UDC 004.93
DOI: 10.15587/2706-5448.2025.332931

Oleksandr Kozyra,
Andrii Fechan,
Vladyslav Daliavskyi

DEVELOPMENT OF A METHOD FOR USING COLOR IN MACHINE-READABLE OPTICAL CODES TO INCREASE THE INFORMATION CAPACITY

The possibility of increasing the capacity of QR codes by using color modules without adding new metadata is studied. A method for automatically determining the number of colors and their palette using image processing is proposed, which ensures compatibility with classical QR codes. A system is proposed that allows creating a QR code that uses 4, 8 or 16 colors in addition to the standard black and white version.

The main problem is the optimal use of the available color space to minimize errors when reading an informative image with an optical camera, compensate for the effects of uneven lighting and poor image quality, and ensure backward compatibility with the black and white version.

In the course of analyzing the use of different color spaces, the most promising perceptually uniform OKLCH space was determined. Algorithms for image preprocessing for correct decoding of information and an algorithm for encoding and decoding information using color have been developed.

The results obtained are explained by the distribution of the color gamut after the test reading of informative images, the number of errors and successful readings. Using the OKLCH color space, it was possible to read 60% of 16-color test images, while in HSL it was not possible to read any image due to color overlap. However, both spaces have a fairly high rate of successful reads in 4 and 8 color codes.

The use of color will allow the introduction of new standards for high-capacity color machine-readable codes without requiring changes to existing ones for additional metadata, while maintaining full backward compatibility and reliability of black and white codes. Increased information capacity in some cases allows to eliminate the need to be connected to the Internet, reduce the size of the code, and make it more visually appealing by using colors.

Keywords: *optical identification, machine-readable code, QR code, image processing, color space.*

Received: 01.03.2025
Received in revised form: 05.05.2025
Accepted: 20.05.2025
Published: 29.08.2025

How to cite
Kozyra, O., Fechan, A., Daliavskyi, V. (2025). Development of a method for using color in machine-readable optical codes to increase the information capacity. Technology Audit and Production Reserves, 4 (2 (84)), 6–12. https://doi.org/10.15587/2706-5448.2025.332931

1. Introduction

Machine-readable codes are extremely popular due to their small size, low cost, and flexibility, providing wide usage in trade, logistics, and medicine. The one-dimensional codes, like UPC, EAN, and GS1-128 are quite simple, which makes them quick to read [1]. Two-dimensional codes including Data Matrix, Aztec Code, and QR Code are more complex but have a much higher information capacity and mechanisms for detecting and correcting errors, which ensures their reliability [2]. The QR code is perhaps the most popular two-dimensional code developed by DENSO WAVE INCORPORATED in 1994. QR code scans reached 41.77 million in 2025 which is a 433% increase over the past four years [3]. The main advantages over other two-dimensional codes are high data density, and the ability to read a rotated, reflected, displaced, and inverted image [4]. Examples of the use of QR codes include services (payment in transport, institutions), authentication (confirmation of login to accounts or access), identification (obtaining information about the purchased goods, the contents of packages in warehouses). This prevalence of this format indicates that it is quite useful and will be relevant for a long time. Despite its high capacity, that is up to 7089 digits, 4296 alphanumeric characters, 2953 bytes, or 1817 kanji characters in single code [5], sometimes even this is not enough for the needs of users. In this case, users usually encode a link to an electronic resource with the necessary information, instead of directly encoding final information. However, such a solution requires the end user to be connected to the Internet, which is not always possible, for example, in combat situations, during fieldwork, or when staying in basements. There may also be security problems when following unknown links encoded in a QR code [6]. To avoid this, increasing the capacity of machine-readable codes is proposed. QR codes are usually located where there is Internet access, precisely because they need it. However, as capacity increases, this can be changed by encoding the necessary information from a website directly into the code. For example, in the field without Internet access, energy workers can read the characteristics of power lines or substations from the code, military personnel can read information about ammunition, medicine, or humanitarian aid received, and tourists can get detailed information about remote landmarks on the spot. While this information could be attempted in a conventional code, depending on its size, it may not fit into a single code, or may require multiple scans of different codes, which can be avoided by squeezing it into fewer color codes. In addition, color codes look much more attractive than black and white, which will have a positive impact on the marketing industry. Another advantage is that the color QR code is denser, so it can be used to identify small items.

One study [7] suggests using more than 2 colors for each of the code modules to increase the capacity of a QR code. The more colors are used in a color code, the more data can be encoded in the same area. This is a direct logarithmic correlation ($\log_2$) between the number of colors and the amount of encoded information. Thus, by using 4 different colors in each module, it is possible to encode 2 bits of information; with 8 colors, 3 bits can be encoded; and with 16 colors, 4 bits. As a result, using 16 colors, it is theoretically possible to accommodate up to 28356 digits, 17184 alphanumeric characters, 11812 bytes, or 7268 kanji characters in a single code. This increase in capacity will allow to encode larger volumes of plain text (the history of an architectural monument, a list of the contents of large packages). Or even small files of arbitrary formats (encryption keys, images, formatted text, melodies). This code is called HCC2D (High Capacity Colored 2-Dimensional) Code. The authors of the study focused on ensuring backward compatibility between the color and standard code. To achieve correct decoding, they introduced a non-standard repeating color palette into the data encoding area, which allowed each color to be accurately mapped to its original bit combination. However, this approach introduces its own risks, in particular, if the palette is damaged, such a QR code would become unreadable. In addition, the problem of color selection, correction, and decoding, as well as version recognition (the number of colors in the code) remains unsolved.

The other papers [8, 9] are more focused on describing color selection within HSV color space. However, it is probably not the best choice due to limitations and complex calculations for selecting colors. Also, color overlaps were present which led to scan failures, so color correction algorithms were still needed.

Another approach [10] proposes to multiplex multiple QR codes into one colored code. That allows it to freely increase capacity as needed, but like in previous examples it is sensitive to color distortions. The other problem is that the entire process is quite complex and demultiplexing of such code would take a long time, which tells us that it is better to encode information directly in a colored module.

The aim of this study is to develop a method for selecting colors for color machine-readable codes, analyze color models for working with code images, and use methods for correcting and decoding selected colors. This will make it possible to expand the existing standards of machine-readable codes to introduce color versions of increased capacity that can be used in a variety of areas of human activity.

2. Materials and Methods

2.1. The object of research

The object of research is the process of color matching for color machine-readable codes to minimize reading errors using the example of a QR code. Several color models that can be used for image processing are considered, as well as image processing itself to compensate for the influence of external factors.

The main problems in the implementation are the compatibility of color versions with black and white and the definition of a palette for decoding. Version compatibility allows to consider a regular QR code as a color code with a two-color palette and, accordingly, use the same algorithms and software for both versions. This approach allows to reuse part of the tested code of existing QR code scanners and easily move on to the stage of writing or reading the data itself. It follows that adding any new non-standard metadata, such as the size of the color palette or the color palette itself, is not possible. Instead, this data must be extracted from the image.

2.2. Image preprocessing

Before working with color information, the image must be processed to compensate for distortions. The input is a cropped, scaled and aligned image containing only the optical code with distorted colors. The output is an image with corrected colors, which is transmitted for decoding.

First, the white and black balancing is performed taking into account the gradient. To achieve this, the QR code's three search patterns located in the lower left, upper left, and upper right corners are used. For each of these patterns, the maximum and minimum values of the RGB components that make up the white and black references are calculated. This is easy to do because these patterns are always constant and have guaranteed light and dark pixels, otherwise it would be impossible to detect such code and get to this stage. Knowing the centers and the white & black reference pair of each of these patterns, each pixel of the image is linearly approximated with its individual white reference and black reference. With relative to these values for each RGB channel of the current pixel, a linear approximation of the balanced value is performed. In other words, the pixel value moves from a space limited to the maximum black and maximum white in the image to a space with true black RGB (0, 0, 0) and true white RGB (255, 255, 255). This allows to perform white and black balancing, gradient compensation, and removal of color characteristic of a certain type of lighting. After that, gamma decoding is performed [11], resulting in a corrected image.

2.3. Choosing a color space and color palette

It is important to choose the right color space for performing operations on image pixels, because the process of image decoding usually comes down to comparing the color of a pixel with reference samples that correspond to certain pre-known bit combinations for decoding. The simplest option is to use the RGB space and compare colors by their Euclidean distance. One way or another, the image will be encoded in this space at some stage of processing. However, such a comparison is unstable with respect to color distortion and is less intuitive.

It is easier to compare colors by a specific component rather than by all of them at once. It is about color spaces that contain the H (hue) component, which avoids unnecessary calculations of Euclidean distances and simplifies the logic by dividing the hue space into equal ranges. Among these color spaces, the most popular are HSL, HSV, and OKLCH, the latter of which is perceptually uniform. This means that it is designed to model colors in a three-dimensional space that are as close as possible to human perception [12]. Fig. 1 demonstrates the OKLCH color space, where luminance and hue change along the two horizontal axes, and chroma changes along the vertical axis, with the maximum value depending on the previous two indicators [13].

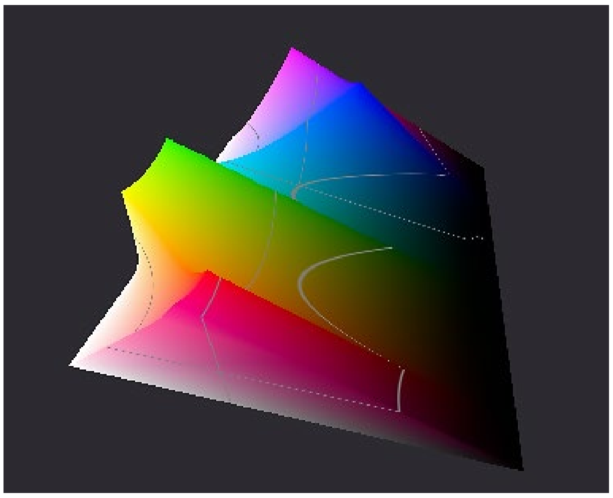
Fig. 1. OKLCH color space

The further choice of color space depends on the features of the color palette for machine-readable code. It is assumed that there are 2, 4, 8, and 16-color versions of the code, where the capacity increases with the number of colors. The 2-color version is the standard black-and-white version that is widely used, so when developing software to work with machine-readable codes of increased capacity, it is advisable to make it backward compatible and treat standard codes as 2-color. It follows that the palette of color versions should include black and white. Thus, the task is to select 0, 2, 6, and 14 additional colors. The selection of 14 different colors can be difficult, especially since they can be distorted after the reading cycle, so it is important to optimize it and build on it to select 2 and 6 colors.

It would be possible to use 14 equally distant colors with the same saturation and lightness, but this method does not guarantee reliability due to possible overlap of neighboring colors after reading. To increase reliability, this paper proposes to create a light and dark variation of each color, thus reducing the number of required colors by half, which in turn reduces the probability of overlapping neighboring colors. In this case, 7 equally distant colors are selected, for each of which a pair of final colors is created that have a higher and lower lightness. However, splitting the hue space into 7 introduces its own inconveniences and limitations.

Consider the RGB color space, which is represented by a cube with white, black, and six primary colors on its vertices. Such a palette would be ideal for an 8-color code because all colors are as equidistant as possible. On the other hand, for a 16-color code, it would be more appropriate to choose six equally distant colors that correspond to the primary colors. To complete the palette to the end, let's take gray as the seventh complementary color. It is possible to achieve a more even use of the available color space and reduce the possibility of color overlap.

In the case of color spaces that contain hue, saturation, and lightness values, six equidistant hue values are selected, each of which creates dark and light colors with maximum saturation. To create grays, it is possible to select an arbitrary shade with minimal saturation.

When using OKLCH, it was enough to select shades with values of 30, 90, 150, 210, 270, 330, which correspond approximately to the primary colors, and for each of them to create a dark-light color pair with brightness values of 0.5 and 0.7. These values were chosen to ensure high color saturation, as can be seen in Fig. 1.

Using the HSL color space, which is not perceptually uniform, has its drawbacks. In particular, when choosing six shades with values of 0, 60, 120, 180, 240, 300, which correspond to the original colors, the choice of lightness values to form a pair of dark and light colors must be made individually for each shade. Assuming that smartphone cameras are designed to match human perception as closely as possible, different colors will have different lightnesses after the reading cycle. Fig. 2 demonstrates the dependence of perceptual luminance on lightness and hue in the HSL model. Similar curves can be seen in Fig. 1 – saturated yellows, greens, and cyans colors appear brighter than others.

2.4. Reading process

The reading process requires finding the value of the original color from the distorted value in the image. First of all, black and white pixels are identified, which can be easily separated by their luminance values, which are close to the minimum and maximum values, respectively. The required luminance limits (L) for black and white colors can be set to be constant, since all colors will be "attracted" to black and white after image preprocessing. In this case, it is possible to choose the values of 0.05 and 0.95, respectively. At this stage, it is worth noting that the use of the HSV color model here has a significant disadvantage compared to HSL or OKLCH, because the (V) value alone does not allow to distinguish, for example, white from bright red.

The next step is to separate gray colors, which are characterized by a low saturation or chroma value (S or C), in this study < 0.2. The use of chromaticity (C) is more reliable, as it is based on human perception, than saturation (S), which is not perceptually uniform, so the chromaticity (C) of the OKLCH space was used for both color models. After that, the lightness (L) value is used to determine whether a color is light or dark. The selection of the lightness limit is described below.

The last step is to divide the remaining colors into six groups by their hue (H) and classify them into light and dark by their lightness (L). In turn, the procedure for differentiating light and dark colors is complicated due to their nature. First of all, it is impossible to choose a universal value of the lightness threshold for all colors, because each color may appear lighter or darker than the other after printing or displaying on the gadget screen. Secondly, perceptual luminance depends not only on lightness, but also on the color hue itself. All these factors require individual selection of the lightness limit for each color group. To do this, within each group, the number of colors with a certain lightness value is counted, and a histogram of the lightness distribution is obtained. The resulting histogram is smoothed, after which it will have two peaks – for light and dark colors, and the valley between them is the optimal brightness threshold, which is a trivial task. In this work, for simplicity, let's take the median between the least and most lightly detected color, and use the array of derivatives to search for the nearest valley.

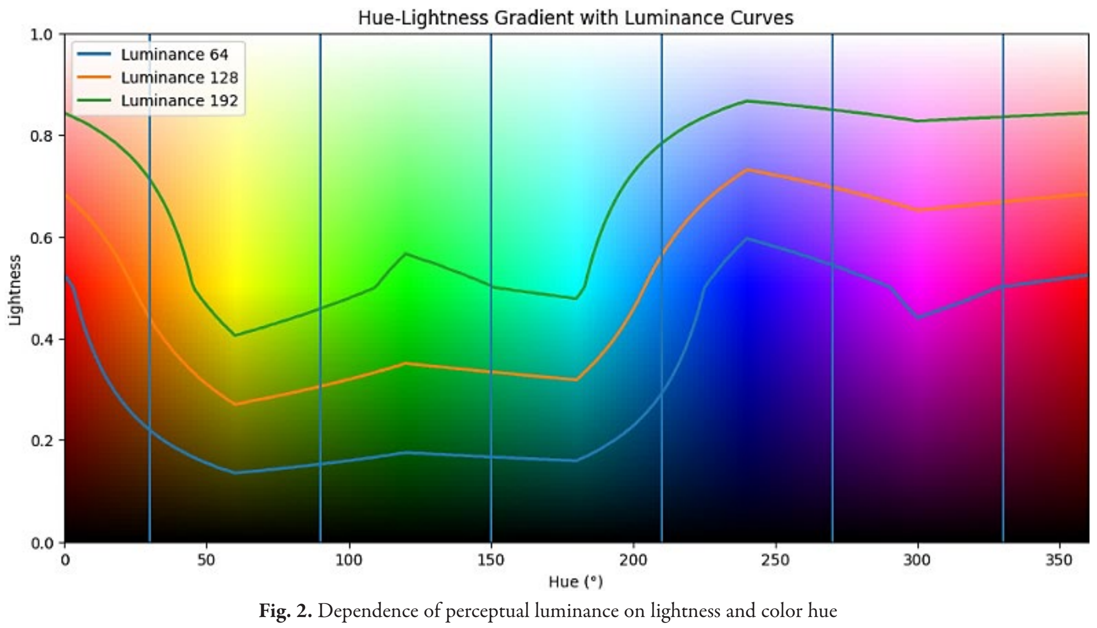

Fig. 2. Dependence of perceptual luminance on lightness and color hue

2.5. Determining the size of the palette (version)

For color codes 4 and 8, instead of selecting full colors, it is possible to select derivative colors that create light and dark variations. This is due to the need to distinguish between versions with different color palette sizes. To do this, it is possible to count the number of colors in each of the six color groups by hue. Next, a boundary is chosen, for example, half the arithmetic mean of the number of colors in each group. Then the number of dominant groups in which the number of colors is higher than the limit is counted. For a 2-color code, there will be no such groups, for a 4-color code there will be 1 dominant group, for an 8-color code there will be 3 groups, and for a 16-color code, ideally all 6 groups. Due to the uneven distribution in the data, some colors may not cross the detection limit, in which case the number of dominant groups is rounded up to the highest number in the series. Hence, it is impossible to select only primary colors for an 8-color code, because then it will be treated as a 16-color code.

It is also worth noting that when scanning a 2-color code, due to distortion, colored pixels may be falsely detected, which can lead to the code being interpreted as a 4-color code. To prevent this, it is possible to make sure that the total number of colored pixels is at least greater than, for example, one-eighth of all pixels in the code. Such a limit allows for unevenness in the encoded data, which causes the dominance of black and white pixels, and is sufficient to detect significant errors in the separation of colors into black, white, and chroma.

2.6. Tools used

For the study, software was developed for working with high-capacity color QR codes using the Dart language and the Flutter cross-platform framework, and reused part of the code from the ready-made library for creating and scanning QR codes Zxing [14].

Using the created software, a data feed was encoded for 2, 4, 8, and 16-color codes with an error correction level of M, for each of them, 10 test images of the printed code and the code displayed on the screen were taken using both HSL and OKLCH. Each snapshot went through the scanning process, after which the number of successful scans and the average number of errors were determined for each series of experiments. To demonstrate and verify the work, 4 screenshots were selected using OKLCH at different stages of decoding.

A Samsung SyncMaster SA100 monitor was used to display the test images. This is an 18.5-inch display with a TN matrix and a resolution of 1366 × 768 pixels. The monitor provides basic image quality with limited viewing angles and mediocre color reproduction, making it suitable for standard office and test tasks, though not for precise graphics work.

The test images were printed using an inkjet printer with a setting of 150 dpi to check the performance of low-quality images.

A Xiaomi Mi A2 Lite smartphone was used to capture the image. The rear camera system includes a dual lens: a 12MP main sensor with f/2.2 aperture and a 5MP depth sensor. The camera captures images with a maximum resolution of 4032 × 3024 pixels, which is suitable for general purpose photography and computer vision. The images were taken in such a way as to select the best lighting conditions for a particular case.

3. Results and Discussion

Fig. 3 demonstrates the generated codes using the OKLCH color distribution, from which it is possible to see the difference in data density in terms of their linear dimensions: 57, 41, 37, and 33 pixels, respectively.

Each code was photographed, only code pixels were selected (Fig. 4), and processed before decoding (Fig. 5).

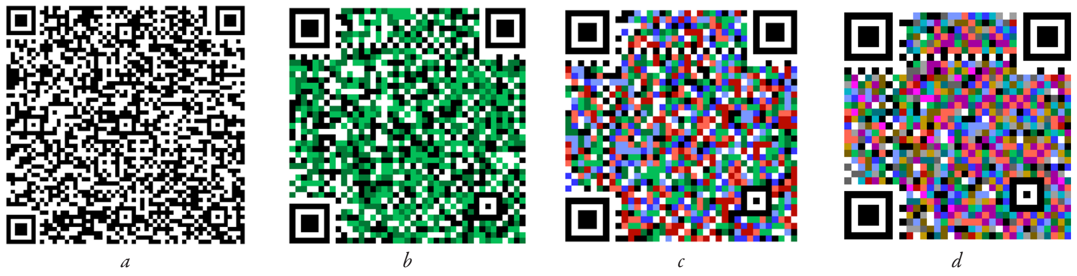

Fig. 3. Generated codes for: *a* – 2 colors; *b* – 4 colors; *c* – 8 colors; *d* – 16 colors

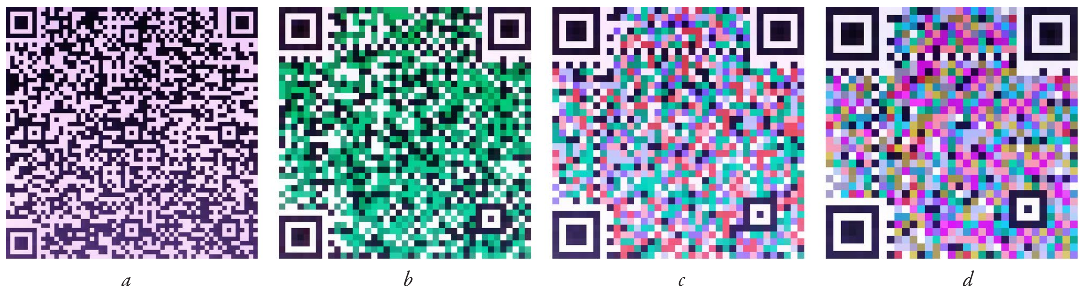

Fig. 4. Color codes after taking a picture with the camera and sampling for: *a* – 2 colors; *b* – 4 colors; *c* – 8 colors; *d* – 16 colors

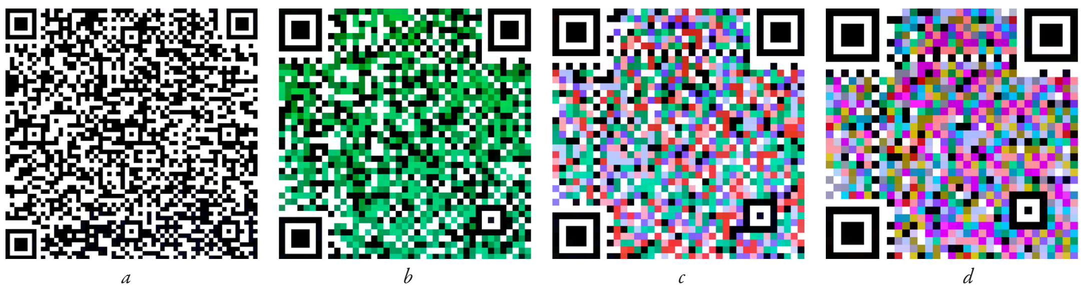

Fig. 5. Color codes after correction: *a* – 2-color code; *b* – 4-color code; *c* – 8-color code; *d* – 16-color code

At this stage, each code is analyzed for color distribution to determine the size of its palette and further decoding. This is demonstrated by the image in Fig. 6, which contains three sections. The upper section shows the distribution of the number of pixels for a certain luminance value for each color group. Typically, this distribution contains two peaks that correspond to dark and light colors. The horizontal line marks the boundary above which colors are considered dark and below which they are considered light. The middle section shows the quantitative distribution of the detected colors by their hue. Vertical lines divide the entire hue space into groups. Each group can contain peaks that correspond to the color used. The closer these peaks are to the middle of the group, the higher the reliability, which means that there is a low probability of overlapping colors in different groups. Between the upper and lower sections, the total number of pixels in a particular group is displayed, which determines the size of the code color palette. For a better understanding of the resulting palette, the lower section shows the distribution of colors by their hue and luminance. A smaller spread of each group indicates good color reproduction and a low probability of errors.

When constructing the image data in Fig. 6, white, black, and gray pixels were not taken into account, as they do not carry useful information in terms of hue and are processed separately.

The reconstructed codes are shown in Fig. 7, with 0 errors detected for the 2-color code, 9 errors for the 4-color code, 20 errors for the 8-color code, and 48 errors for the 16-color code. Scanning errors are visually represented by a pair of read (upper left) – valid (lower right) pixels in Fig. 8. Using these figures, it is easy to identify the overlap of color groups and the locations of the largest error clusters.

A comparison of the number of scanning errors for different codes that use the HSL space is shown in Table 1.

In the case of scanning a 16-color code from paper, there was a constant overlap of color groups, as can be seen in Fig. 9. This phenomenon is usually fatal in scanning, resulting in no code being successfully read. One of the reasons for this is the unevenness of gamma decoding, in this case, the red channel of pink colors is increased much more than the blue channel, which causes a hue shift. The solution may be to fix the hue information before gamma decoding, but this method is not considered in this paper.

The same comparison was made using the OKLCH space, the results of which are shown in Table 2.

When performing scans from 16-color code paper, it was found that most errors occurred due to the misinterpretation of light blue as light gray due to its naturally low saturation. As a result, each scanned code exceeded the maximum number of errors that could be corrected. Therefore, the series of experiments was repeated for the same images with a new gray saturation limit of 0.175 instead of 0.2, as can be seen in Table 3.

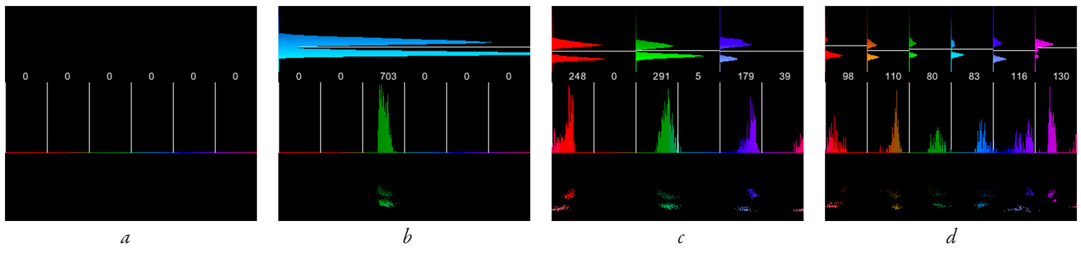

Fig. 6. Color distribution for color codes: *a* – 2-color code; *b* – 4-color code; *c* – 8-color code; *d* – 16-color code

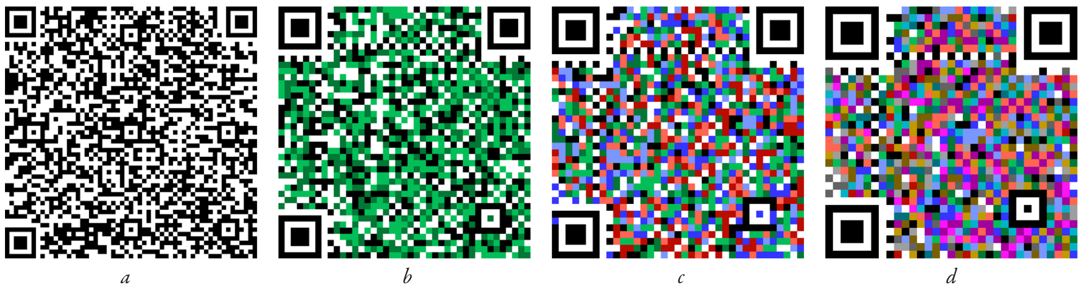

Fig. 7. Reconstructed codes that use: *a* – 2 colors; *b* – 4 colors; *c* – 8 colors; *d* – 16 colors

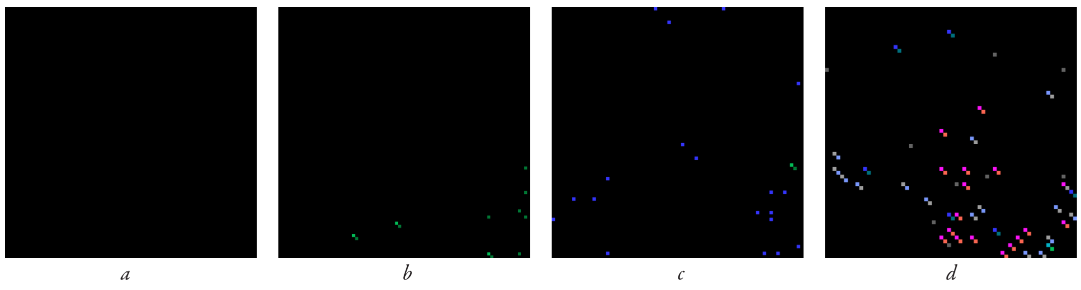

Fig. 8. Scan errors for: *a* – 2-color code; *b* – 4-color code; *c* – 8-color code; *d* – 16-color code

Table 1

Results of a series of code scanning experiments in HSL space

Palette size	2	4	8	16
Successfully scanned codes from the screen	10 (100%)	9 (90%)	10 (100%)	8 (80%)
Average number of scanning errors from the screen	0 (0%)	16.5 (0.98%)	13.9 (1.02%)	34.6 (3.18%)
Successfully scanned codes from paper	10 (100%)	9 (90%)	10 (100%)	0 (0%)
Average number of paper scan errors	0 (0%)	13.2 (0.79%)	16.8 (1.23%)	228.9 (21.02%)

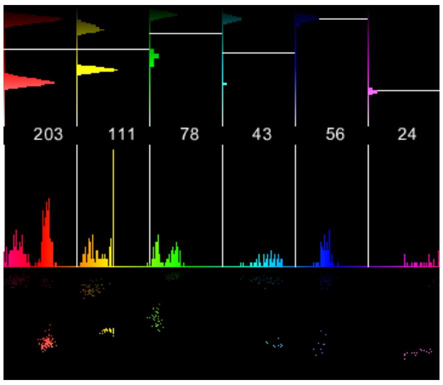

Fig. 9. Demonstration of overlapping magenta and red colors

Table 2

Results of a series of experiments scanning codes in the OKLCH space

Palette size	2	4	8	16
Successfully scanned codes from the screen	10 (100%)	10 (100%)	9 (90%)	7 (70%)
Average number of scanning errors from the screen	0 (0%)	5.5 (0.32%)	33.1 (2.41%)	51.7 (4.75%)
Successfully scanned codes from paper	10 (100%)	10 (100%)	10 (100%)	0 (0%)
Average number of paper scan errors	0 (0%)	17.7 (1.05%)	32.0 (2.34%)	74.9 (6.88%)

Table 3

Comparison of the effect of selecting the saturation limit for gray color detection

Experiment No.	1	2	3	4	5	6	7	8	9	10
Error count for $C_{threshold} = 0.2$	86	59	84	60	98	69	75	78	74	66
Error count for $C_{threshold} = 0.175$	60	28	66	40	77	55	61	66	49	43

Thus, 60% of the codes were successfully scanned, and the average number of errors decreased to 54.5 (5.00%). In turn, choosing a saturation threshold that is too low to detect gray colors can lead to an erroneous interpretation of them as color-containing. Therefore, the choice of the saturation threshold is no less challenging, and the creation of an algorithm for its selection can improve the proposed method.

Both color models can be used to work with 4 and 8 color codes when reading both gadget screens and paper. However, if to plan to use 16-color codes, it is possible to give preference to the OKLCH space, which allows for more stable color recognition under different conditions, unlike the perceptually uneven HSL.

The results obtained in the study offer a solution for the use of color in machine-readable codes to increase their capacity. The proposed method is not limited to machine-readable codes and can be applied to color quantization or color label value recognition [15].

The accuracy of measurements is limited primarily by the lighting conditions or the quality of the display and the capabilities of the camera used, as this directly affects the ability to distinguish between color shades and their luminance. In this case, under poor lighting conditions, this method can only be used if an additional light source is used. To work at enterprises, it is necessary to modernize industrial scanners, which will increase throughput and potentially expand the functionality of such scanners to perform the functions of a camera. However, this will lead to an increase in the cost of equipment and less stable reading. In printing, the advantage will be visual appeal and higher density, but at the expense of reduced reliability and increased sensitivity to lighting conditions.

Further research is possible to develop and evaluate new image processing methods to counteract color distortion in low light conditions, as well as to improve the existing palette recognition and decoding algorithm, including algorithms for determining saturation and luminance limits.

4. Conclusions

In this research, a universal method of increasing the information capacity of machine-readable codes is proposed, which includes rules for choosing colors for encoding information, methods of image preprocessing to compensate for distortions, an algorithm for decoding the processed image and obtaining the original image from it. Software for working with QR codes that use 2, 4, 8, or 16 colors has been developed, and the use of two color spaces has been compared.

It is shown that the perceptually uniform OKLCH color space produces more stable results and is suitable for use with 16-color codes. In contrast, HSL performs worse due to hue groups overlapping. This is explained by the color distribution that should contain distinct peaks for color groups and a clear separation in luminance, which allows accurate decoding of the color codes.

The proposed method will help to implement new standards of high-capacity color machine-readable codes without requiring changes to existing ones for additional metadata, while maintaining full backward compatibility and reliability of black and white codes. This makes it possible to increase the security and expand the application of machine-readable codes.

The proposed method for correcting the color characteristics of an image can be used to improve the accuracy of optical sensors of physical quantities that use the color of the primary transducer as an information parameter.

Conflict of interest

The authors declare that they have no conflict of interest in relation to this research, whether financial, personal, authorship or otherwise, that could affect the research and its results presented in this paper.

Financing

The research was performed without financial support.

Data availability

Data will be made available on reasonable request.

Use of artificial intelligence

The authors confirm that they did not use artificial intelligence technologies when creating the current work.

References

1. *GS1 Barcodes.* Bar Code Graphics, Inc. Available at: https://www.gs1standards.info/gs1-barcodes/

2. *2D Barcode: Types, Use Cases, and Benefits* (2023). Bitly. Available at: https://bitly.com/blog/2d-barcode/

3. Ricson, E. (2025). *61+ QR Code Statistics & Trends 2025 Full Report.* QR TIGER PTE. LTD. Available at: https://www.qrcode-tiger.com/qr-code-statistics-2022-q1

4. *What is a QR Code?* QR Code. Available at: https://www.qrcode.com/en/about/

5. *ISO/IEC 18004:2015: Information technology – Automatic identification and data capture techniques – QR Code bar code symbology specification* (2015). International Organization for Standardization. Geneva. Available at: https://raw.githubusercontent.com/yansikeim/QR-Code/master/ISO%20IEC%2018004%202015%20Standard.pdf

6. David, C. (2024). *QR Codes – what's the real risk?* NCSC. Available at: https://www.ncsc.gov.uk/blog-post/qr-codes-whats-real-risk
7. Grillo, A., Lentini, A., Querini, M., Italiano, G. F. (2010). High Capacity Colored Two Dimensional codes. *Proceedings of the International Multiconference on Computer Science and Information Technology*. Wisla, 709–716. https://doi.org/10.1109/imcsit.2010.5679869
8. Taveerad, N., Vongpradhip, S. (2015). Development of Color QR Code for Increasing Capacity. *2015 11th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS)*. Bangkok, 645–648. https://doi.org/10.1109/sitis.2015.42
9. You, F., Zhang, Q., Welt, B. (2019). Research on color matching model for color QR code. *Journal of Applied Packaging Research, 11 (3),* 57–68. Available at: https://repository.rit.edu/japr/vol11/iss3/5
10. Galiyawala, H. J., Pandya, K. H. (2014). To increase data capacity of QR code using multiplexing with color coding: An example of embedding speech signal in QR code. *2014 Annual IEEE India Conference (INDICON)*. Pune, 1–6. https://doi.org/10.1109/indicon.2014.7030441
11. *Understanding gamma correction*. Cambridge in Colour. Available at: https://www.cambridgeincolour.com/tutorials/gamma-correction.htm
12. Vidra, V., Pešička, O. (2023). *LCH vs OKLCH: what is the difference?* Atmos. Available at: https://atmos.style/blog/lch-vs-oklch
13. *OKLCH Color Picker & Converter*. OKLCH. Available at: https://oklch.com/#0.6,0.1032,210,100
14. *ZXing-Dart*. GitHub, Inc. Available at: https://github.com/shirne/zxing-dart
15. Fechan, A., Khoverko, Y., Daliavskyi, V., Dyhdalovych, T. (2024). Contactless dual-function sensors based on Si-cholesteric liquid crystal systems for optical identification. *Journal of Materials Science: Materials in Electronics, 35 (18)*. https://doi.org/10.1007/s10854-024-13005-5

✉***Oleksandr Kozyra,*** *Department of Software, Lviv Polytechnic National University, Lviv, Ukraine, e-mail: theenery@gmail.com, ORCID: https://orcid.org/0009-0006-5122-6298*

Andrii Fechan, *Doctor of Technical Sciences, Professor, Department of Software, Lviv Polytechnic National University, Lviv, Ukraine, ORCID: https://orcid.org/0000-0001-9970-5497*

Vladyslav Daliavskyi, *Assistant, Department of Software, Lviv Polytechnic National University, Lviv, Ukraine, ORCID: https://orcid.org/0000-0002-4059-1218*

✉*Corresponding author*