CYBERNETICS and COMPUTER TECHNOLOGIES

Lossy image compression algorithms play a crucial role in various domains, including graphics, and image processing. As image information density increases, so do the resources required for processing and transmission. One of the most prominent approaches to address this challenge is color quantization, proposed by Orchard et al. (1991). This technique optimally maps each pixel of an image to a color from a limited palette, maintaining image resolution while significantly reducing information content. Color quantization can be interpreted as a clustering problem (Krishna et al. (1997), Wan (2019)), where image pixels are represented in a three-dimensional space, with each axis corresponding to the intensity of an RGB channel. However, scaling of traditional algorithms like K-Means can be challenging for large data, such as modern images with millions of colors. This paper reframes color quantization as a three-dimensional stochastic transportation problem between the set of image pixels and an optimal color palette, where the number of colors is a predefined hyperparameter. We employ Stochastic Quantization (SQ) with a seeding technique proposed by Arthur et al. (2007) to enhance the scalability of color quantization. This method introduces a probabilistic element to the quantization process, potentially improving efficiency and adaptability to diverse image characteristics. To demonstrate the efficiency of our approach, we present experimental results using images from the ImageNet dataset. These experiments illustrate the performance of our Stochastic Quantization method in terms of compression quality, computational efficiency, and scalability compared to traditional color quantization techniques.

Keywords: non-convex optimization, stochastic optimization, stochastic quantization, color quantization, lossy compression.

A. KOZYRIEV, V. NORKIN

LOSSY IMAGE COMPRESSION WITH STOCHASTIC QUANTIZATION

Introduction. Digital image representation in modern video displays relies on additive color mixing, where the intensity of three primary colors (Red, Green, and Blue) is modulated at each pixel [1, 2]. In this system, each pixel is encoded as a triplet of unsigned integers, with each element corresponding to the intensity of a primary color. For instance, an 8-bit display can represent $2^8 = 256$ values per RGB channel, yielding a total of $2^{8 \cdot 3} = 16\,777\,216$ possible colors. To efficiently store images produced by digital photography, compression algorithms have been developed to balance data storage requirements and image quality. Many of these algorithms fall into the category of lossy compression, where the compressed image experiences a reduction in quality that cannot be fully recovered to its original state.

Color quantization, introduced by Orchard et al. [1], is one approach to lossy image compression that reduces image data without compromising the original resolution. This method comprises two primary stages: (1) selecting an optimal color palette of dominant colors present in the image, and (2) mapping each pixel to the nearest color within this optimal palette. This approach conceptualizes each pixel as a point in three-dimensional space, with each axis representing the intensity of one of the three primary colors. Consequently, pixels with similar color shades cluster more closely in this space. The original paper explores various techniques to address the color quantization problem, including total squared error (TSE) minimization, binary tree palette design, and the Linde-Buzo-Gray (LBG) algorithm, among others. Krishna et al. [3] proposed a solution to the color quantization problem utilizing the K-Means algorithm [4] for optimal color palette selection, where cluster centers represent optimal colors. This approach was further investigated in subsequent works [5, 6].

However, a recent study [7] highlighted the limitations of traditional clustering methods in solving quantization problems, particularly their poor scalability for large datasets. With the rapid increase in image resolution in modern video displays, this scalability bottleneck may significantly impact the performance of these color quantization methods.

In this paper, we present a novel interpretation of color quantization within the domain of stochastic programming, specifically as a non-convex stochastic transportation problem [8]. We employ the Stochastic Quantization algorithm [9, 7] to determine the optimal color palette by minimizing the distance to the set of colors in the original image.

We provide experimental results of this approach using test images from the ImageNet dataset [10], implemented using a Python implementation of the Stochastic Quantization algorithm.

The problem setting. The transportation problem (see, e.g., [8, 9, 7]) is utilized to approximate one discrete probability distribution $\{\xi_i\} \in \Xi \subset \mathbb{R}^n$ with another discrete distribution containing fewer elements $\{y_k\} \in Y \subset \mathbb{R}^n$. The optimal positioning of each element $\{y_1, \ldots, y_K\}$ is determined by minimizing the Wasserstein (Kantorovich–Rubinstein) distance:

$$\min_{y=\{y_1,\ldots,y_K\}\in Y^K \subset \mathbb{R}^{nK}} \min_{q=\{q_1,\ldots,q_K\}\in \mathbb{R}_+^K} \min_{x=\{x_{ij}\geq 0\}} \sum_{i=1}^{I}\sum_{k=1}^{K} d(\xi_i, y_k)^r x_{ik} \tag{1}$$

subject to the constraints:

$$\sum_{k=1}^{K} x_{ik} = p_i, \quad \sum_{k=1}^{K} q_k = 1, \quad i = 1, \ldots, I . \tag{2}$$

where $p_i > 0$, $\sum_{i=1}^{I} p_i = 1$ represent normalized supply volumes, x_{ik} denote transportation volumes, $d(\xi_i, y_k) = \sqrt{\sum_{j=1}^{n} |\xi_{ij} - y_{kj}|^2}$ is the metric between elements in the objective function (1), $Y \subset \mathbb{R}^n$ is a common constraint set for variables $\{y_k, k = 1, \ldots, K\}$, and $n, I, K \in \mathbb{N}$, $r \geq 1$. We chose the Euclidean norm as a distance metric $d(\xi_i, y_k)$, as the similar colors would have close primary colors magnitude, thus they have small distance in Euclidean space between each other.

In the context of color quantization, the distribution $\{\xi_i\}$ represents a set of pixels from the original image, while $\{y_k\}$ denotes an optimal color palette with the number of colors K set manually as a hyperparameter. Both $\{\xi_i\}$ and $\{y_k\}$ are subsets of the space $\{(x, y, z) \in \mathbb{Z}^3 : 0 \leq x, y, z \leq 255\}$, representing the combined intensity of three primary colors for each image pixel. Prior to solving the problem (1), we normalize the original pixel distribution $\{\xi_i\}$ to the unit cube $[0,1]^3 \subset \mathbb{R}^3$.

Stochastic Color Quantization. Paper [7] proposed a solution to the problem (1) by reformulating it as a nonconvex non-smooth stochastic transportation problem:

$$\min_{y=\{y_1,\ldots,y_K\}\in Y^K \subset \mathbb{R}^{nK}} F(y_1, \ldots, y_k), \tag{3}$$

$$F(y) = F(y_1, \ldots, y_k) = \sum_{i=1}^{I} p_i \min_{1\leq k \leq K} d(\xi_i, y_k)^r == \mathbb{E}_{i \sim p} \min_{1\leq k \leq K} d(\xi_i, y_k)^r ,$$

where $\mathbb{E}_{i \sim p}$ denotes the expected value over a set of image pixels $\{\xi_i\}$ with corresponding probabilities $\{p_i\}$.

To solve the stochastic transportation problem (3), we employed the Stochastic Quantization algorithm, as discussed in [9] and [7]:

$$y_k^{(t+1)} = \pi_Y\left(y_k^{(t)} - \rho_t g_k^{(t)}\right), \quad k = 1,\ldots,K,$$

$$g_k^{(t)} = \begin{cases} r \parallel \tilde{\xi}^{(t)} - y_k^{(t)} \parallel^{r-2} (y_k^{(t)} - \tilde{\xi}^{(t)}), & k = k^{(t)}, \\ 0, & k \neq k^{(t)}, \end{cases}$$

$$k^{(t)} \in S(\tilde{\xi}^{(t)}, y^{(t)}) = \arg\min_{1 \leq k \leq K} d(\tilde{\xi}^{(t)}, y_k^{(t)}), \quad t = 0,1,\ldots,$$

where $y^{(t)} = (y_1^{(t)},\ldots,y_K^{(t)})$, $\tilde{\xi}^{(t)}$ is a random element from the set $\{\xi_i, i = 1,\ldots,I\}$, $\rho_t > 0$ is a learning rate parameter, π_Y is the projection operator onto the set $Y \subseteq \mathbb{R}^n$, $r \geq 1$ is the degree of the norm, K is the number of colors in the optimal palette, and $t = 0,1,\ldots$ is the iteration number. To calculate the update for the element $y_k^{(t)}$ based on the gradient value $g_k^{(t)}$ we used Stochastic Gradient Descent (SGD) algorithm [11] and its nonconvex nonsmooth extension from [12].

In contrast to traditional clustering algorithms such as K-Means [4], which updates all components $y_k^{(t)}$ and uses all elements $\{\xi_i\}$ per iteration t, our approach processes only single component of $y^{(t)}$ and uses one random element $\tilde{\xi} \in \{\xi_1,\ldots,\xi_I\}$ per iteration. This eliminates memory constraints and enhances scalability for high-resolution images. Furthermore, paper [7] establishes local convergence conditions for the algorithm (4) under specific initial conditions that is not typically guaranteed for traditional algorithms.

Given the non-convex nature of the problem, the initialization (seeding) of elements $y^{(0)} = \{y_k, k = 1,\ldots,K\}$ plays a crucial role in the algorithm's convergence. While uniform sampling from the original distribution Ξ is the fastest seeding approach, it can compromise accuracy. We implemented the seeding technique proposed in [13], where the selection probability of a new element in $y_k^{(0)}$ is proportional to its distance from previously selected elements $\{y_k^{(0)}\}$. The seeding algorithm proceeds as follows:

Sample the initial element $y_0^{(0)}$ uniformly from Ξ,

Select the next element $y_k^{(0)}$ with probability $\min_{1 \leq s \leq k} \parallel \xi_j - y_s^{(0)} \parallel^2 \Big/ \sum_{i=1}^{I} \min_{1 \leq s \leq k} \parallel \xi_i - y_s^{(0)} \parallel^2$,

Repeat step 2 until $y^{(0)}$ contains K elements.

The research [7] empirically demonstrates the efficiency of this seeding technique and compares the convergence rates of modified versions of the algorithm (4) with adaptive learning rates.

Numerical Experiments. To evaluate the efficiency of Stochastic Color Quantization, we conducted lossy compression experiments on a subset of images from the ImageNet dataset [10], varying in resolution and stored in JPEG format. The algorithm was implemented in Python, utilizing NumPy [14] for efficient CPU-based tensor operations. All experiments were executed on a virtual machine with a dual CPU core and 8GB RAM. For reproducibility, we set the random seed to 42 and used consistent hyperparameter values ($\rho = 0.001$, $r = 3$) across all experiments. The source code and experimental results are publicly available in our GitHub repository [15].

The table presents the algorithm's convergence time, the optimal value of the objective function (3), and the Mean Squared Error (MSE) distortion metric between original and compressed images. To demonstrate the algorithm's performance, we applied it to a monochrome image labeled ILSVRC2012 val

00023267 from the dataset. The original image (1200×1206 resolution, 248 distinct colors, 790.0KB) was compressed to 207.4KB with an optimal palette of four colors: #707070, #3e3e3e, #d7d7d7, and #a1a1a1 (see Figure). The algorithm converged in 17.7±0.2 seconds, achieving a transport value of F(y) = 129465.2 and a distortion of MSE = 0.0037. Additionally, we performed an experiment on the original colorful image (135917 distinct colors, 1.4MB), which was compressed to 213.5KB with an optimal palette (#8c684a, #ebd3a6, #513926, #c79967), convergence time 18.2±0.2 seconds, transport value F(y) = 170058.8, and a distortion MSE = 0.0055.

TABLE. Comparison of the optimal transport value for images with different resolutions

| Optimal color palette (K)	Image resolution		
	438x500	1200x1206	1606x2400
4	F(y) = 22275.6 t = 2.78±0.07 sec MSE = 0.0043	F(y) = 170058.8 t = 17.83±0.21 sec MSE = 0.0056	F(y) = 419947.9 t = 47.64±0.4 sec MSE = 0.0049
8	F(y) = 14673.1 t = 2.94±0.1 sec MSE = 0.0019	F(y) = 122733.9 t = 19.11±0.45 sec MSE = 0.0029	F(y) = 265572.5 t = 50.97±1.1 sec MSE = 0.0019
12	F(y) = 12138.0 t = 3.16±0.17 sec MSE = 0.0013	F(y) = 98810.9 t = 20.55±0.83 sec MSE = 0.0018	F(y) = 205460.5 t = 54.88±2.08 sec MSE = 0.0012
24	F(y) = 9789.3 t = 4.33±0.45 sec MSE = 0.0009	F(y) = 71381.9 t = 28.36±2.75 sec MSE = 0.0010	F(y) = 146528.3 t = 74.84±7.01 sec MSE = 0.0005
36	F(y) = 8691.3 t = 6.08±0.89 sec MSE = 0.0007	F(y) = 59184.2 t = 39.42±5.83 sec MSE = 0.0007	F(y) = 121956.2 t = 105.84±14.96 sec MSE = 0.0004
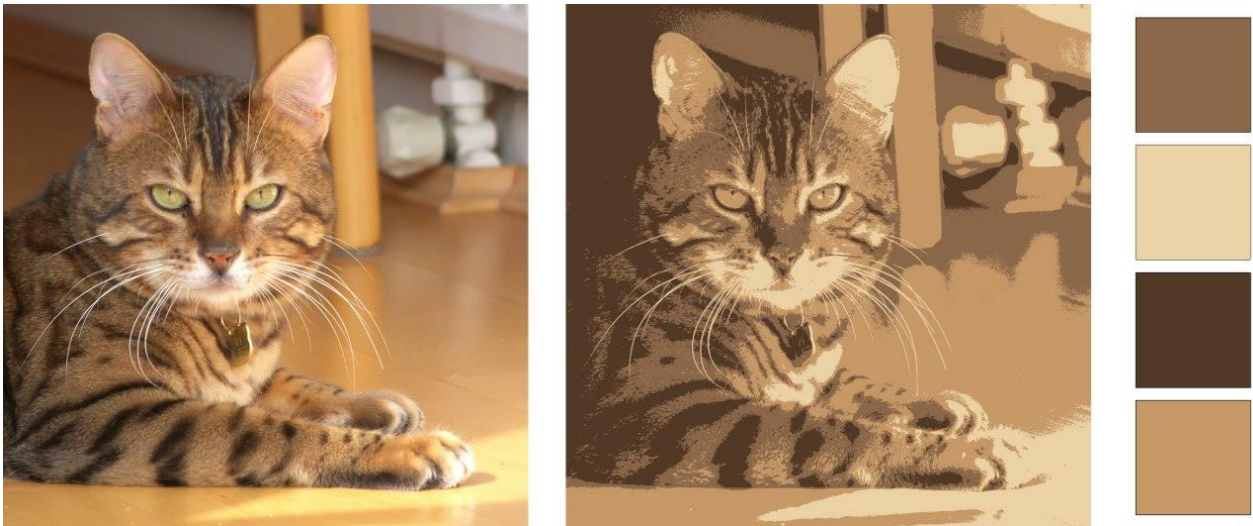
FIGURE. Original image (left) compared with compressed image with lower quality and an optimal color palette with 4 colors (right)

Conclusions. This study introduces a scalable algorithm for solving the color quantization problem without memory constraints, demonstrating its efficiency on a subset of images from the ImageNet dataset [10]. The convergence speed of the algorithm can be further enhanced by modifying the update rule (4) with alternative methods to Stochastic Gradient Descent (SGD) that incorporate adaptive learning rates, as explored in [7]. Moreover, the stochastic nature of the proposed solution enables the utilization of parallelization techniques to simultaneously update the positions of multiple quants $\{y_k, k = 1, \ldots, K\}$, potentially leading to significant performance improvements. This aspect of parallelization and its impact on algorithm efficiency presents a topic for future research. The proposed method not only addresses the limitations of existing color quantization techniques but also opens up new possibilities for optimizing image compression algorithms in resource-constrained environments.

Authorship contribution statement.

Anton Kozyriev: implementation of the stochastic quantization method, the development of lossy image compression example, numerical calculations, writing a draft of the manuscript.

Vladimir Norkin: the development of the stochastic quantization method, editing.

Data availability statement. The data and codes that support the findings of this study are openly available in [15].

References

1. Orchard M.T., Bouman C.A. Color quantization of images. *IEEE Transactions on Signal Processing*. 1991. **39** (12). P. 2677–2690. https://doi.org/10.1109/78.107417
2. Sharma G., Bala R. Digital color imaging handbook. CRC press, 2017.
3. Krishna K., Ramakrishnan K.R., Thathachar M.A.L. Vector quantization using genetic k-means algorithm for image compression. In *Proceedings of ICICS, 1997 International Conference on Information, Communications and Signal Processing. Theme: Trends in Information Systems Engineering and Wireless Multimedia Communications*. 1997. **3**. P. 1585–1587. https://doi.org/10.1109/ICICS.1997.652261
4. Lloyd S. Least squares quantization in pcm. *IEEE Transactions on Information Theory*. 1982. **28** (2). P. 129–137. https://doi.org/10.1109/TIT.1982.1056489
5. Celebi M.E. Improving the performance of *k*-means for color quantization. *Image and Vision Computing*. 2011. **29** (4). P. 260–271. https://doi.org/10.1016/j.imavis.2010.10.002
6. Wan X. Application of k-means algorithm in image compression. *IOP Conference Series: Materials Science and Engineering*. 2019. **563** (5). 052042. https://doi.org/10.1088/1757-899X/563/5/052042
7. Kozyriev A., Norkin V. Robust clustering on high-dimensional data with stochastic quantization. https://arxiv.org/abs/2409.02066 (accessed: 08.09.2024)
8. Kuzmenko V., Uryasev S. Kantorovich–Rubinstein distance minimization: Application to location problems. *Springer Optimization and Its Applications*. 2019. **149**. P. 59–68. https://doi.org/10.1007/978-3-030-22788-3
9. Lakshmanan R., Pichler A. Soft quantization using entropic regularization. *Entropy*. 2023. **25** (10). 1435. https://doi.org/10.3390/e25101435
10. Russakovsky O., Deng J., Su H. et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision (IJCV)*. 2015. **115** (3). P. 211–252. https://doi.org/10.1007/s11263-015-0816-y
11. Ermoliev Y.M. Methods of Stochastic Programming. M.: Nauka, 1976. (in Russian)
12. Ermol'ev Yu.M., Norkin V.I. Stochastic generalized gradient method for nonconvex nonsmooth stochastic optimization. *Cybernetics and Systems Analysis*. 1998. **34** (2). P. 196–215. https://doi.org/10.1007/BF02742069
13. Arthur D., Vassilvitskii S. *k*-means++: the advantages of careful seeding. In *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '07. Society for Industrial and Applied Mathematics, 2007. P. 1027–1035. ISBN 9780898716245.
14. Harris C.R., Millman K.J., van der Walt S.J. et al. Array programming with NumPy. *Nature*. 2020. **585** (7825). P. 357–362. https://doi.org/10.1038/s41586-020-2649-2
15. Kozyriev A. Lossy image compression with stochastic quantization, Aug 2024. https://github.com/kaydotdev/sqic (accessed: 08.09.2024)

Anton Kozyriev,

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

https://orcid.org/ 0009-0007-6692-2162

a.kozyriev@kpi.ua

Vladimir Norkin,

DcS, Leading researcher fellow, V.M. Glushkov Institute of Cybernetics of the NAS of Ukraine, Kyiv,

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

https://orcid.org/ 0000-0003-3255-0405

vladimir.norkin@gmail.com, v.norkin@kpi.ua

Anton Kozyriev [1] [*], Vladimir Norkin [1,2]

Lossy Image Compression with Stochastic Quantization

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

[2] V.M. Glushkov Institute of Cybernetics of the NAS of Ukraine, Kyiv

[*] Correspondence: a.kozyriev@kpi.ua

Introduction. Lossy image compression algorithms play a crucial role in various domains, including graphics, and image processing. As image information density increases, so do the resources required for processing and transmission. One of the most prominent approaches to address this challenge is color quantization, proposed by Orchard et al. (1991). This technique optimally maps each pixel of an image to a color from a limited palette, maintaining image resolution while significantly reducing information content. Color quantization can be interpreted as a clustering problem (Krishna et al. (1997), Wan (2019)), where image pixels are represented in a three-dimensional space, with each axis corresponding to the intensity of an RGB channel.

The purpose of the paper. Scaling of traditional algorithms like K-Means can be challenging for large data, such as modern images with millions of colors. This paper reframes color quantization as a three-dimensional stochastic transportation problem between the set of image pixels and an optimal color palette, where the number of colors is a predefined hyperparameter. We employ Stochastic Quantization (SQ) with a seeding technique proposed by Arthur et al. (2007) to enhance the scalability of color quantization. This method introduces a probabilistic element to the quantization process, potentially improving efficiency and adaptability to diverse image characteristics.

Results. To demonstrate the efficiency of our approach, we present experimental results using images from the ImageNet dataset. These experiments illustrate the performance of our Stochastic Quantization method in terms of compression quality, computational efficiency, and scalability compared to traditional color quantization techniques.

Conclusions. This study introduces a scalable algorithm for solving the color quantization problem without memory constraints, demonstrating its efficiency on a subset of images from the ImageNet dataset. The convergence speed of the algorithm can be further enhanced by modifying the update rule with alternative methods to Stochastic Gradient Descent (SGD) that incorporate adaptive learning rates. Moreover, the stochastic nature of the proposed solution enables the utilization of parallelization techniques to simultaneously update the positions of multiple quants, potentially leading to significant performance improvements. This aspect of parallelization and its impact on algorithm efficiency presents a topic for future research. The proposed method not only addresses the limitations of existing color quantization techniques but also opens up new possibilities for optimizing image compression algorithms in resource-constrained environments.

Keywords: non-convex optimization, stochastic optimization, stochastic quantization, color quantization, lossy compression.

УДК 004.942, 004.895, 519.688

А. Козирев [1][*], В. Норкін [1, 2]

Стиснення зображення з втратами за допомогою стохастичного квантування

[1] *Національний технічний університет України «Київський політехнічний інститут ім. Ігоря Сікорського»*
[2] *Інститут кібернетики імені В.М. Глушкова НАН України, Київ*
[*] *Листування: a.kozyriev@kpi.ua*

Вступ. Алгоритми стиснення зображень із втратами відіграють вирішальну роль у різних областях, включаючи графіку та обробку зображень. Зі збільшенням щільності інформації про зображення зростають і ресурси, необхідні для обробки та передачі. Одним із найвидатніших підходів до вирішення цієї проблеми є квантування кольорів, запропоноване Orchard et al. (1991). Ця техніка оптимально поєднує кожен піксель зображення з кольором з обмеженої палітри, зберігаючи роздільну здатність зображення при значному зниженні вмісту інформації. Квантування кольорів можна інтерпретувати як проблему кластеризації (Krishna et al. (1997), Wan (2019)), де пікселі зображення представлені в тривимірному просторі, де кожна вісь відповідає інтенсивності каналу RGB.

Мета роботи. Масштабування традиційних алгоритмів, таких як K-Means, може бути складним для великих даних, таких як сучасні зображення з мільйонами кольорів. У цій статті квантування кольорів розглядається як тривимірна стохастична транспортна задача між набором пікселів зображення та оптимальною кольоровою палітрою, де кількість кольорів є попередньо визначеним гіперпараметром. Ми використовуємо стохастичне квантування (SQ) із технікою посіву, запропонованою Arthur et al. (2007) для підвищення масштабованості квантування кольорів. Цей метод вводить імовірнісний елемент у процес квантування, потенційно покращуючи ефективність і адаптованість до різноманітних характеристик зображення.

Результати. Щоб продемонструвати ефективність нашого підходу, ми представляємо експериментальні результати, використовуючи зображення з набору даних ImageNet. Ці експерименти ілюструють ефективність нашого методу стохастичного квантування з точки зору якості стиснення, ефективності обчислень і масштабованості порівняно з традиційними методами квантування кольорів.

Висновки. Це дослідження представляє масштабований алгоритм для вирішення проблеми квантування кольорів без обмежень пам'яті, демонструючи його ефективність на підмножині зображень із набору даних ImageNet. Швидкість конвергенції алгоритму можна додатково підвищити, модифікувавши правило оновлення за допомогою методів, альтернативних стохастичному градієнтному спуску (SGD), які включають адаптивні швидкості навчання. Крім того, стохастичний характер запропонованого рішення дозволяє використовувати методи розпаралелювання для одночасного оновлення позицій кількох квантів, що потенційно може призвести до значного підвищення продуктивності. Цей аспект розпаралелювання та його вплив на ефективність алгоритму є темою для майбутніх досліджень. Запропонований метод не тільки усуває обмеження існуючих методів квантування кольорів, але й відкриває нові можливості для оптимізації алгоритмів стиснення зображень у середовищах з обмеженими ресурсами.

Ключові слова: неопукла оптимізація, стохастична оптимізація, стохастичне квантування, кольорове квантування, стиснення з втратами.