

## ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ

УДК 004.032.26 + 519.72

Р. Н. Кветний, Ю. Ю. Іванов, В. В. Півошенко, А. С. Васюра

### АЛГОРИТМ РОЗПІЗНАВАННЯ ТА КОРИГУВАННЯ ПОШКОДЖЕНИХ QR-КОДІВ

Вінницький національний технічний університет

Анотація. Інтенсивний розвиток інформаційних технологій призвів до створення систем обміну даними, які застосовують комбіноване стиснення, захист від пошкоджень та зберігання інформації. Подібні системи зазвичай використовують матричні коди, які дають можливість компактно зберігати великий обсяг інформації та швидко розпізнаються сканувальним обладнанням. У даній статті розв'язано задачу розпізнавання та коригування сильно пошкоджених матричних кодів, а саме QR-кодів, у яких присутній високий рівень шуму, відсутні ключові елементи або накладаються кольори. Такі зображення не розпізнаються декодуючим програмним забезпеченням, оскільки структура детекторів елементів пошкоджена, а коригуюча здатність вбудованих кодів Ріда-Соломона не дозволяє виправити необхідну частину помилок. Тому запропоновано алгоритм, який базується на використанні штучної нейронної мережі Хеммінга-Ліппмана з базою еталонів та обробленні зображень у режимі ковзного вікна, що спрощує процес навчання мережі без використання трудомістких обчислювальних операцій, великих обсягів пам'яті та витрат часу, навіть для зображень високої роздільної здатності та великих розмірів. Процес навчання мережі складається з двох частин: обробка зображень, розпізнавання та корекція зразка. Для того, щоб домогтися коректного розпізнавання, необхідно експериментально виявляти оптимальні параметри навчання, завдяки яким в матриці еталонних образів рядки будуть достатньо відрізнятися один від одного. Для цього авторами на мові C# розроблено програмне забезпечення, за допомогою якого і проведено необхідні експериментальні дослідження. Визначено умови коректної роботи нейронної мережі (оптимальні значення розміру ковзного вікна та порогу для різних розмірів зображень QR-кодів), а також випадки, коли можливі помилки розпізнавання та нестабільності її виходів. Результати досліджень показують, що розроблений алгоритм можна застосовувати як додаткову процедуру розпізнавання та виправлення QR-кодів у різноманітних системах обміну даними.

**Ключові слова:** нейронна мережа Хеммінга-Ліппмана, алгоритм навчання, режим ковзного вікна, розпізнавання зображень, корекція помилок, QR-коди, обчислювальна складність, оптимальні параметри.

Аннотация. Интенсивное развитие информационных технологий привело к созданию систем обмена данными, которые применяют комбинированное сжатие, защиту от поврежденной и хранение информации. Подобные системы обычно используют матричные коды, которые дают возможность компактно хранить большой объем информации и быстро распознаются сканирующим оборудованием. В данной статье решена задача распознавания и коррекции сильно поврежденных матричных кодов, а именно QR-кодов, в которых присутствует высокий уровень шума, отсутствуют ключевые элементы или накладываются цвета. Такие изображения не распознаются декодирующим программным обеспечением, поскольку структура детекторов элементов повреждена, а корректирующая способность встроенных кодов Рида-Соломона не позволяет исправить необходимую часть ошибок. Поэтому предложен алгоритм, основанный на использовании искусственной нейронной сети Хемминга-Липпмана с базой эталонных образцов и обработке изображений в режиме скользящего окна, что упрощает процесс обучения сети без использования трудоемких вычислительных операций, больших объемов памяти и затрат времени, даже для изображений с высоким разрешением и большим размером. Процесс обучения сети состоит из двух частей: обработка изображений, распознавание и коррекция образца. Для того, чтобы добиться корректного распознавания, необходимо экспериментально выявлять оптимальные параметры обучения, благодаря которым в матрице эталонных образцов строки будут достаточно отличаться друг от друга. Для этого авторами на языке C# разработано программное обеспечение, с помощью которого и проведены необходимые экспериментальные исследования. Определены условия корректной работы нейронной сети (оптимальные значения размера скользящего окна и порога для различных размеров изображений QR-кодов), а также случаи, когда возможны ошибки распознавания и нестабильности ее выходов. Результаты исследований показывают, что разработанный алгоритм можно применять как дополнительную процедуру распознавания и исправления QR-кодов в различных системах обмена данными.

**Ключевые слова:** нейронная сеть Хемминга-Липпмана, алгоритм обучения, режим скользящего окна, распознавание изображений, коррекция ошибок, QR-коды, вычислительная сложность, оптимальные параметры.

Abstract. The intensive development of information technology has led to the creation of data exchange systems, that using combined compression, protection against damage and information storage. Such systems usually use matrix codes, that allow to store a large amount of information compactly and recognized it quickly by scanning equipment. In this article has been solved the task of recognition and correction of high damaged matrix codes, namely QR codes, where there is a high level of noise, there are no key elements or colors are overlaid. Such images are not recognized by decoding software as the structure of the detectors of elements is damaged and the correcting ability of the built-in Reed-Solomon codes doesn't allow to correct the necessary part of errors. That's why the algorithm based on the usage of artificial Hamming-Lippmann neural network with the base of samples and image processing in the sliding window mode is offered, which simplifies the learning process of the network without appliance of labor-intensive computational operations, large volumes of memory and high time consumption, even for images with high resolution and big size. The network learning process consists of two parts: image processing, sample recognition and correction. In order to achieve correct recognition, it is necessary to identify experimentally the optimal parameters of learning, thanks to which the matrix of samples the rows will differ sufficiently from each other. For this purpose, the authors developed software in C# language, with the helping of which the necessary experimental researches were performed. The conditions of correct work of the neural network (optimal values of the sliding window size and the threshold for different sizes of QR-code images) has been determined, as well as cases, when errors of recognition and instabilities of its outputs are possible. The results of the researches show, that the developed algorithm can be applied as an additional procedure of recognition and correction of QR codes in different data exchange systems.

**Key words:** Hamming-Lippmann neural network, learning algorithm, sliding window mode, images recognition, error corrections, QR codes, computational complexity, optimal parameters.

DOI: <https://doi.org/10.31649/1999-9941-2019-45-2-25-32>.

## Вступ

Розвиток сучасних інформаційних технологій сприяє створенню та поширенню систем обміну даними, які застосовують комбіноване стиснення, захист від пошкоджень та зберігання інформації. Зазвичай подібні системи включають матричні коди або двовимірні штрих-коди, такі як *Data Matrix*, *Aztec* і QR-коди, які є найбільш поширеними. Їхньою основною перевагою є легке розпізнавання сканувальним обладнанням (наприклад, камерою смартфона) і великий обсяг компактно збереженої інформації (до 7089 цифр або 4296 букв), що дозволяє використовувати QR-коди в торгівлі, виробництві, логістиці тощо. За даними веб-порталу *comScore* більше 20 мільйонів громадян США використовують смартфони для сканування цих кодів; в Японії вони розміщуються на всіх буклетах, довідниках і практично всіх товарах [1].

## Актуальність

Математична основа та принципи кодування і декодування QR-кодів представлені в стандарті ISO 18004 [2]. Для коригування помилок у даних кодах застосовують вбудовані недвійкові циклічні коди Ріда-Соломона з алгоритмом декодування Берлекемпа-Мессі (альтернатива – алгоритми Судана та Гао), пошуком положення помилок та оцінювання їх значень за Ченом та Форні [3, 4], що дозволяє відновити інформацію, якщо пошкоджено до  $R \leq 30\%$  матриці даних QR-коду [5, 6]. Саме тому ряд досліджень направлено на розв'язання задачі відновлення матриці даних для сильно пошкоджених QR-кодів ( $R > 30\%$ ), коли коригуюча здатність кодів Ріда-Соломона не дозволяє виправити необхідну частину помилок, тобто присутній високий рівень шуму або накладаються кольори. Для цього науковці застосовують модифіковані конструкції вбудованих кодів [7] та алгоритми виявлення ключових ознак [8], що вимагає використання досить трудомістких обчислювальних операцій у режимі реального часу. Це може бути недоцільним у ситуаціях, коли на матриці QR-коду відсутні ключові елементи-детектори або їхня структура значно пошкоджена, оскільки у цьому випадку декодує програму забезпечення не зможе навіть зчитати QR-код, а відповідно не буде намагатись розпізнати його. У даній роботі пропонується інший підхід до розв'язання задачі розпізнавання та коригування QR-кодів, який базується на додатковому використанні бази еталонів та штучної нейронної мережі Хеммінга-Ліппмана (НМХЛ) [9-11] з обробленням зображень у режимі ковзного вікна. Запропонований алгоритм можна застосовувати у системах обміну даними різного функціонального призначення у випадку, якщо матриця даних QR-коду пошкоджена настільки сильно, що немає можливості зчитати або розпізнати даний код звичайним чином.

## Мета

*Метою роботи* є розробка алгоритму розпізнавання та коригування сильно пошкоджених QR-кодів, його експериментальне дослідження і пошук оптимальних параметрів роботи.

## Задачі

Для досягнення мети необхідно виконати наступні задачі:

- 1) проаналізувати математичний апарат НМХЛ;
- 2) розробити програмне забезпечення, в якому реалізовано запропоновані алгоритми;
- 3) провести експериментальне дослідження розпізнавання та коригування сильно пошкоджених QR-кодів з використанням запропонованих розробок.

## Алгоритми навчання нейронної мережі, розпізнавання та коригування QR-кодів

Нехай існує еталонний набір зображень (база даних), призначений для навчання мережі. Необхідно порівняти зображення із рівнем шуму, подане на вхід нейронної мережі, з усіма еталонними зразками і зробити висновок, що воно є одним із зразків, або його не можна розпізнати точно.

Використаємо НМХЛ (рис. 1), яка складається з  $m$  вхідних ( $S$ ),  $n$  прихованих ( $Z, A$ ),  $n$  вихідних ( $Y$ ) нейронів і 4 прошарків. Значення  $n$  дорівнює кількості опорних зображень, збережених мережею і представлених у вигляді  $m$ -мірних біполярних векторів  $S = (s_1, \dots, s_m)$ . Також у цій мережі використовуються значення зміщення  $b$ . Матриця ваг  $W$  розмірністю  $m \times n$  містить значення синаптичних зв'язків мережі прямого поширення, матриця  $E$  розмірності  $n \times n$  зберігає значення синаптичних зв'язків нейронів мережі *MaxNet*. Функція активації нейронів є пороговою.

*Алгоритм навчання НМХЛ* включає в себе наступну послідовність кроків:

*Крок 1.1.* Проаналізувати матрицю  $P$  пікселів розміром  $H \times B$ , де  $H$  і  $B$  – висота і ширина зображення відповідно, рахуючи кількість темних пікселів. Щоб визначити, чи є аналізований піксель темним, необхідно зробити перетворення у відтінки сірого за формулою [12]:

$$Y = 0,298839 \cdot R + 0,586811 \cdot G + 0,11435 \cdot B, \quad (1)$$

де  $R, G, B$  – значення відповідних кольорних складових.

*Крок 1.2.* Вибрати певне граничне значення  $L$ .

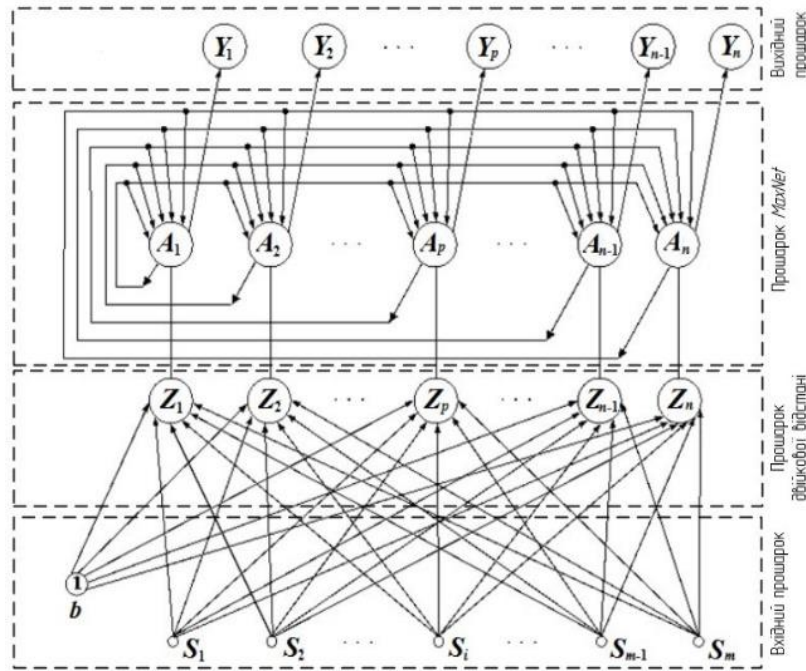


Рисунок 1 – Архітектура НМХЛ

Крок 1.3. Порівняти  $Y$  із  $L$ . Якщо  $0 \leq Y \leq L$ , тоді відповідний піксель вважається темним, інакше – білим.

Крок 1.4. Записати отриману двійкову матрицю зображення розміром  $m = H \times B$ . Таким чином, для всіх еталонних зображень  $n$  можна визначити відповідні бінарні вектори, які будуть використовуватися пізніше для навчання мережі і розпізнавання невідомих зразків.

Крок 1.5. Вибрати значення  $K$  (розмір ковзного вікна), яке дорівнює кількості рядків або стовпців, які будуть оброблятися (рис. 2). Змінюючи розмір вікна  $K$ , можна змінити точність аналізу зображення.

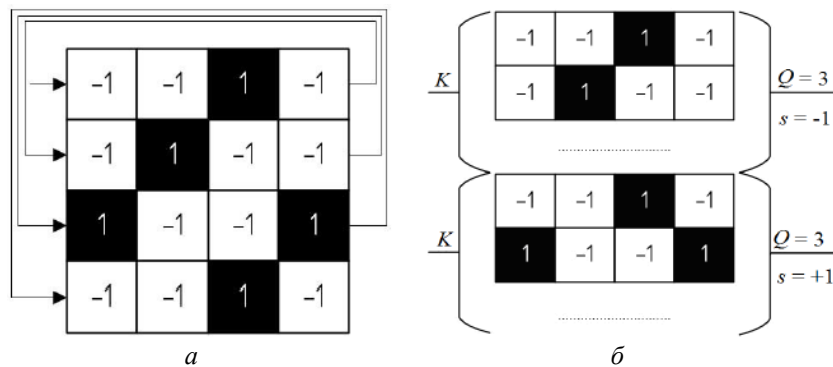


Рисунок 2 – Обробка зображення порядково (а) та у режимі ковзного вікна (б)

Крок 1.6. Вибрати порогову величину  $Q$ .

Крок 1.7. Обробити  $K$  рядків та обчислити  $N_i$  – кількість темних пікселів. Якщо  $N_i \geq Q$ , тоді  $i$ -ий елемент бінарного вектору  $S_i$  дорівнює «+1», інакше – «-1». Цей крок повторюється доки всі рядки не будуть оброблені. Тепер маємо вектор  $S$  із  $m$  елементами, для подачі на вхідні нейрони.

Крок 1.8. Повторити кроки 1.1-1.7 для всіх еталонних зображень із бази даних QR-кодів.

Крок 1.9. Базуючись на результатах, отриманих на кроці 1.8 (для всіх зображень), створити матрицю ваг  $W$  розмірністю:

$$M = m \times n = \lceil X / K \rceil \times n, \quad (2)$$

де  $X$  – висота  $H$  або ширина  $B$  даного зображення.

Модифікація, представлена в кроках 1.5-1.7 та 1.9, значно зменшить довжину вхідного вектору (особливо для зображень великих розмірів), що, в свою чергу, спростить обчислювальну складність навчання мережі.

Алгоритм розпізнавання та коригування QR-кодів можна сформулювати у такому вигляді:

Крок 2.1. Вибрати вхідне зображення для розпізнавання та виконати кроки 1.1-1.7.

Крок 2.2. Запустити процес розпізнавання [13, 14]. Вихідний сигнал для  $S$  нейронів має вигляд:

$$U_{outputS_i} = f_s(U_{inputS_i}) = s_i. \quad (3)$$

Крок 2.3. При поданні вхідного вектору кожен із  $Z$  нейронів від 1 до  $n$  обчислює свій вхідний сигнал відповідно до формули:

$$U_{inputZ_k} = b + \sum_{i=1}^m w_{ik} \cdot U_{outputS_i} = \frac{m}{2} + \frac{1}{2} \sum_{i=1}^m w_{ik} \cdot s_i, \quad (4)$$

де  $b$  – зміщення;  $w_{ik}$  – значення ваги від  $i$  вхідного нейрона до  $k$  нейрона  $Z$  (біполярне значення +1 або -1).

Крок 2.4. Обчислити значення функції активації для прихованих  $Z$  нейронів, використовуючи наступне правило:

$$U_{outputZ_k} = f_z(U_{inputZ_k}) = \begin{cases} 0, & \text{якщо } U_{inputZ_k} \leq 0; \\ v \cdot U_{inputZ_k}, & \text{якщо } 0 < U_{inputZ_k} \leq U; \\ U, & \text{якщо } U_{inputZ_k} > U, \end{cases} \quad (5)$$

де  $v$  – параметр нормалізації;  $U = 1/v$  – константа.

Крок 2.5. Вихідний сигнал від  $Z$  нейронів є вхідним сигналом підмережі *MaxNet*:

$$U_{inputA_k} = U_{outputZ_k}. \quad (6)$$

Крок 2.6. Дана підмережа використовує ітераційний процес отримання максимального вхідного сигналу в наступному вигляді:

$$U_{outputA_k}(0) = U_{outputZ_k}, \quad (7)$$

$$U_{inputZ_k}(t+1) = \sum_{i=1}^n w_{ik} \cdot U_{outputA_i}(t) = U_{outputA_k}(t) - \varepsilon \cdot \sum_{i=1, i \neq k}^n U_{outputA_i}(t), \quad (8)$$

де  $t$  і  $(t+1)$  – попередня та наступна ітерації;  $0 < \varepsilon \leq 1$ ;  $w_{ik} = 1$  для *MaxNet*, якщо  $i = k$  (для мережі Хопфілда  $w_{ik} = 0$ ) та  $-\varepsilon$  – в іншому випадку.

Крок 2.7. Обчислити значення функції активації для прихованих  $A$  нейронів, використовуючи наступне правило:

$$U_{outputA_k}(t) = f_A(U_{inputA_k}(t)) = \begin{cases} U_{inputA_k}(t), & \text{якщо } U_{inputA_k} > 0; \\ 0, & \text{якщо } U_{inputA_k} \leq 0. \end{cases} \quad (9)$$

Крок 2.8. Ітераційний процес (кроки 2.6 і 2.7) повторюється до тих пір, поки норма двох векторів не стане більшою, ніж значення  $eps_1$ . Отже, умова стабілізації формулюється наступним чином [9]:

$$\|U_{outputA_k}(t+1) - U_{outputA_k}(t)\| = \sqrt{\sum_{k=1}^n (U_{outputA_k}(t+1) - U_{outputA_k}(t))^2} \leq eps_1. \quad (10)$$

Крок 2.9. Вихідні сигнали нейронів  $A$  надходять, як вхідні в нейрони  $Y$ , що можна представити у формі:

$$U_{outputY_k} = U_{outputA_k}. \quad (11)$$

Крок 2.10. Обчислити значення функції активації для вихідних  $Y$  нейронів, використовуючи відхилення  $eps_2 = 10^{-5} - 10^{-6}$  або нуль в ідеальному випадку. Це можна зобразити в такому вигляді:

$$U_{outputY_k} = f_Y(U_{inputY_k}) = \begin{cases} 1, & \text{якщо } U_{inputY_k} > eps_2; \\ 0, & \text{якщо } U_{inputY_k} \leq eps_2, \end{cases} \quad (12)$$

або, якщо потрібно виконати багато ітерацій, то необхідно вибрати максимальне значення  $U_{inputY}$ :

$$U_{outputY_k} = \max_{1,n} = (\{U_{inputY_k}\}). \quad (13)$$

Крок 2.11. Відкоригувати QR-код. Індекс позитивного елемента  $k = \overline{1, n}$  вихідного сигналу нейронів  $U_{output_k}$  вказує номер еталонного зображення, яке найкраще підходить вхідному тестовому сигналу.

Виходом мережі в ідеальному випадку повинен бути вектор з одним позитивним елементом. Але іноді вхідна вибірка може бути дуже зашумленою, і на виході мережі може з'явитися кілька позитивних значень. У цьому випадку вважається, що мережа не може однозначно класифікувати вхідне зображення. Але вдало підібрані параметри алгоритму навчання можуть допомогти подати цей недолік. У наступному розділі статті продемонстрована працездатність запропонованих алгоритмів на практиці.

### Результати експериментальних досліджень

Для вирішення поставленої задачі на мові програмування C# нами було розроблене універсальне програмне забезпечення. Щоб визначити оптимальні значення порогової величини  $Q$  і розміру ковзного вікна  $K$  для запропонованого алгоритму, були проведені експерименти, в яких використано зображення QR-кодів розміром  $200 \times 200$ ,  $512 \times 512$  та  $1024 \times 1024$  з високим рівнем шуму та пошкоджень, відсутністю деталей або злиттям кольорів.

*Приклад.* На рис. 3 і 4 представлено 5 навчальних і тестових зображень (QR-коди із пошкодженнями). Дані зображення не розпізнаються програмним забезпеченням для декодування, оскільки структура детекторів елементів пошкоджена, а коригуюча здатність вбудованих кодів Ріда-Соломона не дозволяє виправити необхідну частину помилок. Отже, застосуємо додаткову процедуру виправлення помилок, використовуючи НМХЛ, базу еталонних QR-кодів та запропонований алгоритм. Експериментальні результати розпізнавання зображень QR-кодів різних розмірів представлені в табл. 1-3. Використовувалась лінійна залежність  $Q = \alpha \cdot K + \beta$  для порогової величини  $Q$  і розміру ковзного вікна  $K$ , для якої експериментально вибрані значення констант  $[\alpha; \beta] = [(50; 0), (200; 0), (400; 100)]$ .



Рисунок 3 – Навчальні зображення

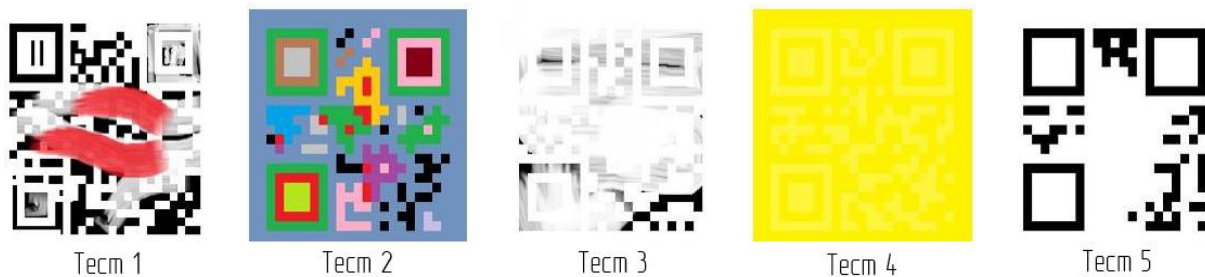


Рисунок 4 – Тестові зображення

Таблиця 1 – Результати експериментів (лінійна залежність  $Q = 50 \cdot K$ , зображення розміром  $200 \times 200$ )

$K$	$Q$	Тест 1	Тест 2	Тест 3	Тест 4	Тест 5	Правильно розпізнані	Результат
1-7	50-350	ВІРНО	ВІРНО	ВІРНО	ВІРНО	ВІРНО	5	ДОБРЕ
<b>8</b>	<b>400</b>	<b>ВІРНО</b>	<b>ВІРНО</b>	<b>ВІРНО</b>	<b>ВІРНО</b>	<b>ВІРНО</b>	<b>5</b>	<b>ДОБРЕ</b>
9	450	ВІРНО	НЕВІРНО	ВІРНО	ВІРНО	НЕВІРНО	4	НЕПОГАНО
10	500	ВІРНО	НЕВІРНО	ВІРНО	ВІРНО	НЕВІРНО	4	НЕПОГАНО
11	550	ВІРНО	НЕВІРНО	НЕВІРНО	НЕВІРНО	НЕВІРНО	1	ПОГАНО
12	600	ВІРНО	ВІРНО	НЕВІРНО	ВІРНО	НЕВІРНО	3	ПОГАНО
13	650	ВІРНО	НЕВІРНО	НЕВІРНО	НЕВІРНО	НЕВІРНО	1	ПОГАНО
14	700	ВІРНО	НЕВІРНО	НЕВІРНО	НЕВІРНО	НЕВІРНО	1	ПОГАНО
15	750	НЕВІРНО	ВІРНО	НЕВІРНО	НЕВІРНО	НЕВІРНО	1	ПОГАНО
16	800	ВІРНО	НЕВІРНО	НЕВІРНО	НЕВІРНО	НЕВІРНО	1	ПОГАНО

Таблиця 2 – Результати експериментів (лінійна залежність  $Q = 200 \cdot K$ , зображення розміром  $512 \times 512$ )

$K$	$Q$	Тест 1	Тест 2	Тест 3	Тест 4	Тест 5	Кількість правильно розпізнаних	Результат
9-15	1800-3000	ВІРНО	ВІРНО	ВІРНО	ВІРНО	ВІРНО	5	ДОБРЕ
<b>16</b>	<b>3200</b>	<b>ВІРНО</b>	<b>ВІРНО</b>	<b>ВІРНО</b>	<b>ВІРНО</b>	<b>ВІРНО</b>	<b>5</b>	<b>ДОБРЕ</b>
17	3400	ВІРНО	НЕВІРНО	ВІРНО	ВІРНО	ВІРНО	4	НЕПОГАНО
18	3600	ВІРНО	НЕВІРНО	ВІРНО	ВІРНО	ВІРНО	4	НЕПОГАНО
19	3800	НЕВІРНО	ВІРНО	ВІРНО	ВІРНО	ВІРНО	4	НЕПОГАНО
20	4000	ВІРНО	НЕВІРНО	ВІРНО	ВІРНО	ВІРНО	4	НЕПОГАНО
21	4200	ВІРНО	НЕВІРНО	НЕВІРНО	ВІРНО	ВІРНО	3	ПОГАНО
22	4400	ВІРНО	НЕВІРНО	ВІРНО	ВІРНО	ВІРНО	4	НЕПОГАНО
23	4600	ВІРНО	НЕВІРНО	ВІРНО	ВІРНО	ВІРНО	4	НЕПОГАНО
24	4800	ВІРНО	ВІРНО	НЕВІРНО	ВІРНО	ВІРНО	4	НЕПОГАНО
25	5000	ВІРНО	НЕВІРНО	ВІРНО	НЕВІРНО	ВІРНО	3	ПОГАНО
26	5200	ВІРНО	НЕВІРНО	ВІРНО	ВІРНО	ВІРНО	4	НЕПОГАНО
27	5400	НЕВІРНО	ВІРНО	ВІРНО	НЕВІРНО	НЕВІРНО	2	ПОГАНО
28	5600	НЕВІРНО	ВІРНО	ВІРНО	НЕВІРНО	НЕВІРНО	2	ПОГАНО

Таблиця 3 – Результати експериментів (залежність  $Q = 400 \cdot K + 100$ , зображення розміром  $1024 \times 1024$ )

$K$	$Q$	Тест 1	Тест 2	Тест 3	Тест 4	Тест 5	Кількість правильно розпізнаних	Результат
1-25	500-10100	ВІРНО	ВІРНО	ВІРНО	ВІРНО	ВІРНО	5	ДОБРЕ
<b>26</b>	<b>10500</b>	<b>ВІРНО</b>	<b>ВІРНО</b>	<b>ВІРНО</b>	<b>ВІРНО</b>	<b>ВІРНО</b>	<b>5</b>	<b>ДОБРЕ</b>
27	10900	ВІРНО	ВІРНО	ВІРНО	ВІРНО	НЕВІРНО	4	НЕПОГАНО
28	11300	ВІРНО	ВІРНО	ВІРНО	ВІРНО	НЕВІРНО	4	НЕПОГАНО

На основі даних з табл. 1-3 можна зробити висновок, що оптимальними значеннями параметрів  $K$  і  $Q$  для тестових зображень з розмірами  $H$  на  $B \in [H \times B; K; Q] = [(200 \times 200; 8; 400), (512 \times 512; 16; 3200), (1024 \times 1024; 26; 10500)]$ . Використовуючи ці значення, НМХЛ правильно розпізнає всі тестові QR-коди, але для досягнення меншої обчислювальної складності навчання і більшої точності розпізнавання може бути виконана процедура налаштування параметра  $K_t$ , коли  $(n-1)$  тестових зображень були розпізнані, наприклад,  $[H \times B; \{K_t\}] = [(200 \times 200; \{9; 10\}), (512 \times 512; \{17-20; 22-24; 26\}), (1024 \times 1024; \{27, 28\})]$ .

У табл. 4 демонструється той факт, що, коли порогова величина  $Q$  менша за значення розміру ковзного вікна  $K$  або дорівнює йому, то НМХЛ не може правильно виконати розпізнавання зображень.

Таблиця 4 – Результати експериментів (зображення розміром  $200 \times 200$ ,  $K \geq Q$ )

$K$	$Q$	Тест 1	Тест 2	Тест 3	Тест 4	Тест 5	Кількість правильно розпізнаних	Результат
1	1	ВІРНО	ВІРНО	НЕВІРНО	НЕВІРНО	НЕВІРНО	2	ПОГАНО
4	2	ВІРНО	ВІРНО	НЕВІРНО	НЕВІРНО	НЕВІРНО	2	ПОГАНО
4	4	ВІРНО	ВІРНО	НЕВІРНО	НЕВІРНО	НЕВІРНО	2	ПОГАНО
9	4	ВІРНО	НЕВІРНО	НЕВІРНО	НЕВІРНО	НЕВІРНО	1	ПОГАНО
9	9	ВІРНО	НЕВІРНО	НЕВІРНО	НЕВІРНО	НЕВІРНО	1	ПОГАНО
16	8	ВІРНО	ВІРНО	НЕВІРНО	НЕВІРНО	НЕВІРНО	2	ПОГАНО
16	16	ВІРНО	НЕВІРНО	НЕВІРНО	НЕВІРНО	НЕВІРНО	1	ПОГАНО
25	12	ВІРНО	НЕВІРНО	НЕВІРНО	НЕВІРНО	НЕВІРНО	1	ПОГАНО
25	25	ВІРНО	ВІРНО	НЕВІРНО	НЕВІРНО	НЕВІРНО	2	ПОГАНО
36	18	ВІРНО	НЕВІРНО	НЕВІРНО	НЕВІРНО	НЕВІРНО	1	ПОГАНО
36	36	ВІРНО	НЕВІРНО	НЕВІРНО	НЕВІРНО	НЕВІРНО	1	ПОГАНО
49	24	ВІРНО	НЕВІРНО	НЕВІРНО	НЕВІРНО	НЕВІРНО	1	ПОГАНО
64	32	ВІРНО	НЕВІРНО	НЕВІРНО	НЕВІРНО	НЕВІРНО	1	ПОГАНО
81	40	ВІРНО	НЕВІРНО	НЕВІРНО	НЕВІРНО	НЕВІРНО	1	ПОГАНО
100	50	ВІРНО	НЕВІРНО	НЕВІРНО	НЕВІРНО	НЕВІРНО	1	ПОГАНО

У табл. 5 показано приклад процедури налаштування, заснованої на інтервалах із табл. 1, для розміру ковзного вікна  $K = 9$  (розмір зображення  $200 \times 200$ ), яке більше за оптимальне значення, при якому всі QR-коди успішно розпізнаються.

Результати даних експериментів показали ефективність запропонованої процедури корекції, тобто її можна використовувати у системах обміну даними для виправлення сильно пошкоджених QR-кодів.

Таблиця 5 – Результати процедури налаштування в програмному забезпеченні

$K_i$	$Q$	Тест 1	Тест 2	Тест 3	Тест 4	Тест 5	Кількість правильно розпізнаних	Результат	Оптимальне значення
9	400	ВІРНО	ВІРНО	ВІРНО	НЕВІРНО	НЕВІРНО	3	ПОГАНО	$K = K_i - 1 = 8$
	455	ВІРНО	НЕВІРНО	ВІРНО	ВІРНО	НЕВІРНО	4	ПОГАНО	
	460	ВІРНО	ВІРНО	ВІРНО	ВІРНО	НЕВІРНО	4	ПОГАНО	
	465	ВІРНО	НЕВІРНО	ВІРНО	ВІРНО	НЕВІРНО	3	ПОГАНО	
	470	ВІРНО	ВІРНО	ВІРНО	ВІРНО	НЕВІРНО	4	ПОГАНО	
	475	НЕВІРНО	ВІРНО	ВІРНО	ВІРНО	НЕВІРНО	3	ПОГАНО	
	480	ВІРНО	ВІРНО	ВІРНО	ВІРНО	НЕВІРНО	4	ПОГАНО	
	485	ВІРНО	ВІРНО	НЕВІРНО	ВІРНО	НЕВІРНО	3	ПОГАНО	
	490	ВІРНО	ВІРНО	НЕВІРНО	ВІРНО	НЕВІРНО	3	ПОГАНО	
	495	ВІРНО	ВІРНО	НЕВІРНО	ВІРНО	НЕВІРНО	3	ПОГАНО	
	500	ВІРНО	ВІРНО	ВІРНО	ВІРНО	НЕВІРНО	4	ПОГАНО	
510	ВІРНО	ВІРНО	НЕВІРНО	ВІРНО	НЕВІРНО	3	ПОГАНО		

#### Висновки

У статті запропоновано алгоритм розв'язання задачі розпізнавання та коригування сильно пошкоджених QR-кодів, який базується на використанні НМХЛ з базою еталонних зразків та обробці зображень у режимі ковзного вікна, що не потребує використання трудомістких обчислювальних операцій, великих обсягів пам'яті та значних витрат часу. У ході експериментальних досліджень з використанням розробленого програмного забезпечення визначено набір оптимальних параметрів алгоритму (розмір ковзного вікна  $K$  та порогового значення  $Q$ ) для різних розмірів зображень QR-кодів. Розроблений алгоритм можна використовувати як додаткову процедуру виправлення QR-кодів у системах обміну даними тоді, коли скануючому обладнанню не вдається зчитати та розпізнати QR-код звичайним чином.

#### Список літератури

- [1] QR Code. [Electronic resource]. Available: <https://www.comscore.com/Insights/Infographics/20-Million-Americans-Scanned-a-QR-Code-in-October>. Accessed: 01-Oct-2019.
- [2] ISO/IEC 18004:2015. Estonian Centre for Standardization. [Electronic resource]. Available: <https://www.evs.ee/products/iso-iec-18004-2015>. Accessed: 01-Oct-2019.
- [3] T.K. Moon, *Error Correction Coding: Mathematical Methods & Algorithms*, John Wiley & Sons, 2005.
- [4] R.H. Morelos-Zaragoza, *The Art of Error Correcting Coding*. Chichester: Wiley, 2006.
- [5] G. Barland, Error Correction and QR Codes. [Electronic resource]. Available: <https://www.stthomas.edu/media/collegeofartsandsciences/mathematics/pdf/2017AndersonGroupBarlandCAMReport.pdf>. Accessed: 01-Oct-2019.
- [6] QR Code Tutorial from Thonky. [Electronic resource]. Available: <https://www.thonky.com/qr-code-tutorial/>. Accessed: 01-Oct-2019.
- [7] I. Szentandrási, A. Herout, M. Dubska, "Fast Detection and Recognition of QR codes in High-Resolution Images", *Published in SCCG 2012*, pp. 129-136, 2012.
- [8] K. Suran, "QR Code Image Correction based on Corner Detection and Convex Hull Algorithm", *Journal of Multimedia*, vol. 8, no. 6, pp. 662-668, 2013.
- [9] R.P. Lippmann, B. Gold, M.L. Malpass, *A Comparison of Hamming and Hopfield Neural Nets for Pattern Classification*, Ft. Belvoir: Defense Technical Information Center, pp. 1-35, 1987.
- [10] K. Koutroumbas, N. Kalouptsidis, "Generalized Hamming Networks and Applications", *Neural Networks*, vol. 18, pp. 896-913, 2005.
- [11] А.А. Ежов, А.С. Новиков, "Сеть Хэмминга и ее применение для решения задачи распознавания подписей", *Известия ТулГУ*, №11, С. 56-64, 2016.
- [12] R.C. Gonzalez, R.E. Woods, *Digital Image Processing*. Prentice Hall, 2008.
- [13] Ю.Ю. Іванов, *Методи штучного інтелекту та наука про дані: лекції, лабораторні роботи, алгоритми та задачі*, Вінниця, Україна: ВНТУ, 2018.
- [14] L.V. Fausett, *Fundamentals of Neural Networks: Architectures, Algorithms, and Applications*, Delhi: Pearson Education, 2004.

## References

- [1] QR Code. [Electronic resource]. Available: <https://www.comscore.com/Insights/Infographics/20-Million-Americans-Scanned-a-QR-Code-in-October>. Accessed: 01-Oct-2019.
  - [2] ISO/IEC 18004:2015. Estonian Centre for Standardization. [Electronic resource]. Available: <https://www.evs.ee/products/iso-iec-18004-2015>. Accessed: 01-Oct-2019.
  - [3] T.K. Moon, *Error Correction Coding: Mathematical Methods & Algorithms*, John Wiley & Sons, 2005.
  - [4] R.H. Morelos-Zaragoza, *The Art of Error Correcting Coding*. Chichester: Wiley, 2006.
  - [5] G. Barland, Error Correction and QR Codes. [Electronic resource]. Available: <https://www.stthomas.edu/media/collegeofartsandsciences/mathematics/pdf/2017AndersonGroupBarlandCAMReport.pdf>. Accessed: 01-Oct-2019.
  - [6] QR Code Tutorial from Thonky. [Electronic resource]. Available: <https://www.thonky.com/qr-code-tutorial/>. Accessed: 01-Oct-2019.
  - [7] I. Szentandrasei, A. Herout, M. Dubska, "Fast Detection and Recognition of QR codes in High-Resolution Images", *Published in SCCG 2012*, pp. 129-136, 2012.
  - [8] K. Suran, "QR Code Image Correction based on Corner Detection and Convex Hull Algorithm", *Journal of Multimedia*, vol. 8, no. 6, pp. 662-668, 2013.
  - [9] R.P. Lippmann, B. Gold, M.L. Malpass, *A Comparison of Hamming and Hopfield Neural Nets for Pattern Classification*, Ft. Belvoir: Defense Technical Information Center, pp. 1-35, 1987.
  - [10] K. Koutroumbas, N. Kalouptsidis, "Generalized Hamming Networks and Applications", *Neural Networks*, vol. 18, pp. 896-913, 2005.
  - [11] A.A. Ezhov, A.S. Novykov, "Set Khemmynya y ee pryomenenye dlia resheniya zadachy raspoznavaniya podpysei", *Yzvestyia TulHU*, №11, S. 56-64, 2016.
  - [12] R.C. Gonzalez, R.E. Woods, *Digital Image Processing*. Prentice Hall, 2008.
  - [13] Iu.Iu. Ivanov, *Metody shtuchnogo intelektu ta nauka pro dani: leksii, laboratorni roboty, alhorytmy ta zadachi*, Vinnytsia, Ukraina: VNTU, 2018.
  - [14] L.V. Fausett, *Fundamentals of Neural Networks: Architectures, Algorithms, and Applications*, Delhi: Pearson Education, 2004.
- Стаття надійшла: 20.09.2019.

## Відомості про авторів

**Кветний Роман Наумович** – д. т. н., професор, завідувач кафедрою автоматизації та інтелектуальних інформаційних технологій ВНТУ, член-кореспондент Національної академії педагогічних наук України, заслужений діяч науки і техніки, тел.: (0432) 59-82-43.

**Іванов Юрій Юрійович** – к. т. н., доцент кафедри автоматизації та інтелектуальних інформаційних технологій.

**Пивошенко Володимир Володимирович** – студент кафедри автоматизації та інтелектуальних інформаційних технологій.

**Васюра Анатолій Степанович** – к. т. н., професор кафедри автоматизації та інтелектуальних інформаційних технологій, академік Української технологічної академії.

R. N. Kvetnyy, V. V. Pivoshenko, Yu. Yu. Ivanov, A. S. Vasyura

## АЛГОРИТМ РАСПОЗНАВАНИЯ И КОРРЕКЦИИ ПОВРЕЖДЕННЫХ QR-КОДОВ

Винницкий национальный технический университет

R. N. Kvyetnyy, V. V. Pivoshenko, Yu. Yu. Ivanov, A. S. Vasyura

## ALGORITHM FOR RECOGNITION HIGHLY CORRUPTED QR-CODES

Vinnitsa National Technical University