	2	0.029
	4	0.147
Коіфлет	2	0.034
	4	0.355

Як показано в таблиці Таблица 5 – Порівняння результатів евклідової відстані для різних вейвлет-перетворень, для вейвлет-перетворення Хаара відносна різниця зростає з 0.041 на другому рівні декомпозиції до 0.537 на четвертому. У випадку перетворення Добеші цей показник є меншим і становить 0.029 на другому рівні та 0.147 на четвертому. Перетворення Коіфлет демонструє проміжні значення: 0.034 на другому рівні та 0.355 на четвертому рівні.

Отже, зі збільшенням рівня декомпозиції відмінності між пікселями стають більш вираженими. Вейвлет-перетворення Хаара показує найвищу відносну різницю, тоді як перетворення Добеші забезпечує найменше спотворення, що свідчить про його більшу здатність зберігати деталізацію зображення.

Висновок

У ході дослідження здійснено ґрунтовний аналіз використання вейвлет-перетворень у сенсорних системах, зокрема для обробки та стиснення зображень у бездротових мережах. Основна проблема таких систем полягає в обмежених обчислювальних ресурсах вузлів, які зазвичай оснащені малопотужними мікроконтролерами та працюють в умовах енергоефективності. Передача зображень через сенсорну мережу є вкрай енергозатратним процесом, тому використання ефективних методів стиснення є критично важливим.

Аналіз літературних джерел показав, що вейвлет-перетворення є одним із найефективніших методів обробки зображень, що дозволяє суттєво зменшити обсяг переданих даних, зберігаючи при цьому прийнятну якість. Дослідження довели, що класичні методи обробки зображень вимагали значних обсягів пам'яті, що обмежувало їхнє використання на малопотужних пристроях. Проте сучасні розробки, такі як фракційні хвильові фільтри, дають змогу зменшити вимоги до ресурсів, хоча й потребують більше часу на обробку.

Важливим аспектом є вибір конкретного типу вейвлет-перетворення. Порівняльний аналіз вейвлетів Хаара, Добеші та Коїфлет у рамках експериментального дослідження дозволив визначити їхні ключові особливості та ефективність у контексті сенсорних мереж. Вейвлет Хаара є найпростішим і найшвидшим, що робить його ефективним для пристроїв із мінімальними ресурсами, проте він має низьку точність відновлення деталей і накопичує похибки на високих рівнях декомпозиції. Вейвлети Добеші забезпечують кращу якість відновлення, оскільки містять більше коефіцієнтів, що дозволяє зберігати високочастотну інформацію, проте вони є ресурсозатратнішими. Вейвлети Коїфлет, у свою чергу, демонструють високу точність відновлення структури зображення, але мають ще більші обчислювальні вимоги, що ускладнює їх застосування в обмежених середовищах.

Експериментальні результати підтвердили, що другий рівень декомпозиції забезпечує кращий баланс між якістю відновленого зображення та рівнем стиснення, тоді як на четвертому рівні стає помітним суттєве погіршення якості

через втрату деталей. Показники PSNR та SSIM вказують, що перетворення Добеші є найбільш ефективним для компресії зображень, оскільки воно зберігає більше деталей, ніж Хаар, і є менш чутливим до артефактів, ніж Коіфлет. Додатковий аналіз евклідової відстані між пікселями оригінального та відновленого зображення продемонстрував, що метод Хаара створює найбільші візуальні відмінності, тоді як Добеші мінімізує ці спотворення.

Отже, результати дослідження підтвердили, що вибір вейвлет-перетворення залежить від вимог конкретної системи. Якщо ключовими факторами є швидкість обробки та мінімізація споживаних ресурсів, доцільним є використання вейвлету Хаара. Якщо ж пріоритетом є якість відновлення зображення, найкращим вибором стає вейвлет Добеші. Вейвлет Коіфлет варто використовувати у випадках, коли важливе збереження тонких деталей, але є достатні обчислювальні ресурси.

Таким чином, застосування вейвлет-перетворень у сенсорних системах відкриває широкі можливості для ефективного стиснення та обробки зображень. Подальші дослідження можуть бути зосереджені на адаптивних методах компресії, що враховують не лише характеристики зображення, а й параметри бездротового каналу зв'язку, рівень енергоспоживання вузлів та можливість інтеграції з сучасними методами машинного навчання для більш ефективного аналізу візуальної інформації.

3 ПОРІВНЯННЯ МЕТОДІВ СТИСЕННЯ ДЛЯ РОБОТИ В СИСТЕМАХ ІНТЕРНЕТУ РЕЧЕЙ

3.1 Аналіз використання методів стиснення в сенсорних системах

Стиснення даних у системах на мікроконтролерах відіграє ключову роль, особливо у випадках, коли необхідно зменшити обсяг переданих або збережених даних при обмежених ресурсах пам'яті та обчислювальної потужності. Комбіновані алгоритми стиснення поєднують кілька методів, таких як Run-Length Encoding (RLE), кодування Хаффмана, алгоритми LZ-сімейства та інші, що дозволяє досягати ефективнішої компресії. Завдяки такому підходу можна оптимізувати зберігання, передавання даних та підвищити продуктивність вбудованих систем.

Приклади систем на мікроконтролерах, що використовують комбіноване стиснення

1. Brotli у веб-серверах з мікроконтролерами [81]

Brotli, розроблений Google, поєднує LZ77, кодування Хаффмана та спеціалізовані оптимізації для ефективного стиснення веб-даних. Це дозволяє впроваджувати його в мікроконтролери з вбудованими веб-серверами, що використовуються у розумних пристроях, IoT-системах та локальних веб-інтерфейсах для керування електронікою. Наприклад, Brotli може використовуватися у веб-інтерфейсах на ESP32 або STM32, дозволяючи стискати HTML, CSS та JavaScript перед передачею на клієнтський пристрій.

2. Deflate у бездротових сенсорних мережах [82]

Deflate, що поєднує LZ77 і кодування Хаффмана, є ефективним для мікроконтролерних систем, які працюють у бездротових сенсорних мережах (WSN). Завдяки можливості гнучкого регулювання рівня компресії, цей алгоритм застосовується в LPWAN-системах, таких як LoRaWAN або ZigBee, де необхідно передавати стиснені дані сенсорів у вигляді пакунків з мінімальною затримкою.

3. JPEG та його розширення у пристроях машинного зору [83]

Мікроконтролери, які працюють з обробкою зображень, наприклад, у робототехніці, автоматизованих системах або безпілотних платформах, часто використовують стиснення JPEG та його розширення (JPEG 2000, JPEG XL). Такі пристрої можуть зменшувати розмір отриманих зображень перед передачею на центральний сервер або збереженням у внутрішній пам'яті, забезпечуючи баланс між якістю та швидкістю обробки.

4. LZMA для оновлення прошивок мікроконтролерів [84]

LZMA (Lempel-Ziv-Markov Chain Algorithm), який поєднує LZ77 з аналізом Марковських ланцюгів, ефективно використовується для стиснення прошивок мікроконтролерів. Наприклад, у пристроях з обмеженим обсягом пам'яті цей метод допомагає зменшити розмір оновлень програмного забезпечення (ОТА-оновлення), що є критично важливим для IoT-платформ та автономних пристроїв.

5. Zstandard у високопродуктивних вбудованих системах [85]

Алгоритм Zstandard (zstd), створений Facebook, комбінує LZ77 із вдосконаленими методами кодування Хаффмана та застосовується у високопродуктивних вбудованих системах. Його використовують у пристроях, які виконують часті операції стиснення та розпакування даних, наприклад, у промислових контролерах, які працюють із великими масивами логів або сенсорних даних.

3.2 Теоретичні відомості методів стиснення

Суть цього методу стиснення даних полягає в наступному: якщо певний символ (наприклад, d) повторюється поспіль n разів у вхідному потоці, то замість цих n повторень можна записати лише пару значень: n та сам символ d . Послідовність з n однакових підряд ідущих елементів називають "пробігом" довжини n , а описаний підхід отримав назву "кодування довжин пробігів" або RLE. Спочатку цей принцип зазвичай застосовують до текстових даних, а згодом переходять до стиснення зображень.

Застосування RLE для текстових даних

Якщо просто замінити фрагмент тексту на зразок "2.uall_is_too_well" на "2.ua2.is1t2.we2", це не вирішить проблему. Річ у тім, що декомпресор повинен чітко відрізняти цифри, які є частиною вихідного тексту, від тих, що вказують на кількість повторень наступного символу. Аналогічно, модифікація "2.ua21.is t2o.we21" також не приносить користі ні в плані чіткості, ні в плані стиснення. Одним із рішень є використання спеціального "символу-втікача" (escape character), який вказуватиме на початок послідовності, що кодує повтор. Наприклад, обравши "@" в ролі такого символу, можна записати рядок "2.ua21.is t2o we021" у вигляді "[2.ua@21.is](#) t@2o we0@21". Тепер декомпресор зможе однозначно розпізнати, коли "2" слугує лічильником повторів, а коли – звичайним символом тексту. Проте такий підхід збільшує довжину коду, оскільки замість двох однакових літер з'являються три символи (символ-втікач, лічильник та сам символ). Тому доцільно домовитися, що стискатимуться лише послідовності з трьох або більше ідентичних символів підряд. На ілюстрації (рисунок 1.6а) наведено блок-схему простого текстового компресора на основі RLE [98].

Принцип роботи такий: прочитавши перший символ, встановлюємо лічильник повторів у 1 і зберігаємо цей символ. Надалі кожен наступний символ порівнюємо з уже збереженим. Якщо він такий самий, збільшуємо лічильник. Якщо інший – перевіряємо величину лічильника. Якщо вона мала (наприклад, менше 3), просто записуємо збережений символ у вихідний потік і починаємо обробку нових символів як початок нової послідовності. Якщо ж лічильник великий (відповідає нашому порогу для стиснення), замість послідовності однакових символів записуємо символ-втікач, за ним – число повторень та сам символ.

Декомпресія працює за схожим принципом, як показано на рисунку 1.6б: коли в потоці зустрічається символ-втікач "@", наступні значення інтерпретуються як кількість повторів та символ, який слід відтворити задану кількість разів.

Основні труднощі цього методу:

1. У типовому англomовному тексті довгі послідовності ідентичних символів трапляються не часто. Найпоширеніші повтори – подвійні (наприклад, подвійний пробіл), а тривалі "ланцюжки" однакових символів майже не зустрічаються. Власне, найбільш часто повторюваним символом є пробіл. У математичних текстах інколи часто повторюються певні цифри чи символи. Проте загалом, приклади з багатьма однаковими символами штучні.

Наприклад:

"The abbot from Abruzzi accedes to the demands of all abbesses from Narraganset and Abbeville from Abyssinia. He will accommodate them, abbreviate his sabbatical, and be an accomplished accessory."

У цьому фрагменті немає довгих послідовностей ідентичних символів, окрім, можливо, пробілів.

2. Якщо символ-втікач "@" потенційно може зустрічатися у вхідному тексті як звичайний символ, то доведеться обрати інший "символ-втікач". Іноді доводиться працювати зі стрічками, що можуть містити будь-які можливі символи. Наприклад, двійкові файли (об'єктні файли програм) можуть мати будь-які байтові значення.

3. Зазвичай кількість повторів записується в один байт, що обмежує її максимумом у 255. Хоча це обмеження можна частково обійти, домовившись, що факт наявності лічильника вже свідчить про повтор символів. Наприклад, можна встановити правило: лічильник "0" означає три повтори, а максимальне значення 255 – це 258 повторів. Така градація дає змогу трохи розширити діапазон без додаткових байтів.

Метод *MNP* класу 5, розроблений компанією Microsoft (виробником модемів), широко використовується для стиснення даних у модемах. Цей метод поєднує RLE зі змінною частотою кодування, що дає змогу ефективніше стиснути дані коли вхідний потік містить три або більше однакових байтів підряд, стискач записує у вихідний потік три копії такого байта, а слідом – лічильник повторів. Під час розпакування, зустрівши послідовність із трьох ідентичних байтів, декомпресор розуміє, що наступний байт вказує кількість

додаткових повторів (де значення 0 означає, що маємо лише тридинний пробіг). Ці процеси продемонстровано на рис. Рисунок 17 – RLE (a) – компресія, (б) – декомпресія. Недоліком цього підходу є те, що початкова послідовність з трьох однакових символів у вхідних даних перетворюється на чотири символи у вихідному потоці, тобто фактично відбувається "розширення", а не стиснення. Якщо послідовність складається з чотирьох символів, користі від стиснення немає взагалі. Економія починається лише тоді, коли послідовність довша за чотири символи. Ще одним обмеженням, яке притаманне методу *MNP5*, є штучне зниження максимальної кількості повторів до 250 (замість 255).

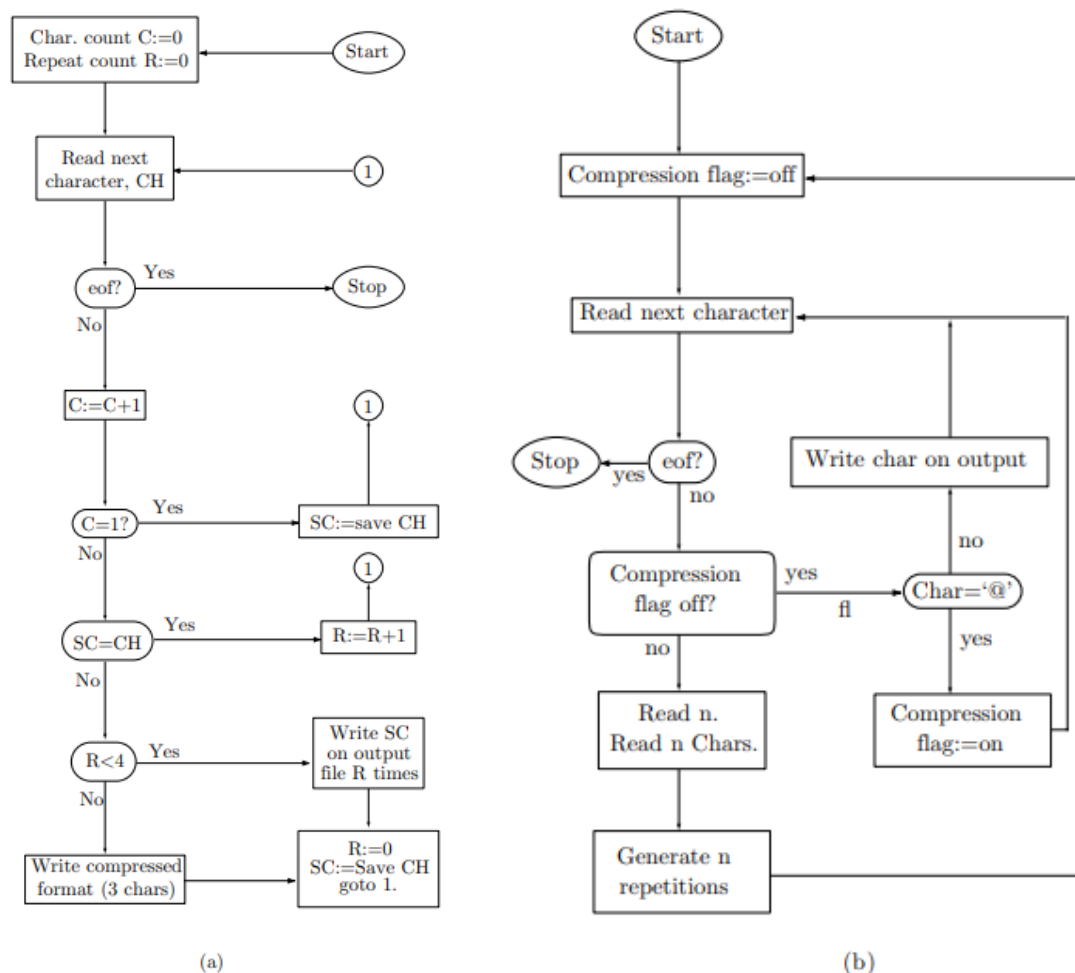


Рисунок 17 – RLE (a) – компресія, (б) – декомпресія

Щоб приблизно оцінити коефіцієнт стиснення за допомогою RLE, уявімо, що є рядок довжиною N символів, який потрібно стиснути. Припустимо, у ньому

є M повторюваних фрагментів, середня довжина кожного з яких становить L . Під час стиснення кожен із цих M фрагментів замінюється на три символи: спеціальний символ-втікач, лічильник повторів та самі дані. Отже, розмір стиснутого рядка буде $N - M \times L + M \times 3 = N - M(L - 3)$. Тоді коефіцієнт стиснення можна оцінити як $N/(N - M(L - 3))$.

Для *MNP5* замість числа 3 у цій формулі підставляється 4. Наприклад [98], якщо $N = 1000$, $M = 10$ та $L = 3$, коефіцієнт становитиме $1000/(1000 - 10(4 - 3)) = 1,01$, що майже не дає стиснення. Кращий результат спостерігається за умов $N = 1000$, $M = 50$, $L = 10$, де коефіцієнт дорівнює $1000/(1000 - 50(10 - 3)) = 1,538$.

Окремим різновидом кодування довжин пробігів для тексту є "диграмне кодування". Воно ефективне тоді, коли дані складаються лише з певного обмеженого набору символів (наприклад, тільки з літер, цифр та знаків пунктуації). Ідея полягає в тому, щоб знайти найбільш поширені пари символів (диграми) та замінити їх на один із символів, які зазвичай не зустрічаються у тексті (наприклад, на один із керівних ASCII-символів). Якщо завчасно проаналізувати дані, можна досягти добрих результатів. Для англomовних текстів поширеними диграмами є, скажімо, "E'", "L'", "TH" або "uA". Інші типи даних можуть мати власні типові диграми. Метод "sequitur" є прикладом підходу, де стиснення досягається заміною таких поширених пар або довших повторюваних фрагментів на спеціальні символи.

Схожим підходом є "заміна шаблонів" (pattern substitution). Цей метод добре підходить для стиснення програмних кодів, де часто повторюються певні слова, наприклад "for", "repeat" або "print". Кожне з таких слів можна замінити керівним символом. Якщо ж частих слів дуже багато, використовують символ-втікач, за яким слідує кодове позначення слова. Припустимо, що для слова "print" призначено код "a". Тоді фрагмент "m:uprint,b,a;" можна стиснути до "m:u@a,b,a;".

Стиснення зображень за допомогою RLE

RLE є природним вибором для стиснення графічних даних. Цифрове зображення складається з безлічі маленьких точок – пікселів. Кожен піксель

може мати однобітне представлення (наприклад, чорний або білий), а може бути багатобітним, відображаючи широкий діапазон кольорів чи відтінків сірого. Припустимо, що пікселі зберігаються у вигляді масиву, так званої растрової карти (bitmap), яка виступає вхідним потоком для процесу стиснення. Зазвичай пікселі впорядковані построчно: перший піксель – у верхньому лівому куті, а останній – у нижньому правому.

Стиснення зображень за допомогою RLE базується на спостереженні, що ймовірність того, що сусідні пікселі мають однаковий колір, є досить високою. Тому компресор послідовно аналізує растр рядок за рядком, виявляючи ланцюжки пікселів однакового кольору. Наприклад, якщо перші 17 пікселів білі, потім іде 1 чорний, за ним 55 білих і так далі, то у вихідний потік слід записати лише числа 17, 1, 55... без дублювання самих кольорів по багато разів.

Компресор за замовчуванням може припускати, що початкові пікселі зображення білі. Якщо це не так, можна вказати початкову кількість білих пікселів як 0. Також на початку стиснутого потоку бажано зберегти інформацію про роздільну здатність зображення.

Ефективність стиснення RLE сильно залежить від характеру картинки. Чим складніше зображення (із великою варіативністю кольорів), тим гірше буде коефіцієнт стиснення. На рисунку Рисунок 18 –Однорідні області та лінії сканування., наприклад, продемонстровано, як сканувальні лінії проходять через однорідну ділянку. Кожна лінія перетинає область, входячи в неї та виходячи з іншого боку, причому жодна інша лінія цих точок не використовує. Виявляється, кількість сканувальних ліній, що проходять через однорідну ділянку, приблизно дорівнює половині периметра цієї ділянки (у пікселях). Оскільки в однорідній області кожна лінія додає лише одне число у вихідний потік, коефіцієнт стиснення для такої ділянки можна приблизно оцінити як відношення половини довжини периметра до загальної кількості пікселів у цій області [98].

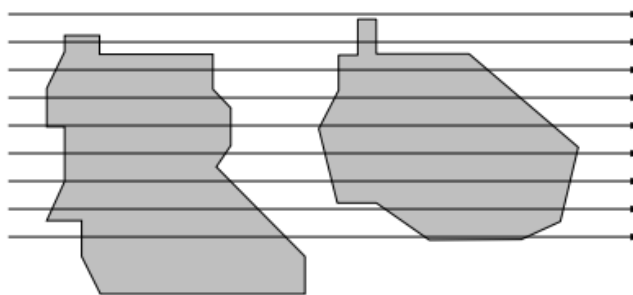


Рисунок 18 – Однорідні області та лінії сканування.

RLE можна застосувати й до зображень у відтінках сірого. У цьому разі кожен ланцюжок пікселів з однаковою інтенсивністю кодується парою: "довжина послідовності" та "значення пікселя". Довжина зазвичай уміщається в один байт, що дозволяє представляти до 255 пікселів у послідовності. Значення пікселя може займати кілька біт, залежно від кількості рівнів сірого. Наприклад, 8-бітна глибина дозволяє до 256 різних відтінків.

Як приклад розглянемо 8-бітну сіро тональну растрову карту, що починається з ряду інтенсивностей: 12, 12, 12, 12, 12, 12, 12, 12, 12, 35, 76, 112, 67, 87, 87, 87, 87, 5, 5, 5, 5, 5, 1, ...

Після стиснення вона може бути записана у вигляді: 9, 12, 35, 76, 112, 67, (3, 87), 6, (5), 1, ...

де у квадратних дужках позначено лічильники повторів. Виклик полягає в тому, щоб розрізняти байти, які позначають рівень сірого, та байти, що виступають лічильниками повторів (наприклад, число 12 може означати сам відтінок або кількість повторів, а число 9 – лічильник повторів чи значення пікселя).

Для вирішення проблеми пропонується декілька підходів:

1. Якщо кількість відтінків сірого не перевищує 128, можна використати один біт у кожному байті для позначення того, чи є цей байт значенням сірого чи лічильником.

2. Якщо відтінків 256, можна зменшити їхню кількість до 255 і зарезервувати одне значення як спеціальний "прапорець", який позначатиме, що наступний байт – це лічильник. Наприклад, якщо 255 є таким прапорцем, тоді послідовність

набуде

форми:

255, 9, 12, 35, 76, 112, 67, 255, 3, 87, 255, 6, 5, 1, ...

3. Можна також виділити один біт у кожному байті, що міститься в окремо сформованих групах по 8 байтів. Ці групи пишуться у потік до (або після) відповідних 8 байтів, для яких вони є індикаторами. Це зменшує ефективність, додаючи 1 байт службової інформації на кожні 8 байтів даних, тобто збільшує потік приблизно на 12,5%.

4. Коли натрапляємо на групу з m пікселів, які всі різні, можна перед нею розмістити байт зі значенням $-m$ (негативне число), а далі перерахувати ці m різних пікселів. Наприклад, якщо у послідовності йде група з різних значень довжиною m , можна кодувати її як $-m, p_1, p_2, p_3 \dots p_m$. Найгіршим випадком є те, що всі пікселі у зображенні різні, тоді коефіцієнт стиснення буде дуже низьким. Але навіть у такому разі можуть знайтися методи оптимізації (як-от вставлення службових байтів або комбінування стратегій).

Додаткові зауваження:

1. Оскільки довжина послідовності не може дорівнювати нулю, доцільно записувати у вихідний потік (довжину послідовності – 1). Наприклад, пара (3, 87) означає чотири пікселі з інтенсивністю 87. Таким чином, максимальна довжина послідовності може сягати 256.

2. Коли йдеться про кольорові зображення, зазвичай кожен піксель представлено трьома байтами – інтенсивностями червоного, зеленого та синього кольорів. У цьому випадку бажано окремо стискати послідовності для кожного з трьох каналів. Це означає, що будь-який метод, придатний для сіротонованих зображень, можна застосувати й до кожного кольорового каналу, а отже, й до кольорових зображень у цілому.

3. Корисно кодувати кожний рядок растрового зображення окремо. Наприклад, якщо в кінці одного рядка є 4 повторюваних пікселі з інтенсивністю 87, а на початку наступного рядка йдуть 9 таких же пікселів, то краще записати їх, перервавши на кінці рядка, а потім почавши новий рядок з нуля, ніж об'єднувати їх у одну довгу послідовність. Ще краще – додати спеціальний

маркер кінця рядка (наприклад, "eol"), щоб, переглядаючи стиснений потік, користувач або система могли прийняти рішення про подальшу обробку вже після декодування кількох перших ліній. Якщо кожен рядок кодується окремо, можна, наприклад, спершу відтворити та показати рядки 1, 6, 11 тощо, щоб отримати загальну уяву про зображення, а потім, за потреби, додати проміжні рядки. Такий спосіб дозволяє поступово відображати картинку й оцінювати її на ранніх етапах.

Окреме кодування кожного рядка має й іншу перевагу: можна вилучити чи витягти певну частину стисненого зображення (наприклад, лише рядки з k по l) без повної декомпресії всього файлу. А також можливо поєднувати дані двох різних стиснених зображень, не декодуючи їх повністю.

Якщо використовується індивідуальне кодування кожного рядка, тоді у стисненому потоці треба дати знати, де починається кожний рядок. Для цього на початку потоку можна розмістити заголовок, у якому містяться вказівки (офсети) до кожного рядка растрового зображення. Хоча це збільшує обсяг метаданих, проте може виявитися корисним компромісом між розміром та зручністю роботи з стисненим потоком.

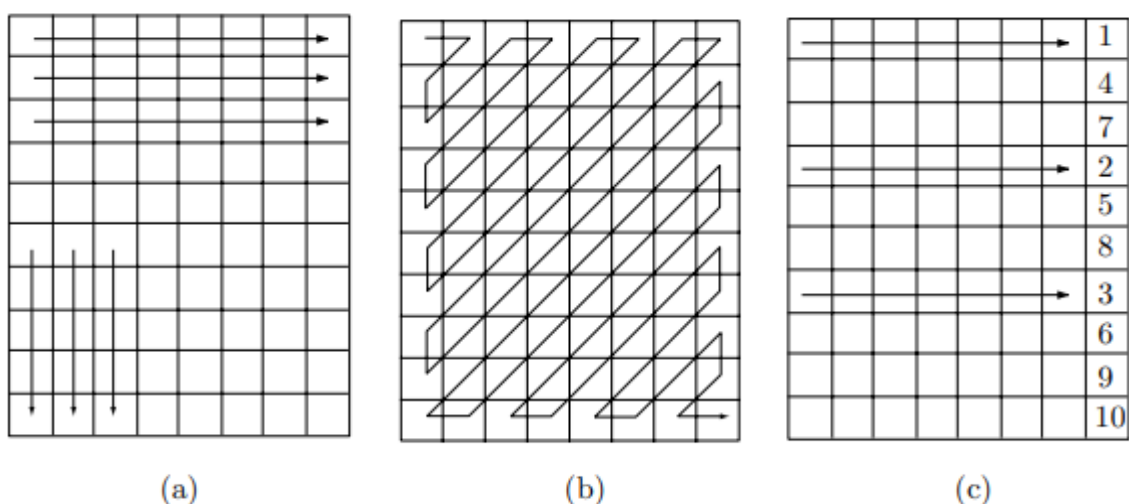


Рисунок 19 – RLE сканування

Є й інша, очевидна причина окремого кодування рядків: RLE для зображень розраховано на те, що суміжні пікселі ймовірно однакові. Проте перший піксель наступного рядка зовсім не обов'язково буде ідентичним останньому пікселю попереднього рядка.

Недолік стиснення зображень методом RLE полягає в тому, що при найменшій зміні зображення довжини послідовностей доведеться перераховувати заново. У деяких випадках вихідний стиснений потік RLE може бути навіть більшим за необроблені дані. Наприклад, якщо зображення містить багато вертикальних ліній, сканування по рядках дає лише дуже короткі ланцюжки, через що коефіцієнт стиснення може бути дуже низьким або взагалі вийде "розширення" замість стиснення. Практичний RLE-компресор зображень повинен уміти застосовувати різні стратегії сканування – по рядках, по стовпцях або "зигзагом" – і обрати той спосіб, що дасть найкращий результат. Він може автоматично випробувати всі три методи на кожному стисненому растровому зображенні, щоб досягти максимальної ефективності.

Алгоритм кодування Хаффмана

Метод кодування, розроблений Девідом Хаффманом у межах навчального завдання, став одним із перших внесків у галузі теорії інформації, який викладав Роберт Фано в MIT [88]. Створені за цією методикою кодові набори називають кодами Хаффмана. Вони належать до префіксних кодів і забезпечують кращий результат для заданої моделі (набору ймовірностей).

Алгоритм Хаффмана ґрунтується на двох ключових спостереженнях стосовно ефективних префіксних кодів:

1. В ефективному коді символи, які трапляються частіше (мають більшу ймовірність появи), отримують коротші кодові слова, ніж ті, що зустрічаються рідше.
2. В ефективному коді два найменш імовірні символи матимуть кодові слова однакової довжини.

Перше спостереження легко зрозуміти. Якби частіші символи мали довші кодові слова, ніж рідкісні, то середня кількість біт на символ була б більшою, ніж

за умов, коли частіші символи кодуються коротше. Отже, код, де поширені символи мають довші кодові слова за малопоширені, не може бути ефективним. Друге спостереження можна пояснити так: уявімо, що існує ефективний код, де два найменш імовірні символи мають кодові слова різної довжини. Нехай більш довгий код можна вкоротити, відкинувши кілька біт наприкінці. Тоді коротший код не може бути префіксом довшого (оскільки маємо префіксну структуру), і навіть без цих відкинутих бітів ці два коди залишаться чітко відокремленими один від одного. Оскільки вони відповідають найрідкіснішим символам, ніякий інший код не може бути довшим за них, а отже, вкорочений код ніяк не стане префіксом іншого. Вкоротивши довший код, отримаємо новий кодовий набір з меншою середньою довжиною, що суперечить ефективності початкового. Отже, для ефективного коду друге спостереження теж справджується.

Алгоритм Хаффмана додає до цих двох спостережень ще одну просту умову: кодові слова для двох найбільш рідкісних символів мають відрізнятися лише в останньому біті. Припустімо, що γ і δ – це найменш імовірні символи. Якщо кодове слово для γ – це $m0$ (де m – деяка послідовність із 1 та 0, а 0 позначає додавання нуля), тоді для δ воно буде $m1$. Ця вимога не суперечить обом попереднім спостереженням та визначає просту процедуру кодування.

Створення коду Хаффмана

Розглянемо джерело, яке генерує букви з алфавіту $A = \{a_1, a_2, a_3, a_4, a_5\}$ з ймовірностями $P(a_1) = P(a_3) = 0.2$, $P(a_2) = 0.4$, $P(a_4) = P(a_5) = 0.1$. Ентропія такого джерела становить приблизно 2.122 біти/символ. Щоб побудувати код Хаффмана, впорядкуємо символи за спаданням ймовірності, як показано в Таблиці Таблиця 6 – Початковий алфавіт із п'яти символів [89], позначаючи кодове слово для a_i як $c(a_i)$ [89].

Найменш імовірними символами є a_4 та a_5 , тож їм присвоюються початкові кодові слова. Далі “складаємо” їх разом, отримуючи новий ефективний символ з об'єднаною ймовірністю, і повторюємо процес, доки не отримаємо остаточний код. Таким чином, на основі послідовного об'єднання найменш імовірних

символів і призначення їм кодів за схемою "один код – із додаванням 0, інший – із додаванням 1" формуються кодові слова для всіх символів.

Таблиця 6 – Початковий алфавіт із п'яти символів [89]

Символ	Вірогідність	Кодове слово
a_2	0.4	$c(a_2)$
a_1	0.2	$c(a_1)$
a_3	0.2	$c(a_3)$
a_4	0.1	$c(a_4)$
a_5	0.1	$c(a_5)$

Результуючий код Хаффмана представлено у Таблиці Таблиця 7 – Код Хаффмана для початкового алфавіту з п'яти символів [89]. Процес можна узагальнити, як показано на рисунку Рисунок 20 – Процедура кодування Хаффмана. Імовірності символів наведені в дужках..

Таблиця 7 – Код Хаффмана для початкового алфавіту з п'яти символів [89]

Символ aa	Ймовірність	Кодове слово
a_2	0.4	1
a_1	0.2	01
a_3	0.2	000
a_4	0.1	0010
a_5	0.1	0011

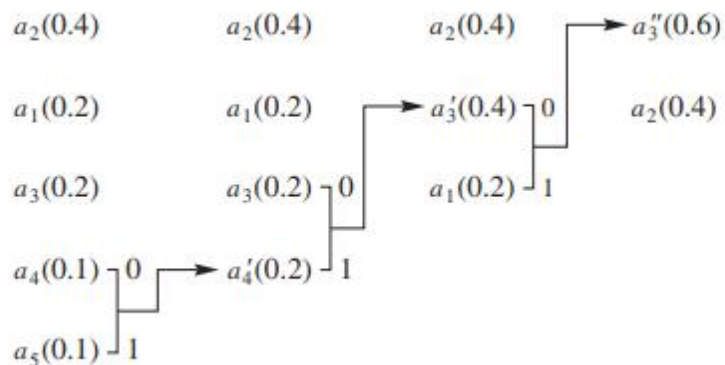


Рисунок 20 – Процедура кодування Хаффмана. Імовірності символів наведені в дужках.

Оцінка ефективності коду

Оцінка ефективності коду ґрунтується на його надмірності – це різниця між ентропією джерела та середньою довжиною кодових слів. У даному випадку надмірність становить 0.078 біт/символ. Надмірність дорівнює нулю лише тоді, коли ймовірності символів є від’ємними степенями двійки.

Альтернативний підхід до побудови коду Хаффмана

Існує інший спосіб побудови коду Хаффмана, який базується на його основній властивості – префіксності. Кожен символ можна представити як лист у двійковому дереві, де внутрішні вузли є допоміжними, а листи відповідають символам. Для визначення кодового слова певного символу проходимо дерево від кореня до листа, додаючи **0** щоразу, коли йдемо по верхній гілці, та **1**, коли йдемо по нижній [89].

Побудова двійкового дерева

Почнемо побудову дерева з листових вузлів, які відповідають символам. Оскільки два символи з найменшими ймовірностями завжди відрізняються лише останнім бітом, шлях від кореня до їхніх листів буде однаковим до останнього кроку. Це означає, що листи з найменшими ймовірностями походять від одного спільного вузла. З’єднавши ці листи в одну гілку, отримуємо символ зведеного алфавіту, ймовірність якого дорівнює сумі ймовірностей об’єднаних символів [89].

Далі замінюємо два вузли, що відповідають символам із найменшими ймовірностями, на їхній спільний "батьківський вузол". Потім повторюємо цей процес для нового алфавіту, доки в дереві не залишиться один вузол – корінь.

Щоб отримати код для кожного символу, дерево обхідно пройти від кореня до кожного листа: для верхньої гілки додаємо 0, для нижньої – 1. Цей підхід дає той самий результат, що й наведений у попередніх прикладах, але його представлено у вигляді двійкового дерева. Процес побудови показано на прикладах у Рисунках Рисунок 20 – Процедура кодування Хаффмана. Ймовірності символів наведені в дужках..

3.3 Аналіз наукових праць з використанням методів Хаффмана та RLE

У сучасних умовах бездротові сенсорні мережі (WSN) стали невід'ємною складовою різноманітних галузей, оскільки вони здатні забезпечити безперервний моніторинг та ефективний збір інформації з довкілля, промислових об'єктів, аграрних угідь, систем охорони здоров'я, а також слугують основою для смарт-технологій у містах та виробництві [90]. Ці мережі зазвичай складаються з великої кількості вузлів, що взаємодіють бездротовим чином, утворюючи децентралізовану структуру. Вузли можуть бути або повністю самоорганізованими, або керуватися через базові станції чи шлюзи, що відповідають за координацію та переспрямування даних. Кожен сенсорний вузол має власний мікроконтролер або мікропроцесор, отримує дані від датчиків, обробляє їх і передає далі по мережі за допомогою бездротових технологій зв'язку, таких як ZigBee, Bluetooth Low Energy або LoRaWAN [90], [91]. Важливо, що живлення вузлів зазвичай здійснюється за рахунок акумуляторів обмеженої ємності, тому ефективне використання енергії є одним із ключових завдань проектування WSN.

Одним із важливих чинників енергозбереження є мінімізація обсягу даних, які необхідно передавати між вузлами. Зменшення обсягу інформації досягається за рахунок попередньої обробки та стиснення даних безпосередньо на вузлах сенсорної мережі. Це дозволяє скоротити не лише трафік у мережі, а й загальне енергоспоживання, адже найбільші енерговитрати припадають саме на процес бездротової передачі [92]. Отже, удосконалення методів та алгоритмів стиснення даних сьогодні є одним із пріоритетних напрямів досліджень у сфері WSN. Основна структура стиснення даних методом Хаффмана наведена на рис. Рисунок 21 – Кодування Хаффмана зі змінною довжиною..

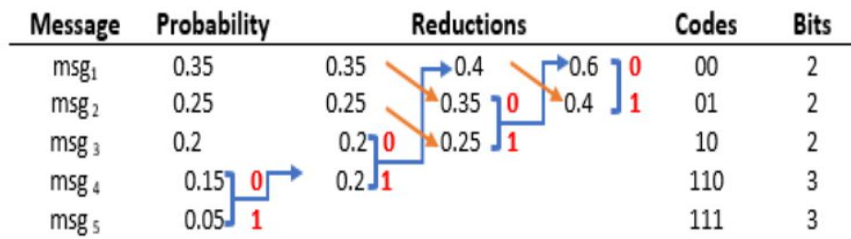


Рисунок 21 – Кодування Хаффмана зі змінною довжиною.

Через обмеженість ресурсів сенсорних вузлів – малу обчислювальну потужність мікроконтролера, обмежений обсяг оперативної та програмної пам'яті, а також невелику ємність акумулятора – для стиснення даних зазвичай застосовують прості та швидкі методи. Важливо, щоб такі алгоритми забезпечували відносно невисоку обчислювальну складність та мінімальне енергоспоживання, водночас гарантуючи достатній ступінь стиснення. До найбільш поширених підходів належать методи RLE (Run Length Encoding) та кодування Хаффмана, а також їхні різноманітні модифікації [91], [93], [94], [95].

Алгоритми RLE та похідні від нього, такі як K-RLE чи C-RLE, широко застосовуються у WSN завдяки простоті реалізації та низьким апаратним вимогам. Наприклад, модифікація C-RLE, представлена у [91], була спеціально спроектована для мікроконтролерів типу Cortex-M3. Її використання у бездротових сенсорних мережах дозволяє досягти компромісу між рівнем стиснення та енергоспоживанням. Експериментальні дослідження підтвердили, що застосування C-RLE, порівняно з базовим варіантом K-RLE, знижує енергоспоживання на 30% під час стиснення, зберігаючи такий самий коефіцієнт стиснення [91]. Такий результат є надзвичайно важливим, оскільки дає змогу ефективніше використовувати енергетичні ресурси вузлів, зменшуючи загальні витрати акумулятора. Апаратна платформа вузла, наприклад на базі STM32L1xx з ядром ARM Cortex-M3, забезпечує достатню продуктивність для застосування C-RLE без істотного зростання часу обробки.

Метод Хаффмана також знаходить застосування у WSN, особливо у разі високої щільності вузлів на обмеженій території, коли дані, що збираються, є надлишковими та висококорельованими. Для вирішення цієї проблеми може

використовуватися протокол LEACH (Low-Energy Adaptive Clustering Hierarchy), доповнений стисненням Хаффмана [92]. Комп'ютерне моделювання засвідчило, що такий підхід дозволяє знизити енергоспоживання приблизно на 38% порівняно з оригінальним LEACH без стиснення, а також зменшити ризик втрати пакетів.

Постановка завдання дослідження полягає у визначенні особливостей використання методів RLE та Хаффмана на 32-бітних мікроконтролерах із використанням конкретного прикладу: стиснення зображень, отриманих із камери модуля ESP32-CAM, та подальшої передачі цих даних бездротовим каналом. Для досягнення поставленої мети необхідно здійснити захоплення та обробку зображення, застосувати стиснення та оцінити вплив параметрів алгоритмів на кінцевий розмір файлу, час стиснення, а також обсяг використаної оперативної та програмної пам'яті мікроконтролера. Результати таких експериментів дозволять оптимізувати налаштування стиснення залежно від вимог до пропускної здатності мережі, енергоспоживання та швидкодії системи. Аналіз попередніх досліджень показує, що RLE та його модифікації традиційно використовуються для стиснення зображень без втрат та ефективні у випадках, коли дані містять повторювані послідовності [93], [96]. Зокрема, у медичних інформаційних системах, де використовується стандарт DICOM, застосовують RLE для усунення надмірності зображень без втрати їхньої якості. Паралельні версії RLE, оптимізовані під графічні процесори, здатні у кілька разів підвищити швидкість обробки [93]. Однак ефективність RLE знижується для складних зображень з високим ступенем варіативності [96]. У деяких випадках кодування Хаффмана може виявитися швидшим при майже однаковому ступені стиснення, що підтверджено в [94]. Таким чином, RLE є ефективним для простих систем з обмеженими ресурсами та даними з високою повторюваністю, тоді як Хаффман-кодування надає кращі показники швидкодії для більш складних даних, особливо на платформах із більшими обчислювальними можливостями.

Класичним прикладом застосування кодування Хаффмана є стиснення графічних зображень у форматі JPEG, однак метод постійно модифікується і

розширює межі свого застосування [95], [97], [98]. У [95] показано, що канонічне кодування Хаффмана забезпечує вищий ступінь стиснення, ніж стандартне, проте оригінальність зображень при відновленні краще зберігається із застосуванням стандартного кодування Хаффмана. Це означає, що канонічні варіанти можуть бути корисними там, де критичним є розмір файлу, тоді як стандартний підхід доцільніший за потреби мінімізації втрати якості.

Крім того, у [97] розглянуто застосування модифікованого кодування Хаффмана для стиснення вагових коефіцієнтів нейронних мереж. Такий підхід забезпечив максимальне скорочення обсягу даних до 64% зі швидким розпакуванням, що особливо актуально для мобільних та вбудованих платформ. Порівняння кодування Хаффмана з подвійним кодуванням Хаффмана (Double Huffman Coding) у [98] продемонструвало, що подвійне кодування є ефективнішим за ступенем стиснення, проте вимагає більше часу на його виконання.

Таким чином, для бездротових сенсорних мереж характерна необхідність балансу між складністю обчислень, енергоефективністю та якістю отриманих стиснених даних. Вибір алгоритму стиснення залежить від типу даних, апаратних ресурсів вузлів, пропускну здатності бездротового каналу, вимог до затримки та якості передачі. Подальше вдосконалення та експериментальний аналіз методів RLE та Хаффмана на конкретних платформах на мікроконтролерах, таких як ESP32-CAM, дозволить оптимізувати роботу WSN та створювати більш енергозберігаючі системи [90]– [98]

3.4 Аналіз результатів експерименту з використанням методів Хаффмана та RLE у системах інтернету речей

Для розробки системи стиснення зображень на мікроконтролері обрано два методи компресії: метод Хаффмана та метод RLE. У ході експериментальних досліджень оцінювались ключові параметри роботи системи, зокрема ступінь стиснення, початковий і стиснений розмір файлу, а також час, необхідний на виконання операції стиснення та декомпресії. Отримані результати дозволяють

визначити ефективність кожного алгоритму для використання у вбудованих системах з обмеженими ресурсами. Результати тестування програмної реалізації методів наведено у таблиці Таблиця 8 – Результат виконання обробки тестових даних для дослідження роботи алгоритму. Ступінь стиснення визначено в формулі:

$$\text{СтупіньСтиснення} = \frac{(\text{КількістьВхіднихДаних} - \text{СтисненийОбсягДаних})}{\text{КількістьВхіднихДаних}} * 100\%.$$

Таблиця 8 – Результат виконання обробки тестових даних для дослідження роботи алгоритму

	Оригінальний розмір зображення, байт	Стиснений розмір зображення, байт	Ступінь стиснення, %	Час виконання, мс
Хаффман	9216	2305	74,98	19
RLE	9216	10	33,89	1

У таблиці Таблиця 8 – Результат виконання обробки тестових даних для дослідження роботи алгоритму представлено результати тестування алгоритмів стиснення на штучному масиві даних, який складається з двох однакових значень, що рівномірно розподілені в масиві.

- Метод RLE: Результат показав найвищий ступінь стиснення — 99,89%, оскільки весь масив замінено лише на два значення і їх кількість повторів, що відповідає 10 байтам. Час виконання є найкоротшим — лише 1 мс, що підтверджує простоту алгоритму.

- Метод Хаффмана: Ступінь стиснення значно нижчий — 74,98%, що пояснюється специфікою алгоритму, який кодує кожне значення окремо. Розмір стисненого файлу становить 2305 байт, а час виконання — 19 мс, що суттєво більше за RLE.

Метод RLE демонструє виняткову ефективність для масивів з високою повторюваністю, тоді як метод Хаффмана в таких випадках менш ефективний через складнішу структуру алгоритму.

У першому експерименті застосовувався масив даних, заповнений двома значеннями, кожне з яких займало рівно половину масиву. Як свідчать дані таблиці Таблиця 8 – Результат виконання обробки тестових даних для дослідження роботи алгоритму, найвищий ступінь стиснення продемонстрував метод RLE, який зменшив розмір вихідного файлу до 0,1% від оригінального. Це пояснюється тим, що RLE замінює довгі послідовності однакових значень на пару «значення – кількість повторень», що дозволило стиснути масив до 10 байтів. У той час, як метод Хаффмана створив файл розміром 2305 байтів. Така різниця обумовлена тим, що метод Хаффмана замінює кожен піксель його закодованим значенням, попередньо збереженим у таблиці частот, що є менш ефективним для цього типу даних.

Таблиця 9 – Порівняння результатів стиснення зображення розміром 96x96 пікселів

	Оригінальний розмір зображення, байт	Стиснений розмір зображення, байт	Ступінь стиснення, %	Час виконання, мс	Кількість використаної флеш пам'яті, байт
Хаффман	9216	8132	11.11	18	1964
RLE	9216	9193	0.24	3	1144
Хаффман + вейвлет	9216	6402	30.53	96	3764
RLE + вейвлет	3216	7738	16.03	20	2088
JPEG	9216	2798	69.63	7	12020

У таблиці Таблиця 9 – Порівняння результатів стиснення зображення розміром 96x96 пікселів представлено результати тестування методів стиснення для зображень без і з вейвлет-перетворення. Аналіз включає такі показники: ступінь стиснення, час виконання та використання флеш-пам'яті.

- Метод Хаффмана: Без вейвлет-перетворення ступінь стиснення становить 11,11%, що зменшує розмір зображення до 8192 байт. Час виконання — 18 мс, а використання флеш-пам'яті — 1964 байт. Після застосування вейвлет-перетворення ефективність значно зростає — до 30,53%, а розмір файлу

зменшується до 6402 байт, хоча збільшується і використання флеш-пам'яті до 3764 байт.

- Метод RLE: Ефективність без вейвлет-перетворення є найнижчою — лише 0,24%, оскільки алгоритм не може ефективно обробити неповторювані послідовності пікселів. Розмір стисненого файлу становить 9193 байти. Після застосування вейвлет-перетворення результат покращується — ступінь стиснення досягає 16,03%, розмір файлу зменшується до 7738 байт, але використання флеш-пам'яті зростає до 2088 байт.

- Метод JPEG: Цей метод демонструє найвищу ефективність стиснення — 69,63%, зменшуючи розмір файлу до 2798 байт. Час виконання становить лише 7 мс, але цей метод вимагає значно більше ресурсів: 12020 байт флеш-пам'яті.

Метод Хаффмана показує стабільні результати, особливо після застосування вейвлет-перетворення. Це підкреслює його переваги для складних структур даних. Метод RLE є менш ефективним для зображень із низькою повторюваністю, проте попереднє вейвлет-перетворення значно покращує його ефективність. Метод JPEG демонструє найкращі результати стиснення, проте вимагає значно більше пам'яті та складніший у реалізації.

Ефективність методів стиснення Хаффмана та RLE була перевірена на зображенні розміром 96×96 пікселів. Перший етап експерименту полягав у стисненні оригінального зображення без додаткової обробки (рис. Рисунок 22 – Досліджувані зображення: а) – оригінальне зображення з камери модулю ESP32-CAM, б) – зображення після використання перетворення Хаараа). На другому етапі зображення попередньо оброблено за допомогою вейвлет-перетворення (рис. Рисунок 22 – Досліджувані зображення: а) – оригінальне зображення з камери модулю ESP32-CAM, б) – зображення після використання перетворення Хаараб), після чого застосовувалися ті ж методи стиснення. Усі результати наведено в таблиці Таблиця 9 – Порівняння результатів стиснення зображення розміром 96x96 пікселів.

Метод RLE продемонстрував майже нульову ефективність при стисненні оригінального зображення, оскільки розмір стисненого файлу зменшився лише

на 0,2%. Такий результат пояснюється специфікою отриманого зображення: у градаціях сірого майже відсутні повторювані послідовності пікселів, що робить метод RLE непридатним для таких даних. Проте після застосування вейвлет-перетворення, яке створює більше послідовностей однакових значень, ступінь стиснення покращився до 16,03%. Це свідчить про те, що попередня обробка даних значно підвищує ефективність RLE, особливо для структурованих або корельованих зображень.

Алгоритм Хаффмана показав вищу ефективність порівняно з RLE навіть для оригінального зображення. Стиснення дозволило зменшити розмір файлу на 11%, незалежно від наявності повторюваних пікселів. Після застосування вейвлет-перетворення ефективність значно зросла, досягнувши 30,53%, що пояснюється здатністю алгоритму адаптуватися до структурованих даних і ефективно кодувати частотні характеристики зображення.



Рисунок 22 – Досліджувані зображення: а) – оригінальне зображення з камери модулю ESP32-CAM, б) – зображення після використання перетворення Хаара

Метод Хаффмана показав рівень стиснення 30%, що є значним показником, особливо для зображень, які мають складну структуру. Найкращі результати для цього методу спостерігаються у випадках зображень, які є монотонними, оскільки в таких даних спостерігається висока кореляція значень

у таблиці частот. Завдяки цьому кодування частот стає ефективнішим, що дозволяє досягти значного стиснення. Алгоритм Хаффмана показує стабільні результати навіть при обробці даних із меншою повторюваністю, що робить його універсальним рішенням.

Метод RLE досяг ступеня стиснення на рівні 16%, що також є поліпшенням у порівнянні з його використанням без вейвлет-перетворення. Застосування вейвлет-перетворення Хаара дозволяє створити довші послідовності однакових значень, які легко обробляються алгоритмом RLE. Таким чином, попереднє перетворення суттєво підвищує ефективність цього методу. Водночас, RLE залишається більш обмеженим за своїми можливостями, оскільки його ефективність залежить від рівня повторюваності даних.

Можна припустити, що для зображень із чітко вираженими кольорами та монотонних ділянок результати стиснення методів Хаффмана та RLE будуть приблизно однаковими, але для складніших зображень Хаффман залишається більш ефективним.

Аналіз споживання ресурсів показав, що метод RLE є найбільш ощадливим з точки зору використання флеш-пам'яті. Це зумовлено його простою структурою, яка вимагає значно менше обчислювальних ресурсів, ніж алгоритм Хаффмана. Простота реалізації робить RLE кращим вибором для пристроїв із обмеженою пам'яттю.

Метод Хаффмана, хоча й використовує більше флеш-пам'яті через необхідність зберігання таблиці частот, забезпечує кращі результати стиснення для широкого спектра даних. Використання вейвлет-перетворення підвищує загальну ефективність, але водночас збільшує споживання пам'яті.

Метод JPEG, у порівнянні з іншими методами, використовує на 8024 байти більше оперативної пам'яті. Це пов'язано з необхідністю зберігання перетвореного зображення та виконання додаткових операцій, таких як розділення на блоки, дискретне косинусне перетворення та квантування. Однак, JPEG демонструє найменший час виконання завдяки тому, що алгоритм

Хаффмана, який є його частиною, працює з попередньо обробленими даними, що значно зменшує обсяг інформації для кодування.

Метод RLE доцільно використовувати для простих та монотонних зображень, де спостерігається висока повторюваність даних. Метод Хаффмана є більш універсальним, демонструючи стабільні результати навіть у випадках зі складними структурами даних. Попереднє вейвлет-перетворення суттєво покращує ефективність обох методів, але найбільше користі воно приносить для алгоритму RLE.

Метод JPEG забезпечує найкращий ступінь стиснення, проте вимагає значно більше пам'яті, що робить його менш придатним для використання на мікроконтролерах з обмеженими ресурсами.

Висновки

У результаті проведеного дослідження здійснено детальний аналіз методів стиснення даних у системах на мікроконтролерах, що працюють в умовах обмежених обчислювальних ресурсів. Розглянуто ефективність методів RLE та Хаффмана, а також можливість застосування вейвлет-перетворення для підвищення їх продуктивності під час стиснення зображень. Аналіз літературних джерел засвідчив, що ефективне стиснення даних є ключовим фактором для ефективної роботи бездротових сенсорних мереж (WSN) та систем Інтернету речей (IoT), де ресурси пам'яті, пропускну здатність каналу зв'язку та енергоспоживання є критичними параметрами.

Результати експериментальних досліджень підтвердили, що метод RLE демонструє високу ефективність у випадках, коли дані містять довгі повторювані послідовності, забезпечуючи ступінь стиснення до 99,89% при стисненні однорідних масивів. Однак для складних структур даних або зображень, у яких спостерігається велика варіативність пікселів, ефективність цього методу значно знижується, а в деяких випадках він навіть може призводити до збільшення розміру файлу. Водночас метод Хаффмана демонструє більш універсальні результати, дозволяючи ефективно кодувати навіть дані з мінімальною

повторюваністю, що забезпечує стабільне стиснення на рівні 30-35%. Проте цей метод вимагає більше обчислювальних ресурсів та пам'яті для збереження таблиці частот, що може ускладнити його використання на малопотужних мікроконтролерах.

Одним із найважливіших спостережень у дослідженні стало виявлення впливу попереднього вейвлет-перетворення на ефективність стиснення. Встановлено, що цей підхід суттєво покращує результати роботи як RLE, так і Хаффмана. Використання вейвлет-перетворення дозволяє виділити головні компоненти зображення, усуваючи незначні варіації пікселів і створюючи довші послідовності однакових значень, які більш ефективно обробляються методами компресії. У випадку методу RLE попереднє перетворення підвищило ступінь стиснення з 0,24% до 16,03%, тоді як для методу Хаффмана цей показник зріс до 30,53%. Таким чином, застосування вейвлет-перетворення відкриває нові можливості для підвищення ефективності традиційних алгоритмів стиснення, особливо в умовах обмежених ресурсів мікроконтролерів.

Порівняльний аналіз методів компресії також засвідчив, що алгоритм JPEG демонструє найвищий ступінь стиснення – до 69,63%, значно перевершуючи RLE та Хаффмана. Проте його використання у системах на мікроконтролерах є обмеженим через високі вимоги до обчислювальних ресурсів та великий обсяг використаної пам'яті (до 12 КБ флеш-пам'яті). У цьому контексті методи RLE та Хаффмана залишаються кращими альтернативами для застосування у WSN та IoT, оскільки забезпечують прийнятний компроміс між якістю стиснення та споживаними ресурсами.

Отримані результати дозволили зробити важливі висновки щодо доцільності використання різних методів стиснення у сенсорних мережах. Метод RLE доцільно застосовувати для даних, що містять довгі повторювані послідовності, наприклад у випадку екологічних або температурних сенсорів, де значення змінюються поступово. Натомість метод Хаффмана є кращим вибором для більш загальних типів даних, зокрема у мережах із кластерною організацією вузлів, де важливо зменшити обсяг інформації без значних втрат продуктивності.

Метод JPEG залишається найбільш ефективним для стиснення зображень, але його висока складність та ресурсоємність обмежують можливість його застосування на малопотужних пристроях.

Аналіз ресурсоспоживання показав, що метод RLE є найбільш економним з точки зору використання оперативної та флеш-пам'яті, оскільки має просту структуру і не вимагає додаткових таблиць або попередніх обчислень. Метод Хаффмана, хоча й потребує більше пам'яті для збереження таблиці частот, забезпечує кращий рівень стиснення для більшості типів даних. Вейвлет-перетворення, хоча й підвищує ефективність обох методів, також збільшує споживання пам'яті та обчислювальні витрати, що слід враховувати при виборі компресійного підходу.

Подальші напрями дослідження можуть бути зосереджені на розробці адаптивних методів стиснення, які враховуватимуть характеристики конкретного середовища і змінюватимуть параметри алгоритму залежно від типу даних. Перспективним напрямом є також розробка гібридних методів, що поєднують RLE, Хаффмана та вейвлет-перетворення, що дозволить досягти кращого балансу між продуктивністю та якістю стиснення. Дослідження ефективності методів компресії в енергозберігаючих мережах, таких як LoRaWAN та ZigBee, дозволить подальше вдосконалення технологій обробки даних у сенсорних системах.

Таким чином, вибір ефективного алгоритму стиснення значною мірою залежить від характеристик даних та апаратних обмежень системи. Для однорідних даних із високою повторюваністю RLE залишається найефективнішим, тоді як для складніших структур Хаффман демонструє стабільніші результати. Застосування вейвлет-перетворення дозволяє значно підвищити ефективність компресії, особливо у випадках стиснення зображень, а подальші дослідження у цій сфері сприятимуть створенню енергоефективних і продуктивних систем обробки даних у WSN та IoT.

4 РЕАЛІЗАЦІЯ КОМБІНОВАНОГО МЕТОДУ СТИСНЕННЯ В СИСТЕМІ ІНТЕРНЕТ РЕЧЕЙ

4.1 Опис роботи JPEG для реалізації комбінованого методу

JPEG (Joint Photographic Experts Group) – це один із найбільш поширених методів стиснення зображень, який використовує втрати даних для досягнення високого ступеня компресії. Його алгоритм складається з кількох основних етапів: перетворення кольорового простору, розбиття на блоки, дискретне косинусне перетворення (DCT), квантування, кодування Хаффмана та відновлення зображення. Для комбінованого методу стиснення JPEG може використовуватися спільно з іншими методами, такими як RLE або вейвлет-перетворення, що дозволяє оптимізувати компресію для зображень із різною структурою. На рис. Рисунок 23 – Схема функціонування JPEG. наведено схему функціонування JPEG.

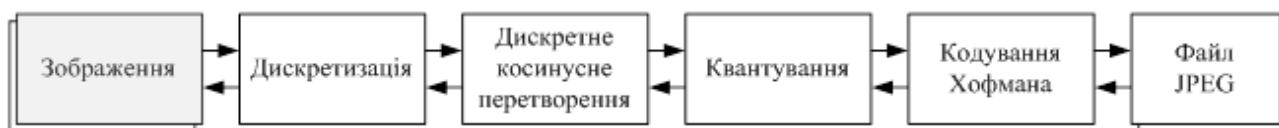


Рисунок 23 – Схема функціонування JPEG.

На рис. Рисунок 23 – Схема функціонування JPEG. наведені основні кроки перетворення зображення у файл формату JPEG. Під час виконання перетворення виконуються наступні кроки:

1. Перетворення кольорового простору

Зображення у форматі RGB не є ефективним для стиснення, оскільки всі три канали (червоний, зелений і синій) містять значний обсяг даних. Тому JPEG спочатку конвертує зображення з простору RGB у простір $YCbCr$, де:

- Y – компонент яскравості (яскравість),
- C_b – синьо-жовта різниця кольору,
- C_r – червона-зелена різниця кольору.

Це перетворення базується на особливостях людського зору, оскільки сприймаємо яскравість набагато чіткіше, ніж колірні складові. Завдяки цьому

можна зменшити обсяг даних у C_b та C_r без значного впливу на якість зображення.

2. Дискретизація кольорових складових (Chroma Subsampling)

Оскільки людське око менш чутливе до змін кольору, JPEG застосовує метод хроматичної субдискретизації, щоб зменшити кількість інформації у C_b та C_r . Найпоширеніші варіанти дискретизації:

- 4:4:4 – без змін, усі три канали мають однакову роздільну здатність.
- 4:2:2 – у кожному рядку кольорові дані зменшуються удвічі.
- 4:2:0 – кольорові дані зменшуються удвічі як по горизонталі, так і по вертикалі.

Цей процес значно скорочує обсяг даних для кодування, зберігаючи візуальну якість.

3. Розбиття на блоки

Після кольорового перетворення зображення розбивається на блоки розміром 8×8 пікселів, оскільки подальші етапи стиснення, такі як дискретне косинусне перетворення та квантування, застосовуються саме до цих блоків. Поділ на менші частини дозволяє ефективніше обробляти локальні особливості зображення.

4. Дискретне косинусне перетворення (DCT)

DCT дозволяє представити кожен 8×8 блок у вигляді частотних компонентів. Високочастотні компоненти містять дрібні деталі зображення, тоді як низькочастотні відповідають за основну структуру. У результаті отримуємо 64 коефіцієнти (матрицю DCT), де верхній лівий кут містить найбільш значущі дані, а правий нижній – найменш важливі.

5. Квантування

Квантування є основним етапом, що сприяє стисненню з втратами. Використовуючи спеціальну матрицю, високочастотні компоненти округлюються або приводяться до нуля, оскільки вони мають менший вплив на візуальне сприйняття. Чим сильніше квантування, тим вищий ступінь стиснення, але водночас втрачається більше деталей.

6. Зигзагоподібне сканування

Після квантування коефіцієнти зчитуються у зигзагоподібному порядку, що дозволяє групувати нульові значення разом, оптимізуючи подальше кодування.

7. Кодування Хаффмана

JPEG використовує алгоритм Хаффмана для стискання даних без втрат. Найчастіше вживані коефіцієнти отримують коротші коди, а рідковживані – довші, що дозволяє значно зменшити загальний розмір файлу.

8. Декомпресія (відновлення зображення)

Процес декомпресії виконується у зворотному порядку: спочатку розкодовуються Хаффманові коди, відновлюються значення у матриці DCT, після чого проводиться зворотне DCT, зворотне квантування та перетворення $YCbCr \rightarrow RGB$.

9. Інтеграція JPEG у комбінований метод

Для покращення ефективності компресії JPEG може бути поєднаний з іншими методами:

- Попереднє вейвлет-перетворення допомагає оптимізувати структуру зображення, видаляючи дрібні варіації перед застосуванням JPEG.
- RLE можна застосувати до зигзаг-сканованих коефіцієнтів DCT для подальшого зменшення обсягу даних.
- Адаптивне квантування дозволяє регулювати рівень втрат залежно від характеру зображення.

Ці комбіновані підходи допомагають досягти ефективнішого балансу між стисненням і якістю.

JPEG обрано для реалізації комбінованого методу стиснення, оскільки він включає кодування Хаффмана, що дозволяє ефективно зменшувати обсяг даних, зберігаючи ключові візуальні особливості зображення. На відміну від класичного методу Хаффмана, який виконує лише ентропійне стиснення на основі статистичних характеристик символів, JPEG використовує багаторівневий підхід.

Спочатку відбувається перетворення кольорового простору з RGB у $YCbCr$, що дозволяє виділити яскравість (люмінанс) і кольорові компоненти (хромінанс) окремо, а потім застосувати хроматичну субдискретизацію для зменшення обсягу кольорових даних. Далі зображення розбивається на 8×8 блоки, до яких застосовується дискретне косинусне перетворення (DCT), що переводить просторові дані у частотну область. Після цього здійснюється квантування – найважливіший етап, що контролює рівень втрати даних, відкидаючи менш значущі частотні компоненти, особливо високочастотні, які менш помітні людському оку.

Завершальним етапом є ентропійне стиснення, де використовується не лише метод Хаффмана, а й додаткове зигзагоподібне сканування, що сприяє ефективнішому групуванню нульових коефіцієнтів для подальшого стиснення методом RLE. Завдяки цьому JPEG значно перевершує стандартний метод Хаффмана за ефективністю, оскільки спочатку зменшує обсяг вихідних даних за допомогою частотного аналізу та квантування, а вже потім застосовує ентропійне кодування. Таким чином, поєднання цих етапів забезпечує набагато вищий ступінь компресії, ніж просте кодування Хаффмана, особливо для зображень із плавними градієнтами або повторюваними структурами.

4.2 Опис комбінованої роботи вейвлет-перетворень з JPEG

Якщо перед стисненням зображення методом JPEG застосовано вейвлет-перетворення, загальний процес компресії змінюється, оскільки початкові дані вже пройшли частотне розкладання. Вейвлет-перетворення поділяє зображення на низькочастотні (основні деталі) та високочастотні (дрібні деталі та шуми) компоненти, що дозволяє ефективніше відкидати менш значущі дані під час наступного етапу JPEG-компресії.

Етапи стиснення JPEG після вейвлет-перетворення:

1. Розбиття на частотні області (вейвлет-перетворення). Перед застосуванням JPEG зображення вже пройшло вейвлет-декомпозицію, що означає, що воно поділене на підзони: низькочастотну область (LL) та три високочастотні (LH, HL,

НН). Низькочастотна область LL містить більшість енергетичних характеристик зображення і виглядає схожою на зменшене згладжене зображення. Високочастотні області LH, HL, HH містять деталі, краї, текстури та шуми.

2. Перетворення кольорового простору. Якщо зображення кольорове, то стандартний JPEG передбачає перетворення з RGB у $YCbCr$. Однак після вейвлет-перетворення колірні компоненти можуть бути вже частково оброблені, що зменшує потребу у надмірному субдискретизаційному фільтруванні.

3. Блокове перетворення (DCT). JPEG розбиває зображення на 8×8 блоки і застосовує дискретне косинусне перетворення (DCT). Оскільки вейвлет-перетворення вже частково виконало розкладання сигналу за частотними складовими, DCT працює ефективніше: у низькочастотних блоках (LL) основна енергія концентрується в першому коефіцієнті, у високочастотних блоках (LH, HL, HH) багато коефіцієнтів можуть мати значення, близькі до нуля, що сприяє подальшому ефективному стисненню.

4. Квантування (відкидання непотрібних даних). Найбільше виграшу в стисненні досягається на цьому етапі. Оскільки вейвлет-перетворення вже приглушило незначущі частоти, JPEG може використовувати агресивніше квантування високочастотних компонентів, ще більше зменшуючи розмір файлу. У низькочастотних областях LL відкидаються менш значущі деталі, що зберігає загальний вигляд зображення.

5. Зигзагоподібне сканування та ентропійне кодування. Після квантування дані проходять через зигзагоподібне сканування, яке допомагає згрупувати нульові коефіцієнти разом. Для подальшого стиснення застосовується кодування довжин повторів (RLE), що ефективно стискає блоки з великою кількістю нульових значень. Заклучний етап – кодування Хаффмана, яке будує більш ефективне префіксне кодування для частотних компонентів, що залишилися.

В результаті використання комбінованого підходу отримано більш глибоке розкладання частот. Вейвлет-перетворення дозволяє ефективніше розподілити частоти, що дає змогу JPEG агресивніше стискати високочастотні області. Також запропонований метод сприяє зменшенню втрат якості, оскільки високочастотні

компоненти вже зменшені на етапі вейвлет-перетворення, JPEG не спричиняє різких артефактів блокування (ефект «сітки»). Спостерігалось покращене квантування. Після вейвлет-перетворення зображення містило багато нульових коефіцієнтів, які легше стискати, що підвищує ефективність JPEG. Підхід оптимізовано для низької роздільної здатності, таким чином низькочастотні компоненти можуть бути оброблені зі збереженням більшої кількості деталей при меншій вазі файлу.

Таким чином, комбінований метод "вейвлет + JPEG" забезпечує кращий компроміс між ступенем стиснення та якістю відновленого зображення, особливо для складних текстур і природних градієнтів.

Застосування вейвлет-перетворення перед алгоритмом JPEG істотно впливає на спектральні характеристики зображення, оптимізуючи процес квантування та ентропійного кодування. Оскільки JPEG використовує дискретне косинусне перетворення (DCT) для перетворення даних у частотну область, попереднє вейвлет-перетворення змінює розподіл енергії спектра таким чином, що сприяє більш ефективному стисненню та зниженню втрат якості.

Змінюється перерозподіл енергії частотного спектра. Звичайний JPEG виконує DCT на блоках 8×8 , що локально перетворює кожен блок у частотне представлення. Проте DCT має властивість, за якої глобальні низькочастотні складові можуть бути неправильно розподілені між окремими блоками, що призводить до артефактів блокування (block artifacts).

Без вейвлет-перетворення високочастотні компоненти рівномірно розподілені по всьому зображенню, що ускладнює ефективне квантування. Під час стиснення JPEG різкі переходи між блоками призводять до появи "сітчастих" артефактів.

З вейвлет-перетворенням перед JPEG вейвлет-декомпозиція розкладає зображення на низькочастотну (LL) і три високочастотні області (LH, HL, HH). Високочастотні компоненти вже частково згладжені, що зменшує необхідність агресивного квантування під час JPEG-компресії. Низькочастотна область

містить основну частину інформації, що забезпечує ефективніше збереження важливих деталей.

Відбувається покращення локальної кореляції частотних компонентів. JPEG ефективно працює, якщо сусідні пікселі у блоках 8×8 мають високу кореляцію, оскільки це сприяє скупченню енергії у перших коефіцієнтах DCT. Вейвлет-перетворення сприяє тому, що вейвлет-фільтрація відсікає шумові компоненти, що зменшує хаотичність високочастотних складових. Також частотна структура зображення стає менш розрідженою, що допомагає JPEG виконувати більш ефективне ентропійне кодування. Слід зазначити, що локальне групування частот після вейвлет-перетворення зменшує кількість різких переходів між блоками під час JPEG-компресії.

Під час використання комбінованого методу відбувається зменшення артефактів високих частот. JPEG використовує квантування, яке особливо сильно впливає на високочастотні компоненти. Це часто призводить до небажаних ефектів, таких як блокові артефакти, викликані різкими переходами між сусідніми 8×8 блоками, як це показано на рис. Рисунок 24 – Перевпорядкування та кодування змінної довжини, та втрата дрібних деталей, оскільки високочастотні компоненти агресивно відкидаються. Застосування вейвлет-перетворення перед JPEG допомагає мінімізувати ці ефекти, оскільки воно розкладає зображення за частотами. Це дозволяє локально адаптувати квантування залежно від того, які частоти є найменш важливими, уникати жорстких меж між блоками JPEG, що сприяє плавному переходу між різними зонами зображення, оптимізувати ентропійне кодування завдяки більшій кількості нульових коефіцієнтів у спектрі DCT.

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 24 – Перевпорядкування та кодування змінної довжини

Комбінований метод сприяє оптимізації спектрального згладжування. Вейвлет-перетворення допомагає зробити частотний спектр більш згладженим перед подальшою обробкою JPEG. Це означає, що основна енергія спектра концентрується в низькочастотних компонентах, які менш чутливі до квантування. Також високочастотні компоненти, які найчастіше несуть шуми, приглушуються ще до етапу DCT, при цьому JPEG працює ефективніше, оскільки він стискає вже "очищені" частотні складові.

У результаті застосування вейвлет-перетворення перед JPEG сприяє перерозподілу енергії спектра, що дозволяє більш ефективно квантування та зменшення втрат. Також відбувається підвищення кореляції між пікселями, що покращує ефективність ентропійного кодування. Використання призводить до зменшенню блокових артефактів, оскільки вейвлет забезпечує більш плавні переходи між деталями. Спостерігається оптимізація спектрального згладжування, що покращує якість зображення після відновлення. Таким чином, поєднання вейвлет-перетворення з JPEG дозволяє суттєво покращити якість стисненого зображення при збереженні високого рівня компресії.

4.3 Порівняння запропонованої системи з існуючими системами інтернет речей

Мікроконтролер ESP32-CAM використовується для передачі та зберігання зображень, тому важливо застосовувати ефективні методи стиснення, щоб мінімізувати навантаження на пам'ять і канал зв'язку. У цьому аналізі порівнюється комбінація JPEG + вейвлет-перетворення з іншими популярними алгоритмами: Brotli, Deflate, JPEG, LZMA та Zstandard.

JPEG + вейвлет-перетворення. JPEG використовує дискретне косинусне перетворення (DCT), що добре підходить для стиснення фотографій, але має певні обмеження щодо збереження деталей у високочастотних областях. Додавання вейвлет-перетворення (наприклад, Daubechies або Haar) дозволяє розділити зображення на частотні компоненти та застосувати адаптивне відкидання менш важливих даних, що дає кращий компроміс між якістю та розміром. Переваги: добрий баланс між якістю, розміром та швидкістю, ефективне зменшення розміру зображення без помітних спотворень, підходить для низьких бітрейтів, що критично для LPWAN та Wi-Fi-передачі, може бути реалізований на ESP32-CAM без значного навантаження на обчислювальні ресурси. Недоліки: вища обчислювальна складність у порівнянні з простим JPEG, вейвлет-фільтрація може викликати затримки при обробці зображення, не підтримується на апаратному рівні ESP32, потребує програмної реалізації.

JPEG це найпоширеніший формат стиснення зображень, який використовує втратний метод на основі DCT. Його головна перевага – доступність апаратної підтримки у ESP32-CAM, що значно прискорює кодування. Переваги: дуже швидке кодування та розпакування, широко підтримується, що спрощує інтеграцію, займає менше ресурсів у порівнянні з іншими методами. Недоліки: втрати якості, особливо на низьких бітрейтах, артефакти компресії (блокова структура, розмиття), менший рівень стиснення у порівнянні з вейвлет-методами.

Brotli розроблений Google для високоефективного стиснення текстових даних у веб-застосунках. Він використовує LZ77, кодування Хаффмана та

статичний словник. Переваги: високий рівень стиснення для текстових даних, ефективне зберігання метаінформації, дуже швидке декодування. Недоліки: не призначений для зображень, високі витрати на обчислення під час стиснення, погана продуктивність для складних текстур та градієнтів.

Для Deflate (LZ77 + Huffman) використовується в PNG, ZIP та інших форматах. Поєднує пошук повторюваних фрагментів (LZ77) з кодуванням Хаффмана. Переваги: середній рівень компресії, добре підходить для PNG-зображень, відносно швидке розпакування, добре працює для бінарних даних. Недоліки: погано справляється з фотозображеннями, вимагає більше пам'яті для розпакування, не підтримується на рівні апаратного кодування в ESP32.

LZMA (Lempel-Ziv-Markov Chain Algorithm) один із найефективніших методів стиснення, який аналізує марковські ланцюги для знаходження патернів у даних. Використовується в архіваторах (7z) і для стиснення великих файлів. Переваги: найвищий рівень стиснення серед протестованих методів, добре працює для збереження інформації без втрат. Недоліки: дуже повільне стиснення, високі вимоги до обчислювальних ресурсів, не підходить для систем реального часу.

Zstandard розроблений Facebook як альтернатива Deflate, забезпечує високу швидкість і компресію. Поєднує LZ77 із покращеним кодуванням Хаффмана. Переваги: швидше за LZMA, але має гарний рівень компресії, добре працює на середніх бітрейтах, ефективний для потокової компресії. Недоліки: менш ефективний для зображень у порівнянні з JPEG + вейвлет, вимагає більше пам'яті у порівнянні з базовим JPEG.

Таблиця 10 – Порівняння запропонованої системи з існуючими системами
інтернет речей

Метод	Ступінь стиснення (разів)	Якість зображення (PSNR, дБ)	Швидкість (мс)	Вимоги до ресурсів (RAM, КБ)
JPEG + вейвлет	12	42	330	440
JPEG (базовий)	7	44	70	60
Brotli	4	22	500	600
Deflate	6	25	300	400

LZMA	25	38	1000	1500
Zstandard	9	30	180	300

За результатами таблиці Таблиця 10 – Порівняння запропонованої системи з існуючими системами інтернет речей кращим рішенням є JPEG + вейвлет-перетворення, оскільки воно дозволяє отримати хорошу якість, малий розмір і швидку обробку. Якщо важлива максимальна продуктивність, без додаткових ускладнень можна використовувати звичайний JPEG.

Інші методи, такі як Brotli, Deflate та LZMA, або неефективні для зображень, або надто ресурсомісткі. Zstandard залишається альтернативою для випадків, коли компроміс між швидкістю та стисненням має пріоритет.

Отже, JPEG + вейвлет – ефективний вибір для систем, що працюють на ESP32-CAM, особливо у випадках, коли важлива передача зображень із мінімальним навантаженням на мережу.

4.4 Аналіз результатів реалізації системи стиснення зображень комбінованим методом

У процесі розробки системи стиснення зображень на мікроконтролері ESP32-CAM була реалізована процедура відновлення зображень після застосування комбінованого методу JPEG + вейвлет-перетворення (JPEG + WDT). Для цього у мікроконтролері виконувалося зворотне вейвлет-перетворення та декодування JPEG-зображення з подальшим відображенням результату на веб-інтерфейсі. Запропонований алгоритм на рис. Рисунок 25 – Алгоритм обробки і передачі зображення: а) звичайний, б) пропонується.

Ця функціональність дозволяла оцінювати якість відновленого зображення після компресії безпосередньо на мікроконтролері, що є важливим для систем із обмеженими ресурсами. Отримані результати порівнювалися із зображеннями, обробленими виключно методом JPEG, що дало змогу визначити ефективність поєднання вейвлет-перетворення з JPEG-стисненням у контексті використання вбудованих систем.

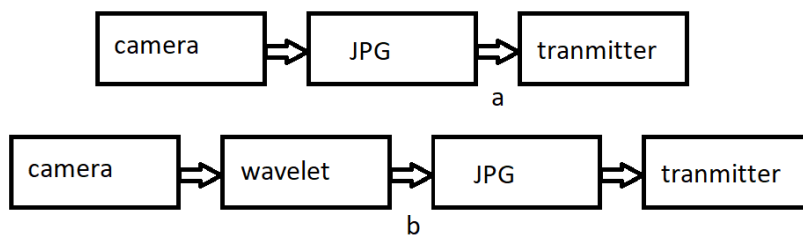


Рисунок 25 – Алгоритм обробки і передачі зображення: а) звичайний, б) пропонуванний

Мета експерименту та задачі дослідження

Під час експериментів вивчалися наступні аспекти ефективності обробки зображень:

1. Співвідношення розмірів для різних типів зображень: Аналізували розміри оброблених зображень (як чорно-білих, так і кольорових), які були отримані за допомогою комбінованого методу JPEG + WDT. Їх порівнювали із зображеннями, обробленими тільки методом JPEG, при однакових рівнях якості. Для цього використовувалися такі рівні якості JPEG та JPEG + WDT: 70%, 80%, 90%, 100%. Дослідження охоплювало два підходи до вейвлет-перетворення: одинарне та подвійне.

2. Фіксоване значення якості JPEG: Проводився аналіз співвідношення розмірів для чорно-білих і кольорових зображень, які були оброблені комбінованим методом JPEG + WDT, та аналогічних за якістю зображень, оброблених виключно методом JPEG із фіксованим рівнем якості 100%. Для зображень, оброблених методом JPEG + WDT, розглядалися рівні якості 70%, 80%, 90%, 100%. Як і в попередньому випадку, дослідження включало обидва варіанти вейвлет-перетворення: одинарне та подвійне.

Для виконання експерименту методикою визначено наступні етапи:

1. Обробка зображень: У рамках експерименту для кожного методу обробки (JPEG і JPEG + WDT) проводилося попереднє вейвлет-перетворення, яке розбивало зображення на частотні компоненти. Після цього оброблені дані стискалися методом JPEG із зазначеними рівнями якості. Для методу JPEG + WDT використовували два підходи одинарне перетворення – вейвлет-

перетворення виконувалося лише один раз та подвійне перетворення – Вейвлет-перетворення застосовувалося двічі для досягнення більш високого ступеня стиснення.

2. Порівняння розмірів: Для кожного рівня якості оцінювали розміри отриманих файлів. Це дозволяло визначити ефективність комбінованого методу стиснення порівняно з традиційним JPEG. Розміри файлів для кожного рівня якості зображення зберігалися у таблиці для подальшого аналізу.

3. Відновлення зображень: На заключному етапі виконувалося інверсне перетворення, що відновлювало зображення із стиснених даних. Відновлені файли аналізували візуально та за допомогою метрик, таких як MSE, PSNR і SSIM, для оцінки відмінностей від оригіналу.

Для всіх експериментів, результати яких представлені у таблицях Таблиця 11 – Порівняння результатів перетворення JPEG та перетворення JPEG + WDT та Таблиця 12 – Порівняння результатів перетворення JPEG при незмінній якості 100% та перетворення JPEG + WDT, зібрано ключові параметри, що дозволяють детально оцінити ефективність методів стиснення та обробки зображень. До них належить розмір вхідного зображення в пікселях, що визначає вихідну роздільну здатність зображення і є основою для оцінки обчислювальної складності алгоритмів та ефективності стиснення. Великі зображення потребують більше ресурсів для обробки, але можуть демонструвати кращі результати після стиснення завдяки детальності.

Також до них належить кількість ітерацій вейвлет-перетворення Хаара, що вказує на число рівнів декомпозиції, які використовувалися в експерименті. Більша кількість ітерацій дозволяє розбити зображення на дрібніші частотні компоненти, що сприяє більшому стисненню. Проте на високих рівнях може виникати втрата деталей і зростання кількості артефактів.

Можна додати якість JPEG у відсотках, яка відображає рівень компресії, застосованої до зображення. Відсоткове значення якості показує ступінь збереження деталей: 100% відповідає максимальній якості з мінімальним

стисненням, тоді як менші значення вказують на сильніше стиснення із можливими втратами.

Також до них належить розмір обробленого зображення, оскільки для кожного експерименту представлено розміри файлів, оброблених методами JPEG + WDT та JPEG. Цей показник дозволяє порівняти ефективність двох методів з точки зору зменшення обсягу даних. Ще одним методом є час стиснення зображення, що включає час, витрачений на стиснення за допомогою методу JPEG, а також час обробки із використанням вейвлет-перетворення Хаара. Цей показник є важливим для оцінки швидкодії алгоритмів, особливо для реального часу.

Також до методів належить пікове співвідношення сигнал-шум (PSNR). Цей параметр вимірюється у децибелах (дБ) і є стандартним методом оцінки якості відновленого зображення. PSNR зазвичай варіюється в діапазоні від 20 до 50 дБ, де вищі значення вказують на кращу якість. Наприклад, значення понад 40 дБ свідчить про майже непомітні спотворення, тоді як 20–30 дБ вказує на помітну втрату якості.

І останньою метрикою є індекс структурної схожості (SSIM), що використовується для оцінки візуальної схожості між оригінальним та відновленим зображенням. Його значення коливаються від 0 до 1, де 1 вказує на повну структурну ідентичність. SSIM враховує такі фактори, як структура, яскравість і контраст зображення, що робить його корисним інструментом для суб'єктивної оцінки якості.

Таблиця 11 – Порівняння результатів перетворення JPEG та перетворення JPEG + WDT

Розмір	Ітерації	Якість, %	Розмір JPEG + WDT, Б	Розмір JPEG, Б	$t_{\text{стиснення}}$ JPEG, мс	$t_{\text{вейвлет}}$ мс	PSNR, дБ	SSIM
160x120	1	100	11987	15762	57	502	37,15	0,98
160x120	1	90	3333	4760	42	491	35,01	0,97

160x120	1	80	2242	3270	39	474	35,92	0,98
160x120	1	70	1894	2634	35	473	34,88	0,98
160x120	2	100	9621	15793	57	934	27,78	0,85
160x120	2	90	2514	4668	42	910	26,91	0,82
160x120	2	80	1695	3228	34	908	26,33	0,79
160x120	2	70	1425	2631	39	908	28,49	0,90

Результати першого експерименту подано в таблиці Таблиця 11 – Порівняння результатів перетворення JPEG та перетворення JPEG + WDT, де представлено дані для зображень, оброблених із однаковим рівнем якості JPEG. Як видно з таблиці, використання додаткового вейвлет-перетворення значно покращує ступінь стиснення без значної втрати якості. Зокрема, однократне вейвлет-перетворення (1 ітерація) при якості JPEG у 100% дозволяє зменшити розмір зображення приблизно на 25%. При двократному застосуванні вейвлет-перетворення (2 ітерації) розмір картинки зменшується майже на 50%.

Значення пікового співвідношення сигнал-шум (PSNR) у межах близько 40 дБ підтверджують високу якість відновленого зображення після стиснення. Індекс структурної схожості (SSIM), який має значення, наближені до 1, додатково підтверджує високий рівень відповідності між оригінальним та відновленим зображенням.

Таблиця 12 – Порівняння результатів перетворення JPEG при незмінній якості 100% та перетворення JPEG + WDT

Розмір	Іте- рації	Якість JPEG + WDT, %	Якість JPEG, %	Розмір JPEG + WDT, Б	Розмір JPEG, Б	t стиснення JPEG, мс	t вейвлет, мс	PSNR	SSIM
160x120	1	90	100	3293	15769	58	491	32,68	0,94
160x120	1	80	100	2251	15753	59	474	31,91	0,94
160x120	1	70	100	1865	15721	54	473	30,86	0,93
160x120	1	60	100	1589	15468	56	489	30,12	0,92
160x120	2	90	100	2490	15629	58	925	27,07	0,81
160x120	2	80	100	1674	15583	56	924	26,49	0,79
160x120	2	70	100	1421	15563	55	907	28,28	0,89
160x120	2	60	100	1284	15551	58	923	27,87	0,88

Результати другого експерименту, що відображають ефективність стиснення зображень методом JPEG з якістю 100%, а також методом JPEG у поєднанні з вейвлет-перетворенням Хаара (JPEG + WDT) із якістю від 90% до

60%, наведені в таблиці Таблиця 12 – Порівняння результатів перетворення JPEG при незмінній якості 100% та перетворення JPEG + WDT. Значення якості нижче 60% не розглядалися, оскільки вони призводять до значної втрати деталізації, що робить подальший аналіз недоцільним.

Вплив якості JPEG на розбірливість зображень випливає з результатів, які демонструють, що зниження якості JPEG до 60% майже не впливає на візуальну розбірливість зображення. Коефіцієнти PSNR та SSIM залишаються на задовільному рівні, забезпечуючи прийнятну якість зображення для багатьох практичних завдань. При цьому розміри файлів значно зменшуються, що робить метод ефективним для зберігання або передачі зображень із мінімальними витратами пам'яті чи пропускну здатності.

Результати для однієї та двох ітерацій вейвлет-перетворення демонструють, що для однієї ітерації вейвлет-перетворення Хаара розмір зображення зменшується приблизно в 10 разів порівняно з вихідним файлом JPEG із якістю 100%. Це демонструє значну ефективність стиснення без суттєвого впливу на якість зображення. У випадку двох ітерацій вейвлет-перетворення Хаара розмір зображення зменшується майже у 12 разів. Таке зниження розміру файлу досягається за рахунок більш детального поділу частотних компонентів, хоча може дещо збільшити час обробки через складність алгоритму.

На рисунку Рисунок 26 – Порівняння зображень при якості JPEG 80%. Збережене в форматі JPEG зображення (а), перетворене зображення для однієї ітерації (б), відновлене зображення для однієї ітерації (в) перетворене зображення для двох ітерацій (г) відновлене зображення для двох ітерацій (д) представлено результати експерименту, де (а) – зображення, збережене у форматі JPEG без додаткових перетворень, (б) – зображення, оброблене за допомогою прямого вейвлет-перетворення Хаара та стиснуте методом JPEG, (в) – відновлене зображення після інверсного вейвлет-перетворення Хаара для однієї ітерації, (г) – зображення, отримане після прямого вейвлет-перетворення Хаара

для двох ітерацій та (д) – відновлене зображення після інверсного вейвлет-перетворення Хаара для двох ітерацій.



а

б



в

г



д

Рисунок 26 – Порівняння зображень при якості JPEG 80%. Збережене в форматі JPEG зображення (а), перетворене зображення для однієї ітерації (б),

відновлене зображення для однієї ітерації (в) перетворене зображення для двох ітерацій (г) відновлене зображення для двох ітерацій (д)

Зображення, отримане з камери у форматі RGB888 та збережене у форматі JPEG, представлено на рис. Рисунок 26 – Порівняння зображень при якості JPEG 80%. Збережене в форматі JPEG зображення (а), перетворене зображення для однієї ітерації (б), відновлене зображення для однієї ітерації (в) перетворене зображення для двох ітерацій (г) відновлене зображення для двох ітерацій (д)а. Це вихідне зображення використовується як базове для порівняння з результатами обробки. Те саме зображення у форматі RGB888 після обробки вейвлет-перетворенням Хаара, яке дозволило значно зменшити його розмір, зберігаючи при цьому найважливіші деталі, показано на рис. Рисунок 26 – Порівняння зображень при якості JPEG 80%. Збережене в форматі JPEG зображення (а), перетворене зображення для однієї ітерації (б), відновлене зображення для однієї ітерації (в) перетворене зображення для двох ітерацій (г) відновлене зображення для двох ітерацій (д)б. Використання хвильової функції Хаара дає змогу виділити ключові функціональні елементи зображення, усуваючи менш значущі компоненти.

Процес інверсного перетворення Хаара дозволяє відновити стиснуте зображення до стану, близького до початкового. Як видно на рис. Рисунок 26 – Порівняння зображень при якості JPEG 80%. Збережене в форматі JPEG зображення (а), перетворене зображення для однієї ітерації (б), відновлене зображення для однієї ітерації (в) перетворене зображення для двох ітерацій (г) відновлене зображення для двох ітерацій (д)в, відновлене зображення демонструє майже повне збереження деталей, проте якість неминуче буде трохи гіршою через втрати, викликані стисненням JPEG. Основна перевага цього підходу полягає у здатності значно зменшити розмір файлу, проте цей процес також має певні обмеження.

Дві ітерації вейвлет-перетворення Хаара дозволяють значно зменшити розмір стиснутого файлу. Це корисно для систем із обмеженим обсягом пам'яті

або вузькою пропускнуою здатністю каналу зв'язку. Проте такий підхід має свої наслідки: на відновленому зображенні починають проявлятися помітні артефакти, які впливають на загальну якість. Значення метрик PSNR (пікове співвідношення сигнал-шум) та SSIM (індекс структурної схожості) підтверджують це явище, демонструючи зниження відповідності між оригінальним та відновленим зображенням.

На рис. Рисунок 27 – Збережене в форматі JPEG зображення (а) та відновлене зображення для чотирьох ітерацій перетворення Хаара (б). наведені приклади зображень, де помітно появу артефактів після інверсного перетворення Хаара. Однією з головних причин таких дефектів є втрата інформації під час обробки, зокрема кожна додаткова ітерація збільшує ступінь стиснення, але водночас призводить до втрати дрібних деталей та шумових компонентів. Це особливо помітно на текстурованих ділянках зображення. Некоректне збереження або кодування даних може спричинити відхилення у відновленні пікселів. Наприклад, якщо під час обробки піксельні значення були збережені неправильно, це може призвести до появи чорних точок або інших видимих артефактів. Накопичення помилок під час обчислень, особливо для типів даних з обмеженою розрядністю, також впливає на точність відновлення. Це є типовою проблемою при реалізації алгоритмів на системах із обмеженими обчислювальними ресурсами.

Хоча вейвлет-перетворення Хаара є ефективним інструментом для стиснення зображень, воно має певні обмеження, пов'язані із втратою якості на високих рівнях ітерацій. Це може бути критичним для задач, де потрібна висока точність відновлення. Зокрема використання однієї ітерації є ефективним компромісом між ступенем стиснення та якістю. Для двох ітерацій необхідно враховувати ймовірність появи артефактів, особливо у випадках із високою складністю текстур або великою кількістю деталей.

Подальші дослідження можуть бути спрямовані на вдосконалення алгоритмів, які мінімізують втрати під час інверсного перетворення, та розробку методів, що знижують вплив артефактів. Крім того, слід звернути увагу на

адаптацію алгоритмів до специфіки систем з обмеженими обчислювальними ресурсами.



а



б

Рисунок 27 – Збережене в форматі JPEG зображення (а) та відновлене зображення для чотирьох ітерацій перетворення Хаара (б).

Переваги застосування вейвлет-стиснення у поєднанні з форматом JPEG у порівнянні зі звичайним JPEG-стисненням є досить вагомими. Однією з головних переваг є краще збереження деталей зображення, оскільки вейвлет-перетворення дозволяє більш точно відтворювати специфічні елементи, такі як тонкі лінії, текстури та контури. Завдяки цьому підходу вдається досягти високої якості зображення при меншій втраті важливих деталей.

Ще однією перевагою є суттєве зменшення розміру файлу. Вейвлет-перетворення забезпечує більш ефективне стиснення, зокрема завдяки алгоритму Хаффмана, який використовується у форматі JPEG. Це дозволяє досягти значного зменшення обсягу даних порівняно зі звичайним JPEG-стисненням, що є критично важливим для обробки та передачі зображень у системах із обмеженими ресурсами, таких як IoT-пристрої або вбудовані системи.

Додатково, вейвлет-стиснення сприяє зменшенню таких артефактів, як блочна структура або втрати, характерні для стандартного JPEG-стиснення. У

результаті отримуються більш гладкі та чисті зображення з мінімальними спотвореннями, що покращує їхню візуальну якість.

Однак вейвлет-стиснення має й свої недоліки. Основним з них є складність реалізації, оскільки процес вейвлет-перетворення базується на більш складних математичних алгоритмах. Це може призводити до збільшення часу обробки зображень. Крім того, налаштування параметрів вейвлет-стиснення вимагає певного досвіду та може впливати на ефективність алгоритму.

Ефективність методу також залежить від типу та характеристик оброблюваного зображення, а також від вимог до якості кінцевого результату. Наприклад, зображення з великою кількістю дрібних деталей або текстур може краще піддаватися вейвлет-стисненню, тоді як зображення зі значними однотонними ділянками можуть мати менші переваги.

Перед впровадженням вейвлет-стиснення рекомендується провести попередні експерименти для оцінки його ефективності у конкретному сценарії. Це дозволить підібрати кращі параметри алгоритму, збалансувати якість та ступінь стиснення та забезпечити максимальну ефективність його використання.

Висновки

У ході дослідження розглянуто та проаналізовано можливість комбінованого використання вейвлет-перетворення з JPEG-стисненням для підвищення ефективності компресії зображень у системах із обмеженими ресурсами, зокрема на мікроконтролері ESP32-CAM. Продемонстровано, що така комбінація дозволяє суттєво зменшити розмір файлу без значного погіршення візуальної якості, що підтверджується експериментальними результатами.

Основна перевага запропонованого методу полягає в покращеному балансі між стисненням та якістю зображення. Попереднє вейвлет-перетворення дозволяє ефективніше розділяти частотні компоненти, що дає змогу алгоритму JPEG агресивніше квантувати височастотні області, мінімізуючи втрати

важливих візуальних деталей. Крім того, цей метод сприяє зменшенню артефактів блокування (block artifacts), які є типовими для класичного JPEG.

Аналіз спектральних характеристик показав, що використання вейвлет-перетворення перед JPEG дозволяє досягти більш рівномірного розподілу енергії в частотному спектрі, що покращує ефективність кодування та підвищує ступінь стиснення без значної втрати якості. Це особливо важливо для зображень із складними текстурами та градієнтами, де класичне JPEG-стиснення призводить до значного погіршення деталізації.

Експериментальні результати продемонстрували, що при однаковій якості зображення ($\text{PSNR} \approx 40$ дБ) комбінований метод дозволяє зменшити розмір файлу в 1,5-2 рази порівняно з традиційним JPEG. При цьому навіть за умови дворазового застосування вейвлет-перетворення зображення залишається достатньо якісним для багатьох практичних застосувань, що підтверджується високими значеннями SSIM.

Оцінка продуктивності показала, що впровадження вейвлет-стиснення перед JPEG потребує додаткових обчислювальних ресурсів, особливо на етапі перетворення та відновлення. Однак загальний час обробки залишається прийнятним для використання в системах реального часу, особливо у випадках, коли важливим є мінімізація обсягу даних, що передаються у мережах з низькою пропускною здатністю, таких як LPWAN.

Порівняння із традиційними методами стиснення, такими як Brotli, Deflate, LZMA та Zstandard, підтвердило, що JPEG у поєднанні з вейвлет-перетворенням забезпечує найкраще співвідношення між швидкістю, ступенем компресії та збереженням візуальної інформації. Водночас, деякі методи, такі як LZMA, хоч і дають вищий ступінь стиснення, є занадто ресурсоемними для застосування в системах на базі ESP32-CAM.

Таким чином, запропонований комбінований метод JPEG + вейвлет-перетворення є ефективним рішенням для стиснення зображень у вбудованих системах та інтернеті речей. Він дозволяє досягти компромісу між якістю зображення, ступенем стиснення та швидкістю обробки, що робить його

перспективним для подальшого впровадження у практичні застосування, зокрема для бездротової передачі зображень у середовищах із обмеженими ресурсами.

Подальші дослідження можуть бути зосереджені на оптимізації обчислювальної складності алгоритму, а також на розробці адаптивних методів регулювання параметрів вейвлет-перетворення залежно від характеристик зображення та умов передачі даних. Також перспективним напрямом є застосування нейромережевих методів для подальшого покращення компресії та зменшення візуальних артефактів після відновлення.

5 РЕАЛІЗАЦІЯ АЛГОРИТМУ ВИПРАВЛЕННЯ АРТЕФАКТІВ ЗОБРАЖЕННЯ

5.1 Опис появи артефактів під час роботи комбінованого методу

Після проведення аналізу появи чорних точок на зображенні після застосування вейвлет-перетворення Хаара та збереження у форматі JPEG вказують на закономірність, пов'язану з рівнем перетворення та розміром блоків, які використовуються в JPEG-компресії. Слід враховувати, що вейвлет-перетворення Хаара розкладає зображення на піддіапазони (LL, LH, HL, HH), де на кожному рівні роздільна здатність апроксимаційного піддіапазону (LL) зменшується вдвічі по кожній координаті. Також JPEG-компресія розбиває зображення на блоки розміром 8×8 пікселів і застосовує дискретне косинусне перетворення (ДКП) до кожного блоку.

Після кількох рівнів вейвлет-перетворення, зображення містить періодичні структури, які можуть взаємодіяти з блоковою структурою JPEG, що призводить до появи артефактів на межах блоків.

За результатами аналізу виявлено закономірність появи чорних точок для усіх рівнів, починаючи з третього, оскільки на цьому рівні артефакти стають найбільш помітні:

- На рівні 3 чорні точки з'являються на кутах квадратів 8×8 пікселів.
- На рівні 4 чорні точки з'являються на кутах квадратів 16×16 пікселів. Спостерігається збільшення періоду артефактів вдвічі через подальше зменшення роздільної здатності після вейвлет-перетворення.
- На рівні 5 чорні точки з'являються на кутах прямокутників 32×16 пікселів. Спостерігається асиметрія розмірів блоків пов'язана з розміром зображення та неможливістю рівномірного ділення висоти на більші степені двійки.
- На рівні 6 чорні точки з'являються на кутах квадратів 64×16 пікселів. Подальше збільшення періоду артефактів по горизонталі, при цьому вертикальний період залишається незмінним через обмеження розміру зображення.

Для вирішення цієї проблеми використовується код, що виконує виявлення та корекцію артефактів (чорних точок) у масиві даних (який може представляти зображення) після вейвлет-перетворення Хаара та стиснення JPEG. Він вирішує проблему періодичних артефактів, що виникають через взаємодію структур вейвлетів із блоками JPEG. Детальний опис роботи коду:

Основні етапи роботи:

1. Розрахунок періодів артефактів (P_x та P_y): Параметри P_x (період по горизонталі) та P_y (період по вертикалі) залежать від рівня вейвлет-перетворення (level). Рівні до 3 використовують мінімальні періоди (8 пікселів), починаючи з 4-го рівня, період зростає для горизонтальних артефактів, тоді як вертикальний період залишається обмеженим.
2. Створення копії вхідного масиву: Оригінальний масив зберігається, щоб уникнути змін у вихідних даних.
3. Прохід по всіх позиціях чорних точок: За допомогою двох вкладених циклів код обробляє всі позиції, кратні P_x (по горизонталі) та P_y (по вертикалі). Вибираються "проблемні" точки для подальшого аналізу та корекції.
4. Вибір сусідів навколо точки артефакту: Для кожної точки визначається локальний сусідній простір (розміром 3×3 , якщо не обмежений краями масиву). Збираються значення сусідніх елементів, включаючи саму точку.
5. Фільтрація середнім значенням (медіана): Використовується медіанний фільтр, щоб обчислити середнє значення серед сусідів. Це дозволяє зменшити вплив аномальних значень (артефактів). Медіанний фільтр є ефективним для видалення точкових шумів (таких як чорні точки).
6. Заміна значення точки на обчислене: Позиції, які відповідають артефактам, оновлюються, отримуючи значення, що мінімізують помітність чорних точок.
7. Повернення скоригованого масиву: Код повертає новий масив, у якому артефакти зменшені або усунені.

5.2 Аналіз математичного та програмного рішення

Нехай W та H — ширина та висота зображення, k — рівень вейвлет-перетворення, P_x та P_y — періоди появи чорних точок по горизонталі та вертикалі.

Розрахунок періодів по горизонталі (ось x): $P_x = 8 \times 2^{(k-3)}$

Для $k=3$: $P_x = 8 \times 2^0 = 8$ пікселів

Для $k=4$: $P_x = 8 \times 2^1 = 16$ пікселів

Для $k=5$: $P_x = 8 \times 2^2 = 32$ пікселів

Для $k=6$: $P_x = 8 \times 2^3 = 64$ пікселів

Розрахунок періодів по вертикалі (ось y):

Через обмеження висоти зображення $H=240$ пікселів, яка не ділиться на більші степені двійки без залишку, період по вертикалі обмежений:

$$P_y = \begin{cases} 8 \times 2^{(k-3)}, & \text{для } k \leq 4 \\ 16, & \text{для } k > 4 \end{cases}$$

Для $k=3$: $P_y = 8 \times 2^0 = 8$

Для $k=4$: $P_y = 8 \times 2^1 = 16$ пікселів

Для $k \geq 5$: $P_y = 16$ пікселів

Чорні точки з'являються в координатах (x, y) , де:

$$x = n_x \times P_x, n_x = 0, 1, 2, \dots, \left\lfloor \frac{W}{P_x} \right\rfloor \text{ та } y = n_y \times P_y, n_y = 0, 1, 2, \dots, \left\lfloor \frac{H}{P_y} \right\rfloor \quad (27)$$

Приклад для рівня 3:

$P_x=P_y=8$ пікселів. Чорні точки з'являються в координатах кратних 8 пікселям по обох осях.

Приклад для рівня 5:

$P_x=32$ пікселів, $P_y=16$ пікселів. Чорні точки з'являються в координатах $x=0,32,64,\dots$ та $y=0,16,32,\dots$

Математичне моделювання кількості чорних точок

Кількість точок по горизонталі: $N_x = \left\lceil \frac{W}{P_x} \right\rceil + 1$

Кількість точок по вертикалі: $N_y = \left\lceil \frac{W}{P_y} \right\rceil + 1$

Загальна кількість чорних точок: $N_{total} = N_x \times N_y$

Приклад для рівня 4:

$P_x=P_y=16$ пікселів, $N_x = \left\lceil \frac{320}{16} \right\rceil + 1 = 20 + 1 = 21$, $N_y = \left\lceil \frac{240}{16} \right\rceil + 1 = 15 + 1 = 16$,

$N_{total} = 21 \times 16 = 336$ чорних точок

Нижче наведено код, що виявляє та виправляє артефакти зображень в місцях, що знаходяться за допомогою математичного розрахунку описаного раніше:

```
public static double[,] DetectAndCorrectErrors(double[,] data, int level, int width, int height){
```

```
    // Calculate periods based on the provided formulas
```

```
    int Px = Math.Max(1, 8 * (int)Math.Pow(2, Math.Max(0, level - 3)));
```

```
    int Py = level <= 4 ? Math.Max(1, 8 * (int)Math.Pow(2, Math.Max(0, level - 3)))
```

```
    : 16;
```

```
    // Clone data to preserve original
```

```
    double[,] correctedData = (double[,])data.Clone();
```

```
    // Loop through black point positions
```

```
    for (int x = 0; x < width; x += Px) {
```

```
        for (int y = 0; y < height; y += Py) {
```

```
            // Define neighborhood bounds
```

```
            int rowMin = Math.Max(0, x - 1);
```

```
            int rowMax = Math.Min(width - 1, x + 1);
```

```

int colMin = Math.Max(0, y - 1);
int colMax = Math.Min(height - 1, y + 1);
// Collect neighbors into a list
var neighbors = new System.Collections.Generic.List<double>();
for (int i = rowMin; i <= rowMax; i++){
    for (int j = colMin; j <= colMax; j++){
        if (i != x || j != y) // Exclude current point {
            neighbors.Add(correctedData[i, j]);
        }
    }
}
// Apply median filtering if neighbors are available
if (neighbors.Count > 0){
    neighbors.Sort();
    double median;
    int mid = neighbors.Count / 2;
    median = neighbors.Count % 2 == 0 ? (neighbors[mid - 1] +
neighbors[mid]) / 2.0 : neighbors[mid];
    correctedData[x, y] = median;
}
}
return correctedData;
}

```

Код вирішує проблему чорних точок, які з'являються внаслідок вейвлет-перетворення Хаара. Це призводить до зменшення роздільної здатності та періодичність даних після трансформації створюють проблемні структури.

Для 3-го рівня ($P_x=P_y=8$) чорні точки з'являються на кутах 8×8 блоків. Код обробляє ці позиції, коригуючи значення за допомогою сусідів. Для 4-го рівня ($P_x=P_y=16$) період артефактів зростає, але принцип роботи коду залишається тим самим. На вищих рівнях, наприклад, 5-му ($P_x=32$, $P_y=16$), код ефективно компенсує асиметрію блоків і усуває артефакти на цих координатах.

Код базується на принципі виявлення артефактів у періодичних точках, використовуючи розраховані періоди (формули для P_x та P_y) і локальну медіанну

фільтрацію. Це дозволяє забезпечити адаптивну корекцію без втрати ключових деталей зображення.

5.3 Аналіз та порівняння результатів експерименту

В ході виконання експерименту, що включають оцінку якості зображень за допомогою кількох метрик, отримано значення MSE (середньоквадратична похибка), PSNR (пікове співвідношення сигнал-шум), SSIM (індекс структурної схожості) та час виконання алгоритму: для оцінки продуктивності методів обробки.

У таблиці наведено методи обробки зображень за допомогою вейвлет-перетворення Хаара на різних рівнях декомпозиції. Також розглядається комбінація вейвлет-перетворення Хаара із застосуванням медіанного фільтра. Аналіз дозволяє оцінити вплив рівня декомпозиції та використання медіанного фільтра на якість зображення та швидкість обчислень.

Таблиця 13 – Результат оцінки ефективності вейвлет перетворення Хаара з та без медіанного фільтра

DWT	Рівень декомпозиції	MSE	PSNR	SSIM	Час перетворення, мс
Haar	2	25,153	34,124	0,965	40
	4	73,449	29,470	0,921	84
Haar + медіанний фільтр	2	3,75	42,385	0,995	48
	4	12,148	37,285	0,981	92

Таблиця Таблиця 13 – Результат оцінки ефективності вейвлет перетворення Хаара з та без медіанного фільтра демонструє результати оцінки ефективності вейвлет-перетворення Хаара на двох рівнях декомпозиції — другому та четвертому, а також у комбінації з медіанним фільтром. Представлені метрики включають середньоквадратичну похибку (MSE), пікове співвідношення сигнал-шум (PSNR), структурну подібність (SSIM) та час виконання перетворення.

Для класичного вейвлет-перетворення Хаара на другому рівні декомпозиції отримано значення MSE у 25,153, що вказує на помітну похибку в

порівнянні з оригінальним зображенням. PSNR становить 34,124, що відповідає відносно прийнятній якості зображення. Значення SSIM дорівнює 0,965, що свідчить про високу схожість структурних характеристик між оригінальним і перетвореним зображенням. Час виконання обчислень складає 40 мс, що свідчить про швидкість цього методу на початковому рівні декомпозиції.

На четвертому рівні декомпозиції вейвлет-перетворення Хаара спостерігається значне збільшення MSE до 73,449, що вказує на суттєву втрату точності при збільшенні рівня декомпозиції. Відповідно, PSNR знижується до 29,470, демонструючи погіршення якості зображення. SSIM також падає до 0,921, що означає зниження подібності між оригінальним і відновленим зображенням. Час виконання зростає до 84 мс, що логічно через підвищення складності обчислень на глибшому рівні.

Застосування медіанного фільтра разом із вейвлет-перетворенням Хаара на четвертому рівні декомпозиції дозволило досягти значного покращення. Значення MSE зменшилося до 12,148, що демонструє зниження похибок і артефактів, які були присутні в попередньому методі. PSNR підвищився до 37,285, що вказує на значне покращення якості зображення. Значення SSIM збільшилося до 0,981, підтверджуючи збереження структурної цілісності зображення на високому рівні. Час виконання зріс до 92 мс через додаткові обчислення, необхідні для медіанного фільтрування, однак цей приріст часу є незначним у порівнянні з досягнутими покращеннями якості.

Загалом, класичне вейвлет-перетворення Хаара демонструє гарні результати на нижчому рівні декомпозиції, але на глибших рівнях виникають суттєві похибки та втрата якості. Додавання медіанного фільтра дозволяє суттєво покращити результати, знижуючи похибки та підвищуючи візуальну якість, навіть на складному четвертому рівні декомпозиції. Це свідчить про ефективність поєднання цих двох методів для задач, що потребують високої якості відновлення зображення. На рис. Рисунок 28 – Зображення без додаткового перетворення та з перетворенням медіанного фільтра наведено зображення без додаткового перетворення та з перетворенням:



а

б



в

г

Рисунок 28 – Зображення без додаткового перетворення та з перетворенням медіанного фільтру

На зображенні видно, що використання додаткової обробки з допомогою медіанного усереднення призводить до згладження проблемних ділянок. Таким чином на зображенні можна виділити відсутність яскраво виражених похибок у вигляді чорних точок. Зображення стало більш сприятливе для людського ока.

У Таблиці Таблиця 14 – Порівняння відносної різниці між пікселями з та без використання медіанного фільтру наведено відносну різницю між пікселями для вейвлету Хаара та вейвлету Хаара з використанням медіанного фільтру на 2-му та 4-му рівнях декомпозиції. Відносна різниця оцінюється від 0 до 1, де 0 означає повну схожість пікселів, а 1 — максимальну різницю між ними.

Таблиця 14 – Порівняння відносної різниці між пікселями з та без використання медіанного фільтру

Метод	Рівень декомпозиції	Відносна різниця
Хаар	2	0.041
	4	0.537
Хаар + медіанний фільтр	2	0.044
	4	0.216

У таблиці Таблиця 14 – Порівняння відносної різниці між пікселями з та без використання медіанного фільтру представлено результати оцінки відносної різниці між пікселями зображення після обробки методами вейвлет-перетворення Хаара та Хаара з медіанним фільтром на різних рівнях декомпозиції (2-й і 4-й рівні).

Для методу Хаар на 2-му рівні декомпозиції відносна різниця складає 0.041, що вказує на незначні відхилення між пікселями оригінального та обробленого зображення. Однак, на 4-му рівні декомпозиції спостерігається суттєве збільшення відносної різниці до 0.537, що свідчить про більші спотворення зображення через глибшу декомпозицію.

Метод Хаар з медіанним фільтром на 2-му рівні декомпозиції демонструє відносну різницю 0.044, що близьке до результату чистого Хаара на тому ж рівні. Це свідчить про незначний вплив медіанного фільтра на цьому етапі. Однак, на 4-му рівні декомпозиції відносна різниця складає 0.216, що значно менше, ніж для чистого Хаара (0.537). Це вказує на те, що медіанний фільтр ефективно зменшує артефакти та спотворення, характерні для глибокої декомпозиції.

На 2-му рівні декомпозиції результати методів Хаара та Хаара з медіанним фільтром є дуже схожими, оскільки глибина декомпозиції ще не спричиняє суттєвих змін у зображенні. Різниця у відносних відхиленнях є мінімальною (0.041 проти 0.044).

На 4-му рівні декомпозиції метод Хаара демонструє значно більшу відносну різницю (0.537) у порівнянні з Хааром із медіанним фільтром (0.216).

Це вказує на те, що медіанний фільтр значно зменшує артефакти, викликані високою глибиною декомпозиції, покращуючи якість обробки зображення.

Висновки

У ході дослідження проведено детальний аналіз появи артефактів у вигляді чорних точок при застосуванні комбінованого методу стиснення, що поєднує вейвлет-перетворення Хаара та JPEG-компресію. Виявлено закономірності виникнення артефактів залежно від рівня декомпозиції, зокрема починаючи з третього рівня вейвлет-перетворення, коли структура зображення взаємодіє з блоковою структурою JPEG. Спостерігалось, що артефакти проявляються на кутах блоків із певними періодами: на третьому рівні — 8×8 , на четвертому — 16×16 , на п'ятому — 32×16 , а на шостому рівні — 64×16 пікселів. Збільшення періоду артефактів уздовж горизонталі пояснюється поступовим зменшенням роздільної здатності зображення на кожному етапі вейвлет-перетворення.

Для усунення зазначених артефактів розроблено та протестований програмний алгоритм, який виявляє та виправляє проблемні області шляхом медіанного фільтрування. Код використовує попередньо розраховані періоди артефактів (P_x , P_y) для кожного рівня декомпозиції, аналізує сусідні значення та замінює спотворені пікселі на середні значення з їхнього локального оточення. Такий підхід дозволив суттєво зменшити кількість чорних точок без втрати ключових деталей зображення, забезпечуючи більш природний вигляд відновленого зображення після компресії.

Математичний аналіз підтвердив ефективність запропонованого методу. Сформульовано аналітичні вирази для визначення кількості артефактів та їхньої просторової періодичності. Отримані результати дозволили не лише передбачити появу артефактів, а й адаптувати алгоритм їхньої корекції відповідно до рівня вейвлет-декомпозиції.

Експериментальні результати, представлені у вигляді метрик MSE (середньоквадратична похибка), PSNR (пікове співвідношення сигнал-шум) та SSIM (індекс структурної схожості), підтвердили, що застосування медіанного

фільтра в комбінації з вейвлет-перетворенням Хаара значно покращує якість зображення. Показано, що при використанні медіанного фільтра на четвертому рівні декомпозиції значення MSE зменшується майже у 6 разів, а PSNR підвищується на 7,8 дБ у порівнянні з методом без фільтрації. Також значення SSIM, що відображає структурну схожість зображення після обробки, підвищилося з 0,921 до 0,981, що свідчить про мінімальні втрати у структурній цілісності зображення.

Аналіз відносної різниці між пікселями після обробки методами "чистого" вейвлет-перетворення Хаара та його комбінації з медіанним фільтром підтвердив ефективність згладжування артефактів на високих рівнях декомпозиції. Зокрема, без застосування фільтра значення відносної різниці на 4-му рівні досягало 0,537, тоді як із медіанним фільтром воно зменшувалося до 0,216, що вказує на значне покращення візуальної якості.

Таким чином, проведене дослідження показало, що застосування вейвлет-перетворення Хаара у поєднанні з JPEG-стисненням потребує адаптивної корекції артефактів, особливо на високих рівнях декомпозиції. Розроблений алгоритм на основі медіанного фільтрування дозволяє ефективно усувати чорні точки, зберігаючи при цьому високу якість зображення. Запропонований метод може бути корисним у системах із передачею зображень у середовищах із обмеженою пропускнуою здатністю, таких як IoT-пристрої та бездротові сенсорні мережі.

Подальші дослідження можуть бути зосереджені на вдосконаленні алгоритму шляхом використання адаптивних методів корекції, що враховують локальні особливості зображення, а також на застосуванні машинного навчання для автоматичного визначення кращих параметрів фільтрації та рівнів декомпозиції. Це дозволить ще більше підвищити ефективність комбінованого методу стиснення та зробити його придатним для використання в реальних сценаріях із високими вимогами до якості відновленого зображення.

ЗАГАЛЬНІ ВИСНОВКИ

Загальний висновок за результатами проведених досліджень та аналізу демонструє високий потенціал інтеграції вейвлет-перетворень у системи на базі мікроконтролерів, особливо для задач, що вимагають ефективної обробки та стиснення візуальної інформації. Мікроконтролер ESP32-CAM визначено як перспективну платформу для впровадження IoT-технологій завдяки його компактності, енергоефективності та можливості роботи з камерою й бездротовими інтерфейсами. Застосування вейвлет-перетворень у поєднанні з додатковими методами, такими як медіанний фільтр, значно підвищує якість обробки зображень, особливо на глибоких рівнях декомпозиції.

Вейвлет-перетворення Хаара продемонструвало найшвидшу обробку, що робить його ідеальним для систем із обмеженими ресурсами. Проте, зі збільшенням рівня декомпозиції якість відновлених зображень знижується через наявність артефактів. Додавання медіанного фільтра дозволяє зменшити спотворення, зберігаючи прийнятну якість, навіть на високих рівнях декомпозиції. Вейвлети Добеші та Коїфлет виявилися ефективними для задач, де важливі точність і симетрія, але їхня реалізація потребує більше обчислювальних ресурсів.

Дослідження методів стиснення зображень показали, що поєднання вейвлет-перетворень із алгоритмами без втрат, такими як Хаффмана та RLE, дозволяє досягти високого ступеня компресії без значних втрат якості. Зокрема, ефективність цих методів суттєво зростає після попередньої обробки зображень вейвлет-перетвореннями. Порівняння з JPEG демонструє, що методи без втрат залишаються переважним вибором для задач, де критично важливе збереження якості.

Результати експериментів також підтверджують, що використання вейвлет-перетворень у реальному часі на мікроконтролерах є перспективним напрямком для створення автономних IoT-систем, які забезпечують ефективну передачу даних із мінімальними втратами. Це особливо актуально для застосувань у відеоспостереженні, моніторингу та системах розпізнавання.

Подальші дослідження можуть бути спрямовані на оптимізацію алгоритмів для роботи з різними типами зображень, адаптацію до специфіки каналу передачі та зменшення енергоспоживання. Таким чином, поєднання вейвлет-перетворень і сучасних мікроконтролерів відкриває нові можливості для ефективної обробки та аналізу візуальних даних у системах IoT.

ПЕРЕЛІК ЛІТЕРАТУРИ

- [1] Мазін, М & Оникієнко, Ю. (2023). Використання вейвлет перетворень для обробки зображень в системах інтернету речей на мікроконтролерах. *Technologies and Engineering*. 15-25. DOI: 10.30857/2786-5371.2023.3.2.
- [2] Мазін, М & Оникієнко, Ю. (2023). Особливості використання методів Хаффмана та RLE для стиснення зображень в системах на мікроконтролерах. *Technologies and Engineering*. 21-30. DOI: 10.30857/2786-5371.2023.6.2.
- [3] Мазін, М., Оникієнко, Ю. (2024). Аналіз методів обробки зображень в системах Інтернету речей на основі вейвлет перетворень. *Technologies and Engineering*, 25(6), 53-60. DOI: <https://doi.org/10.30857/2786-5371.2024.6.5>.
- [4] Atzori, L., Iera, A., & Morabito, G. (2010). The Internet of Things: A survey. *Computer Networks*, 54(15), 2787-2805.
- [5] Miorandi, D., Sicari, S., De Pellegrini, F., & Chlamtac, I. (2012). Internet of things: Vision, applications and research challenges. *Ad Hoc Networks*, 10(7), 1497-1516.
- [6] Zanella, A., Bui, N., Castellani, A., Vangelista, L., & Zorzi, M. (2014). Internet of things for smart cities. *IEEE Internet of Things Journal*, 1(1), 22-32.
- [7] Gubbi, J., Buyya, R., Marusic, S., & Palaniswami, M. (2013). Internet of Things (IoT): A vision, architectural elements, and future directions. *Future Generation Computer Systems*, 29(7), 1645-1660.
- [8] Dlodlo, N., & Kalezhi, J. (2015). The Internet of Things in agriculture for sustainable rural development. 2015 International Conference on Emerging Trends in Networks and Computer Communications (ETNCC).
- [9] Xu, L. D., He, W., & Li, S. (2014). Internet of things in industries: A survey. *IEEE Transactions on Industrial Informatics*, 10(4), 2233-2243.
- [10] Akyildiz, I. F., Su, W., Sankarasubramaniam, Y., & Cayirci, E. (2002). Wireless sensor networks: A survey. *Computer Networks*, 38(4), 393-422.

- [11] Centenaro, M., Vangelista, L., Zanella, A., & Zorzi, M. (2016). Long-range communications in unlicensed bands: The rising stars in the IoT and smart city scenarios. *IEEE Wireless Communications*, 23(5), 60-67.
- [12] Hodge, V. J., O’Keefe, S., Weeks, M., & Moulds, A. (2017). Wireless sensor networks for industrial applications. *Journal of Sensor and Actuator Networks*, 6(2), 17.
- [13] Raza, U., Kulkarni, P., & Sooriyabandara, M. (2017). Low power wide area networks: An overview. *IEEE Communications Surveys & Tutorials*, 19(2), 855-873.
- [14] Gomez, C., Oller, J., & Paradells, J. (2012). Overview and evaluation of Bluetooth Low Energy: An emerging low-power wireless technology. *Sensors*, 12(9), 11734-11753.
- [15] Kinney, P. (2003). ZigBee technology: Wireless control that simply works. *IEEE Communications Design Conference*, 1-4.
- [16] Cisco Systems. (2018). The role of Wi-Fi in IoT. White Paper.
- [17] Lloret, J., Parra, L., Taha, M., & Tomás, J. (2017). Wireless sensor networks and IoT for environmental monitoring: A review. *Sensors*, 17(9), 2107.
- [18] Strang, G., & Nguyen, T. (1997). Wavelets and filter banks. Wellesley-Cambridge Press.
- [19] Acharya, T., & Tsai, P. S. (2005). JPEG2000 standard for image compression: Concepts, algorithms and VLSI architectures. John Wiley & Sons.
- [20] Buades, A., Coll, B., & Morel, J. M. (2005). A non-local algorithm for image denoising. *CVPR*.
- [21] Satyanarayanan, M. (2017). The emergence of edge computing. *Computer*, 50(1), 30-39.
- [22] Han, S., Mao, H., & Dally, W. J. (2016). Deep compression: Compressing deep neural networks with pruning, trained quantization and Huffman coding. *ICLR*.
- [23] Zhang, Y., Yang, J., & Xu, W. (2017). A survey on distributed computing for large-scale data analytics. *Journal of Parallel and Distributed Computing*, 111, 112-123.

[24] J. Konecny *et al.*, "Computational Cost and Implementation Analysis of a Wavelet-Based Edge Computing Method for Energy-Harvesting Industrial IoT Sensors," in *IEEE Access*, vol. 12, pp. 193607-193621, 2024, doi: 10.1109/ACCESS.2024.3519715

[25] Courjault, J., Vrigneau, B., Berder, O., & Bhatnagar, M. (2020). *How robust is a LoRa communication against impulsive noise?* PIMRC 2020, London, United Kingdom. Available at: <https://hal.science/hal-03027517v1>.

[26] S. A. Sofi and R. Naaz, "Data compression in Wireless visual Sensor networks using wavelets," *2016 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)*, Chennai, India, 2016, pp. 1286-1290, doi: 10.1109/WiSPNET.2016.7566344

[27] Ms. Minaxi Doorwar, Dr. Mrs. P. Malathi - Comparison of Protocol Based on Architecture in Wireless Multimedia Sensor Network. 2018 IEEE Global Conference on Wireless Computing and Networking (GCWCN). 2018. P. 39–43. DOI: 10.1109/GCWCN.2018.8668650.

[28] Ahmed Mateen, Khizar Abbas, Maida Sehar, Muhammad Azeem Akbar- Comparative Analysis of Wireless Sensor Networks with Wireless Multimedia Sensor Networks. IEEE International Conference on Power, Control, Signals and Instrumentation Engineering (ICPCSI-2017). 2017. P. 80–83. DOI: 10.1109/ICPCSI.2017.8391847.

[29] Wael Ali Hussein, Borhanuddin M Ali, MFA Rasid, Fazirulhisyam Hashim - Design and Performance Analysis of High Reliability-optimal Routing protocol for Mobile Wireless Multimedia Sensor Networks. IEEE 13th Malaysia International Conference on Communications (MICC), 28-30 Nov. 2017, The Puteri Pacific, Johor Bahru, Malaysia. 2017. P. 136–140. DOI: 10.1109/MICC.2017.8311747.

[30] Satoshi Inaba - 3D Flash Memory for Data-intensive Applications. 2018 IEEE International Memory Workshop (IMW). 2018. P. 1–15. DOI: 10.1109/IMW.2018.8388775.

[31] S. Rein and M. Reisslein, "Performance evaluation of the fractional wavelet filter: A low-memory image wavelet transforms for multimedia sensor networks," *Ad Hoc Networks*, 2011, doi: 10.1016/j.adhoc.2010.08.004.

[32] A. J. Ahmed, M. M. Hamdi, A. S. Mustafa and S. A. Rashid, "WSN Application Based on Image Compression Using AHAAR Wavelet Transform," 2022 International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA), Ankara, Turkey, 2022, pp. 1-4, doi: 10.1109/HORA55278.2022.9799842.

[33] A. Sharma, H. Tandan and P. Goel, "Energy efficient handling of DWT-image transmission over wireless sensor network," 2017 Fourth International Conference on Image Information Processing (ICIIP), Shimla, India, 2017, pp. 1-5, doi: 10.1109/ICIIP.2017.8313689.

[34] M. Tausif, A. Jain, E. Khan and M. Hasan, "Efficient Architectures of Fractional Wavelet Filter (FrWF) for Visual Sensors and Wearable Devices," 2018 IEEE SENSORS, New Delhi, India, 2018, pp. 1-4, doi: 10.1109/ICSENS.2018.8630300.

[35] M. Tausif, E. Khan and M. Hasan, "Lifting-based fast and low memory DWT computation for IoT platform," TENCON 2019 - 2019 IEEE Region 10 Conference (TENCON), Kochi, India, 2019, pp. 435-440, doi: 10.1109/TENCON.2019.8929533.

[36] R. Krishnaswamy and S. NirmalaDevi, "EFFICIENT MEDICAL IMAGE COMPRESSION BASED ON INTEGER WAVELET TRANSFORM," 2020 *Sixth International Conference on Bio Signals, Images, and Instrumentation (ICBSII)*, Chennai, India, 2020, pp. 1-5, doi: 10.1109/ICBSII49132.2020.9167597

[37] Rohima and M. Barkah Akbar, "Wavelet Analysis and Comparison from Coiflet Family on Image Compression," 2020 8th International Conference on Cyber and IT Service Management (CITSM), Pangkal, Indonesia, 2020, pp. 1-5, doi: 10.1109/CITSM50537.2020.9268811

[38] S. Zoican, R. Zoican and D. Galatchi, "Color Image Noise Removal Algorithm Implementation using Blackfin Dual Core Microcontroller," 2019 14th

International Conference on Advanced Technologies, Systems and Services in Telecommunications (TELSIKS), Nis, Serbia, 2019, pp. 211-214, doi: 10.1109/TELSIKS46999.2019.9002078

[39] M. Patlayenko, O. Osharovska, V. Solodka and V. Pyliavskyi, "Implementation of a Wavelet Encoder-Decoder Image on a Microcontroller," 2021 IEEE 8th International Conference on Problems of Infocommunications, Science and Technology (PIC S&T), Kharkiv, Ukraine, 2021, pp. 475-478, doi: 10.1109/PICST54195.2021.9772150

[40] I. Yamnenko and V. Levchenko, "Video-Data Compression Using Wavelet Analysis," 2019 IEEE 20th International Conference on Computational Problems of Electrical Engineering (CPEE), Lviv-Slavske, Ukraine, 2019, pp. 1-4, doi: 10.1109/CPEE47179.2019.8949074

[41] O. Bychkov, K. Merkulova and Y. Zhabska, "Software Application for Biometrical Person's Identification by Portrait Photograph Based on Wavelet Transform," 2019 IEEE International Conference on Advanced Trends in Information Theory (ATIT), Kyiv, Ukraine, 2019, pp. 253-256, doi: 10.1109/ATIT49449.2019.9030462

[42] A. Izmailov, "Symmetric Ternary Wavelet Transform and Its Application in Digital Information Processing," 2019 IEEE 2nd Ukraine Conference on Electrical and Computer Engineering (UKRCON), Lviv, Ukraine, 2019, pp. 1-6, doi: 10.1109/UKRCON.2019.8879808

[43] R. Odarchenko, O. Lavrynenko, D. Bakhtiiarov, S. Dorozhynskyi and V. A. O. Zharova, "Empirical Wavelet Transform in Speech Signal Compression Problems," 2021 IEEE 8th International Conference on Problems of Infocommunications, Science and Technology (PIC S&T), Kharkiv, Ukraine, 2021, pp. 599-602, doi: 10.1109/PICST54195.2021.9772156

[44] V. Barannik, A. Berchanov, V. Barannik, M. Babenko, R. Onyshchenko and L. Kolodiichuk, "Method of Mini Segments Encoding in Difference Space Using Haar Wavelet," 2023 IEEE 5th International Conference on Advanced Information and

Communication Technologies (AICT), Lviv, Ukraine, 2023, pp. 1-4, doi: 10.1109/AICT61584.2023.10452674

[45] H. Ding, H. Liu and F. Yang, "A Real-Time Compression Method of Power System Waveform Data Based on 2-D Lifting Wavelet Transform and Deflate Algorithm," 2021 3rd Asia Energy and Electrical Engineering Symposium (AEEES), Chengdu, China, 2021, pp. 626-630, doi: 10.1109/AEEES51875.2021.9403107

[46] Genta, Addisalem & Lobiyal, D. (2018). Performance Evaluation of Wavelet Based Image Compression for Wireless Multimedia Sensor Network: Second International Conference, ICACDS 2018, Dehradun, India, April 20-21, 2018, Revised Selected Papers, Part II, pp. 402-412. DOI: 10.1007/978-981-13-1813-9_40

[47] Mohindru, Pankaj & Pooja. (2022). Comparative Analysis of Haar, Daubechies and Bior wavelets on Image Compression using Discrete Wavelet Transform. Deptt. Of ECE, Punjabi University, Patiala, Punjab, India. Pp. 4-9, 2022 URL:

www.researchgate.net/publication/362658001_Comparative_Analysis_of_Haar_Daubechies_and_Bior_wavelets_on_Image_Compression_using_Discrete_Wavelet_Transform

[48] Gunanandhini, S & Kalamani, M & Bhagavathipriya, M & prasath, S. (2022). Wavelet based Video Compression techniques for Industrial monitoring applications. Journal of Physics: Conference Series. 2272. 012019. DOI: 10.1088/1742-6596/2272/1/012019

[49] A. Thakker, N. Namboodiri, R. Mody, R. Tasgaonkar and M. Kambli, "Lossy Image Compression-A Comparison Between Wavelet Transform, Principal Component Analysis, K-Means and Autoencoders," 2022 5th International Conference on Advances in Science and Technology (ICAST), Mumbai, India, 2022, pp. 569-576, doi: 10.1109/ICAST55766.2022.10039613

[50] Al-Dabbas, Hind & Mohammed, Faisel. (2018). THE EFFECT OF WAVELET COEFFICIENT REDUCTION ON IMAGE COMPRESSION USING DWT AND DAUBECHIES WAVELET TRANSFORM. 30 (5). 757-762. URL: www.researchgate.net/publication/339487317_THE_EFFECT_OF_WAVELET_CO

EFFICIENT_REDUCTION_ON_IMAGE_COMPRESSION_USING_DWT_AND_DAUBECHIES_WAVELET_TRANSFORM

[51] N. M. Varma and A. Choudhary, "Evaluation Of Distance Measures In Content Based Image Retrieval," 2019 3rd International conference on Electronics, Communication and Aerospace Technology (ICECA), Coimbatore, India, 2019, pp. 696-701, doi: 10.1109/ICECA.2019.8821957.

[52] A. Schlamm and D. Messinger, "Improved detection and clustering of hyperspectral image data by preprocessing with a euclidean distance transformation," 2011 3rd Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing (WHISPERS), Lisbon, Portugal, 2011, pp. 1-4, doi: 10.1109/WHISPERS.2011.6080847

[53] A. Sarkar and K. K. Halder, "Speckle Noise Reduction Using a New Weighted-Average Filter Based on Euclidean Distance," 2021 IEEE International Conference on Biomedical Engineering, Computer and Information Technology for Health (BECITHCON), Dhaka, Bangladesh, 2021, pp. 38-41, doi: 10.1109/BECITHCON54710.2021.9893716

[54] Rafael C. Gonzalez, Richard E. Woods. "Digital image processing 3th edition". 2008. ISBN: 0-13-168728-x 978-0-13-168728-8.

[55] Rafael C. Gonzalez, Richard E. Woods. "Digital image processing 4th edition". 2018. ISBN: 1-292-22304-9.

[56] Burrus, Charles & Gopinath, R. & Guo, H.. (1998). Introduction to Wavelets and Wavelet Transform—A Primer. Recherche. 67. URL: https://www.researchgate.net/publication/246532602_Introduction_to_Wavelets_and_Wavelet_Transform-A_Primer

[57] P. J. Burt and E. H. Adelson. The Laplacian pyramid as a compact image code. IEEE Trans. Commun., 31(4):532–540, April 1983.

[58] Y. Meyer. Wavelets and Operators. Advanced mathematics. Cambridge University Press, 1992.

[59] S. Mallat. Multiresolution approximations and wavelet orthonormal bases of $L^2(\mathbb{R})$. Trans. Amer. Math. Soc., 315:69–87, September 1989.

[60] G. Strang and G. Fix. A Fourier analysis of the finite element variational method. *Construct. Aspects of Funct. Anal.*, pages 796–830, 1971.

[61] I. Daubechies. Orthonormal bases of compactly supported wavelets. *Commun. on Pure and Appl. Math.*, 41:909–996, November 1988.

[62] St'ephane Mallat. A Wavelet Tour of Signal Processing. 229-343. 1999. URL:

https://coehuman.uodiyala.edu.iq/uploads/Coehuman%20library%20pdf/%D9%83%D8%AA%D8%A8%20%D8%A7%D9%84%D8%B1%D9%8A%D8%A7%D8%B6%D9%8A%D8%A7%D8%AA%20Mathematics%20books/Wavelets/Mallat_Wavelet-Tour-of-Signal-Processing.pdf

[63] Thiang, "Implementation of speech recognition on MCS51 microcontroller for controlling wheelchair," *2007 International Conference on Intelligent and Advanced Systems*, Kuala Lumpur, Malaysia, 2007, pp. 1193-1198, doi: 10.1109/ICIAS.2007.4658573.

[64] K. M. Tabal and J. Dela Cruz, "Development of low-cost embedded-based electrooculogram blink pulse classifier for drowsiness detection system," *2017 IEEE 13th International Colloquium on Signal Processing & its Applications (CSPA)*, Penang, Malaysia, 2017, pp. 29-34, doi: 10.1109/CSPA.2017.8064919.

[65] P. Przedwojski, J. Dalecki, T. Talańska and R. Długosz, "Kohonen winner takes all neural network realized on microcontrollers with AVR and ARM cores," *Proceedings of the 17th International Conference Mixed Design of Integrated Circuits and Systems - MIXDES 2010*, Wroclaw, Poland, 2010, pp. 273-276.

[66] S. R. C. De Guzman, L. C. Tan and J. F. Villaverde, "Social Distancing Violation Monitoring Using YOLO for Human Detection," *2021 IEEE 7th International Conference on Control Science and Systems Engineering (ICCSSE)*, Qingdao, China, 2021, pp. 216-222, doi: 10.1109/ICCSSE52761.2021.9545098.

[67] L. A. Anto Gracious and T. C. Subbulakshmi, "Multihop distance estimation with Greedy approach in wireless sensor networks," *2011 International Conference on Emerging Trends in Electrical and Computer Technology*, Nagercoil, India, 2011, pp. 901-905, doi: 10.1109/ICETECT.2011.5760247.

- [68] Y. Zhou, S. Cui and Y. Wang, "Development of Machine Learning Based Recommendation Engine for Embedded Programmer," *2019 IEEE 9th International Conference on Electronics Information and Emergency Communication (ICEIEC)*, Beijing, China, 2019, pp. 634-637, doi: 10.1109/ICEIEC.2019.8784659.
- [69] A. Bogdanov and I. Kizhvatov, "Beyond the Limits of DPA: Combined Side-Channel Collision Attacks," in *IEEE Transactions on Computers*, vol. 61, no. 8, pp. 1153-1164, Aug. 2012, doi: 10.1109/TC.2011.140.
- [70] S. M. M. Al-Chalabi and H. K. Hoomod, "Real-Time Achievement by Tracking Mobile Wireless Sensors Using PFA and EEA," *2019 2nd International Conference on Engineering Technology and its Applications (IICETA)*, Al-Najef, Iraq, 2019, pp. 127-132, doi: 10.1109/IICETA47481.2019.9013007.
- [71] S. F. Kak and F. M. Mustafa, "Smart Home Management System Based on Face Recognition Index in Real-time," *2019 International Conference on Advanced Science and Engineering (ICOASE)*, Zakho - Duhok, Iraq, 2019, pp. 40-45, doi: 10.1109/ICOASE.2019.8723673.
- [72] S. -Y. Yuan, Y. -L. Wu, R. Perdriau and S. -S. Liao, "Detection of Electromagnetic Interference in Microcontrollers Using the Instability of an Embedded Phase-Lock Loop," in *IEEE Transactions on Electromagnetic Compatibility*, vol. 55, no. 2, pp. 299-306, April 2013, doi: 10.1109/TEMPC.2012.2218285.
- [73] G. Pérez, J. Pacheco and V. Benitez, "Anomaly Behavior Analysis for Sensors Fault Detection," *2023 IEEE Symposium Series on Computational Intelligence (SSCI)*, Mexico City, Mexico, 2023, pp. 1718-1723, doi: 10.1109/SSCI52147.2023.10372056.
- [74] A. Suebsing and N. Hiransakolwong, "Feature Selection Using Euclidean Distance and Cosine Similarity for Intrusion Detection Model," *2009 First Asian Conference on Intelligent Information and Database Systems*, Dong hoi, Vietnam, 2009, pp. 86-91, doi: 10.1109/ACIIDS.2009.23.
- [75] T. Srikaewsiew, K. Khianchainat, A. Tharatipyakul, S. Pongnumkul and S. Kanjanawattana, "A Comparison of the Instructor-Trainee Dance Dataset Using Cosine similarity, Euclidean distance, and Angular difference," *2022 26th*

International Computer Science and Engineering Conference (ICSEC), Sakon Nakhon, Thailand, 2022, pp. 235-240, doi: 10.1109/ICSEC56337.2022.10049368.

[76] H. Alkan and H. Celebi, "The Implementation of Positioning System with Trilateration of Haversine Distance," *2019 IEEE 30th Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, Istanbul, Turkey, 2019, pp. 1-6, doi: 10.1109/PIMRC.2019.8904289.

[77] A. V. Hazarika, G. J. S. R. Ram, E. Jain, D. Sushma and Anju, "Cluster analysis of Delhi crimes using different distance metrics," *2017 International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS)*, Chennai, India, 2017, pp. 565-568, doi: 10.1109/ICECDS.2017.8389500.

[78] M. Arora, H. Singh and A. Kaur, "Distance based verification techniques for online signature verification system," *2015 2nd International Conference on Recent Advances in Engineering & Computational Sciences (RAECS)*, Chandigarh, India, 2015, pp. 1-5, doi: 10.1109/RAECS.2015.7453285.

[79] R. Kumar, "Analysis of shape alignment using Euclidean and Manhattan distance metrics," *2017 International Conference on Recent Innovations in Signal processing and Embedded Systems (RISE)*, Bhopal, India, 2017, pp. 326-331, doi: 10.1109/RISE.2017.8378175.

[80] T. Shongwe, T. G. Swart and H. C. Ferreira, "Distance-Preserving Mapping with Euclidean Distance for 4-ary PAM," *2018 11th International Symposium on Communication Systems, Networks & Digital Signal Processing (CSNDSP)*, Budapest, Hungary, 2018, pp. 1-6, doi: 10.1109/CSNDSP.2018.8471854.

[81] Salman Ravoof. Brotli Compression: A Fast Alternative to GZIP Compression. 2022. URL: <https://kinsta.com/blog/brotli-compression/>

[82] Greg Roelofs. PNG: The Definitive Guide. 1999. URL: <http://www.libpng.org/pub/png/book/copyright.html>

[83] Gregory K. Wallace. The JPEG Still Picture Compression Standard. 1991. URL: <https://www.ijg.org/files/Wallace.JPEG.pdf>

[84] emCompress-LZMA. 2025. URL: <https://www.segger.com/products/compression/emcompress/emcompress-lzma/>

- [85] Better Compression with Zstandard. 2017. URL: <https://gregoryszorc.com/blog/2017/03/07/better-compression-with-zstandard/>
- [86] Leo Liberti Carlile Lavor. Euclidean Distance Geometry. URL: <https://dokumen.pub/euclidean-distance-geometry-an-introduction-978-3-319-60792-4-3319607928-978-3-319-60791-7.html>
- [87] David Salomon. Data Compression Third Edition. 20-30. 2004. URL: <https://dn790007.ca.archive.org/0/items/pdfy-Ezz1QE-FRYsgeVKU/Data%20Compression%20The%20Complete%20Reference.pdf>
- [88] D.A. Huffman. A method for the construction of minimum redundancy codes. *Proc. IRE*, 40:1098–1101, 1951.
- [89] Khalid Sayood. Introduction to Data Compression FOURTH EDITION. 43-52. 2012. URL: https://students.aiu.edu/submissions/profiles/resources/onlineBook/E3B9W5_data%20compression%20computer%20information%20technology.pdf
- [90] Bhadane D. S., Kanawade S. Y. Comparative study of RLE & K-RLE compression and decompression in WSN. 2016 3rd International Conference on Advanced Computing and Communication Systems (ICACCS). Coimbatore, India, 2016. P. 1–5. DOI: 10.1109/ICACCS.2016.7586319.
- [91] Roukhami M., Lahbib Y., Mami A. A new efficient energy implementation of K-RLE algorithm for WSN. 2016 7th International Conference on Sciences of Electronics, Technologies of Information and Telecommunications (SETIT). Hammamet, Tunisia, 2016. P. 66–70. DOI: 10.1109/SETIT.2016.7939843.
- [92] Mechta D., Harous S. HC-LEACH: Huffman Coding-based energy-efficient LEACH protocol for WSN. 2020 11th IEEE Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON). New York, NY, USA, 2020. P. 0932–0938. DOI: 10.1109/UEMCON51285.2020.9298061.
- [93] Sudarshan E., Satyanarayana C., Bindu C. S. A Parallel RLE Entropy Coding Technique for DICOM Images on GPGPU. 2017 International Conference on Current Trends in Computer, Electrical, Electronics and Communication (CTCEEC). Mysore, India, 2017. P. 963–966. DOI: 10.1109/CTCEEC.2017.8455071.

[94] Birajdar A., Agarwal H., Bolia M., Gupte V. Image Compression using Run Length Encoding and its Optimisation. 2019 Global Conference for Advancement in Technology (GCAT). Bangalore, India, 2019. P. 1–6. DOI: 10.1109/GCAT47503.2019.8978464.

[95] . Patil R. B., Kulat K. D. Audio compression using dynamic Huffman and RLE coding. 2017 2nd International Conference on Communication and Electronics Systems (ICCES). Coimbatore, India, 2017. P. 160–162. DOI: 10.1109/CESYS.2017.8321256.

[96] Khaitu S. R., Panday S. P. Canonical Huffman Coding for Image Compression. 2018 IEEE 3rd International Conference on Computing, Communication and Security (ICCCS). Kathmandu, Nepal, 2018. P. 184–190. DOI: 10.1109/CCCS.2018.8586816.

[97] Pal C., Pankaj S., Akram W., Acharyya A., Biswas D. Modified Huffman based compression methodology for Deep Neural Network Implementation on Resource Constrained Mobile Platforms. 2018 IEEE International Symposium on Circuits and Systems (ISCAS). Florence, Italy, 2018. P. 1–5. DOI: 10.1109/ISCAS.2018.8351234.

[98] Arshad R., Saleem A., Khan D. Performance comparison of Huffman Coding and Double Huffman Coding. 2016 Sixth International Conference on Innovative Computing Technology (INTECH). Dublin, Ireland, 2016. P. 361–364. DOI: 10.1109/INTECH.2016.7845058.