

УДК 621.382

МАЗІН М. Ю., ОНИКІЄНКО Ю. О.

Національний технічний університет України "Київський політехнічний інститут імені Ігоря Сікорського", Україна

ОСОБЛИВОСТІ ВИКОРИСТАННЯ МЕТОДІВ ХАФФМАНА ТА RLE ДЛЯ СТИСНЕННЯ ЗОБРАЖЕНЬ В СИСТЕМАХ НА МІКРОКОНТРОЛЕРАХ

Мета: Дослідження ефективності використання методів стиснення RLE та Хаффмана для стиснення зображень з урахуванням обмежених ресурсів мікроконтролерів і, відповідно, необхідності зменшення часу передачі зображення та зменшення витрат енергії в бездротових сенсорних мережах.

Методика: Експериментальне дослідження ступеня стиснення зображень з використанням методів Хаффмана та RLE, реалізованих програмно для мікроконтролерного модулю ESP32-CAM з вбудованою камерою. Оцінка та порівняння ступеню стиснення зображень, виконаного за допомогою алгоритмів Хаффмана та RLE.

Результати: Створено програмний модуль для реалізації алгоритмів стиснення Хаффмана та RLE на мікроконтролері ESP32-CAM. Досліджена ефективність використання алгоритмів Хаффмана та RLE, для стиснення зображень, використовуючи обмежені ресурси мікроконтролера, порівняна ефективність стиснення та розміри стиснених файлів. В результаті досліджень встановлено, що метод RLE показав низьку ефективність стиснення зображень (менше 1%) у порівнянні з методом Хаффмана, який забезпечив ступінь стиснення для вибраного зображення 11%. Попереднє використання вейвлет перетворення Хаара позитивно впливає на результати стиснення для обох досліджених методів: для RLE ступінь стиснення зріс до 16%, Хаффмана до 31% без втрат якості зображення. Для порівняння використано метод JPEG, який забезпечує ступінь стиснення того ж зображення до 70%, однак з втратою якості зображення.

Наукова новизна: Виконано аналіз ефективності стиснення даних за допомогою методів Хаффмана та RLE, реалізованих з урахуванням обмежених обчислювальних можливостей 32-х бітних мікроконтролерів з використанням додаткової зовнішньої пам'яті. Досліджено вплив вибору методу на час обробки зображень та кількість задіяної пам'яті.

Практична значимість: Створено програмне забезпечення, яке реалізує алгоритми стиснення даних Хаффмана та RLE. Підтверджена доцільність використання наведених методів в системах на 32-х бітних мікроконтролерах для зменшення споживання енергії при виконанні перетворень і передачі зображень, що дає змогу подовжити термін служби батареї сенсорного вузла. При цьому досліджені алгоритми для стиснення зображення можуть забезпечити якість вихідного зображення з найменшими можливими втратами даних.

Ключові слова: стиснення зображення; метод Хаффмана; метод RLE; Run-length encoding.

Вступ. У наш час бездротові сенсорні мережі (WSN) мають широке коло сфер застосування. Це і моніторинг навколишнього середовища, промисловий моніторинг, розумне сільське господарство, спостереження та відстеження об'єктів, автоматизація процесів, моніторинг здоров'я та багато інших застосувань. Базовою одиницею WSN є вузли, які спілкуються один з одним за допомогою бездротових технологій і можуть бути, як самоорганізованими, так і керованими через базові станції або шлюзи. Кожен вузол має у складі мікроконтролер або мікропроцесор, який отримує інформацію з датчиків, обробляє цю інформацію і передає її через бездротовий канал зв'язку. Для забезпечення живлення вузла зазвичай використовується акумулятор. Найбільше енергії споживається в процесі передачі даних, що важливо враховувати через обмежену ємність акумуляторів. Одним із можливих рішень, що можуть призвести до економії енергії, це зменшення обсягу даних, що передаються між вузлами сенсорної мережі. Зменшення обсягу даних можна досягти використанням

ефективної техніки їх стиснення. Тому удосконалення методів та алгоритмів стиснення даних наразі є активним напрямком досліджень [5].

Вузли сенсорних мереж мають певні обмеження по споживаній енергії, і відповідно продуктивність мікроконтролера та його пам'ять теж обмежені. Тому для стиснення даних використовують методи, що потребують відносно невеликих затрат мікроконтролерних ресурсів з одного боку і забезпечують достатній ступінь стиснення даних. На сьогодні до найбільш вживаних можна віднести такі методи як Run Length Encoding (RLE) та стиснення Хаффмана (Huffman) та їх модифікації.

Метод стиснення RLE та його модифікації типу K-RLE використовують в сенсорних мережах в першу чергу орієнтуючись на можливості мікроконтролера. Модифікація варіанту метода K-RLE під назвою C-RLE, запропонована в [1] і спроектована саме з урахуванням особливостей мікроконтролерів Cortex-M3. Алгоритм K-RLE, реалізований у WSN, забезпечує компроміс між споживанням енергії та ефективністю стиснення. Експериментальні результати показали, що запропонований підхід зменшує час обробки та зменшує споживання енергії під час стиснення, зберігаючи однаковий коефіцієнт стиснення K-RLE. Апаратна платформа вузла WSN базується на мікроконтролері STM32L1xx на базі ARM Cortex-M3. Запропонований метод стиснення даних C-RLE забезпечує зниження споживання енергії в процесі стиснення на 30% у порівнянні з K-RLE.

Стиснення Хаффмана також використовується у бездротових сенсорних мережах. Наприклад, у випадку, коли мережа містить велику кількість вузлів на обмеженій площі. В силу близького розташування вузли оперують схожими даними, що призводить до надмірного споживання енергії та може спричинити втрату пакетів даних. В таких випадках для збору інформації використовують спеціальний протокол LEACH (Low-Energy Adaptive Clustering Hierarchy). Додавання в алгоритм обробки стиснення Хаффмана дозволяє вирішити проблему [6]. Результати комп'ютерного моделювання показують ефективність запропонованого алгоритму та зниження енергоспоживання приблизно на 38% порівняно з LEACH протоколом без стиснення.

Постановка завдання. Метою роботи є дослідження особливостей використання методів RLE та Хаффмана у системах на 32-х бітних мікроконтролерах на прикладі стиснення зображення з відеокамери мікроконтролерного модулю ESP32-CAM для передачі через бездротовий канал зв'язку. Для досягнення мети необхідно здійснити захоплення та обробку зображення, стиснути його та оцінити вплив параметрів стиснення на розмір отриманого файлу. Під час експерименту необхідно також оцінити час виконання операції стиснення та об'єм задіяної оперативної пам'яті та пам'яті програм мікроконтролера.

Результати попередніх досліджень. На сьогодні метод RLE та його модифікації широко використовуються у комп'ютерних та вбудованих системах. В медичній галузі для збереження зображень використовується стандарт DICOM. Для стиснення таких зображень використовують техніку ентропійного кодування без втрат, щоб усунути надмірність зображення, зберігши його якість на достатньо високому рівні. RLE кодування – це основний метод ентропійного кодування, який використовується в зображеннях DICOM. В роботі [2] запропоновано метод паралельного RLE, який при використанні графічних процесорів дозволяє до 5-ти разів підвищити швидкість обробки зображень. Переваги та недоліки RLE кодування зображень розглянуто в [7], оцінена ефективність різних варіацій алгоритму RLE кодування, підкреслено, що ефективність стиснення зменшується для зображень з великою кількістю інформації в них. Ефективність використання RLE кодування на комп'ютері для стиснення аудіо файлів у порівнянні з методом динамічного стиснення Хаффмана оцінена в роботі [3]. В результаті ступінь стиснення забезпечувалась приблизно однаковою, але перетворення Хаффмана виконувалось в десятки разів швидше. Таким чином, варто

підкреслити, що використання RLE кодування найбільш доцільне в системах з обмеженими обчислювальними ресурсами [1], а ефективність залежить від типу даних, що обробляються. Ефективність кодування збільшується у випадках наявності послідовностей незмінних даних.

Класичним використанням кодування Хаффмана є стиснення зображень у форматі JPEG. Однак стиснення Хаффмана має і інші модифікації та застосування. В роботі [4] виконано порівняльний аналіз стандартного та канонічного методів кодування Хаффмана. Встановлено, що ступінь стиснення канонічного кодування Хаффмана кращий, ніж у стандартного кодування Хаффмана, але оригінальність зображення краще зберігається в методі стандартного кодування Хаффмана. Таким чином, реконструкція зображень із стисненого файлу призведе до більшої втрати даних у випадку канонічного кодування Хаффмана. В роботі [8] представлено модифікований метод кодування Хаффмана, який було застосовано до вагових коефіцієнтів нейронної мережі на прикладі декількох мереж. Такий підхід забезпечує максимальне стиснення нейронної мережі приблизно на 64 % і мінімальний час розпакування одного модуля 0,33 секунди, що важливо для мобільних платформ з обмеженим обсягом пам'яті. Порівняння кодування Хаффмана та подвійного кодування Хаффмана виконано в [9]. Автори прийшли до висновку, що подвійне кодування Хаффмана є більш ефективним методом стиснення, ніж кодування Хаффмана, оскільки надмірність у подвійному кодуванні Хаффмана менша. Але подвійне кодування вимагає більше часу.

Особливості стиснення даних за допомогою алгоритму RLE. Алгоритм RLE – це простий метод стиснення даних, який замінює повторювані символи на пари символ-кількість. Наприклад, рядок “AAAAABBBCCD” можна стиснути за допомогою RLE до “A5B3C2D1”. Математичну формулу алгоритму RLE можна записати так:

$$RLE(S) = \sum_{i=1}^n (s_i, c_i), \quad (0)$$

де S – вихідний рядок;

n – кількість унікальних символів в S ;

s_i – i -й символ в S ;

c_i – кількість його повторень.

Особливості стиснення даних за допомогою алгоритму Хаффмана. Алгоритм Хаффмана – це метод стиснення даних, який замінює символи на коди змінної довжини, залежно від їх частотності в повідомленні. Цей алгоритм гарантує оптимальність кодування, тобто мінімальну середню довжину коду для даного алфавіту. Математичну формулу алгоритму Хаффмана можна записати так:

$$L = \sum_{i=1}^n p_i l_i, \quad (2)$$

де L – середня довжина коду;

n – кількість символів в алфавіті;

p_i – ймовірність появи i -го символу в повідомленні;

l_i – довжина коду i -го символу.

Алгоритм Хаффмана складається з двох етапів: побудови оптимального кодового дерева та побудови відображення код-символ на основі цього дерева. Основна ідея алгоритму полягає в тому, що об'єднуються два символи з найменшими ймовірностями в один новий символ з сумарною ймовірністю, і цей процес повторюється до тих пір, поки не залишиться один символ. Кожен крок об'єднання відповідає створенню нового вузла дерева, який має двох

потомків – об'єднані символи. Код кожного символу отримується шляхом проходження від кореня дерева до листка, що відповідає цьому символу, і запису біт 0 або 1 на кожному розгалуженні.

Для прикладу, нехай є наступна таблиця (табл. 1) частот повторення символів.

Таблиця 1

Таблиця частот повторення символів послідовності Хаффмана

Символ	A	B	C	D	E
Частота повторення	15	7	6	6	5

Застосовуючи алгоритм Хаффмана, отримуємо таке кодове дерево (рис. 1).

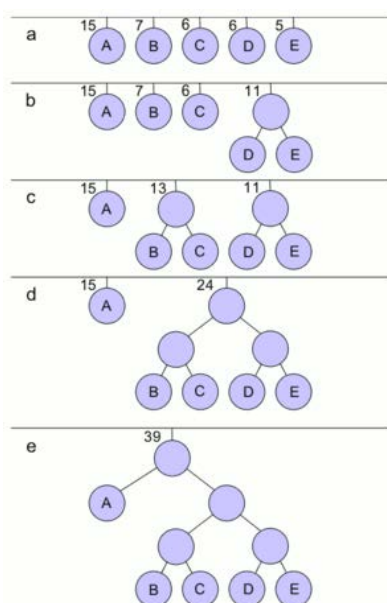


Рис. 1. Вигляд кодового дерева для заданих частот повторення символів алгоритму Хаффмана

Тоді коди кожного символу будуть такими як в табл. 2.

Таблиця 2

Таблиця з кодами Хаффмана для визначеної таблиці символів

Символ	A	B	C	D	E
Код	0	100	101	110	111

Опис апаратного та програмного забезпечення для дослідження алгоритмів стисненні зображень. Для обробки зображень часто використовують 32-х бітні мікроконтролери, що мають достатньо високі тактові частоти і відповідну продуктивність. Але захоплення та обробка зображень вимагає використання великих об'ємів оперативної пам'яті, якої недостатньо у більшості мікроконтролерів загального призначення. Підключення до мікроконтролера додаткової зовнішньої оперативної пам'яті з інтерфейсом SPI дозволяє цю проблему вирішити. Однак використання зовнішньої пам'яті знижує швидкість обробки зображень. Для експериментальних досліджень алгоритмів стиснення використано модуль ESP32-CAM [10], який складається з плати мікроконтролера з підключеною камерою та плати з USB інтерфейсом та колами живлення. На платі мікроконтролера використано двоядерний мікроконтролер ESP32 з мікропроцесорним ядром Tensilica Xtensa LX6 та вбудованим

каналом Wi-Fi, мікросхему пам'яті PSRAM об'ємом 4 Мбайт та 2-мегапіксельну камеру. Зображення передавались безпосередньо на web-сторінку в браузері, завдяки наявності у мікроконтролера ESP32 вбудованого Wi-Fi. Основні характеристики плати мікроконтролера:

- Роздільна здатність камери – 2 мегапікселі
- Максимальна частота тактового генератора – 240 МГц
- Об'єм оперативної пам'яті (внутрішня + зовнішня) – 4+0,5 Мбайт
- Об'єм пам'яті програм – 4 Мбайт

Зовнішній вигляд модулю ESP-CAM з підключеною камерою наведено на рис. 2.



Рис. 2. Зовнішній вигляд модулю ESP32-CAM, камери та плати USB інтерфейсу

Алгоритм RLE відомий як один з найпростіших алгоритмів архівації графіки. Такий алгоритм орієнтований на зображення з невеликою кількістю кольорів, що підходить при проведенні експерименту з чорно-білими зображеннями. До позитивних сторін алгоритму можна додати що він не вимагає додаткової пам'яті при архівації, а також швидко працює, що важливо при використанні мікроконтролерів з обмеженими обчислювальними ресурсами. При використанні алгоритму Хаффмана потрібно мати на увазі те, що об'єм задіяної оперативної пам'яті залежить від частоти повторення символів. Відповідно при реалізації алгоритму Хаффмана на мікроконтролері завжди має бути зарезервовано певний об'єм оперативної пам'яті.

Основною метою проведення експериментів було порівняти алгоритми стиснення Хаффмана та RLE, та оцінка доцільність їх використання для підвищення ефективності стиснення зображення з камери без втрати якості самого зображення. Чим більше ступінь стиснення, тим менше часу необхідно для того, щоб зображення було передано від мікроконтролера на інші пристрої. В ході проведення експерименту зображення з камери отримувалось з використанням функцій, що входять в бібліотеку по роботі з зображеннями для мікроконтролера ESP32-CAM [11]. Далі зображення оброблялось за допомогою двох реалізованих алгоритмів стиснення Хаффмана та RLE. З метою оцінки впливу складності змісту зображення та досягнення найбільшого ступеня стиснення зображень методами стиснення Хаффмана та RLE проведено експеримент з додатковою попередньою обробкою зображень з використанням перетворення Хаара. Експеримент по обробці зображень за допомогою перетворення Хаара на мікроконтролері ESP32 докладно описано в [12]. Для спрощення та прискорення процесу порівняння результатів створено html-сторінку, яка зберігалась у пам'яті мікроконтролера і на яку передавалось вихідне зображення та початковий та кінцевий розміру обробленого файлу.

В ході проведення експериментів досліджувалось наступне:

1. Тестування програмного коду, який реалізує методи стиснення Хаффмана та RLE для масиву даних з заданими значеннями, що повторюються.

2. Співвідношення розмірів оригінального та стисненого зображень, оброблених за допомогою методів Хаффмана та RLE. Експеримент виконувався для зображення розміром 96x96 пікселів.

3. Співвідношення розмірів оригінального та стисненого зображень, оброблених попередньо за допомогою перетворення Хаара, а потім методів Хаффмана та RLE. Експеримент виконувався для зображення розміром 96x96 пікселів.

4. Порівняння ефективності стиснення за допомогою методів Хаффмана та RLE, з попередньою обробкою зображення методом Хаара і без неї. Порівняння реалізованих методів з JPEG.

Аналіз результатів експерименту. Для усіх експериментів наведено наступні данні: використаний метод стиснення для зображення, а також відповідні наведеним даним значення ступеня стиснення, розміри файлів до стиснення та після, а також наведено час виконання.

Ступінь стиснення визначено в формулі (3) відповідно до [13]:

$$\text{СтупіньСтиснення} = \frac{(\text{КількістьВхіднихДаних} - \text{СтисненийОбсягДаних})}{\text{КількістьВхіднихДаних}} * 100\% . \quad (3)$$

Таблиця 3

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	99,89	1

Результати тестування програмного коду реалізації методів стиснення Хаффмана та RLE наведені в таблиці 3. Для даного експерименту використовувався масив даних з двома числами, кожне з яких заповнювала половину тестового масиву. Як видно з таблиці 3, найкращий результат отримано для RLE, всього 0,1% оригінального зображення, оскільки на виході отримано лише 2 значення з їх кількісними значеннями, що в байтовому еквіваленті займає всього 10 байт. Для методу стиснення Хаффмана розмір вихідної зображення є набагато більшим за результати, отримані для стиснення RLE, і становить 2305 байт. Такі результати пояснюються тим, що алгоритм Хаффмана заміняє кожен піксель зображення на його закодоване значення, що заздалегідь обраховується та зберігається у таблиці частот.

Таблиця 4

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	8192	11.11	18	1964
RLE	9216	9193	0.24	3	1144
Хаффман + вейвлет	9216	6402	30.53	96	3764
RLE + вейвлет	9216	7738	16.03	20	2088
JPEG	9216	2798	69.63	7	12020

Оцінка ефективності стиснення методів Хаффмана та RLE проводилась на зображенні розміром 96x96 пікселів, наведеному на рис. 3. Спершу оцінювалась ефективність стиснення самого зображення (рис. 3а), а потім ефективність стиснення вейлет перетворення вказаного зображення (рис. 3б). Результати експериментів по стисненню зображень за методами Хаффмана та RLE наведені в таблиці 4. Як видно з таблиці, розмір стисненого зображення методом RLE практично співпадає з оригіналом (менше на 0,2%). Такий результат пов'язаний з тим, що для отриманого з камери модуля ESP32-CAM зображення градації сірого складають неповторювані послідовності пікселів, що робить стиснення методом RLE неефективним. Метод Хаффмана показало кращу ефективність, порівняно з методом RLE. Розмір стисненого файлу менше розміру оригінального файлу на 11%, а ступінь стиснення не залежить критично від повторюваності пікселів, що є перевагою алгоритму Хаффмана.



Рис. 3. Досліджувані зображення: а) – оригінальне зображення з камери модулю ESP32-CAM, б) – зображення після використання перетворення Хаара

Результати експерименту для зображення розміром 96x96 пікселів за допомогою методів стиснення Хаффмана та RLE з використанням перетворення вейвлету Хаара наведені в таблиці 4. Під час використання методу Хаффмана отримано результати стиснення на рівні 30%. Найкращі результати стиснення можна отримати для монотонних зображень, оскільки вони мають найбільш корельовані значення таблиці частот. Застосування методу RLE показало стиснення на рівні 16%, тобто попереднє вейвлет перетворення підвищує ефективність RLE стиснення. Можна припустити, що ступені стиснення зображень з чітко виділеними кольорами та монотонних зображень будуть близькими.

Щодо використання ресурсів мікроконтролера експеримент показав, що метод стиснення RLE використовує найменше флеш пам'яті обраного мікроконтролера. Це пояснюється тим, що метод RLE має значно простіший алгоритм у порівнянні з алгоритмом методу Хаффмана. Результати використання флеш пам'яті для обраних методів стиснення у поєднанні з вейвлетом Хаара показують схожі результати. Слід зазначити що у порівнянні з усіма протестованими методами, метод JPEG використовує на 8 024 байт більше оперативної пам'яті у порівнянні з іншими методами, оскільки зберігає перетворене зображення. Також JPEG використовує найбільше флеш пам'яті, оскільки алгоритм містить додаткове обробку

даних перед стисненням: розділення на блоки, дискретне косинусне перетворення, квантування та інші операції.

Варто зауважити, що JPEG перетворення має найменший час виконання, хоча також використовує метод Хаффмана для стиснення даних. Однак у випадку JPEG метод Хаффмана оперує набагато меншим об'ємом даних, завдяки попередньому квантуванню, тобто зменшенню масштабу даних. Відповідно операція стиснення займає менше часу.

Висновки. В результаті експериментальних досліджень визначено, що використання методів Хаффмана та RLE для обробки зображень, отриманих з камери мікроконтролера, з метою подальшої передачі в системі має свої переваги, а саме, дозволяє досягти певного ступеню стиснення зображень без втрати якості. Як встановлено в результаті експериментів, стиснення може сягати до 11% для методу стиснення Хаффмана. Метод стиснення RLE є неефективним для обробки оригінального зображення з камери. Після використання вейвлет перетворення ефективність методів Хаффмана та RLE збільшилась до 31% та 16% відповідно. Слід враховувати, що обидва алгоритми можуть мати низьку ефективність стиснення у випадках, коли вхідний файл не містить повторюваних послідовностей (для RLE) або якщо розподіл символічних вихідних даних рівномірний (для Хаффмана). У порівнянні з перетворенням JPEG, методи Хаффмана та RLE показали гірші результати, але при цьому важливо підкреслити, що зазначені методи виконують стиснення без втрат, що має певні переваги при необхідності збереження роздільної здатності зображення. Методи Хаффмана та RLE дозволяють зберігати всю інформацію при зменшенні розміру файлу, що дозволяє в системах на мікроконтролерах передавати зображення з високою розбірливістю та точністю.

References

Література

1. Bhadane, D. S., Kanawade, S. Y. (2016). Comparative study of RLE & K-RLE compression and decompression in WSN. *2016 3rd International Conference on Advanced Computing and Communication Systems (ICACCS)*. Coimbatore, India, pp. 1–5, DOI: 10.1109/ICACCS.2016.7586319.
2. Roukhami, M., Lahbib, Y., Mami A. (2016). 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, pp. 66–70, DOI: 10.1109/SETIT.2016.7939843.
3. Mechta, D., Harous, S. (2020). 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, pp. 0932–0938, DOI: 10.1109/UEMCON.2020.9298061.
4. Sudarshan, E., Satyanarayana, C., Bindu, C. S. (2017). 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, pp. 963–966, DOI: 10.1109/CTCEEC.2017.8455071.

1. 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.
2. 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.
3. 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/UEMCON.2020.9298061.
4. 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*

5. Birajdar, A., Agarwal, H., Bolia, M., Gupte, V. (2019). Image Compression using Run Length Encoding and its Optimisation. *2019 Global Conference for Advancement in Technology (GCAT)*. Bangalore, India, pp. 1–6, DOI: 10.1109/GCAT47503.2019.8978464.
6. Patil, R. B., Kulat, K. D. (2017). Audio compression using dynamic Huffman and RLE coding. *2017 2nd International Conference on Communication and Electronics Systems (ICCES)*, Coimbatore, India, pp. 160–162, DOI: 10.1109/CESYS.2017.8321256.
7. Khaitu, S. R., Panday, S. P. (2018). Canonical Huffman Coding for Image Compression. *2018 IEEE 3rd International Conference on Computing, Communication and Security (ICCCS)*, Kathmandu, Nepal, pp. 184–190, DOI: 10.1109/CCCS.2018.8586816.
8. Pal, C., Pankaj, S., Akram, W., Acharyya, A., Biswas, D. (2018). 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, pp. 1–5, DOI: 10.1109/ISCAS.2018.8351234.
9. Arshad, R., Saleem, A., Khan, D. (2016). Performance comparison of Huffman Coding and Double Huffman Coding. *2016 Sixth International Conference on Innovative Computing Technology (INTECH)*, Dublin, Ireland, pp. 361–364, DOI: 10.1109/INTECH.2016.7845058.
10. ESP32-CAM camera development board. Ai-Thinker Technology. URL: <https://docs.ai-thinker.com/en/esp32-cam>.
11. ESP32 Camera Driver. *espressif.com*. URL: <https://github.com/espressif/esp32-camera/tree/master/conversions>.
12. Mazin, M. Yu., Onykienko, Yu. O. (2023). Vykorystannya veyvlet peretvoren' dlya obrobky zobrazen' v systemakh Internetu rechey na mikrokontrolerakh [Wavelet Transform Application for Image Processing in Microcontroller Based Internet of Things Systems]. *Technologies and Engineering*, No. 3 (14), DOI: 10.30857/2786-5371.2023.3.2 [in Ukrainian].
13. Ren Weizheng, Wang Haobo, Xu Lianming, Cui Yansong (2011). Research on a Quasi-Lossless Compression Algorithm Based on Huffman Coding. *2011 International Conference on Transportation, Mechanical, (CTCEEC)*. Mysore, India, 2017. P. 963–966. DOI: 10.1109/CTCEEC.2017.8455071.
5. 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.
6. 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.
7. 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.
8. 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.
9. 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.
10. ESP32-CAM camera development board. Ai-Thinker Technology. URL: <https://docs.ai-thinker.com/en/esp32-cam>.
11. ESP32 Camera Driver. *espressif.com*. URL: <https://github.com/espressif/esp32-camera/tree/master/conversions>.
12. Мазін М. Ю., Оникієнко Ю. О. Використання вейвлет перетворень для обробки зображень в системах Інтернету речей на мікроконтролерах. *Технології та інжиніринг*. 2023. № 3 (14). DOI: 10.30857/2786-5371.2023.3.2.
13. Ren Weizheng, Wang Haobo, Xu Lianming, Cui Yansong. Research on a Quasi-Lossless Compression Algorithm Based on Huffman Coding. *2011 International Conference on*

and Electrical Engineering (TMEE), Beijing 100876, China, pp. 1729–1732, DOI: 10.1109/TMEE.2011.6199546.

Transportation, Mechanical, and Electrical Engineering (TMEE). Beijing 100876, China, 2011. P. 1729–1732. DOI: 10.1109/TMEE.2011.6199546.

MAZIN MAKSYM

Post graduate student,
Department of Acoustic and
Multimedia Electronic Systems,
National Technical University of Ukraine
"Igor Sikorsky Kyiv Polytechnic Institute", Ukraine
<https://orcid.org/0000-0001-9566-6662>
E-mail: maksimalnuu@ukr.net

ONYKIIENKO YURI

PhD, Associate Professor,
Department of Acoustic and
Multimedia Electronic Systems,
National Technical University of Ukraine
"Igor Sikorsky Kyiv Polytechnic Institute", Ukraine
<https://orcid.org/0000-0001-7508-8391>
Scopus Author ID: 55819618700
Researcher ID: J-6589-2017
E-mail: yurionik@gmail.com

MAZIN M. Yu., ONYKIIENKO Yu. O.

National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", Kyiv, Ukraine

**PECULIARITIES OF USING HUFFMAN AND RLE METHODS FOR IMAGE
COMPRESSION IN MICROCONTROLLER SYSTEMS**

Goal: Purpose of this paper is Study of the efficiency of using RLE and Hoffman compression methods for image compression, taking into account the limited resources of microcontrollers and, accordingly, the need to reduce image transmission time and reduce energy consumption in the wireless sensor networks.

Method: An experimental study of the degree of image compression was conducted using the Hoffman and RLE methods. The algorithms are implemented in software for the ESP32-CAM microcontroller module with a built-in camera. The degree of image compression performed using the Hoffman and RLE algorithms was evaluated and compared.

Results: A software module was created to implement the Huffman and RLE compression algorithms on the ESP32-CAM microcontroller. The effectiveness of using Huffman and RLE algorithms for image compression with limited microcontroller resources is investigated, compression efficiency and compressed file sizes are compared. As a result of the research, it was found that the RLE method showed a low compression efficiency (less than 1%) compared to the Huffman method, which provided a compression ratio of 11% for the image under test. The previous use of the Haar wavelet transform has a positive effect on the compression results for both investigated methods: for RLE, the degree of compression increased to 16%; for Huffman to 31% without loss of image quality. The JPEG method was used for efficiency comparison, which provides a degree of compression of the same image up to 70%, but with a loss of compressed data quality.

Scientific innovation: An analysis of the data compression efficiency using the Huffman and RLE methods, implemented taking into account the limited computing capabilities of 32-bit microcontrollers with the use of additional external memory, was performed. The influence of the choice of method on the time of image processing and the amount of memory used was studied.

Practical significance: Software module has been created that implements Huffman and RLE data compression algorithms. The expediency of using the above methods in systems on 32-bit microcontrollers to reduce energy consumption during image transformations and transmission, which makes it possible to extend the battery life of the sensor node, has been confirmed. At the same time, the investigated algorithms for image compression can ensure the quality of the original image with the least possible data loss.

Keywords: image compression; Huffman; RLE; Run-length encoding.