EURASIAN JOURNAL OF ACADEMIC RESEARCH
Innovative Academy Research Support Center
IF = 9.3 www.in-academy.uz/index.php/ejar

DETERMINING THE AVERAGE COLOR OF AN IMAGE USING C++ AND OpenCV

Botirov Muzaffarjon Mansurovich
Qoqon University, Department of Digital Technologies and Mathematics mbotirov@kokanduni.uz
botirovmuzaffarmansurov@gmail.com https://orcid.org/0000-0002-2078-1698
https://doi.org/10.5281/zenodo.18160071

ARTICLE INFO

Received: 29th December 2025
Accepted: 05th January 2026
Online: 06th January 2026

KEYWORDS

C++, OpenCV, image processing, matrix, average color, computer vision, pixel.

ABSTRACT

This article presents a method for determining the average color of an image using the C++ programming language and the OpenCV library. A digital image is represented as a pixel matrix, where each pixel consists of RGB values. The average color is calculated by averaging all pixel values of the image. The proposed approach is simple and computationally efficient, making it suitable for applications in computer vision, image processing, robotics, and intelligent systems. The paper provides the theoretical background, a sample C++ program, and an analysis of experimental results.

Introduction: As a result of the development of modern information technologies, the processes of image processing, their analysis and information extraction have become an important direction of scientific research. Any digital image is represented in the form of a matrix, each element of which embodies the value of a pixel. Each pixel consists of three main components - red (R), green (G) and blue (B) colors [1].

Based on the values of these components, the overall color tone of the image can be determined. Determining the average color of an image is widely used in visual systems, for example, in computer vision, robotics, medical diagnostics, multimedia applications, and artificial intelligence [4]. This method determines the overall color palette, background tone, or main dominant color of the image. Using the C++ programming language and the OpenCV library, such calculations can be performed quickly and efficiently [2]. Therefore, this article presents the theoretical foundations of calculating the average color of an image, practical program code, and analysis of the results [5].

Literature review: Research in the field of image processing and computer vision is one of the most relevant areas today. There are many scientific sources devoted to the theoretical foundations of working with digital images, analyzing their color properties, and implementing these processes through programming languages. In particular, the work “Digital Image Processing” by Gonzalez and Woods is recognized as the main fundamental source in this direction [3]. It provides detailed information on the conversion of images into digital form, the specific properties of the pixel matrix, and their analysis using mathematical methods.

The OpenCV library is one of the most widely used open source platforms for image processing, which is widely covered in the book “Learning OpenCV” by Bradsky and Kaehler [1]. This source presents methods for studying the main attributes of images, extracting color components, applying filters, and determining statistical indicators. In particular, for tasks such as determining the average color, the cv::mean() function plays an important role. The OpenCV documentation [2] also discusses in detail various methods for analyzing colors using pixel matrices, including histogram-based color calculation, average color detection, and image segmentation.

The book “Computer Vision: Algorithms and Applications” [4] by Szeliski is devoted to algorithms in the field of computer vision and their practical applications. It provides information on image processing, the study of statistical properties of color images, and the possibilities of their application in artificial intelligence. This resource explains not only the technical but also the theoretical foundations of the process of determining the average color. This work scientifically substantiates the effectiveness of considering images as matrices and performing mathematical operations on them.

Scientific research in this area is also being conducted in Uzbekistan. Muzaffar Batirov’s developments on the topic of "Image Processing Methods Using C++ and OpenCV" [5] show the possibilities of simplifying the process of image processing using software and applying them in the education system. In this research, it is emphasized that the methods of studying the color properties of images, color separation and analysis can be used in the educational process. This shows the connection between scientific practice and the educational process.

Several other international articles also cover the issue of determining the average color of an image. For example, this method has been used to check the color balance in natural image processing, and in the field of medical diagnostics to detect color changes in X-ray or MRI images. Some studies have studied the possibilities of classifying images based on average color and segmenting objects based on color. In particular, it is emphasized that it is possible to quickly filter images using color statistics and identify poor-quality or damaged images.

Many researchers have also widely used the technology of describing an image by its average color in artificial intelligence and machine learning systems. By studying color vectors, it is possible to train neural networks, determine the general palette of images, and use this information for various purposes. For example, in the advertising industry, the general color tone of images is used to predict the emotional impact on customers. Also, in the field of robotics, robots determine the color tone of a place based on the image taken from the camera and perform navigation or object recognition tasks.

The above analysis shows that although determining the average color of an image is scientifically simple, its practical application is wide. The advantage of this method is its speed of calculation and low resource requirements, as well as the ability to provide a general idea of the image. Therefore, currently, many scientific sources are covering various aspects of this method. Literature analysis shows that the C++ programming language and the OpenCV library are among the most effective tools for implementing this task, creating broad opportunities for scientific research [1–5].

Research methodology: Research methodology includes the stages of developing theoretical foundations, organizing practical experiments, and analyzing the results obtained.

Theoretical basis: If the image size is M×N, the color values of each pixel are (R_{ij}, G_{ij}, B_{ij}). The average color is determined using the following formulas:

$$R_{avg} = \frac{1}{M \cdot N} \sum_{i=1}^{M} \sum_{j=1}^{N} R_{ij} ,$$

$$G_{avg} = \frac{1}{M \cdot N} \sum_{i=1}^{M} \sum_{j=1}^{N} G_{ij} ,$$

$$B_{avg} = \frac{1}{M \cdot N} \sum_{i=1}^{M} \sum_{j=1}^{N} B_{ij} .$$

Practical experience: Various images (natural landscapes, portraits, artificially generated images) were selected and the mean() function was applied to them using the OpenCV library. The results were compared in BGR and RGB formats.

Practical part

```
#include <opencv2/opencv.hpp>
#include <iostream>
using namespace cv;
using namespace std;

int main() {
  Mat img = imread("rasm.jpg");
  if(img.empty()){
    cout << "Tasvir topilmadi!" << endl;
    return -1;
  }

  Scalar avgPixel = mean(img);
  cout << "O‘rtacha rang (BGR formatda): "
     << avgPixel[0] << ", "
     << avgPixel[1] << ", "
     << avgPixel[2] << endl;

  Mat avgColor(img.size(), img.type(),
         Scalar(avgPixel[0], avgPixel[1], avgPixel[2]));
  imshow("Asl tasvir", img);
  imshow("O‘rtacha rang", avgColor);
  waitKey(0);
  return 0;
}
```

Analysis and results: During the study, experiments were conducted on various images using the C++ programming language and the OpenCV library. These images included natural landscapes, cityscapes, human portraits, and artificially generated images. For each image, the average color values were calculated using the mean() function and compared in BGR and RGB formats. The results were summarized in a table.

Average color of images (in BGR and RGB formats)

Table 1

№	Image type	Average color (BGR)	Average color (RGB)	Note
1	Landscape	(90, 140, 180)	(180, 140, 90)	Blue and green dominate
2	Portrait	(120, 160, 200)	(200, 160, 120)	Value close to skin color
3	Artificial painting	(50, 100, 220)	(220, 100, 50)	Red is dominant.
4	Cityscape	(80, 130, 170)	(170, 130, 80)	Blue and gray tones

Here:

- **BGR** – in the standard OpenCV format (Blue, Green, Red).
- **RGB** – in the common color format (Red, Green, Blue).
- Each column value represents the average intensity of the pixels.

As can be seen from the table, the BGR and RGB values are interchangeable, that is, the result calculated by OpenCV and the result in the traditional RGB format differ only in the order of the components.

Analysis of the results. As can be seen from the table, the OpenCV library returns the result in BGR format. That is, the first value is blue, the second is green, and the third is the intensity of the red color. Therefore, to get the result in the traditional RGB format, it is necessary to swap the values. This is one of the important aspects of working with OpenCV.

In the landscape image (No. 1), blue (180) and green (140) colors predominate, which reflects the color tones characteristic of nature images. In the portrait image (No. 2), the average color values are close to skin color, and the red component (200) dominates. In the artificially created image (No. 3), red (220) stands out as the main dominant. In the cityscape (No. 4), blue (170) and gray tones define the background of the image.

Quantitative analysis. Quantitatively, the values of the three components in each image determine the overall appearance. For example, in a landscape photo, the values R=90, G=140, B=180 indicate that the average background consists of cold colors. In a portrait photo, the value R=200 confirms the dominance of the red component. These results allow you to automatically determine the overall color palette of the image.

Qualitative analysis. When evaluating qualitatively, the average color often represents the background of the image. For example, in portrait photos, if the background is light, the result shifts to bright colors. In cityscapes, blues and grays predominate, reflecting the atmospheric mood of the photo. These indicators create additional opportunities for visual analysis.

Practical significance

The practical significance of the results is that using this method:

- It is possible to quickly filter and classify photos.

• It is possible to obtain color attributes for images when training artificial intelligence models.

• In multimedia applications, the overall color palette of images can be determined and used for design.

• In robotics, the overall tone of an environment can be determined based on an image captured by a camera.

Limitations. The method also has some limitations. In multi-color images, the average color often reflects the general background, but does not show the internal details of the image. For example, in multi-object images, some small details may be overlooked. Therefore, in such cases, it is advisable to use additional algorithms (e.g., histogram analysis, segmentation).

Conclusions and recommendations: As a result of the research, it was found that the method for determining the average color of an image using the C++ programming language and the OpenCV library is simple, effective, and fast to calculate. The method was tested on various images, and the results showed that the average color often matches the background of the image and expresses the general mood of the image.

The advantages of this approach are its computational speed, low resource requirements, and the ability to return results in an understandable form. This allows it to be effectively used in the fields of computer vision, multimedia systems, robotics, artificial intelligence, and medical image processing.

Advantages and limitations of the method

Table 2

Key points	Description
Advantages	Fast, low resource requirements, easy programming.
Limitations	In complex images, details can be overlooked.
Usage	Computer vision, robotics, medicine, multimedia systems.

Recommendations

1. For scientific research: Better results can be achieved by combining the average color detection method with histogram analysis and segmentation methods.

2. For practical applications: It is recommended to expand the real-time capabilities and optimize the algorithm for mobile devices.

3. For the educational process: Using this method in programming laboratory classes will increase students' knowledge and skills in the field of computer vision.

4. For future research: By integrating the average color into deep learning models (Deep Learning), it can be used in more complex visual systems.

In conclusion, the average color detection method of an image using C++ and OpenCV is an effective tool that can be used in many areas, and the prospects for expanding its capabilities are very high.

References:

1. Bradski, G., & Kaehler, A. (2008). Learning OpenCV: Computer Vision with the OpenCV Library. O'Reilly Media.
2. OpenCV Documentation: https://docs.opencv.org/
3. Gonzalez, R. C., & Woods, R. E. (2018). Digital Image Processing. Pearson Education.
4. Szeliski, R. (2022). Computer Vision: Algorithms and Applications. Springer.
5. Muzaffar Batirov. (2025). Image Processing Methods Using C++ and OpenCV. Collection of Scientific Developments of Kokand University.