Protecting machine-learning datasets with digital watermarking based on modified QR decomposition*

Oles Hospodarskyy[1,*,†], Natalia Kukharska[1,†], Orest Polotai[2,†], Oleh Harasymchuk[1,3,†], and Ivan Tyshyk[1,†]

[1] *Lviv Polytechnic National University, 12 Stepan Bandera str., 79000 Lviv, Ukraine*
[2] *Lviv State University of Life Safety, 35 Kleparivska str., 79000 Lviv, Ukraine*
[3] *Ternopil Ivan Puluj National Technical University, 56 Ruska str, 46001 Ternopil, Ukraine*

Abstract

The paper examines an approach to protecting intellectual property rights for machine learning image datasets using digital watermarking based on QR decomposition. The proposed scheme involves embedding hidden identification information into the image structure through controlled modification of the elements of the upper triangular matrix obtained from QR decomposition, ensuring a combination of invisibility, robustness, and computational efficiency. To ensure numerical stability and efficiency in the watermark embedding and extraction processes, the method employs an improved QR decomposition algorithm with diagonal element correction, as well as a fast watermark extraction technique with linear computational complexity. The effectiveness of the approach is evaluated using PSNR, SSIM, and normalized correlation metrics, as well as its resistance to common image processing attacks. Particular attention is paid to analyzing the impact of digital watermarking on the quality of training data for computer vision models used in object classification and detection tasks. Experimental results demonstrate that watermark incorporation does not compromise visual data quality or reduce machine learning model accuracy, thereby confirming the feasibility of using QR decomposition for practical dataset protection.

Keywords

information security, intellectual property protection, machine learning datasets, digital watermarking, QR decomposition–based watermarking, AdamW optimizer, robustness, imperceptibility, computational complexity.

1. Introduction

Machine learning is one of the key directions in the development of modern data analysis technologies, enabling the effective solution of a wide range of tasks — from classification and prediction to pattern recognition and process optimization [1–6]. The performance of machine learning models largely depends on the quality, representativeness, and sufficient size of the training dataset. Complete, reliable, and balanced data allow algorithms to more accurately capture real-world relationships, avoid overfitting or underfitting, and improve the generalization ability of models. Conversely, errors, missing values, or biases in the data can significantly distort prediction results and reduce the accuracy and reliability of the system.

Creating high-quality and representative datasets is a resource-intensive process that requires significant time, financial, and expert investments. At the same time, with the growth of the data market [7] and the active spread of open sources, the risk of their unauthorized use, copying, or (re)sale without the owner's consent is growing. As datasets acquire independent economic value, they become an important object of intellectual property that requires reliable protection.

* *CPITS 2026: Cybersecurity Providing in Information and Telecommunication Systems, February 28, 2026, Kyiv, Ukraine*
* Corresponding author.
† These authors contributed equally.
oles.hospodarskyi.mkbbi.2025@lpnu.ua (O. Hospodarskyy); nataliia.p.kukharska@lpnu.ua (N. Kukharska), orest.polotaj@gmail.com (O. Polotai); oleh.i.harasymchuk@lpnu.ua (O. Harasymchuk); ivan.y.tyshyk@lpnu.ua (I. Tyshyk)
0009-0005-9088-3015 (O. Hospodarskyy); 0000-0002-0896-8361 (N. Kukharska); 0000-0003-4593-8601 (O. Polotai); 0000-0002-8742-8872 (O. Harasymchuk); 0000-0003-1465-5342 (I. Tyshyk)

Legal protection mechanisms, such as GDPR [8] and CCPA [9], SOC 2 [10], while regulating data privacy issues, do not solve the problem of protecting intellectual property rights to the datasets themselves. These regulations focus primarily on protecting the personal data of subjects, not on the rights of dataset owners as intellectual property objects. As a result, a legal gap arises, necessitating the supplementation of legal instruments with technical protection measures.

Related research in the field of secure cloud services and information governance shows that compliance- and policy-oriented frameworks must be complemented by technical mechanisms that ensure data traceability, control, and accountability throughout the entire data lifecycle [11, 12]. This highlights the need for protection approaches that operate directly at the data level rather than relying exclusively on external regulatory or organizational controls.

In this context, one of the traditional technical approaches widely used to protect digital content is encryption. However, its application to machine learning datasets has significant limitations. Although encryption effectively ensures data confidentiality during storage and transmission, it does not provide protection against unauthorized use after decryption. Moreover, encrypted data are unsuitable for direct use in machine learning model training [13], as they require prior decryption.

The mentioned limitations of legal and cryptographic approaches necessitate the search for alternative protection methods capable of exercising control over the use of datasets even after granting access to them. Such methods must ensure identification of the data source, confirmation of ownership rights, and detection of facts of unauthorized use without disrupting the possibility of further application of data in machine learning processes.

One approach that meets these requirements is the use of digital watermarking. Unlike cryptographic methods, digital watermarking does not restrict access to the data but embeds hidden identification features that remain detectable even after the data are used in an open form [14]. This makes it possible to verify ownership of datasets, trace their origin, and detect unauthorized use without negatively affecting the training of machine learning models [15].

A review of modern developments in the field of digital watermarks shows that existing methods can be classified according to several criteria. In particular, methods are distinguished by their embedding domain: spatial domain methods and frequency domain methods.

Spatial domain methods are based on direct modification of components of a digital object. In the case of images, this means direct changes to pixel values [16, 17]. The simplest and at the same time one of the most common examples is the Least Significant Bit (LSB) replacement method. This method also includes the Intermediate Significant Bit (ISB) method [18] and the Patchwork method [19]. Although these methods are simple to implement and characterized by low computational complexity $O(n)$, they generally have insufficient robustness to attacks such as compression or filtering, since even minor image modifications can lead to the loss of embedded information.

Frequency domain methods, unlike spatial methods, embed watermark information in frequency coefficients or components of matrix decompositions, rather than directly in image pixels. Such methods involve:

- Discrete Cosine Transform (DCT) [20];
- Discrete Wavelet Transform (DWT) [21];
- Matrix decompositions (SVD [22, 23], QR [24], LU [28]).

The main advantage of these methods is their high robustness to attacks, as modifications in the frequency domain are generally less sensitive to operations such as compression, rotation, or cropping. However, they usually require higher computational resources compared to spatial-domain methods.

Particular attention should be given to digital watermarking methods based on matrix decompositions. In work [21], the authors investigate the possibility of applying digital watermarking methods, originally created for images, to machine learning datasets to confirm ownership rights. For this purpose, they chose a watermarking scheme based on Singular Value Decomposition (SVD) [22]. The obtained results showed:

- watermark embedding does not lead to significant distortions of the original datasets;

- machine learning models trained on marked data maintain a high level of accuracy;
- the proposed scheme is robust to update, insertion, and data nullification attacks.

Despite the high numerical stability and robustness of SVD-based digital watermarking methods, their computational complexity of $O(n^3)$ limits their applicability in large-scale and time-critical scenarios. In contrast, the QR decomposition method [24] requires fewer computational resources, making it a promising alternative for constructing efficient digital watermarking schemes.

In this regard, the aim of this work is to investigate an approach to protecting intellectual property rights to machine learning datasets based on a modified QR decomposition method for embedding digital watermarks in images, as well as to evaluate the impact of its application on the quality of training computer vision models [25–27].

2. Materials and Methods

Within this study, image datasets intended for training computer vision models were used. The images were represented as matrices of pixel values, which enables the application of linear algebra methods for embedding digital watermarks. The datasets were preprocessed through normalization and converted to a unified format in order to ensure a fair and consistent comparison of experimental results.

To evaluate the impact of digital watermarking on training quality, standard computer vision model architectures were used, as well as commonly accepted classification accuracy metrics. All experiments were conducted under identical training conditions for both original and marked datasets.

2.1. Digital Watermark Embedding Method

The proposed approach to protecting machine learning datasets is based on a modified QR decomposition method [29]. For each image in the training dataset, a matrix of pixel values A is constructed and subjected to QR decomposition:

$$A = QR, \tag{1}$$

where Q is an orthogonal matrix and R is an upper triangular matrix.

The digital watermark is embedded through controlled modification of selected elements of the matrix R, which preserves the structural properties of the original image while minimizing visual distortion. After watermark embedding, a modified matrix R', is obtained, and the watermarked image is reconstructed as:

$$A' = Q R'. \tag{2}$$

This embedding method ensures watermark imperceptibility and robustness to typical image processing attacks.

2.2. Watermark Extraction Method

The digital watermark extraction procedure is based on repeated application of QR decomposition to the marked image. Analysis of modified elements of matrix R allows recovering the embedded identification information and confirming ownership rights to the dataset. The proposed method does not require access to the original image, which makes it suitable for practical use in real scenarios.

2.3. Evaluation of Method Effectiveness

The effectiveness of the proposed approach was evaluated according to the following criteria:

- the level of image distortions after watermark embedding;

- watermark robustness to typical attacks, including scaling, filtering, and partial data modification;
- the impact of marking on the training accuracy of computer vision models.

The obtained results were compared with results for unmarked datasets, which allowed evaluating the feasibility of using QR decomposition for protecting the intellectual property of machine learning datasets.

3. Results

A digital watermark is a technology that involves embedding identification information directly into a digital object in such a way that it is imperceptible to human perception but can be detected using specialized algorithms to confirm authenticity or establish data origin. Watermarks can contain information about the author, creation date, usage conditions, or other metadata necessary for protecting intellectual property rights.

Regarding machine learning datasets, digital watermarks must perform a dual function: ensure protection of intellectual property rights to data while simultaneously preserving the quality and effectiveness of model training.

3.1. Criteria for Evaluating Digital Watermark Effectiveness

Imperceptibility (or transparency) means that the watermark should not noticeably reduce the quality of marked data or interfere with their intended use. Imperceptibility is evaluated through visual perception or objective metrics such as Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index (SSIM):

$$PSNR = 10\log_{10}\left(\frac{MAX^2}{MSE}\right), \tag{3}$$

where MAX is the maximum possible pixel value, MSE is the mean squared error;

$$SSIM(x,y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)}, \tag{4}$$

where μ_x, μ_y are mean values, σ_x^2, σ_y^2 are variances, σ_{xy} is covariance.

For machine learning data, this also means that the watermark should not significantly affect the accuracy of models trained on marked data.

Robustness indicates the watermark's ability to withstand various attacks or manipulations. A watermark is considered robust if it can be recognized even after operations such as compression, filtering, noise addition, cropping, geometric transformations (e.g., scaling or rotation), or their combinations. Robustness is typically evaluated using metrics such as Normalized Correlation (NC) between the original and extracted watermarks.

$$NC = \frac{\sum_{ij} W(i,j) \times W'(i,j)}{\sqrt{\sum_{ij} W(i,j)^2} \times \sqrt{\sum_{ij} W'(i,j)^2}}, \tag{5}$$

where W is the original watermark, W' is the extracted watermark. A higher NC value (close to 1) indicates greater robustness.

Computational complexity is determined by the amount of resources (time, memory, computational power) required for watermark embedding and extraction. A method with lower

computational complexity is typically more practical, especially for large datasets or real-time applications. For example, SVD decomposition requires approximately $11n^3$ floating-point operations for an $n \times n$ matrix, while Schur decomposition requires approximately $\frac{8n^3}{3}$, and QR decomposition requires even less, making it more efficient in terms of computational resources.

It is important to note that there is a certain trade-off between these criteria. For example, increasing watermark robustness typically leads to decreased imperceptibility, as more significant changes to the original data are introduced to achieve greater robustness. Similarly, methods with higher robustness often require greater computational complexity. Therefore, when developing a watermarking system, it is important to balance these criteria according to the requirements of the specific application.

3.2. QR Decomposition for Dataset Protection

Machine learning datasets differ significantly from traditional media files, which creates specific requirements for their protection. The main difference is due to the fact that ML datasets are used for training algorithms. Thus, any changes in the data can potentially affect the accuracy and effectiveness of trained models. Therefore, watermarking technologies must introduce minimal distortions to preserve the usefulness of data for machine learning tasks.

Moreover, machine learning datasets are often multidimensional and structured, which complicates the application of traditional digital watermarking methods developed for images or audio signals. For example, datasets for image classification tasks may contain thousands or millions of samples with diverse characteristics. Under such conditions, watermarks must be evenly distributed throughout the entire dataset to maintain robustness even in case of deletion or modification of its individual parts.

Another feature of machine learning datasets is that they are often represented in matrix form, where rows correspond to individual samples and columns to features. This creates a natural opportunity for applying matrix decomposition methods for watermark embedding.

Among known matrix decomposition methods, QR decomposition is of particular interest. It consists of representing matrix A as the product of an orthogonal matrix Q and an upper triangular matrix R, i.e., $A = QR$. The properties of orthogonality and structure of this decomposition make it suitable for constructing digital watermarking schemes.

At the same time, standard QR decomposition algorithms, particularly the Gram-Schmidt algorithm [30], have certain limitations. Such algorithms compute matrices Q and R element by element and do not check whether the diagonal elements of R are zero or negative. This can lead to numerical instability and complicate correct watermark recovery, as zero or negative values on the diagonal can cause infinite or incorrect results in calculations.

To understand the nature of this problem, let us consider the process of computing diagonal elements when using the classical Gram-Schmidt algorithm for a 4×4 matrix A. The matrix can be represented as a collection of columns $A = [a_1 \ a_2 \ a_3 \ a_4]$, where each column a_i is a vector of dimension 4:

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix} = [a_1 \ a_2 \ a_3 \ a_4].$$

In the orthogonalization process, the diagonal elements of matrix R are computed sequentially, with each subsequent element depending on previous calculations.

The first diagonal element is computed as $r_{11} = \|a_1\|$, which always gives a positive value for a non-zero vector. However, the second diagonal element r_{22} is computed as the norm of a vector

obtained after subtracting the projection of a_2 onto q_1, which can result in a zero-length vector in case of linear dependence of columns. Similarly, the computation of r_{33} and r_{44}can also become problematic in the presence of linear dependencies or near-linear dependencies between matrix columns.

In cases where the columns of matrix A are linearly dependent or nearly linearly dependent, the value of r_{ii} can become very small or equal to zero. This leads to instability not only at the current computation step but also propagates to all subsequent operations, as division by such values can lead to overflow or loss of precision.

3.3. Improved QR Decomposition Algorithm with Diagonal Element Correction

To solve this fundamental problem, work [31] developed an approach based on using the properties of QR decomposition and allowing computation of elements of matrix R independently of matrix Q. The proposed approach uses a matrix equation that follows from the basic QR decomposition relationship.

If $A=QR$, then by multiplying both sides of the equation on the left by A^T and considering that Q is an orthogonal matrix ($Q^T Q=I$), we obtain:

$$A^T A=A^T QR=M$$
$$A^T A=(QR)^T QR=M$$
$$A^T A=R^T Q^T QR=M$$
$$A^T A=R^T R=M$$

This fundamental relationship allows computing matrix R by solving a system of linear equations, without relying on sequential computation of matrix Q and avoiding the instability problems of the traditional approach.

Practical implementation of this approach begins with computing matrix M:

$$M=A^T A.$$

The elements of matrix

$$R=\begin{bmatrix} r_{11} & r_{12} & r_{13} & r_{14} \\ 0 & r_{22} & r_{23} & r_{24} \\ 0 & 0 & a_{33} & r_{34} \\ 0 & 0 & 0 & r_{44} \end{bmatrix}$$

are found sequentially, starting with the diagonal elements.

The first diagonal element is computed directly:

$$r_{11}=\sqrt{m_{11}}. \tag{6}$$

For subsequent diagonal elements, recurrent relationships are used:

$$r_{22}=\sqrt{m_{22}-r_{12}^2}$$

$$r_{33}=\sqrt{m_{33}-r_{13}^2-r_{23}^2} \tag{7}$$

$$r_{44}=\sqrt{m_{44}-r_{14}^2-r_{24}^2-r_{34}^2}$$

Off-diagonal elements are computed using the formulas:

$$\begin{aligned} r_{1j} &= \frac{m_{1j}}{r_{11}}, \quad j=2,3,4 \\ r_{2j} &= \frac{m_{2j} - r_{12} r_{1j}}{r_{22}}, \quad j=3,4 \\ r_{3j} &= \frac{m_{3j} - r_{13} r_{1j} - r_{23} r_{2j}}{r_{33}}, \quad j=4 \end{aligned} \tag{8}$$

The innovation of this method is ensuring control and correction of diagonal elements. Before computing each diagonal element, the value under the square root is checked. If this value turns out to be zero or negative, indicating linear dependence or numerical instability, the diagonal element is set equal to one:

$$r_{ii} = \begin{cases} 1, & m_{ii} - \sum_{k=1}^{i-1} r_{ki}^2 \le 0 \\ \sqrt{m_{ii} - \sum_{k=1}^{i-1} r_{ki}^2}, & elsewhere \end{cases} \tag{9}$$

And corresponding off-diagonal elements:

$$r_{ij} = m_{ij} - \frac{\sum_{k=1}^{i-1} r_{ki} r_{kj}}{r_{ii}}, \quad j = i+1, \ldots n. \tag{10}$$

The advantages of the proposed modified approach manifest in several aspects [29]. Stability is ensured by guaranteed absence of zero diagonal elements and control at each computation step. Algorithm controllability allows detecting and correcting problematic situations before they affect the result quality. Efficiency is achieved through fewer iterative computations compared to the traditional Gram-Schmidt algorithm, and improved quality of watermarked images is confirmed by significant growth in PSNR and SSIM indicators.

3.4. Optimization of Q Matrix Computation

After obtaining a stable R matrix using the modified algorithm, the next step is efficient computation of the orthogonal matrix Q. The traditional approach involves simultaneous computation of Q and R matrices during Gram-Schmidt orthogonalization. However, having an already computed and stabilized R matrix, a more efficient approach can be used, based on the fundamental property $A = QR$.

The main idea of the optimized algorithm is to use the already known values of R matrix for direct computation of Q matrix columns [29]. This approach not only simplifies computations but also guarantees correspondence between Q and R matrices, since the R matrix has already undergone the stabilization procedure.

The process of computing matrix Q begins with the first column. According to the QR decomposition property, the first column of matrix Q can be computed as the normalized first column of the original matrix A. For this, an auxiliary vector is first computed, which equals the first column of matrix A:

$$\widetilde{q_1} = a_1.$$

Then normalization is performed using the corresponding diagonal element of R matrix:$q_1 = \frac{\widetilde{q}_1}{r_{11}}$.

For subsequent columns of matrix Q, the procedure is complicated by the need to account for orthogonality to previously computed columns. Each i-th column of matrix Q is computed by subtracting from the corresponding column of matrix A the projections onto all previous columns of matrix Q:

$$\widetilde{q}_i = a_i - \sum_{k=1}^{i-1} r_{ki} q_k, \tag{11}$$

where a_i represents the i-th column of the original matrix A, and the sum accounts for all projections onto previously computed orthogonal vectors. After computing the auxiliary vector, its normalization is performed:

$$q_i = \frac{\widetilde{q}_i}{r_{ii}}. \tag{12}$$

For the specific case of a 4×4 matrix, the algorithm can be detailed as follows.
The first column is computed directly from the first column of the original matrix:

$$\begin{aligned} \widetilde{q}_1 &= a_1 \\ q_1 &= \frac{\widetilde{q}_1}{r_{11}} \end{aligned} \tag{13}$$

The second column accounts for orthogonality to the first:

$$\begin{aligned} \widetilde{q}_2 &= a_2 - r_{12} q_1 \\ q_2 &= \frac{\widetilde{q}_2}{r_{22}} \end{aligned} \tag{14}$$

The third column is orthogonal to the two previous ones:

$$\begin{aligned} \widetilde{q}_3 &= a_3 - r_{13} q_1 - r_{23} q_2 \\ q_3 &= \frac{\widetilde{q}_3}{r_{33}} \end{aligned} \tag{15}$$

Finally, the fourth column is orthogonal to all previous ones:

$$\begin{aligned} \widetilde{q}_4 &= a_4 - r_{14} q_1 - r_{24} q_2 - r_{34} q_3 \\ q_4 &= \frac{\widetilde{q}_4}{r_{44}} \end{aligned} \tag{16}$$

The advantages of the optimized approach to computing matrix Q manifest in several aspects [29]. Controllability is ensured by the fact that matrix R has already undergone stabilization, which

guarantees stability of matrix Q computations. Efficiency is achieved through fewer operations compared to simultaneous computation of Q and R. Accuracy is improved thanks to using a stable R matrix, which reduces accumulation of rounding errors. Finally, the modularity of the approach allows easy implementation of additional optimizations and adaptation of the algorithm for different processor architectures.

3.5. Watermark embedding scheme

During the embedding stage, the host color image is first partitioned into non-overlapping blocks of size 4×4. The grayscale watermark image subsequently converted into a binary sequence. After that, an improved QR decomposition is applied sequentially to each image block, and the watermark information is embedded into the elements of the R matrix. The overall watermark embedding procedure is described as follows [29]:

1. Partition the host color image H into non-overlapping 4×4 blocks. Each pixel in the image is represented by three color channels (R, G, and B).
2. Convert the grayscale watermark image into a one-dimensional sequence ω_i, where $i=1,2,\ldots,N\times N$, $N\times N$ denotes the size of the watermark image.
3. Apply QR decomposition to each image block in accordance with the method described in Section 3.3-3.4:
    - Extract the B channel of the current block and arrange it into a 4×4 matrix A.
    - Compute the transpose of matrix A, denoted as A^T.
    - Calculate matrix $M=A^TA$.
    - Determine the matrices R and Q by solving the system of equations defined in Eqs. (6)-(8) and (13)-(16).
4. Embed the watermark bit into the upper triangular matrix R following Sun's embedding rule [23]:
    - Select the first diagonal element r_{11}.
    - Compute $z=r_{11} mod\ q$, where q is a positive integer controlling the embedding strength.
    - Modify r_{11} according to the watermark bit ω_i:

        – case "$\omega_i=0$":

$$r'_{11}=\begin{cases} r_{11}+\frac{q}{4}-z, & z\le 3\frac{q}{4} \\ r_{11}+5\frac{q}{4}-z, & elsewhere \end{cases} \tag{17}$$

        – case "$\omega_i=1$":

$$r'_{11}=\begin{cases} r_{11}-\frac{q}{4}-z, & z\le \frac{q}{4} \\ r_{11}+3\frac{q}{4}-z, & elsewhere \end{cases} \tag{18}$$

5. Update the matrix A' using Eq. (2) and assign the modified matrix back to the B channel of the corresponding image block.
6. Repeat Steps 3–5 for all blocks until the entire watermark sequence has been embedded. Finally, reconstruct the modified B channel together with the original R and G channels to obtain the watermarked image H'.

The advantages of the proposed modified approach are reflected in several aspects. Stability is ensured by guaranteed absence of zero diagonal elements and control at each computation step. Algorithm controllability allows detecting and correcting problematic situations before they affect the result quality. Efficiency is achieved through fewer iterative computations compared to the traditional Gram-Schmidt algorithm, and improved quality of watermarked images is confirmed by significant growth in PSNR and SSIM indicators [29].

3.6. Fast Watermark Extraction Algorithm

One of the main disadvantages of traditional QR-based digital watermark methods [23] is the need to perform full QR decomposition for extracting the watermark from each image block. This requirement significantly increases the computational complexity of the extraction process, makes it slow and resource-intensive, which limits the practical application of such methods in real-time systems.

Analysis of the traditional approach shows that to extract one bit of watermark from a 4×4 pixel image block, it is necessary to perform full QR decomposition, obtain element r'_{11}, and then apply the extraction formula. The computational complexity of such an approach is $O(n^3)$ for each block [23], which for large images leads to unacceptably long processing time.

To solve this problem, work [29] developed a new approach based on a key mathematical observation. Analysis of Sun's formula [23] for watermark extraction showed that only the first diagonal element of R matrix is necessary for information extraction, and all other elements of Q and R matrices are not used in the extraction process.

According to the definition of QR decomposition [30], the first diagonal element of R matrix equals the Euclidean norm of the first column of the original matrix:

$$r'_{11}=\|a'_1\|=\sqrt{(a'^T_1 a'_1)}$$

where a'_1 is the first column of the image block matrix.

For practical implementation, consider an image block A' of size 4×4, which can be represented as a matrix with elements , where i and j correspond to row and column. The first column of this matrix forms vector $a'_1=[a'_{11},a'_{21},a'_{31},a'_{41}]^T$. Then the first diagonal element of R matrix is computed using a simple formula:

$$r'_{11}=\sqrt{a'^2_{11}+a'^2_{21}+a'^2_{31}+a'^2_{41}}$$

This formula has computational complexity of $O(n)$, which radically differs from the $O(n^3)$ complexity of full QR decomposition [30].

Therefore, the watermark extraction steps are described as follows [29].

1. The watermarked image H' is first partitioned into non-overlapping 4×4 blocks, where each pixel is represented by three color channels (R, G, and B). The B-channel values of each block are then arranged into a 4×4 matrix A'.
2. Subsequently, the first diagonal element r'_{11} of the matrix R' is computed as the Euclidean norm of the first column vector of matrix:

$$r'_{11}=\sqrt{\sum_{i=1}^{4} a'^2_{i1}} \tag{19}$$

3. Then the extraction formula developed by Sun for SVD decomposition [23] and adapted for QR is applied:

$$z = r'_{11} \bmod q \tag{20}$$

where q is the quantization parameter that determines the watermark embedding strength.

4. Finally, the watermark bit is determined based on the value of z:

$$\omega = \begin{cases} 0, & z \le \frac{q}{2} \\ 1, & elsewhere \end{cases} \tag{21}$$

Comparison of computational complexity of traditional and fast methods shows a significant advantage of the latter. Traditional QR decomposition has computational complexity $O(n^3)$, while the fast method has complexity $O(n)$. For a typical 4×4 block, this means theoretical speedup of 16 times, and for 8×8 blocks – 64 times.

3.7. Experimental Studies and Results Analysis

3.7.1. Testing Watermark Imperceptibility

Evaluation of watermark imperceptibility is critically important for practical application of any digital content protection method. Detailed quality indicators calculated using formulas (3)–(4) for each test image [31] are presented in Table 1.

Table 1
Results of watermark imperceptibility evaluation on test images

Image [31]	PSNR (dB)	SSIM
Lena	44.80	0.999
Baboon	42.15	0.997
Peppers	43.78	0.998
Barbara	45.21	0.998

Analysis of the obtained results indicates exceptional quality of images after the watermark embedding procedure. The average PSNR value is 43.99 dB, which significantly exceeds the generally accepted threshold of 30dB for practical application of digital watermarking methods.

The Structural Similarity Index (SSIM) demonstrates extremely high values, indicating virtually complete preservation of the original image's structural information. SSIM values close to one confirm that watermark embedding does not lead to noticeable structural deformations of the image that could be detected by the human eye or automatic quality analysis systems

3.7.2.Testing Watermark Robustness to Attacks

For comprehensive robustness testing, a representative test image [31] and watermark were selected, shown in Figure 1.

The work examined a range of typical image processing attacks (Figures 2-6) that model both unintentional distortions arising during data storage and transmission, as well as deliberate attempts to violate the integrity of the embedded watermark. Specifically:

- Gaussian Blur attack is applied to evaluate watermark robustness to smoothing filtrations that suppress high-frequency image components and imitate typical distortions caused by data preprocessing.

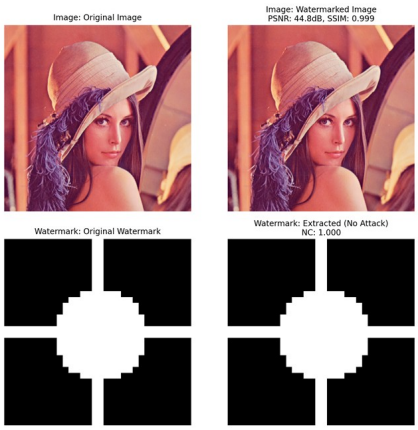


Figure 1. Original image and digital watermark

- Gaussian Noise attack serves to test the digital watermarking scheme's robustness to additive random noise perturbations that model the influence of sensor noise, as well as distortions during data transmission or storage.
- Salt & Pepper attack allows evaluating the watermark's ability to maintain detectability under conditions of impulse noise distortions that lead to local replacement of individual pixels and reflect partial data corruption.
- Scaling attack is designed to test watermark robustness to geometric transformations related to image resizing, and reproduces resampling or data normalization scenarios characteristic of machine learning dataset preparation.
- Sharpening attack enables analysis of watermark robustness to enhancement of high-frequency image components and models the application of sharpness enhancement filters that change local gradients and data structure.

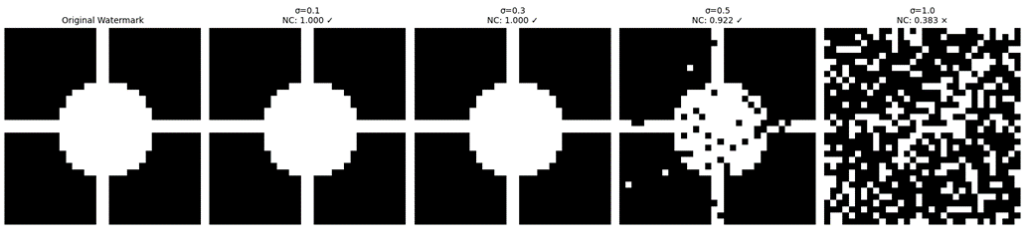


Figure 2. Result after Gaussian Blur attack

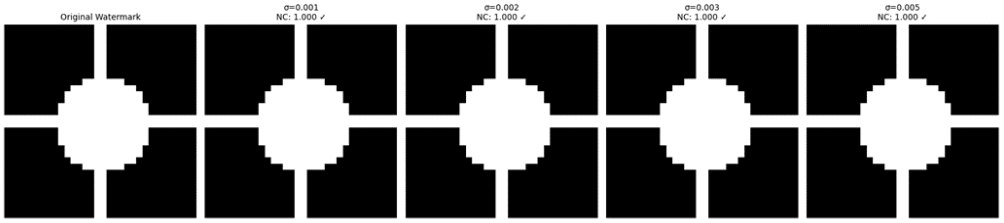


Figure 3. Result after Gaussian Noise attack

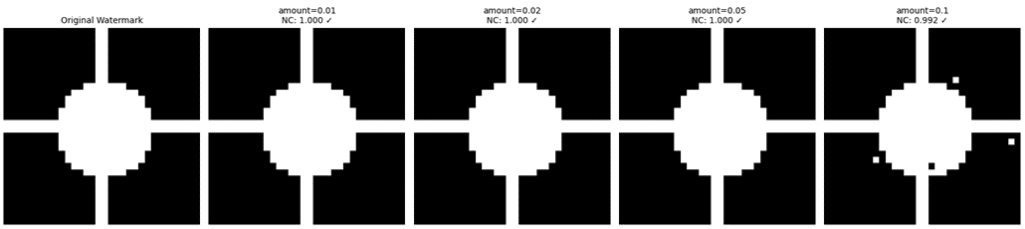


Figure 4. Result after Salt & Pepper attack

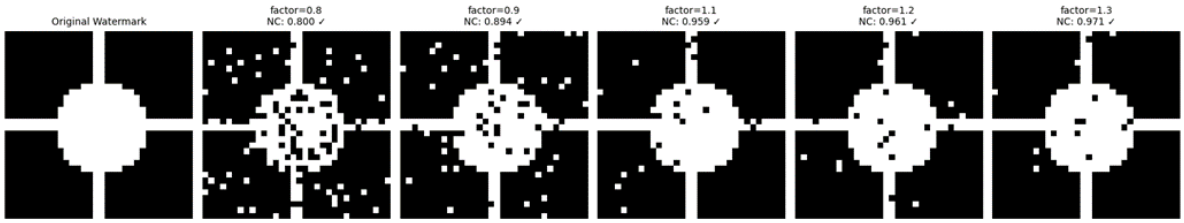


Figure 5. Result after Scaling attack

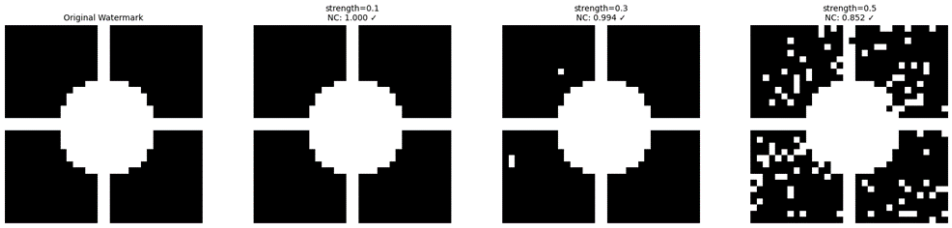


Figure 6. Result after Sharpening attack

The obtained results (Figure 2) demonstrate that the examined digital watermark method is robust to Gaussian blur of weak and medium intensity $(\sigma \leqslant 0.5)$, ensuring high normalized correlation values ($NC \geq 0.92$) and preservation of the watermark's visual structure. At the same time, significant increase of parameter σ leads to critical correlation reduction and loss of watermark suitability for correct identification.

Experimental results (Figure 3) show that the proposed digital watermark embedding method is highly robust to additive Gaussian noise of low and medium intensity. Preservation of $NC = 1.0$ value for all investigated σ parameters indicates that noise does not violate critical structural features of the watermark necessary for its correct extraction. Despite degradation of container image quality, the watermark structure is preserved, and its recovery process demonstrates high robustness to random additive interference.

Analysis of Salt & Pepper attack results (Figure 4) shows that at low and medium values of amount parameter (0.01, 0.02, and 0.05), the normalized correlation value is $NC = 1.0$, indicating complete preservation of the embedded watermark and its error-free detection. Even with increased impulse noise intensity to $amout = 0.1$, the NC value decreases only to 0.992, indicating minor watermark degradation while maintaining high correlation between original and extracted watermarks. The obtained results confirm high robustness of the proposed method to impulse noise attacks: local damage does not lead to complete watermark loss, as the distributed nature of embedding and presence of redundancy allow compensating for partial distortions.

Experimental results (Figure 5) related to scaling attack showed significant dependence of watermark extraction quality on the scaling coefficient. When reducing scale to 0.8, the normalized correlation value decreased to $NC = 0.8$, and at $factor = 0.9$ increased to $NC = 0.894$, indicating significant loss of spatial information. At the same time, when increasing scale in the range $1.1 - 1.3$, NC values remained within 0.959–0.971, ensuring stable and correct watermark recovery. Thus, the examined QR decomposition method demonstrates high robustness to moderate scale increase and reduced robustness to aggressive image size reduction. Nevertheless, it should be noted that the watermark maintains recognizability in all examined cases, indicating certain method invariance to geometric transformations.

The obtained results (Figure 6) indicate that the proposed digital watermarking method is completely robust to weak sharpness enhancement, as evidenced by normalized correlation value $NC = 1.000$ at $strength = 0.1$. With moderate attack intensity ($strength = 0.3$), the method maintains high robustness, and NC value decreases only to 0.994, which does not affect correct watermark identification.

At the same time, with intensive sharpness enhancement ($strength = 0.5$), a more noticeable decrease in normalized correlation indicator to $NC = 0.852$ is observed, indicating partial watermark degradation. However, even in this case, sharpness enhancement does not lead to its critical destruction.

The obtained results confirm that the proposed method is not excessively sensitive to local nonlinear changes in brightness and contrast caused by sharpening attacks. Thus, it is effective for typical image processing scenarios, although its robustness naturally decreases under extreme sharpness enhancement parameters.

3.7.3. Impact of Watermarks on Computer Vision Model Training Quality

Classification Model Training Results

To evaluate the impact of QR decomposition-based digital watermarks on the classification model training process, a series of experiments was conducted using the Cats vs Dogs dataset [32].

Cats vs Dogs is one of the most popular benchmarks used for evaluating binary classification algorithms in computer vision. It contains images of domestic cats and dogs of various breeds, sizes, and under different shooting conditions. The classification task consists of automatically distinguishing between two object classes based on analysis of visual image features. Using this dataset allows evaluating the models' ability to generalize on real photographs with natural lighting and diverse backgrounds.

The experimental study included testing four convolutional neural network architectures: VGG16 [33], ResNet50 [34], MobileNet [35], and EfficientNetB0 [36]. Each model was trained for 10 epochs on two dataset variants: original and modified with embedded watermarks (Figure 7). This allowed for objective comparison of the watermark impact on the training process and final model accuracy. All models used pre-trained weights (transfer learning) and were fine-tuned for the specific binary classification task.

Model training dynamics are presented in Figures 8-11, which demonstrate accuracy and loss curves for training and validation sets throughout the training process. Analysis of the curves shows that using watermarks practically does not affect the convergence speed of training algorithms and does not lead to significant model quality degradation.

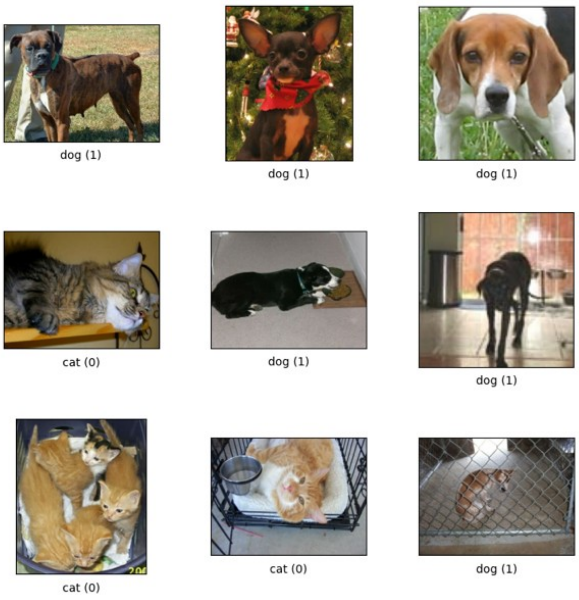


Figure 7. Example from the Cats_vs_Dogs dataset [32]

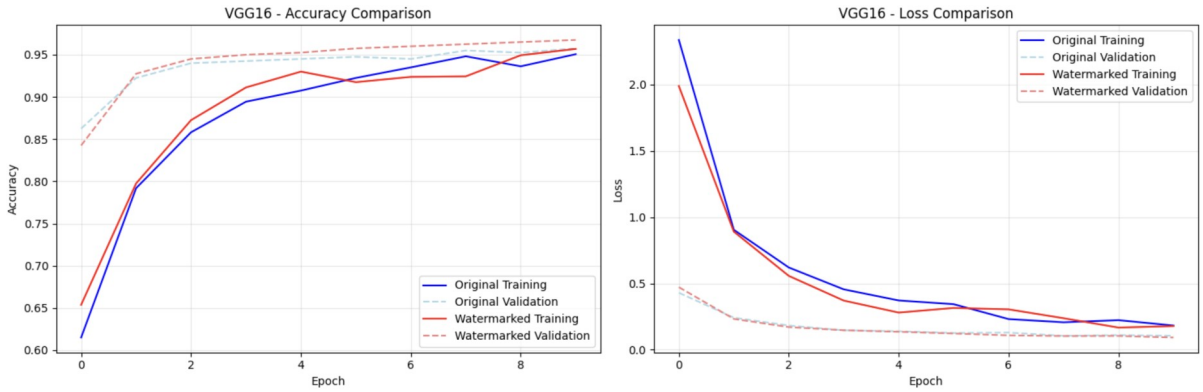


Figure 8. VGG16 model training dynamics: comparison of accuracy and loss for original and watermarked datasets

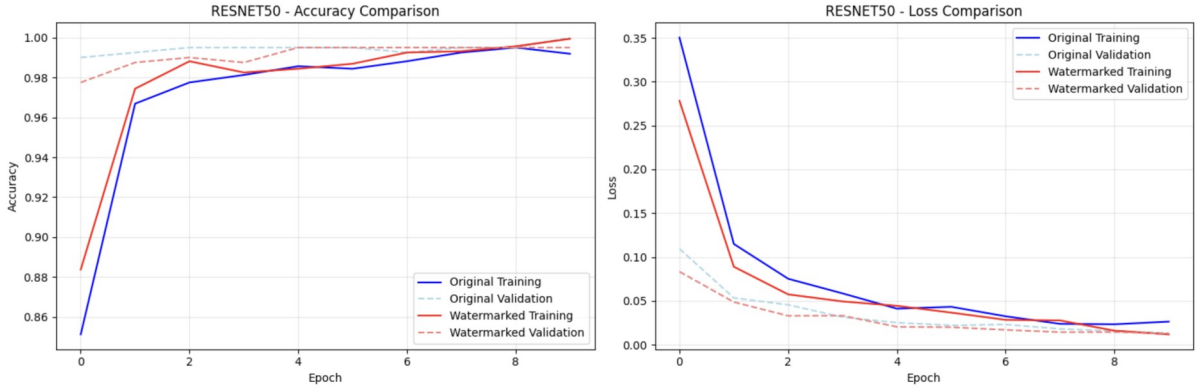


Figure 9. ResNet50 model training dynamics: comparison of accuracy and loss for original and watermarked datasets

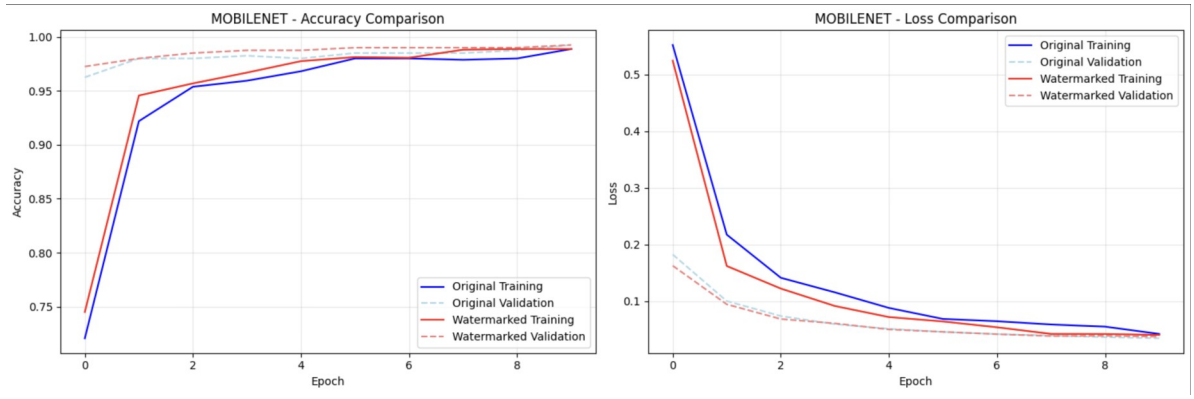


Figure 10. MobileNet model training dynamics: comparison of accuracy and loss for original and watermarked datasets

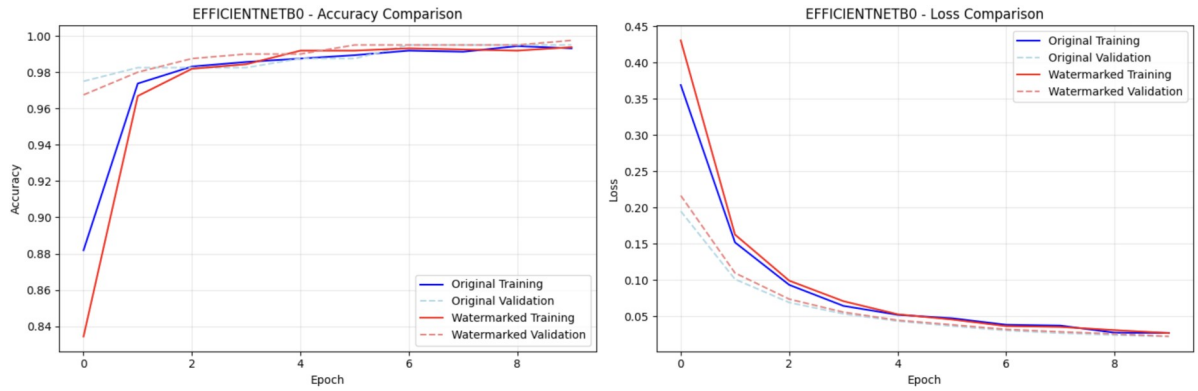


Figure 11. EfficientNetB0 model training dynamics: comparison of accuracy and loss for original and watermarked datasets

Table 2 presents data for comparing classification model training accuracy on the validation set for original data and watermarked datasets.

Analysis of the obtained results shows that watermark embedding does not lead to systematic degradation of model training quality. The most stable results were shown by the MobileNet architecture, where classification accuracy remained absolutely unchanged at 99.25% for both dataset variants, although a slight increase in loss function of 0.0028 is observed.

Table 2

Comparison of classification model accuracy and loss function on Cats vs Dogs dataset

Model Architecture	Original Accuracy (%)	Accuracy with Watermark (%)	Accuracy Difference (%)	Original Loss	Loss with Watermark	Loss Difference
VGG16	95.75	96.75	+1.00	0.1053	0.0918	-0.0135
ResNet50	100.00	99.50	-0.50	0.0134	0.0126	-0.0008
MobileNet	99.25	99.25	0.00	0.0345	0.0373	+0.0028
EfficientNetB0	99.50	99.75	+0.25	0.0219	0.0221	+0.0002

Particularly interesting are the results for VGG16 and ResNet50, where watermarked images led to a reduction in loss function value compared to original images. For VGG16, loss decreased by 12.8% (from 0.1053 to 0.0918), which was accompanied by a 1% accuracy improvement. A similar trend is observed for ResNet50, where a 6% loss reduction correlates with a slight accuracy decrease.

In the case of EfficientNetB0, minimal changes are observed in both accuracy (+0.25%) and loss function (+0.0002), indicating training stability of this architecture regardless of watermark presence.

In addition to accuracy and loss function, the impact of watermarks on model training speed was also analyzed. Table 3 presents data for comparing average processing time per batch during training. Analysis of temporal characteristics shows that the watermark impact on training speed is minimal. The largest processing time increase is observed for VGG16 (+11%), which may be related to architectural features of the model and the method of processing modified images. However, even this increase is not critical for practical application.

Object Detection Model Training Results

For a comprehensive study of watermark impact on various types of computer vision tasks, in addition to classification experiments, research was also conducted concerning object detection tasks.

Detection tasks are characterized by greater complexity compared to classification tasks, as the model must not only recognize objects present in the image but also precisely localize their

position using bounding boxes. Accordingly, evaluation metrics for such tasks differ significantly from those used for classification and include indicators of object localization accuracy.

Table 3

Impact of watermarks on classification model training speed

Model Architecture	Time per batch (original) (ms)	Time per batch (with watermark) (ms)	Difference (ms)	Relative Change (%)
VGG16	218	242	+24	+11.0
ResNet50	117	114	-3	-2.6
MobileNet	36	35	-1	-2.8
EfficientNetB0	44	45	+1	+2.3

One of the most popular and effective architectures for object detection tasks is the YOLO (You Only Look Once) family of models [37]. These models are distinguished by high operating speed and competitive accuracy, making them widely used in both academic research and practical applications. For this experiment, the YOLOv8n architecture [38] was chosen as the most modern and optimized version from the YOLO model family.

Experiments were conducted on the Face Mask Detection dataset (Figure 12), which contains images of people with three types of annotations: “with_mask” (persons wearing masks), “without_mask” (persons without masks), and “mask_incorrect” (persons with incorrectly worn masks). The model was trained for 20 epochs on two data versions with identical hyperparameters and configurations.

The experimental methodology involved using a pre-trained YOLOv8n model with subsequent fine-tuning for the specific mask detection task. Both models used the same optimization parameters (AdamW optimizer [2] with learning rate 0.001429), batch size 16, and image size 640×640 pixels. This configuration ensured objective comparison of watermark impact on the training process.

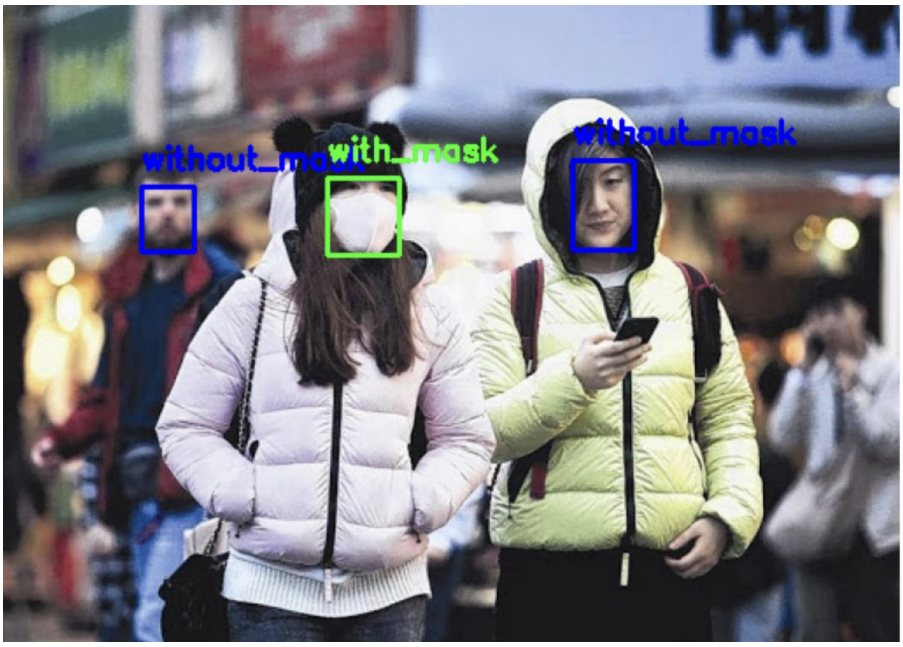


Figure 12. Example from the Face Mask Detection dataset [39]

Main detection model training results are presented in Table 4, which demonstrates comparison of key metrics for evaluating object detection quality.

Table 4

Comparison of detection metrics on Face Mask Detection dataset

Metric	Original Images (%)	Images with Watermark (%)	Absolute Difference (%)	Relative Difference (%)
mAP@0.5	84.0	86.4	+2.4	+2.86
mAP@0.5:0.95	55.7	56.6	+0.9	+1.62
Precision	89.1	87.6	-1.5	-1.68
Recall	74.5	79.9	+5.4	+7.25

Analysis of results shows that the model trained on watermarked data demonstrates improvement in main detection metrics. The most significant improvement is observed for the Recall metric (+5.4%), indicating enhanced model ability to detect all objects present in images. The mAP@0.5 metric, which is a standard detection quality indicator, also showed improvement of 2.4%.

Detailed analysis by individual classes is presented in Table 5, which allows evaluating watermark impact on recognition of different object types.

Table 5. Comparison of detection metrics by object classes

Class	Dataset	Precision (%)	Recall (%)	mAP@0.5 (%)	mAP@0.5:0.95 (%)
with_mask	Original	95.90	91.20	97.30	68.50
	With Watermark	94.10	94.10	97.40	69.00
without_mask	Original	88.90	70.90	81.90	53.10
	With Watermark	78.80	76.70	81.40	52.20
mask_incorrect	Original	82.40	61.50	72.60	45.50
	With Watermark	90.00	68.90	80.40	48.70

Analysis of results by classes demonstrates multi-directional changes for different object types. The “with_mask” class showed stable results with minimal deviations, indicating reliable detection of this object type regardless of watermark presence. For the “mask_incorrect” class, precision improvement from 82.4% to 90.0% and mAP@0.5 from 72.6% to 80.4% is observed, which may indicate positive impact of minimal image modifications on the model's ability to distinguish complex cases.

Comparison of loss functions and temporal training characteristics is presented in Table 6.

Table 6

Comparison of loss functions and training time

Parameter	Original Data	Data with Watermark	Difference
Box Loss	0.146	0.1458	-0.0002
Class Loss	2.474	2.463	-0.011
DFL Loss	0.9905	0.9915	0.001
Training Time (min)	3.78	2.82	-0.96

Analysis of loss functions shows minimal differences between models trained on different dataset versions. Box Loss and Class Loss demonstrate slight improvement for watermarked data, which correlates with improvement in main detection metrics. Particularly important is the fact that training time on the watermarked dataset was 25.4% less, which may indicate better model convergence.

4. Discussion

The experiments conducted on two fundamentally different computer vision tasks – classification and object detection – allow for a comprehensive assessment of the impact of QR decomposition-based watermarks on the model training process. Analysis of results demonstrates consistent trends regardless of task type and model architecture.

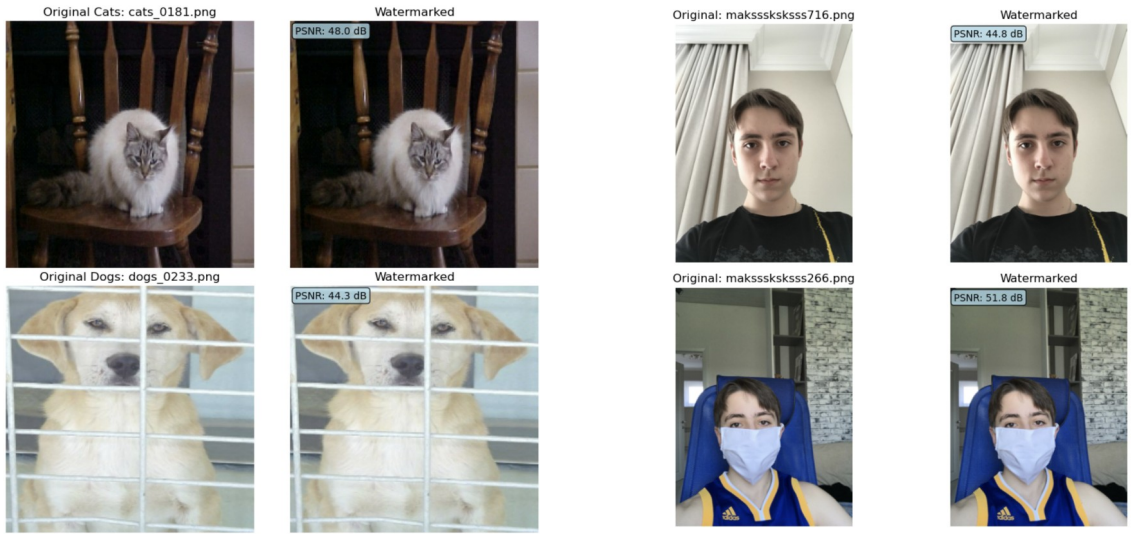
For classification tasks on the Cats vs Dogs dataset, all four tested architectures (VGG16, ResNet50, MobileNet, EfficientNetB0) showed stable results with minimal accuracy deviations within ±1%. Particularly indicative are the MobileNet results, where accuracy remained absolutely unchanged, and VGG16 with EfficientNetB0, which demonstrated even slight improvement on watermarked images.

Similar trends are observed for object detection tasks on the Face Mask Detection dataset. The YOLOv8n model showed improvement in main detection metrics: mAP@0.5 increased by 2.86%, and Recall by 7.25%. Moreover, training time was reduced by 25.4%, which may indicate better model convergence on watermarked images.

Analysis of training curves for both task types demonstrates stable convergence without significant quality metric fluctuations. In the case of classification, smooth loss function reduction is observed with similar dynamics for both original datasets and watermarked arrays. For detection tasks, watermarked images showed even better convergence stability, especially in the final training epochs.

It is important to note that none of the tested models demonstrated signs of overfitting or instability when working with watermarked images. This confirms the hypothesis that minimal modifications introduced by the QR decomposition method do not violate the structural integrity of images necessary for effective model training.

Figure 13 demonstrates examples of original images and images with watermarks for both datasets examined in the work.



a) b)

Figure 13. Visual comparison of original (left) and watermarked (right) images: (a) – examples from Cats vs Dogs; (b) – examples from Face Mask Detection

Visual analysis confirms the quantitative metrics results – the difference between original and watermarked images is practically imperceptible to human perception. This is critically important for practical application of the proposed approach, as it ensures the possibility of protecting datasets without loss of their visual quality and suitability for training models.

Conclusions

This work proposes a robust digital watermarking method for protecting machine learning image datasets based on a modified QR decomposition method. The proposed approach ensures reliable embedding of imperceptible watermarks while preserving the functional suitability of datasets used for training deep learning models.

Experimental results showed that watermarks do not degrade the visual quality of images, which is confirmed by high values of PSNR and SSIM metrics, and also do not reduce the accuracy of machine learning models. On the contrary, in a number of experiments, slight improvement in training stability was observed, which may be related to the effect of weak regularization caused by the embedding process.

The conducted experimental studies showed that the proposed digital watermarking method ensures high robustness to most typical image processing attacks, including Gaussian blur, additive and impulse noise, scaling, and sharpness enhancement, maintaining high normalized correlation values over a wide range of parameters. At the same time, results indicate natural robustness reduction at extreme attack parameter values; however, the watermark remains recognizable, which confirms the method's practical suitability.

Overall, the obtained results confirm that the proposed digital watermarking scheme ensures an effective balance between imperceptibility, robustness, and computational efficiency, making it suitable for practical protection of machine learning image datasets and intellectual property objects in real-world applications.

Declaration on Generative AI

While preparing this work, the authors used the AI programs Grammarly Pro to correct text grammar and Strike Plagiarism to search for possible plagiarism. After using this tool, the authors reviewed and edited the content as needed and took full responsibility for the publication’s content.

References

[1] R. Banakh, A. Piskozub, I. Opirskyy. Devising a method for detecting “evil twin” attacks on IEEE 802.11 networks (Wi-Fi) with KNN classification model. Eastern-European Journal of Enterprise Technologies, 3, 9(123), 2023, 20–32, doi:10.15587/1729-4061.2023.282131

[2] O. Hospodarskyy, et al. Understanding the Adam Optimization Algorithm in Machine Learning. CEUR Workshop Proceedings, 3742, 2024, 235–248.

[3] T. Korobeinikova, et al. Combined ML Approaches for Automated Detection of Server Attack. In Int. Conf. on Advanced computer information technologies ACIT, 2025, 569–572, doi:10.1109/ACIT65614.2025.11185805

[4] M. Tsebak, R. Banakh, A. Piskozub. Enhancing the Efficiency of Secret Detection in Version Control Systems Using Machine Learning Methods. 2025 IEEE 5th Int. Conf. on Smart Information Systems and Technologies, Conference Proceedings, 2025. doi:10.1109/SIST61657.2025.11139348

[5] I. Zhuravel, et al. Stochastic SIR Model for Computer Worm Propagation. In Int. Conf. on Advanced Computer Information Technologies, ACIT, 2025, 505–508, doi:10.1109/ACIT65614.2025.11185628

[6] I. Opirskyy, et al. Modern Methods of Ensuring Information Protection in Cybersecurity Systems using Artificial Intelligence and Blockchain Technology. Technology Center PC, 2025. doi:10.15587/978-617-8360-12-2

[7] S. A. Azcoitia, N. Laoutaris. A survey of data marketplaces and their business models. ArXiv, 2022, doi:10.48550/arxiv.2201.04561

[8] General Data Protection Regulation, 2018. https://gdpr-info.eu/

[9] California Consumer Privacy Act, Cal. Civ. Code § 1798.100 et seq., 2018.

[10] O. Deineka, et al. Designing Data Classification and Secure Store Policy According to SOC 2 Type II. CEUR Workshop Proceedings, 3654, 2024, 398–409.
[11] D. Shevchuk, O. Harasymchuk, A. Partyka, N. Korshun. Designing Secured Services for Authentication, Authorization, and Accounting of Users. CEUR Workshop Proceedings, 3550, 2023, 217–225.
[12] O. Deineka, O. Harasymchuk, A. Partyka, V. Kozachok. Information classification framework according to SOC 2 Type II. CEUR Workshop Proceedings, 3826, 2024, 182–189.
[13] O. Deineka, et al. Detection confidential information by large language models. Informatyka, Automatyka, Pomiary W Gospodarce I Ochronie Środowiska, 15(3), 2025, 91–99, doi:10.35784/iapgos.6910
[14] N. Kukharska, et al. Embedding a digital watermark into an image using a Hamming code and a solver of the square root of a prime number. In Int. Conf. on Advanced Computer Information Technologies, ACIT, 2025, 537–541, doi:10.1109/ACIT65614.2025.11185880
[15] P. Maesen, D. İşler, N. Laoutaris, Z. Erkin. Image watermarking for machine learning datasets. DEC '23: Proceedings of the Second ACM Data Economy Workshop, 2023, 7–13, doi:10.1145/3600046.3600048
[16] N. Kukharska, A. Lagun, O. Yashchyk. Crypto-steganographic system based on the solver of the square root of a prime number. In Int. Conf. on Advanced Computer Information Technologies, ACIT, 2024, 535–538, doi:10.1109/ACIT62333.2024.10712448
[17] N. Kukharska, A. Lagun, O. Polotai. The steganographic approach to data protection using Arnold algorithm and the pixel-value differencing method. In 2020 IEEE 3rd Int. Conf. on Data Stream Mining and Processing, DSMP 2020. Article ID 9204108, 2020, 174–177, doi:10.1109/DSMP47368.2020.9204108
[18] A. Bamatraf, R. Ibrahim, M. N. M. Salleh. A new digital watermarking algorithm using combination of least significant bit (LSB) and inverse bit. arXiv, 2011, doi:10.48550/arxiv.1111.6727
[19] I. K. Yeo, H. J. Kim. Generalized patchwork algorithm for image watermarking. Multimedia Systems 9, 2003, 261–265, doi:10.1007/s00530-003-0097-0
[20] Q. Su, et al. (). Embedding color image watermark in color image based on two-level DCT. SIViP 9, 2015, 991–1007, doi:10.1007/s11760-013-0534-2
[21] H. Agarwal, B. Raman, I. Venkat. Blind reliable invisible watermarking method in wavelet domain for face image watermark. Multimed Tools Appl 74, 2015, 6897–6935, doi:10.1007/s11042-014-1934-1
[22] C. Chang, P. Tsai, C. Lin. SVD-based digital image watermarking scheme. Pattern Recognition Letters, 26(10), 2005, 1577–1586, doi:10.1016/j.patrec.2005.01.004
[23] R. Sun, H. Sun, T. Yao. A SVD- and quantization based semi-fragile watermarking technique for image authentication. In 6th Int. Conf. on Signal Processing, 2, 2002, 1592–1595, doi:10.1109/ICOSP.2002.1180102
[24] Q. Su, et al. Color image blind watermarking scheme based on QR decomposition. Signal Processing, 94, 2013, 219–235, doi:10.1016/j.sigpro.2013.06.025
[25] P. Skladannyi, et al., Adaptive Methods for Embedding Digital Watermarks to Protect Audio and Video Images in Information and Communication Systems, in: Classic, Quantum, and Post-Quantum Cryptography, 4016, 2025, 13–31.
[26] Y. Kostiuk, et al., Application of Statistical and Neural Network Algorithms in Steganographic Synthesis and Analysis of Hidden Information in Audio and Graphic Files, in: Classic, Quantum, and Post-Quantum Cryptography, 4016, 2025, 45–65.
[27] S. Buchyk, et al., Improvement of Steganographic Methods based on the Analysis of Image Color Models, in: Workshop on Cybersecurity Providing in Information and Telecommunication Systems, CPITS, 2923, 2021, 117–124.
[28] D. Wang, F. Yang, H. Zhang. Blind color image watermarking based on DWT and LU decomposition. Journal of Information Processing Systems, 2016. doi:10.3745/jips.03.0055

[29] T. N. Phuong, et al. Consideration of A Robust Watermarking Algorithm for Color Image Using Improved QR Decomposition, 2021. doi:10.21203/rs.3.rs-423766/v1
[30] Vandenberghe, L. (2018). QR factorization, ECE133A, P.6-1-6-39.
[31] Matlab/C/Python/Shell programming and image/video processing/compression https://www.hlevkin.com/hlevkin/06testimages.htm.
[32] W. Cukierski. Dogs vs. cats. Dataset. Kaggle, 2013. https://kaggle.com/competitions/dogs-vs-cats
[33] K. Simonyan, A. Zisserman. Very deep convolutional networks for Large-Scale image recognition. arXiv, 2014, doi:10.48550/arxiv.1409.1556
[34] K. He, X. Zhang, S. Ren, J. Sun. Deep Residual Learning for Image Recognition, 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 2016, 770–778, doi:10.1109/CVPR.2016.90
[35] A. G. Howard, et al. MobileNets: efficient convolutional neural networks for mobile vision applications. arXiv, 2017, doi:10.48550/arxiv.1704.04861
[36] M. Tan, Q. V. Le. EfficientNet: Rethinking model scaling for convolutional neural networks. arXiv, 2019, doi:10.48550/arxiv.1905.11946
[37] Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, 779–788, doi:10.1109/CVPR.2016.91
[38] G. Jocher, A. Chaurasia, J. Qiu. Ultralytics YOLOv8 [Computer software]. GitHub, 2023. https://github.com/ultralytics/ultralytics.
[39] O. Gurav. Face mask dataset [Dataset]. Kaggle, 2020. https://www.kaggle.com/datasets/omkargurav/face-mask-dataset.