JISSE
ISSN: 2636-4425

Journal of International Society for Science and Engineering
Vol. 1, No. 1, XX-YY (2022)

JISSE
E-ISSN:2682-3438

# Solving Of Real-Color Digital Image Processing Problems by Converting Them to Their Corresponding Three Spectral Values to Enhancement the Remote Sensing Satellite Images Classification.

**Eng. Sherif M. Ibrahim,**
**Civil Engineering - Construction Training Institute - Kuwait.**

ARTICLE INFO
*Article history:*
*Received:*
*Accepted:*
*Online:*

Keywords:
Digital images processing
SPOT imagery
Multi-Spectral Images
Satellites Imagery
Remote Sensing

ABSTRACT
*Nowadays, more and more digital images are available in the real-color format. One of the sources of real-color digital images is the real-color digital cameras, which became inexpensive and of high resolutions. Another source is the use of traditional methods of photography and then scanning the photographic film on a digital scanner. Also, remote sensing includes acquiring many types of data such as visible, near-infrared, short-wave infrared images, etc. Many, satellite imagery is available now in real colors, like SPOT imagery. But dealing with these real-color images, from the digital image interpretation point of view, are so difficult, and needs expensive commercial software. Therefore, they are converted and handled with their corresponding spatial-spectral frequencies. Then, the treatment of the 256 color images will be the same as multi-spectral images. The way of converting the color images to their corresponding spectral frequencies is not available in some of the current software dealing with image processing and interpretations. An alternative algorithm has been established in this research, to solve the problem of separating each band of the real-color image alone. This algorithm is applied to a real-color SPOT image. The output results indicate that the developed algorithm is suitable for that kind of work.*

## 1. Introduction

Knowing that colors when mixed in varying portions will yield a perception of all other colors. The technology of cathode ray tubes (CRTs) uses additive primaries of red, green, and blue colors. They are called additive because one starts with an absence of light (black) and progressively adds various combinations of the three primary colors. In an additive system, red and blue combine to form magenta, blue and green combine to form cyan, and red and green combine to form yellow. All these three additive primaries combined together form white **[1].** With the fact that many parts around the world are still deserts, the values of space imagery data promise to be very high, for example; cartographic of buildings; roads; canal and drainage systems; rainfall calculation; ground-water investigation, and interpretation purposes. Also, space imagery has several characteristics that enable it to be very useful for small-scale mapping. Space imagery offers a real opportunity to speed up the progress programs of developing countries, especially in water quantity detection. These satellite imageries have characteristics that include uniformity of view over a wide area, near orthogonal (vertical) image, high geometric fidelity, the superior definition of certain natural features (like water or soil moisture content), and availability at relatively low cost. Also, real-color digital cameras became cheaper and have high resolutions. In this context, the use of real-color digital cameras become more applicable, than using the traditional methods of photography and scanning. Therefore, dealing with these real color images (from the interpreter's point of view) may be converted and handled with their corresponding spatial-spectral frequencies (i.e., red, green, and blue). Then, the treatment of the 256 color (gray level) images will be exactly the same as multi-spectral images. In other words, the produced three images can be used, in the same manner as multi-spectral images, for solving the problems connected with pattern recognition. The way of converting the color images into their corresponding spectral frequencies is not available in some of the current computer software dealing with image processing and interpretations, such as paintbrush, photo editor, EDRISSI, ERDAS imagine, PCI Geomatica ...etc. On the other hand, most of the remaining programs, which are capable of converting real-color images to 256 color (gray level) images, make a worse compilation. Such software processing depends upon taking the average of the frequencies of the three primary color bands, to represent a 256-color (gray level) image. This averaging process will cause significant distortion of some of the existing features of the image. For example, if the pixel frequency is equal to (100, 50, and150) or (150, 100, and 50) in red, blue, and green bands respectively, the computed frequency will be 100. Therefore, an alternative algorithm has been established by the author, to solve the problem of separating each band of the real-color image alone, that is, separating each one of the three primary colors: Red, Green, and Blue, on its own. Also, a computer program for separating red, green, and blue frequencies in a separate file called RGB will be established. The main characteristics of the program RGB, including its input data, program algorithm steps, and flow charts as well as the output results, and running time, will be summarized. Since the problems connected with converting the real-color image to its three spectral values involve certain expressions and associated manipulations, it was found beneficial, before going through the developed algorithms and computer programs, to discuss the digital image structure connected with automatic conversion. Then, the development of computer routines will be presented. After this, a description of the available digital data will be given. Finally, the conclusion of the present investigation will be outlined.

## 2. DIGITAL IMAGE STRUCTURE OF SPACE IMAGERY

A digital image is a function f (x, y) which has been described in both spatial coordinates and brightness. The digital image may be considered as a matrix whose row and column indices identify a point in the image, and the corresponding matrix element value identifies the gray level at the point. The elements of such a digital array are called image elements, pixels **[2], [3] and [5].**

Digital imagery is obtained either directly when digital cameras are used or indirectly when a scanner is employed to transform conventional prints or diapositives into a digital format. In both cases, the digitizing devices have to be calibrated to ensure correct geometry. A rigorous calibration method for digital cameras is given by Chen and Schenk **[4]** and a scanner calibration procedure is provided by Sarjakoski **[14]**. After calibration, the image becomes free of systematic distortions. Any image may be thought of as consisting of tiny equal areas, or picture elements, arranged in regular lines and columns **[6]**. The position of any picture element, or pixel, is determined by an X and Y coordinate system with the origin at the upper left corner for all imagery systems existing at present. The brightness of each pixel has a numerical value ranging from zero for black to some higher number for white. Any image can now be described in strictly numerical terms on a three, dimensional coordinate system (X, Y, Z), with X and Y locating each pixel, and Z is given the grayscale intensity value. An image may be recorded originally in this digital format. An image recorded initially on photographic film may be converted into a numerical format, by a process known as digitization **[15].** On the other hand, the digital image data can be converted into hard-copy images by film writer that operates in reverse fashion to digitizer **[5].** The computer file, that deals with the image having these characteristics, is called a raster file. This raster file must have additional information (like file type, file size, file width, file height, etc.) besides the actual image data. In this context, the raster file internally will be divided into two parts. The first part of the file contains the information attached to define the image perfectly (raster file header). The second part of the file deals with the actual image data (image matrix). The richest and most complete raster file format is the BITMAP raster file format. So, one of the purposes of this research will be the interpretation and understanding of the BITMAP-Raster file format and specifications.

## 3. BITMAP RASTER IMAGE FORMAT FOR REAL-COLOR IMAGES AND 256 COLOR (GRAY LEVEL) IMAGES:

Knowing the fact that, the BITMAP files are raster files comes from the direct output of scanning aerial photos on the scanner, as well as, space imagery captured from satellites by digital cameras **e.g., [9], [11] and [12]**. Hence, the BITMAP files (monochromatic, 256 colors or 256 gray levels or real color) will represent the raster files utilized throughout the present research. Consequently, it can be visualized that, the knowledge and careful analysis of the BITMAP information, represents the essential backbone of the subsequently developed algorithm, which is considered to be the ultimate objective of the present work. This requires the collection of all possible definitions and description connected with the BITMAP information understanding, which have been achieved directly through an Internet search. On the other hand, other definitions and interpretations of the remaining elements of the BITMAP matrix have been performed by the author, through the establishment of more than 100 raster files (BITMAP files), with different specifications. Different combinations of comparisons among those established files yield to a thoughtful understanding of the BITMAP specifications **[7], [8], [10] and [13]**. The BITMAP file has the following important specifications:

- Binary format.
- Header, its first part consists of 54 bytes in which it describes the image characteristics.
- After the header, the degree of brightness of pixels are written row by row starting from the first row at the lower left corner of the image.
- The number of columns must be an integer number multiple of 4 bytes, which means that, for example, if the actual number of columns is equal to 101 (in 256 colors (gray level) BITMAP), then the number of bytes will be equal to 104 in each row.
- No separation between the pixels as well as the rows in the file.
- The degree of brightness of each pixel is equal to the ASCII value of the byte in the pixel position.

The above are some of the BITMAP file characteristics that are very important to any programmer. The header of the BITMAP file will be described in detail in TABLE I. The data structure of the BITMAP file header contains information about the type, size, and layout. It starts from the beginning of the file using the first 14 bytes. Immediately after the first 14 bytes, the BITMAP information structure fully defines the dimensions and color information, using the next 40 bytes. The BITMAP information data structure combines the BITMAP information header structure and a color table, to provide a complete definition of the dimensions and colors. The RGB data structure describes a color consisting of relative intensities of red, green, and blue. The Color field of the BITMAP information data structure consists of an array of RGB data structures. The color field structure contains the following fields:

BYTE (RGB-Blue); BYTE (RGB-Green); BYTE (RGB-Red); BYTE (RGB-Reserved).

| | |
|---|---|
| RGB-Blue | define the frequency of blue color. |
| RGB-Green | define the frequency of green color. |
| RGB-Red | define the frequency of red color. |
| RGB-Reserved | Is not used and must be set to zero. |

It is worth noting that in a monochromatic (black and white) BITMAP file, the total file header will equal 54 bytes + 2*4 (8 bytes) for the monochromatic colors in the image, which equal 62 bytes. In 256 color or 256 gray level BITMAP file, the total file header will equal 54 bytes + 256*4 (1024 byte) for the 256 colors or gray levels in the image, which equals 1078 bytes. In a real color BITMAP file, the total file header will equal 54 bytes + 0 bytes for the real colors in the image, which is equal to 54 bytes.

TABLE I: THE BITMAP FILE HEADER FORMAT

| Byte No. | Abbreviation | Description |
|---|---|---|
| 1,2 | BfType | Specifies the type of bitmap file. It must be BM (ASCII value =19778). |
| 3,4,5,6 | BfSize | Specifies the size of the file |
| 7,8 | BfReserved1 | Is reserved and must be set to zero. |
| 9,10 | BfReserved2 | Is reserved and must be set to zero. |
| 11,12,13,14 | BfOffBits | Specifies in bytes the offset from the BITMAP file header to the actual BITMAP data in the file. |
| 15,16,17,18 | BiSize | Specifies the number of bytes required by the BITMAP information header structure. |
| 19,20,21,22 | BiWidth | Specifies the width of the BITMAP in pixels. |
| 23, 24, 25, 26 | BiHeight | Specifies the height of the BITMAP in pixels. |
| 27, 28 | BiPlanes | Specifies the number of planes for the target device and must be set to 1, in case of dealing with the raw data BITMAP before any processing. |
| 29, 30 | BiBitCount | Specifies the number of bits per pixel. This value must be 1, 4, 8, or 24. |
| 31, 32, 33, 34 | BiCompression | Specifies the method of compression. |
| 35, 36, 37, 38 | BiSizeImage | Specifies the size in bytes of the image. |
| 39, 40, 41, 42 | BiXPelsPerMeter | Specifies the horizontal resolution in pixels per meter. |
| 43, 44, 45, 46 | BiYPelsPerMeter | Specifies the vertical resolution in pixels per meter. |
| 47, 48, 49, 50 | BiClrUsed | Specifies the number of color indices in the color table actually used by the BITMAP. If this value is 0, the BITMAP uses the maximum number of colors corresponding to the value of the biBitCount field. The biClrUsed field must be set to 0 or to the actual size of the color table. |
| 51, 52, 53, 54 | BiClrImportant | Specifies the number of color indices that are considered important for displaying the BITMAP. If this value is 0, then all colors are important. |

## 4. DEVELOPED COMPUTER ALGORITHM NEEDED FOR THE SOLUTION OF THE PROBLEM IN HAND:

This section is devoted to the brief description of the development program RGB, mentioned above for solving the problem of converting the real-color image to its corresponding three spectral values. Accordingly, computer software (called RGB), has been established by the author, to solve the problem. toolbar.

### *4.1. Input Data*

The input data includes the following two files:

- The real-color image file (*.bmp)

- ASCII file go.bat in the following format, which includes one line as follows:

rgb.exe real-color file name red-band file name green-band file name blue-band file name. Styles named

### *4.2. Program Algorithm Steps and Flow-Chart:*

The basic flowchart of program RGB is depicted in figure (1), while the main program steps will be summarized below:

1- Read the names of the input file (real-color image) and the three output files.

2- Open file (real-color.bmp) for input.

3- Read the first part of the header (14 bytes) from the input file.

4- Compute the value of the first and second byte, to ensure that the image is a BITMAP image (the ASCII value must equal to 19778).

5- Read the second part of the header (next 40 bytes) from the input file.

6- Computed the new values of each byte (first 54 bytes) in the 256-color BITMAP header, as mentioned before in section (3).

7- Compute the values of the array of colors (gray levels) for the 256-color BITMAP file.

8- Correct the image width.

9- Compute the size of the 256-colors (gray level) BITMAP file.

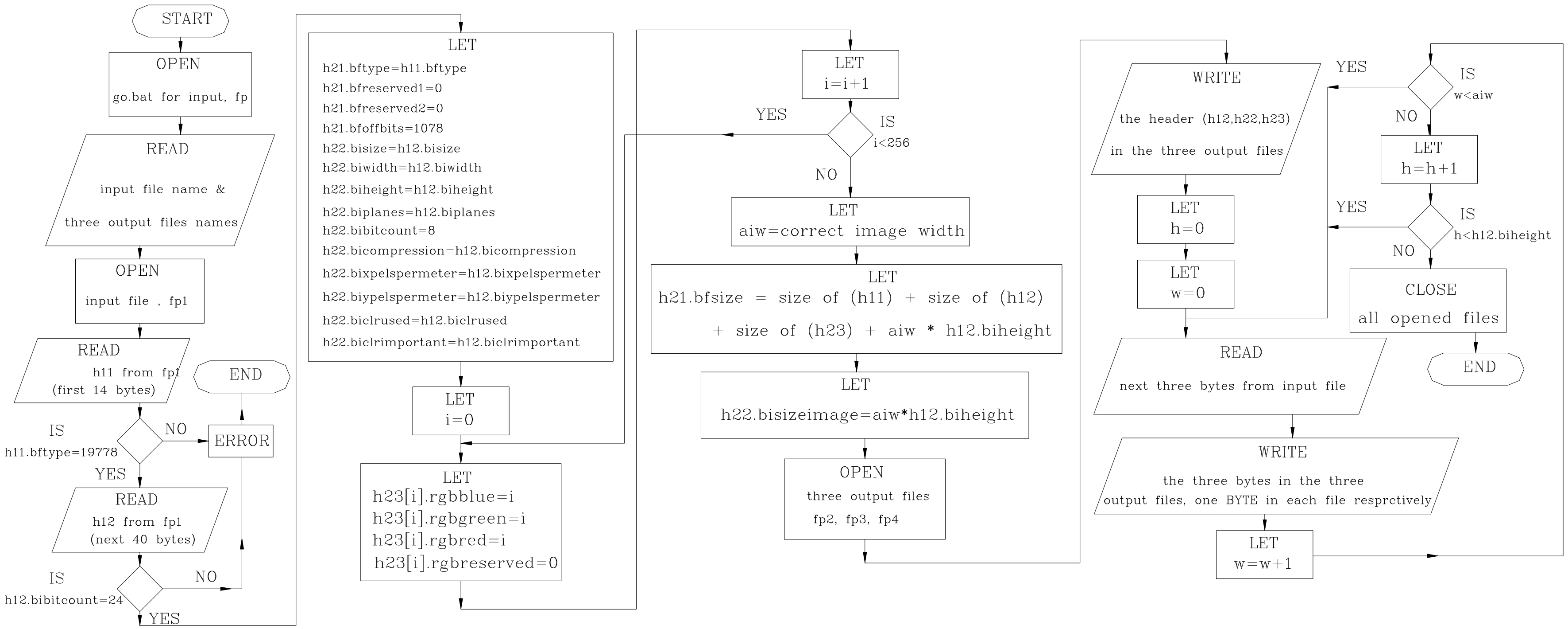


Figure (1) FLOW-CHART OF CONVERTING REAL-COLOUR IMAGE TO ITS CORRESPONDING VALUES

10- Open files (red-band file, green-band file, and blue-band file) for output.

11- Write the header of the output files (256 color (gray level) BITMAP header), which contains: the first part of the computed header (54 bytes) plus the array of colors (gray level) (next 1024 bytes).

12- Scan the input file systematically after the header, until the end of the file (byte by byte), and write in the output files (in the sequence of red, green, and blue) the value of this byte.

13- Close all opened files.

*4.3. Output Results:*

The output of RGB program will be three images 256 color (gray level) BITMAP files as follows:
- The first file is for red-band.
- The second file is for green-band.
- The third file is for the blue-band.

*4.4. Running Time*

With a Pentium 233 computer having 64 MB RAM, for BITMAP real-color file with dimensions of 1000 pixels * 1000 pixels the program running time takes 4 seconds. However, when a Pentium-4 1700 computer has 256 MB RAM, the running time takes less than one second. This time will be linearly proportional to the file dimension size.

## 5. USED DATA AND RESULTS

The used digital data in the present paper is a real-color image file showing a part of Great Cairo in Egypt (figure 2). After applying the previously discussed algorithm to this real-color digital image, three images are obtained, each of which is written with a band of the three spectral bands (red, green, and blue), see figures 3, 4, and 5.

## 6. Conclusions

The obtained results indicate that the RGB computer algorithm presented in this research has very good reliability and resolution in solving the real-color problem efficiency through conversion to the three spectral bands. Therefore, it is recommended to be efficiently used for the solution of the real-color problem in the field of digital image interpretation as the most trustable and reliable up-to-date local software in the subject matter, for mapping applications, and image classification.

**Conflict of Interest**

The authors declare no conflict of interest.

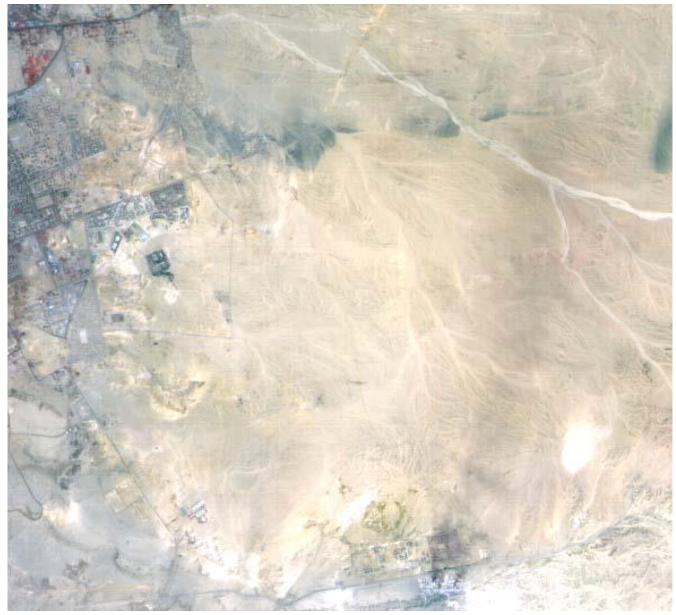

**Figure (2) Real-Color Image Show a Part of Great Cairo Captured by SPOT Satellite**

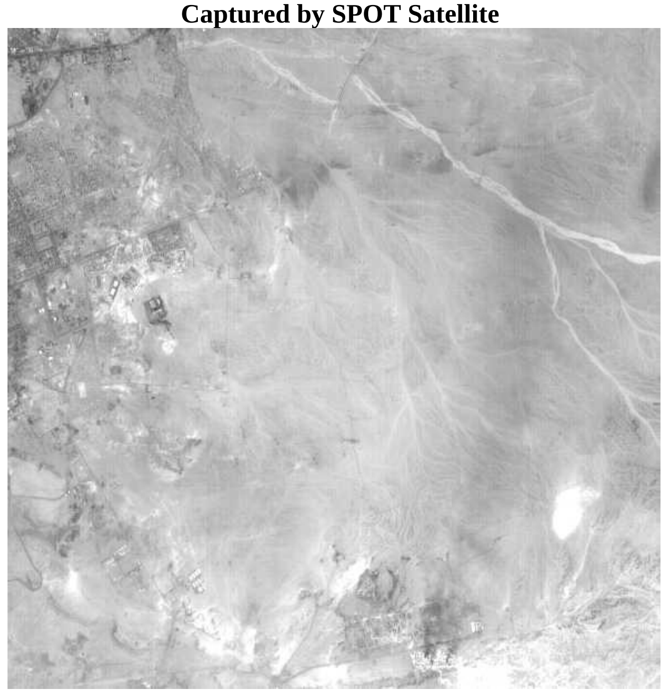

**Figure (3) 256 Color (Gray Level) Image in Blue Band After Converting the Real-Color Image to Its Corresponding Three Spectral Bands.**

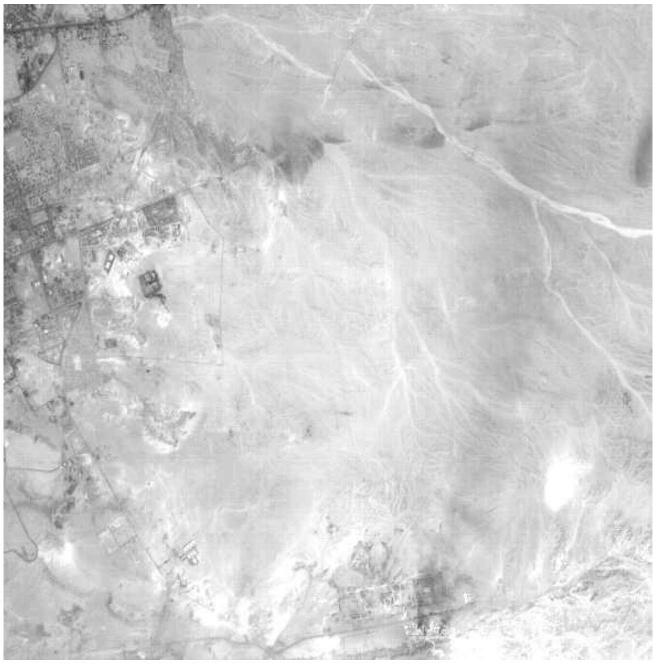

**Figure (4) 256 Color (Gray Level) Image in Green Band After Converting the Real-Color Image to Its Corresponding Three Spectral Bands.**

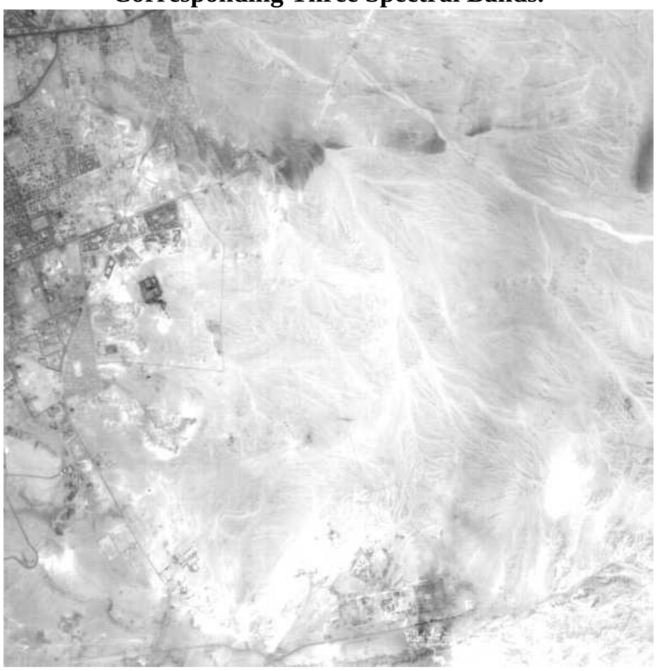

**Figure (5) 256 Color (Gray Level) Image in Red Band After Converting the Real-Color Image To Its Corresponding Three Spectral Bands.**

**References**


[1] Abdel-Gawad, A. K., 2000, "Development of Special Software Package for Automatic Interpretation of Space Imagery and Its Applications for Some Environmental Studies", Ph.D. Thesis, Faculty of Engineering, Ain Shams University, Cairo, Egypt.

[2] Arias, A. B., Cheddad, A., Perez J., C., J., and -Garcia, A., R. (2021) “Digital Image Processing and Development of Machine Learning Models for the Discrimination of Corneal Pathology: An Experimental Model”. Publisher’s Note: MDPI stays neutral with regard to jurisdictional, claims in published maps and institutional affiliations. Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (http://creativecommons.org/licenses/by/4.0/). https://doi.org/10.3390/photonics8040118.

[3] He, J. (2022).” Multimedia Vision Improvement and Simulation in Consideration of Virtual Reality Reconstruction Algorithms”. Hindawi Journal of Electrical and Computer Engineering Volume 2022, Article ID 4968588, 10 pages. https://doi.org/10.1155/2022/4968588.

[4] Chen, Y. and T. Schenk, 1992 "A Rigorous Calibration Method for Digital Camera", The International Archives of photogrammetry and Remote Sensing, Vol.XXIX, part B3, ISPRS Commission III, Washington, U.S.A.

[5] Gonzalez, R. C. and P. Wintz, 1977, "Digital Image Processing", Addision Wesley Publishing Company, London, U.K.

[6] Floyd F. Sabins, Jr., 1978, "Remote Sensing (Principles and Interpretation)", W. H. Freeman and Company, San Francisco, U.S.A.

[7] Jordain, R., 1986, "Programmer's Problem Solver", Prentice Hall Press, U.S.A.M. Uzunoglu, M. S. Alam, “Dynamic modeling, design, and simulation of a combined PEM fuel cell and ultracapacitor system for stand-alone residential applications” IEEE Trans. Ener. Conv., **21**(3), 767–775, 2006. https://doi.org/10.1109/TEC.2006.875468.

[8] Kumar, A. N. (2022). “Learning Programming by Solving Problems”. Conference Paper in IFIP Advances in Information and Communication Technology · January 2002.

[9] Junior, C., P., Cavallieri, G., V., da Silva, F., A., Fernandes, G., L., & Nai, G., A, Salge, A., K., M., Puhle, J., G., Silva, D., T., d., Pereira, D., R., & Mello, Fa., d., Favareto, A., P., A., and Rossi, R., C. (2021). “Digital image processing: a useful tool in the analysis of lung injuries caused by chronic inhalation of agricultural herbicides”. Environmental Science and Pollution Research (2021) 28:57918–57924. https://doi.org/10.1007/s11356-021-14692-x

[10] Lafore, R., 1984,"Assembly Language Primer for The IMB PC and XT", New American Libarary, U.S.A.

[11] Nassar, M., I. Shaker, and A. Ragab, 1997, "Accuracy Assessment of Calcomp 9000 Digitizer and FSS Scanner in The Process of Producing Digital Maps", Proceedings of Al-Azhar, Fifth Engineering International Conference (AEIC), held in Cairo, Dec. 19-22, Egypt.

[12] Ragab, A. F. M., 1996, "A Comparative Study Among Various Sources of Data Within Different Formats for The Production of Digital Mapping", M.Sc. Thesis, Faculty of Engineering, Ain Shams University, Cairo, Egypt.

[13] Saliby. G. Joe. S., (2019). “Design & Implementation of Digital Image Transformation Algorithms”. International Journal of Trend in Scientific Research and Development (IJTSRD) - Volume: 3 | Issue: 3 | Mar-Apr 2019 Available Online: www.ijtsrd.com e-ISSN: 2456 - 6470.

[14] Sarjakoski, T.T., 1992, "Suitability of the Sharp JX-600 Desktop Scanner for the Digitization of Aerial Color Photographs", The International Archives of photogrammetry and Remote Sensing, Vol.XXIX, part B3, ISPRS Commission III, Washington, U.S.A.

[15] Shaker, I.F., 1988, "Beitrage Zur Anwendung der Digitalen Bildverarbeitung in der Nahphotogrammetrie", Ph.D. Thesis, Technical University, Berlin, Germany.

**Abbreviation and symbols**

| | |
|---|---|
| **RGB** | **Red, Green, and Blue** |
| **BM** | **Bitmap** |