Development of a System Based on a Graphical User Interface Using MATLAB Tools

Ahmed A. Thabit
Al-Rafidain University College
Baghdad, Iraq
ahmed.abdelredha.elc@ruc.edu.iq

Enas Alzidi
Al-Turath University College
Baghdad, Iraq
inas.fadel@turath.edu.iq

Mohammed Abdulkreem Mohammed
Al-Noor University College
Nineveh, Iraq
mohammed.abdulkreem@alnoor.edu.iq

Petro Palchyk
Kyiv National University of Construction and Architecture
Kyiv, Ukraine
pppalchik@knuba.edu.ua

Abstract – **Background: Image processing is crucial in different scientific fields and is at the core of digital technology owing to its diverse uses. Due to the extensive usage of digital image processing, user-friendly, efficient picture processing and editing software is needed. Due to the popularity of commercial software like Microsoft's Digital Image Suite and Adobe Photoshop, similar alternatives have emerged.**

Objective: This project aims to create a robust and adaptable picture editor utilising MATLAB capabilities with an intuitive GUI for users of varied competence levels. The article seeks to provide a viable alternative to commercial image processing software that is simple to use and has many features.

Methods: MATLAB's image processing tools and unique methodologies created an image editor with extensive capabilities. The editor's development concentrated on supporting RGB, monochrome, and binary image formats and allowing users to manipulate them. The software's layout, design, message, usability, and cultural appropriateness were considered to improve user experience.

Results: The image editor effectively enhances digital picture quality for medical image restoration and satellite image reconstruction. The program is user-friendly due to its minimal download size, easy setup, and straightforward UI. It shows how digital technology evolves, offering a flexible tool for beginners and experts.

Conclusion: This study successfully developed a user-friendly MATLAB-based picture editor for varied digital image processing demands. The tool improves data analysis success and accuracy in engineering, research, and finance due to its simplicity and powerful features.

INTRODUCTION

Applications in many different areas have led to the widespread adoption of digital image processing in today's technological landscape. Information can be extracted, picture clarity can be enhanced, and data can be visualised in novel ways thanks to our ability to alter and enhance digital photos. There is a growing requirement for simple, effective software that can handle picture processing and offer altering options. Due to their intuitive UI and powerful picture editing capabilities, commercial products like Microsoft's Digital Image Suite, Adobe Photoshop, and Microsoft's Paint have become immensely popular among consumers. As such, this piece aims to create a MATLAB picture viewer with comparable capabilities to commercial offerings [1].

A graphical user interface (GUI) for viewing pictures is necessary when developing an image viewer. Images and the algorithms that modify them need to be displayed and managed within this structure. Constructing a solid infrastructure supporting these algorithms is a prerequisite to studying how they operate. A MATLAB-based image viewer is made possible by combining the features of the MATLAB image processing tools with some unique methods. MATLAB's matrix-based data format is particularly well-suited for picture manipulation, making it an excellent tool for digital image processing [2].

Three standard picture formats end users encounter RGB, monochrome, and binary. The editor's ability to alter images depends on familiarity with the specific editing methods required for each picture form. The image viewer created for this research allows users to perform fundamental manipulations on pictures, such as modifying their hue and saturation and scaling and turning them. In order to provide its users with a more complete picture-altering tool, it also incorporates some novel editing actions. These programs consider picture clarity, processing time, and impact, making them practical and user-friendly [3].

The development of image-processing-based photo editing software provides users with practical, user-friendly tools that are conscious of every aspect, from image clarity and processing time to effects. The suggested image editor is an alternative to popular commercial programs like Microsoft's Digital Image Suite and Adobe Photoshop while providing full features and capabilities [4].

The image editor is built with MATLAB's image processing toolkit and uses secret methods to deliver a unique graphical user interface. A user-friendly interface must be developed to ensure that users can easily access the image processing tools and that the tools are culturally appropriate [5], [6]. Layout, images, message, information, and societal suitability are all areas that need work to make this processing tool more user-friendly. The main objective of this research is to create a robust, user-friendly picture-altering utility using MATLAB image processing tools.

A. The Aim of the Article

The article aims to develop a MATLAB user interface that is intuitive and easy to use so that sophisticated computations and data analysis may be carried out. By offering a platform that is visually intuitive, capable of doing complicated computations, and capable of producing graphical outputs, the objective is to ease the data analysis procedure and increase its accessibility to those who are not data analysis professionals. Users can enter data, choose settings, and perform analyses using the system with just a basic understanding of coding. However, the system will provide sophisticated features and adaptability for those with more expertise. Ultimately, this project aims to enhance the success and accuracy of data analysis in various fields, such as engineering, science, and financial services, by offering a robust and easily accessible tool for conducting complex calculations and visualising data. These fields include engineering, science, and finance.

B. Problem Statement

MATLAB is a vital data modelling and analysis tool, but many customers need more technical skills and expertise to utilise it successfully. Hence, data analysis and modelling processes may be laborious and prone to inaccuracy, reducing the quality and efficacy of the final product. Nevertheless, with a clean and straightforward interface, even proficient MATLAB users can handle massive volumes of data and complicated computations. To make data analysis more approachable for non-experts and more productive for professionals, a graphical user interface (GUI) that streamlines the process is required. To overcome these obstacles, this project will create a graphical user interface (GUI) using MATLAB's capabilities to analyse and model data. The suggested system will allow for simple data entry, parameter specification, and analysis execution while giving more sophisticated options for power users. The end goal of this work is to make MATLAB, a program used for data analysis and modelling, more user-friendly and effective for everyone and to increase the efficiency with which MATLAB experts go about their work.

METHODOLOGY

This article details the creation of a MATLAB image editor that offers many of the same capabilities as industry standard programs like Microsoft's Digital Image Suite, Adobe's Photoshop, and the venerable old standby, Microsoft Paint. Our goal is to create a unique GUI-based system by integrating the capabilities of the MATLAB Image Processing Toolbox with our unique methods [7].

For starters, let us agree that noise is anything that does not belong in an image and thus makes it less transparent and more difficult to decipher. We classify noise as "independent" or "based on the image input data," with the former being the more common. Here, we distinguish between three distinct noises: Gaussian, salt-and-pepper, and speckle. The spatial filtering methods we present can also deal with these various noise sources. Using a spatial filter, you can eliminate the Gaussian noise in an image by averaging the pixel values locally. The average filter can eliminate the salt-and-pepper noise by averaging all the neighbouring pixels. A low-pass filter can eliminate the speckle noise [8], [9].

Next, we discuss adjusting images' luminance, contrast, resolution, pixel count, and size [10], [11]. Contrast is the difference in luminance or colour between an object and its background, while brightness is the subjective perception of an object's luminance. Pixel count measures how many individual picture elements make up the whole, while image resolution measures how many individual picture elements there are. If you resize an image, you can change its overall dimensions while keeping its pixel count and file size unchanged. We advise editing images before resizing and offer instructions on how to do so [12], [13].

We also detail the various colour models, cropping tools, and rotation capabilities for editing images. Cropping eliminates unwanted parts of a photo or illustration, while rotation involves spinning the image in either the clockwise or counterclockwise direction. Here, we will go over the RGB, CMYK, and HSV colour spaces, all of which see regular use in digital image processing. The additive RGB colour model is the most common, while the subtractive CMYK model is widely used in printing [14]. Hue, Saturation, and Value (HSV) is a colour model with three channels rather than just two, and it avoids using primary colours directly. We conclude with a discussion of the YIQ colour model, the most popular model for TV transmission, because of its ability to accurately convey both luminance and chrominance [15].

Using the capabilities of the MATLAB Image Processing Toolbox and our unique methods, we intend to create a user-friendly, specialised GUI-based system. We discuss a wide range of spatial filtering techniques and image editing tools that can deal with various forms of noise. We also detail the various image attributes like luminance [16], [17], contrast, resolution, pixel count, and resizing and review the various colour models typically employed in digital image processing. Insights from this article can be used as a springboard for further research into creating an advanced image editor in MATLAB, complete with flexible editing options and high-powered, user-friendly interface elements as explained in the following equations:

1. Gaussian noise: The Gaussian distribution is given by the equation:

$$f(x) = (1/\sigma\sqrt{(2\pi)}) * e^{(-(x-\mu)^2/(2\sigma^2))}, \quad (1)$$

where f(x) is the probability density function, σ is the standard deviation, and μ is the mean.

2. Spatial filtering: Spatial filtering involves convolving the image with a filter kernel. The output pixel value is calculated as:

$$g(x,y) = \sum\sum h(i,j)f(x-i,y-j), \quad (2)$$

where g(x,y) is the output image, f(x,y) is the input image, h(i,j) is the filter kernel, and the summation is taken over all values of i and j that correspond to the filter kernel.

3. Contrast adjustment: The contrast of an image can be adjusted using the following equation:

$$g(x,y) = (f(x,y) - c) * a + c, \quad (3)$$

where g(x,y) is the output image, f(x,y) is the input image, c is a constant offset, and a is a scaling factor.

4. Resizing: The resizing of an image can be done using the following equation:

$$g(x,y) = f(xs,ys), \quad (4)$$

where g(x,y) is the output image, f(x,y) is the input image, and s is the scaling factor.

5. RGB to grayscale conversion: To convert an RGB image to grayscale, we can use the following equation:

$$g(x,y)=0.2989*R(x,y)+0.5870*G(x,y)+0.1140*B(x,y), \quad (5)$$

where R(x,y), G(x,y), and B(x,y) are the red, green, and blue colour channels, respectively. The coefficients 0.2989, 0.5870, and 0.1140 reflect the human eye's sensitivity to different colours.

A. The Interaction of Noise with the Changes Picture

We define noise as an unwanted image component that damages the image and makes it harder to interpret. There are numerous reasons why photos contain random noise. In general, noise can be divided into two classes: independent noise and noise that depends on the picture input data. White noise, or additive noise [18], is typically randomly dispersed over the frequency range, whereas low-frequency information makes up most of an image. As a result, noise dominates at high frequencies, and low-pass filters of some forms are used to dampen their impacts. A spatial filter or a frequency filter can be used for this.

Gaussian Noise. In essence, adding a value from a zero-mean with a Gaussian distribution to each image pixel creates Gaussian noise by locally averaging the values of the image's pixel values; such noise can be eliminated thanks to the zero-mean-property of the Gaussian function [6]. Additive white-Gaussian noise is an example of noise. At the same time, the original picture values of the noisy image pixels followed the Gaussian curve. We have a spatial filter that can reduce the effect of the Gaussian noise [19]. As the fine-scaled features of interdependent parts block high frequencies in a noisy picture, it has been shown that smoothing the image decreases both the noise and the fine-scaled features.

Salt-and-pepper Noise. Another prominent source of disturbance is noise caused by data corruption (popular names include intensity spikes, speckle, and salt and pepper noise.). The commotion here is due to a breakdown in data transmission [20]. The damaged pixels, resembling snow, have either switched a single bit or are all set to the highest value. The picture may seem salty and peppered if a simple pixel is toggled between zero and the maximum value. There is no change to the state of unscathed pixels. The number of faulty pixels is a standard indicator of noise levels.

Speckle Noise. Speckle noise is known as multiplicative noise because it introduces the same amount of noise into the image as an increase in the signal's power. It is called spatially dependent [21], non-Gaussian, and signal-dependent. Little differences in surface finish within a single frame make the received picture susceptible to random oscillations in phase and amplitude. Distinctly different from the weak intensity generated by destructive interface phase fluctuations, vigorous intensity is generated by constructive ones. This fluctuation in noise level is known as "speckle-noise"[22].

B. Modified Image Noise Reduction Using Spatial Filters

Gaussian Filter. Gaussian smoothing is based on convolutional approaches to generate the "point-spread" function of a 2-dimensional distribution. Keeping in mind that the picture is a collection of individual pixels. First, a discrete approximation of the Gaussian function must be constructed, after which the convolution techniques may be used. As the Gaussian distribution in our study approaches zero at distances greater than about three standard deviations [19] from the average value of the image pixels, we can now stop the kernel from becoming indefinitely more extensive.

Average Filter. An average, linear-class windowed filter results in the smoothest picture. In this case, low-pass-1 is used as the filter setting. The core idea behind this filter is to take an average value for each picture component over all neighbouring pixels.

Brightness. According to the visual perception of light, a source appears to be emitting or reflecting light when it is bright [1]. In other words, brightness is the perception a visual target's luminance elicits. The context of the viewing environment is essential because the perception is not linear to luminance (for example, consider White's illusion) [23].

One of the colour appearance characteristics of various colour appearance models, brightness is a subjective perception of an object being examined. It is commonly expressed by the amount of light that anything appears to emit, referred to as its brightness. This is not the same as the impression of lightness, which is how light something seems compared to a white item that has been equally lighted [19].

Bright is a Middle English adjective derived from an Old English noun with the same meaning, brought. The word comes from the Common Germanic *berhtaz root, which ultimately derives from the PIE root *bhereg, which means "white, dazzling." Previously, "brightness" was used interchangeably with radiometric radiance and photometric luminance. The term "brightness" should only be used for non-quantitative allusions to physiological feelings and perceptions of light, according to the US Federal Glossary of Telecommunication Terms (FS-1037C).

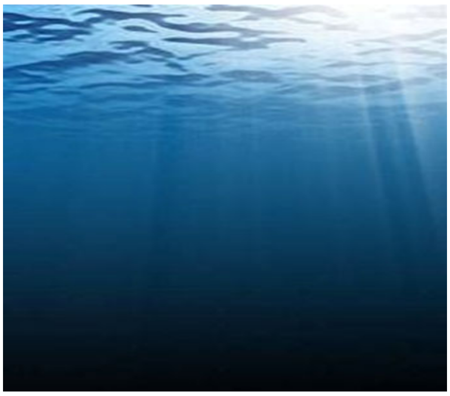


Fig. 1. Decreasing brightness with depth

Contrast. Contrast, or a difference in brightness or colour, allows us to tell an actual thing apart from its representation in a picture or display. The contrast between an item and other objects in the same perspective, in terms of colour and brightness, determines the contrast seen in the actual world. Since humans are more sensitive to contrasts than absolute brightness, we can consistently see the world despite the significant changes in lighting that occur during the day and from place to place. Maximum contrast in a picture may be quantified using the contrast ratio [13], [15]. How much detail is captured in a picture is dependent on its resolution.

Fig. 2. The contrast in the left half of the image is lower than that in the right half.

Image resolution. This term encompasses digital photographs, film images, and other types of pictures. More resolution means sharper images. Image resolution may be evaluated in a variety of ways. Resolution indicates how tight two lines may be together without losing their identities. Measurements, overall picture size, or angular subtense are all potential sources of resolution units. Instead of single lines, line pairs, comprised of a dark line and a bright line next to it, are commonly used. There are two distinct styles of line: those that are dark and those that are light. With a resolution of ten lines per millimetre, there are five sets of two lines, five dark and five light, that alternate every millimetre. Line pairs per millimetre [24] is the standard measurement for the resolution of photography lenses and films.

Pixel count. While "resolution" is often used interchangeably with "pixel count" in digital imaging, international standards in the field of digital cameras specify that "Number of Total Pixels" should be used when discussing image sensors and "Number of Recorded Pixels" should be used when discussing what is wholly captured. Hence, CIPA DCG-001 calls for a note like "Number of Recorded Pixels: 1000, 1500" [19].

The "Number of Effective Pixels" refers exclusively to the pixels sensors that make up the final picture, as opposed to the "Number of Total Pixels," which also counts the useless or light-shielded pixels around the perimeter of an image sensor or digital camera.

An image of N pixels in height and M pixels in width may be shown at any resolution with a minimum of N line per picture height (or N TV lines). When referring to pixel counts as "resolution," it is customary to use a pair of two positive integer numbers to express the pixel resolution, the 1st of which refers [7], [25] to the number of pixel columns and the 2nd of which is the number of pixel rows; for instance, 7680 6876. Resolution may also be defined as the total number of pixels in a picture, which is often stated in terms of megapixels and can be calculated by multiplying the number of pixels in a row by the number of pixels in a column and then dividing the result by one million. Other terminology norms include "pixels per inch" or "pixels per square inch" for describing pixels per length unit or pixels per area unit, respectively.

These resolutions are not accurate but are used as top bounds for images. An illustration of how the same picture may seem at different pixel resolutions if the pixels were not being represented as crisp squares.

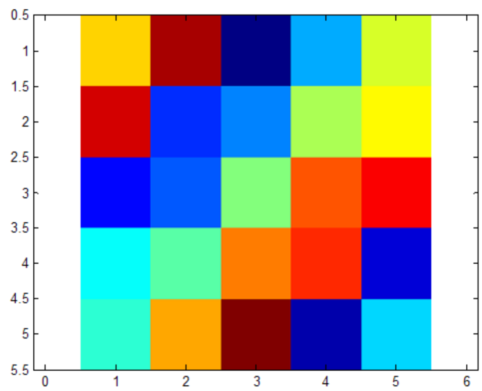


Fig. 3. MATLAB represents Pixel colours.

Resize image. Physically, pixels can have any size. The number of pixels in an image may change as it is scaled, which results in a resampling of the image. The quantity of pixels per inch is referred to as print resolution. Smaller printed pixels and a smaller printed image result in better print resolution.

Reduced print resolution leads to bigger printed pixels and a more expansive printed picture.

- Modify the size and resolution of printed copies without altering the original file size or number of pixels
- adjusts the file size and number of pixels while keeping the exact print resolution and paper size

- Modify the number of pixels, file size, print resolution, and print size
- when choosing a new size for your photographs, keep these tips in mind:
- If you enlarge the photograph, you risk blurriness and a loss of detail.
- It would help if you only resized a picture once. If you mistake while resizing a picture, undo the change and try again.
- Resize photographs after correcting and touching them up.

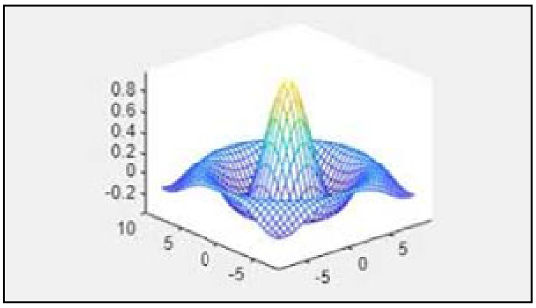


Fig. 4. MATLAB resize image

Rotation. When discussing images or image editors, the rotate function allows you to spin the image in either the clockwise or anticlockwise direction. Some editors, for instance, allow you to rotate photos by 90, 180, or 270 degrees.

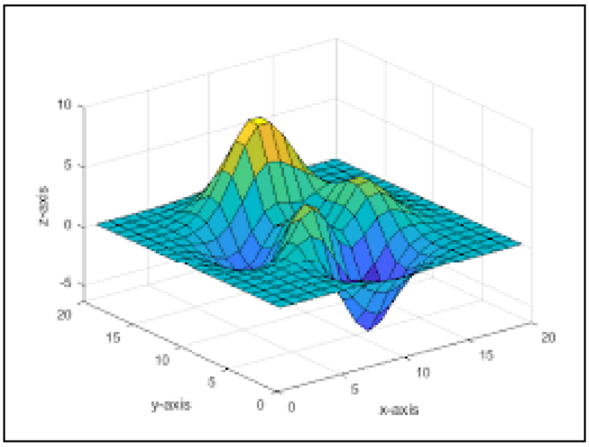


Fig. 5. MATLAB rotation

Crop. To "crop" an image is to remove the unwanted parts of the image. Cropping is removing a piece of an image's border for aesthetic or practical reasons, such as improving the composition, changing the image's aspect ratio, emphasising a specific area, or highlighting a topic by separating it from its backdrop. Depending on the context, this may be done digitally with the use of software for photo editing, or it may be done with a physical photograph, work of art, or film. Cropping is used in many fields, including photography, film processing, television, graphic arts, and printing [26].

Fig. 6. Cropping image

RGB. The RGB colour model is widely used in open CV and digital image processing. The colour picture is composed of three separate channels. One channel per colour. This scheme's three primary colours are red, blue, and green. No other colour can be made without using these three colours in exactly this ratio. An increase in value results in a brighter hue; a value of 0 indicates a dark background.

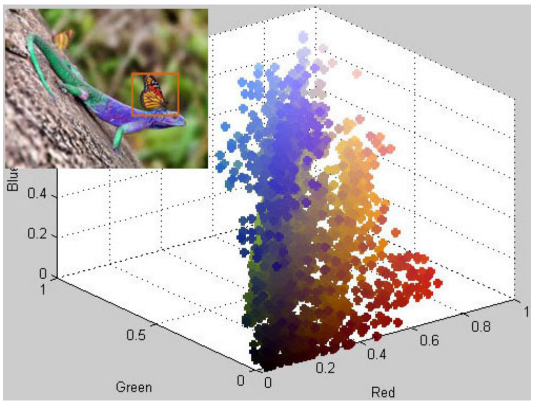


Fig. 7. MATLAB RGB color model

CMYK. For printing, the CMYK colour model is frequently employed. Cyan, Magenta, Yellow, and Black are its acronyms (key). It is a colour-parametric model. The primary colour is represented by 0, and the lightest colour by 1. In this model, the points (1, 1, 1) and (0, 0, 0) stand in for black and white, respectively. Since the model is subtractive, the value is deducted from 1 to range from the least to the most intense colour value [17].

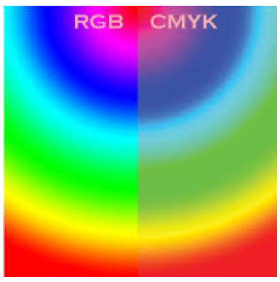


Fig. 8. CMYK colour model

HSV. Three channels made up the image. There are three channels: hue, saturation, and value. The primary colours are not used directly in this colour scheme. It uses colour to alter how people see them, such as when a cone is used to represent HSV colour.

Hue is a part of the colour. The hue represents various colours in various angle ranges since the cone represents the HSV model.

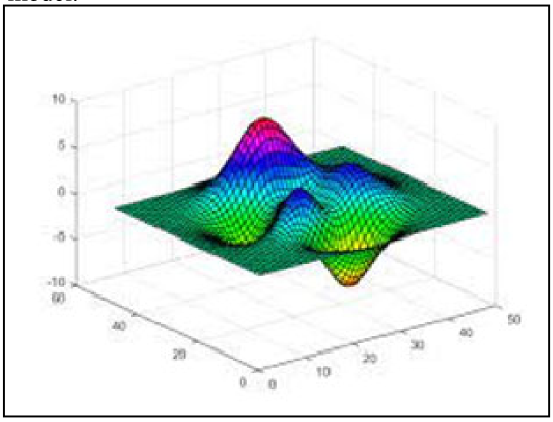
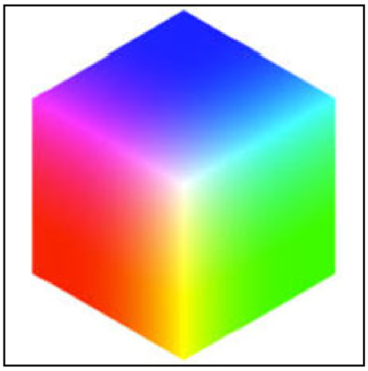


Fig. 9. MATLAB HSV colour model

YIQ. YIQ is the predominant colour model for TV broadcasts. The luminance information is denoted by the letter Y, whereas the letter IQ shows chrominance. The white and black TV only showed the bright half (Y). The y-value may be thought of as the grayscale part. Colour data is encoded in the IQ variable. There is a conversion formula [18] that may be used to convert RGB to YIQ and vice versa.



\Fig. 10. YIQ colour model

RESULTS

Rapid progress in IT has led to increased available image-processing software packages. The sector's fast growth has directly affected the spread of survey-processing systems and image-processing methods. This advancement has shifted its principal platform from the mainframe to the personal computer, enabling users to engage in various operations and processing methods, from the smallest to the most prominent statistical procedures.

The article uses MATLAB to expand and improve the Color Image Processing software library. Imaging enhancement, noise reduction, picture segmentation, and a morphological dilatation operation are just some of the tools in the Biomedical Picture Processing software suite. The software package will use a graphical user interface to display input images, output images, and various click buttons for various image processing techniques, with descriptions of the techniques provided to help users understand how an image is analysed without having to learn the mathematical algorithm for such techniques [27].

This practical training and performance support tool was developed using the ADDIE paradigm, a tried-and-true method for instructional designers and training providers. The program was developed using an iterative cycle that includes analysis, design, development, implementation, and assessment. To begin our study, we surveyed first-year students and undergraduates to gauge their understanding of the material presented in the Biomedical Image Processing course and to identify any weaknesses or concerns with the material's difficulty.

A technical computer environment and programming language, MATLAB, was utilised to develop the Biomedical Image Processing software package. MATLAB was primarily developed to facilitate the development of routines for numerical linear algebra. However, it has subsequently been extended to facilitate the development of numerical algorithms across various domains. To make image processing and other applications more accessible, MATLAB uses a core language similar to traditional linear algebra notation.

We successfully created the Biomedical Image Processing software suite using the MATLAB platform. The software suite includes tools for enhancing images, filtering unwanted noise, segmenting photos, and performing a morphological dilatation operation. Users can grasp how a picture is examined using the software's graphical user interface without studying the underlying mathematical process. The software package was developed using the ADDIE approach, guaranteeing that it is a practical training and performance support tool for students and learners. In order to develop algorithms and routines for many uses, including image processing, the MATLAB programming environment is used [28].

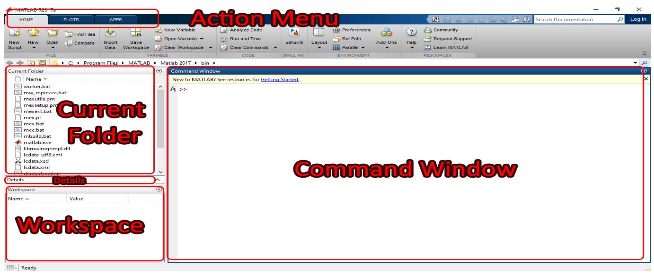


Fig. 11. MATLAB program Interface

Graphical User Interface in MATLAB (GUI): The user can operate a program using a Graphical User Interface (GUI), which is a graphical interface, without knowing the language.

Intuitive controls are provided to accomplish this. The buttons the user clicks to acquire a specific result are called the controls. An event-driven software is GUI. This is so that it can take input anytime exe, execute the program, and produce results using callback functions.

In this post, we will take a closer look at the elements that go into making a graphical user interface. We will also see how to create the components' callback functions, which allow them to operate the controls [29].

The GUI's source codes are produced via MATLAB. This improves it as a tool for designing engineering component GUIs.

In the workspace, type guide to open the GUI.

• Click OK after selecting the Blank GUI (Default) template in the GUIDE Quick Start dialogue box.

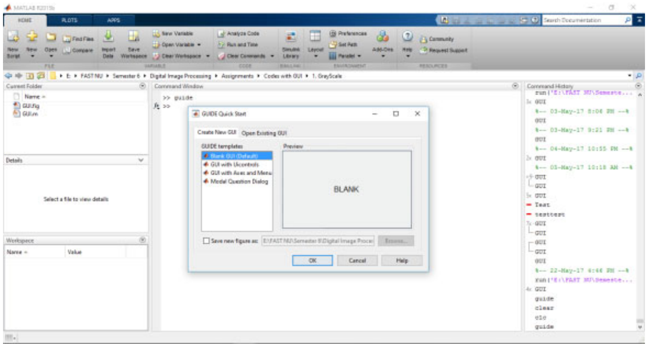


Fig. 12. Guide Quick start window

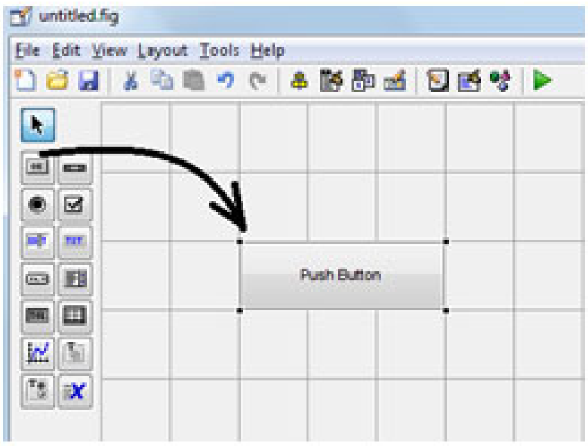


Fig. 13. Adding button at Blank GUI

List the UI elements and their names in the palette: Make sure Display names in the element palette are checked under File> Preferences > GUIDE before clicking OK.

Adjust the width and height of the grid in the Layout Editor to change the size of the UI window. To make a canvas that's 3 inches by 4 inches, click its lower-right corner and drag it to the desired dimensions. Enlarge the canvas if you need to [18].

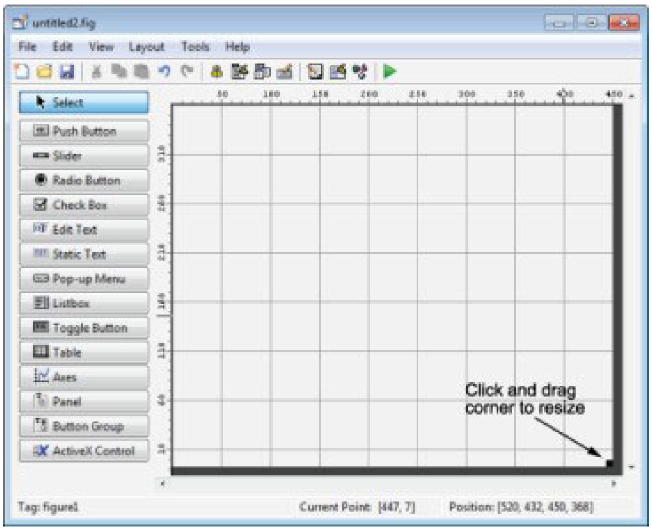


Fig. 14. Windows size selection of the UI

1) When a design is saved in GUIDE, two files are produced: a FIG file and a codes file. Fig files, which end in the extension. Fig is a particular binary File that includes a schematic or layout description. The extended code file includes MATLAB scripts that manage the UI. Save and run your program by selecting Tools > Run.

2) GUIDE displays a dialogue box: “Activating will save changes to your figure file and MATLAB code. Do you wish to continue? Click yes.

3) GUIDE opens a Save As dialogue box in your current folder and prompts you for a FIG file name [20].

4) Browse to any folder for which you have to write privileges, and then enter the file name simple_gui for the FIG file. GUIDE saves both the FIG file and the code file using this name.

5) If the folder in which you save the files is not on the MATLAB path, the current folder.

GUIDE opens a dialogue to allow you to change. GUIDE saves the files simple_gui.fig and simple_gui.m and then runs the program. It also opens the code file in your default editor.

The files must be brought into the MATLAB program at the beginning of work to open it. Click on the Browse button from the folder, as shown in the image. Then, specify the path or location of the folder in the calculator and press the OK button. After fetching the files, we noticed that they had appeared within the MATLAB program, and then we selected the (browse.m) file by clicking on the file name. Double-click on the name of the work and then click on the Run button:

The program interface will appear to us and contain several commands and buttons to control the images as in the picture:

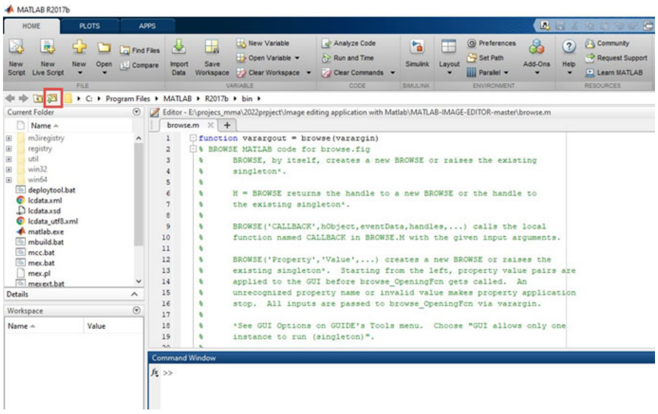


Fig. 15. Adding files into the program

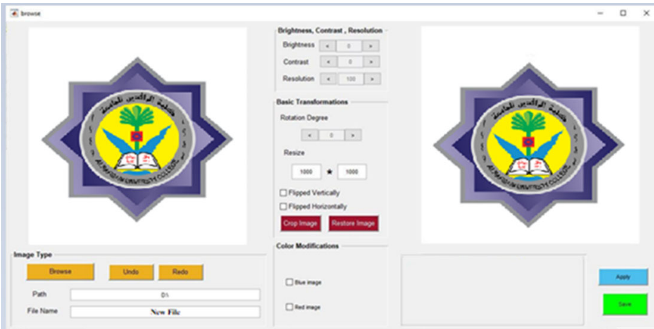


Fig. 16. Connection scheme

The program is divided into several sections, including:

• Control the image brightness, image resolution, and image contrast through the pushbutton:

Fig. 17. Noise Spatial Filtering at the image

Controlling the rotation of the image at different angles by controlling the size of the image by writing the size you want inside the text box, and also controlling the flipping of the image in different directions and cutting the image or returning it to its original. Position without any change. Control over the colours in the image, with two systems.

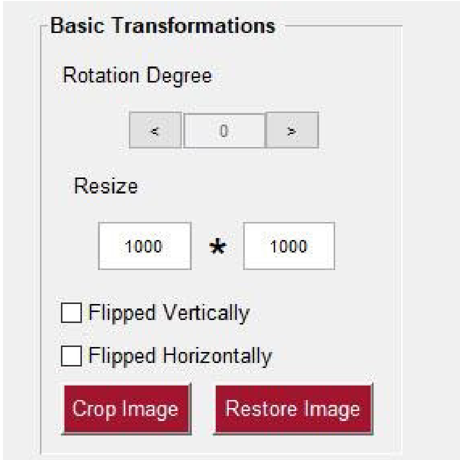


Fig. 18. Window for resizing image

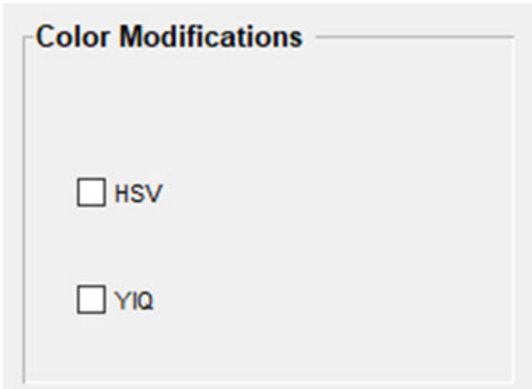


Fig. 19. Color modification window

First, you must bring any image and any extension into the program to be modified.

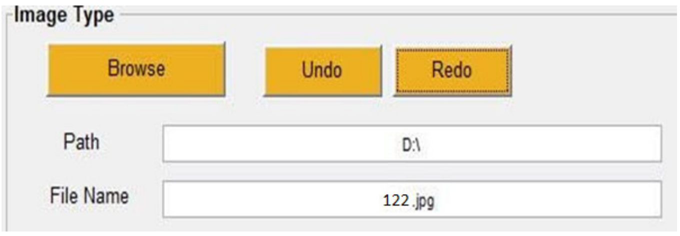


Fig. 20. The window for the selection image type

A platform that is easy to use has been produced due to the system's design. It is based on a graphical user interface utilising tools provided by MATLAB. This platform may be used for data analysis and modelling. Users of the GUI-based system can do complicated computations and data analysis jobs without the requirement for considerable knowledge of programming or coding abilities. Users are provided with an easy-to-navigate interface that allows them to enter data, choose settings, and display the results, utilising various cutting-edge tools and features.

The findings suggest that the newly designed system improves the speed and quality of data modelling and analysis jobs, making these processes available to many users. The system's user interface enables rapid and simple generation of graphical outputs and visualisations, which may help comprehend and share complicated data sets. In addition, the adaptability and scalability of the system render it appropriate for usage in various domains, such as the engineering, scientific, and financial industries.

The acquired findings, taken as a whole, provide evidence that the designed system successfully reduces the complexity of analysing data and modelling while boosting both accuracy and productivity. Users can do complex computations and data analysis activities that are more effectively and efficiently performed with the aid of the user-friendly interface that the system provides, as well as the extensive MATLAB tools and available features.

DISCUSSION

The article provides an in-depth analysis of the development and use of a graphical user interface via MATLAB tools. The GUI system discussed in this article is of considerable importance in several domains, including image processing, software testing, colour correction, and multimedia transmission, among other areas. This paper aims to comprehensively analyse the main aspects and significance of the research at hand, drawing upon pertinent sources included in the references.

The graphical user interface presented in this article provides a flexible platform that may be used across several disciplines. This technology facilitates user engagement with intricate algorithms and processes, hence streamlining activities in several domains, such as image processing [29], software testing [30], and multimedia transmission [13]

The GUI plays a significant role in image processing because it provides a user-friendly interface for various picture-enhancing methods [31]. The accessibility of techniques and tools in this domain is a valuable opportunity for researchers and practitioners, as it can potentially drive progress in image analysis and enhancement [32]

The essay recognises the significance of software testing [30] and highlights the substantial potential of the GUI in this domain. The software provides several functionalities that aid the testing procedure, such as restoring program execution states. This capability could enhance software development's resilience and reliability.

The GUI's significance in colour correction in picture transmission and multimedia applications is emphasised [33]. Ensuring the preservation of colour information in picture and video data is of utmost importance, as it dramatically enhances the quality of multimedia communication systems.

The graphical user interface system is crucial in analysing and enhancing energy efficiency in digital broadcasting [34]. The tool facilitates the representation and control of data about energy use, hence assisting in developing broadcasting systems that are more efficient in their energy consumption.

The paper highlights incorporating the GUI with MATLAB tools, a widely used platform in several scientific and engineering fields [35]. MATLAB's tremendous capabilities are integrated by improving accessibility and usability.

The article cites many research studies, highlighting their significance within the broader scientific and technical framework. The sources included in this statement encompass investigations into picture noise models [18], image perception [24] and colour temperature transformation [10], therefore showcasing the extensive scholarly groundwork behind the invention of the GUI.

The graphical user interface system facilitates innovation across several areas. Researchers and developers may use this platform to experiment with novel algorithms, conduct tests, and effectively evaluate data [9]. Furthermore, it facilitates interdisciplinary cooperation.

Using a GUI guarantees a smooth and user-friendly experience [33], enhancing accessibility to a broader range of individuals, including those without considerable proficiency in programming.

The article will be enhanced by discussing prospective advancements and enhancements of the GUI technology. Potential enhancements to the system include a range of possibilities, such as augmenting its capabilities, integrating it with nascent technologies like the Internet of Things (IoT) [36], or customising it to cater to distinct requirements within various industries.

The paper presents a novel GUI system with MATLAB tools. This system can transform diverse disciplines by streamlining intricate procedures, improving user satisfaction, and fostering advancements in research and innovation. The tool's versatility and seamless integration with MATLAB [6] make it an essential asset for researchers, practitioners, and developers in several sectors. Its utilisation has significantly contributed to developments in software testing, image processing, colour correction, and other related areas.

CONCLUSIONS

Designing an image-processing-based photo editor from scratch is a challenging feat. Several difficulties were faced in the construction of the application, including obtaining the codes for each tool from the internet and the main MATLAB website and arranging each tool in the right area on the screen. Nonetheless, we overcame these obstacles and made a completely functioning software by staying on task and working diligently.

Image resizing, photo utility, photo reversal, colour correction, and image flipping (up, down, left, and right) are some of the core capabilities we included in our picture editor software package. Essential colour picture processing is performed in this system through the integrated GUI. Users may get a deeper insight into each procedure by pushing a single, straightforward button. This GUI allows the creation of any generic-colored picture.

By modifying the callbacks, our image editing program may be used for various purposes while maintaining the same user interface. Including this function improves the software's usability and makes it more malleable to individual preferences. There is room for improvement and further study in most functions covered here. Future studies should focus on expanding the range of applications that use image processing methods and quantifying the qualitative and quantitative gains that result from their use.

The MATLAB development platform was the foundation for the final product of the image editing software package. Originally designed to facilitate the development of routines for numerical linear algebra, MATLAB is a programming language with many other valuable applications. As a result of its subsequent growth, it is often used to develop numerical algorithms for a wide range of uses. A few additions might be confusing at first, but otherwise, the terminology used is similar to what you would find in a textbook on linear algebra.

Key features of the image editor program we created include authenticity, effect, and reliability. The outcome of grading, a fine correction of the collected picture, is usually the cutoff point at which validity is judged to be in place. Image enhancement, noise filtering, segmentation, and morphological dilatation operation may be accomplished using this software package for biomedical image analysis. Furthermore, it may be used to educate students on the basics of image processing without requiring them to study the corresponding mathematical procedure.

During the design phase, a software program's layout is paramount. The MATLAB functions were used to develop the image editing program's graphical user interface, making it simple to use and modify any image. In order to facilitate the user's learning and use of the software's many functions, it was intended to be both simple and straightforward.

Designing an image-processing-based photo editor from scratch is a challenging feat. Nonetheless, we overcame these obstacles and made a completely functioning software by staying on task and working diligently. The software suite we created includes many core capabilities, such as picture resizing, photo utility, photo reversal, colour correction, and photo flipping (up, down, left, and right). The MATLAB programming environment served as the basis for this software's creation, and the program's interface and functionality are meant to be intuitive and straightforward. This program can analyse biological pictures and teach students the foundations of image processing. It contains several essential features, such as validity, impact, and reliability. Further work is needed to expand the range of applications that may benefit from image processing methods and to quantify the statistical uptick that results from their use.

REFERENCES

[1] R. Kabilan, G. P. Devaraj, U. Muthuraman, N. Muthukumaran, J. Z. Gabriel, and R. Swetha: 'Efficient Color Image Segmentation Using Fastmap Algorithm', in Editor (Ed.)^(Eds.): 'Book Efficient Color Image Segmentation Using Fastmap Algorithm' (2021, ed.), pp. 1134-41

[2] Y. Huang: ‘'Overview of Research Progress of Digital Image Processing Technology’', *Journal of Physics: Conference Series*, 2386, (1), 2022, pp. 012034

[3] S. Zhang, Z. Siran, Y. Feng, and J. Chen: 'Research on Visualisation Structure Representation of Image Processing Technology', in Editor (Ed.)^(Eds.): 'Book Research on Visualization Structure Representation of Image Processing Technology' (2021, ed.), pp. 683-86

[4] Y. Sun, Zhao, Z., Jiang, D., Tong, X., Tao, B., Jiang, G., Kong, J., Yun, J., Liu, Y., Liu, X., Zhao, G., & Fang, Z. (2022). : ‘'Low-Illumination Image Enhancement Algorithm Based on Improved Multi-Scale Retinex and ABC Algorithm Optimization. '', *Frontiers in Bioengineering and Biotechnology* 10, 2022

[5] G. Ijemaru, Nwajana, A., Oleka, E., Otuka, R., Ihianle, I., Ebenuwa, S., & Obi, E.: '' Image processing system using MATLAB-based analytics. '', *Bulletin of Electrical Engineering and Informatics.* , 2021

[6] K. Ujjwal, S. M. Sheshadri, D. Sabhari, A. Dhanuka, and A. Singh: 'Processing of Thermal Satellite Images Using MATLAB', in Editor (Ed.)^(Eds.): 'Book Processing of Thermal Satellite Images Using MATLAB' (2022, ed.), pp. 1-10

[7] N. Brancati, G. D. Pietro, D. Riccio, and M. Frucci: ‘'Gigapixel Histopathological Image Analysis Using Attention-Based Neural Networks’', *IEEE Access*, 9, 2021, pp. 87552-62

[8] F. Octaviany, Anam, C., Sutanto, H., & Naufal, A. . ‘'Comparison of Several Methods for Automated Noise Measurements in Computed Tomography’', *International Journal of Scientific Research in Science and Technology*, 2022

[9] A. P. Wibowo, A. Adha, I. F. Kurniawan, and I. Laory: ‘'Wall Crack Multiclass Classification: Expertise-Based Dataset Construction and Learning Algorithms Performance Comparison’', *Buildings*, 12, (12), 2022, pp. 2135

[10] N. Qasim, and V. Pyliavskyi: ‘'Color temperature line: forward and inverse transformation’', *Semiconductor physics, quantum electronics and optoelectronics*, 23, 2020, pp. 75-80

[11] S. Yelmanov, and Y. Romanyshyn: 'Assessing the Information Capacity of Image Based on the Theory of Contrast', in Editor (Ed.)^(Eds.): 'Book Assessing the Information Capacity of Image Based on the Theory of Contrast' (2022, ed.), pp. 1018-23

[12] N. Singh, and A. K. Bhandari: ‘'Principal Component Analysis-Based Low-Light Image Enhancement Using Reflection Model’', *IEEE Transactions on Instrumentation and Measurement*, 70, 2021, pp. 1-10

[13] N. Hashim, A. H. Mohsim, R. M. Rafeeq, and V. Pyliavskyi: ‘'Color correction in image transmission with multimedia path’', 2006

[14] H. Wu, G. Liu, Y. Yao, and X. Zhang: ‘'Watermarking Neural Networks With Watermarked Images’', *IEEE Transactions on Circuits and Systems for Video Technology*, 31, (7), 2021, pp. 2591-601

[15] P. Wang, Z. Wang, D. Lv, C. Zhang, and Y. Wang: "Low illumination colour image enhancement based on Gabor filtering and Retinex theory", *Multimedia Tools and Applications*, 80, (12), 2021, pp. 17705-19

[16] P. Li, H. Wang, M. Yu, and Y. Li: ‘'Overview of Image Smoothing Algorithms’', *Journal of Physics: Conference Series*, 1883, (1), 2021, pp. 012024

[17] S. Tongbram, B. A. Shimray, L. S. Singh, and N. Dhanachandra: "A novel image segmentation approach using fcm and whale optimisation algorithm", *Journal of Ambient Intelligence and Humanized Computing*, 2021

[18] J. Liu, and J. Feng: ‘'Design of embedded digital image processing system based on ZYNQ’', *Microprocessors and Microsystems*, 83, 2021, pp. 104005

[19] S. V. Suryanarayana, D. B. L. Deekshatulu, D. K. L. Kishore, and Y. Kumar: 'Estimation and Removal of Gaussian Noise in Digital Images', in Editor (Ed.)^(Eds.): 'Book Estimation and Removal of Gaussian Noise in Digital Images' (2011, ed.), pp.

[20] M. Le Breton, N. Bontemps, A. Guillemot, L. Baillet, and É. Larose: ‘'Landslide monitoring using seismic ambient noise correlation: challenges and applications’', *Earth-Science Reviews*, 216, 2021, pp. 103518

[21] S. Sharma, and R. Kulkarni: ‘'Noise reduction in speckle interferometry fringe pattern using adaptive Kalman filter’', *Optical Engineering*, 60, (12), 2021, pp. 124105

[22] L. Pan, C. Deng, C. Yu, S. Yue, W. Gong, and S. Han: ''Influence of the sparsity of random speckle illumination on ghost imaging in a noise environment'', *Chinese Optics Letters*, 19, (4), 2021, pp. 041103

[23] E. Y. Chan: ''Brightness Motivates Healthy Behaviors: The Role of Self-Accountability'', *Environment and Behavior*, 54, (2), 2021, pp. 363-82

[24] M. Muniraj, and V. Dhandapani: "Underwater image enhancement by combining colour constancy and dehazing based on depth estimation," *Neurocomputing*, 460, 2021, pp. 211-30

[25] T. Imamura, M. Baba, N. Hoshikawa, H. Nakayama, T. Ito, and A. Shiraki: ''A New Algorithm for Displaying Images With High Resolution Using a Directional Volumetric Display With Threads and a Projector'', *IEEE Access*, 10, 2022, pp. 15288-97

[26] X. Ma, Li, X., Zhou, Y., & Zhang, C.: " Image smoothing based on global sparsity decomposition and a variable parameter. "*Computational Visual Media* 7, 2021, pp. 483 - 97

[27] T. Naskar, & Kumar, J.: "MATLAB codes for generating dispersion images for ground exploration using different MASW transforms. "*GEOPHYSICS.* , 2022

[28] G. K. I. e. al.: "Image processing system using MATLAB-based analytics." *Bulletin of Electrical Engineering and Informatics* 2021

[29] R. C. Gonzalez, Richard Eugene Woods, and Steven L. Eddins. : '' Digital image processing using MATLAB.'', *Pearson Education India*, 2004

[30] H. H. a. S. K. Y. Otani: " A Software Testing Tool with the Function to Restore the State at Program Execution of a Program under Test," *International Journal of Education and Information Technologies*, 2012, pp. 71-78.

[31] L. Yang, R. Parton, G. Ball, Z. Qiu, A. H. Greenaway, I. Davis, and W. Lu: ''An adaptive non-local means filter for denoising live-cell images and improving particle detection'', *J Struct Biol*, 172, (3), 2010, pp. 233-43

[32] L. W. Kang, C. W. Lin, and Y. H. Fu: ''Automatic single-image-based rain streaks removal via image decomposition'', *IEEE Trans Image Process*, 21, (4), 2012, pp. 1742-55

[33] P. Mishra, J. M. Roger, D. N. Rutledge, A. Biancolillo, F. Marini, A. Nordon, and D. Jouan-Rimbaud-Bouveresse: ''MBA-GUI: A chemometric graphical user interface for multi-block data visualisation, regression, classification, variable selection and automated pre-processing'', *Chemometrics and Intelligent Laboratory Systems*, 205, 2020, pp. 104139

[34] N. Qasim, Y. P. Shevchenko, and V. Pyliavskyi: "Analysis of methods to improve the energy efficiency of digital broadcasting", *Telecommunications and Radio Engineering*, 78, (16), 2019

[35] H. Huang, Z. Liu, C. Chen, and Y. Zhang: ''Hyperspectral image classification via active learning and broad learning system'', *Applied Intelligence*, 53, 2022

[36] M. Valinejadshoubi, O. Moselhi, A. Bagchi, and A. Salem: ''Development of an IoT and BIM-based automated alert system for thermal comfort monitoring in buildings'', *Sustainable Cities and Society*, 66, 2021, pp. 102602