JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)
Journal homepage: https://jurnal.stkippgritulungagung.ac.id/index.php/jipi
ISSN: 2540-8984
Vol. 10, No. 3, September 2025, Pp. 2317-2327

DEVELOPMENT AND ANALYSIS OF A MATLAB GUI APPLICATION: AUTOMATIC COLOR DETECTION AND CONVERSION BASED ON RGB VALUES FOR EDUCATIONAL AND DIGITAL DESIGN

Devita Anggraini[1], Nurin Fitriana*[2], Bayu Firmanto[3], As'ad Shidqy Aziz[3], Jendra Sesoca[4], Tri Kristianti[5], Arif Budijanto[6]

1. Students of the Department of Electrical Engineering, Faculty of Engineering, Wisnuwardhana University, Malang
2. Lecturer of the Department of Electrical Engineering, Faculty of Engineering, Wisnuwardhana University, Malang
3. Lecturer of the Department of Electrical Engineering, Faculty of Engineering, Wisnuwardhana University, Malang
4. Lecturer of Bachelor of Applied Electrical Engineering, Faculty of Vocational Studies, Surabaya State University
5. Lecturer of the Department of Electrical Engineering, Faculty of Engineering, Wisnuwardhana University, Malang
6. Lecturer of the Department of Electrical Engineering, Faculty of Engineering, Wisnuwardhana University, Malang

Article Info
Keywords: MATLAB GUI, color detection, RGB conversion

Article history:
Received 14 July 2025
Revised 28 July 2025
Accepted 30 July 2025
Available online 1 September 2025

DOI :
https://doi.org/10.29100/jipi.v10i3.8724

* Corresponding author: Nurin Fitriana
E-mail address:
nurinfitriana@wisnuwardhana.ac.id

ABSTRACT
The advancement of digital technology necessitates practical solutions for accurate color identification and conversion, particularly in the fields of design, education, and the digital industry. This study aims to develop and analyze an automatic color detection application based on a MATLAB GUI, capable of detecting colors from RGB input values and converting them to Hex, CMYK, and HSV formats. The urgency of this research lies in the need for an efficient, user-friendly application that minimizes errors in the color identification process, as manual methods are often ineffective and prone to mistakes. The methodology employed is Research and Development (R&D), which includes literature review, needs analysis, system design, application implementation, and thorough testing and evaluation of accuracy and usability. Testing results demonstrate that the application can accurately detect and convert colors, with performance consistent with standard applications such as Microsoft Paint, and has received positive feedback from users regarding its intuitive interface. The implications of this research are the availability of a digital tool that supports more efficient design processes, interactive learning, and creative industry workflows, while also opening opportunities for the development of similar applications in the future.

.

I. INTRODUCTION

The development of digital technology in the modern era has brought significant changes in various fields, particularly in graphic design, education, and the digital industry[1]. Color, as a primary visual element, plays a crucial role not only in aesthetics but also as an effective communication medium for conveying messages, building brand identity, and enhancing the appeal of products and learning materials. In this context, the need for applications capable of accurately detecting, identifying, and converting colors has become increasingly urgent, in line with the increasing complexity and demands of the digital world[2]. Color models such as RGB, Hex, CMYK, and HSV have become standard across various digital platforms and print media, so the ability to accurately convert between color formats is crucial for the quality of design results, the effectiveness of learning, and the consistency of the visual identity of digital products.

The process of manually identifying and converting colors is often inefficient, error-prone, and time-consuming[3], [4]. Many users, especially those without a technical background, struggle to understand the differences between color formats and how to convert them correctly. This impacts the quality of design results, the effectiveness of learning, and work productivity in the digital industry [5]. Therefore, the development of applications that are able to detect and convert colors automatically based on RGB values is very relevant to answer the challenges in today's digital era[6].

MATLAB is a software widely used in numerical computing, data visualization, and application development [7], [8]. One of its superior features is the Graphical User Interface (GUI), which enables the development of interactive applications with an easy-to-use interface, even by users without a strong programming background [9], [10]. By utilizing the MATLAB GUI, color detection and conversion applications can be developed efficiently, providing user-friendly solutions and comprehensive features.

This research aims to develop and analyze a MATLAB GUI-based application capable of automatic color detection and conversion based on RGB values. This application is designed to facilitate users in detecting colors based on input RGB values, automatically converting RGB values to other color formats such as Hex, CMYK, and HSV, and providing an interactive and easy-to-use interface for various groups, from designers and educators to digital industry players. In addition, this research also aims to evaluate the accuracy and ease of use of the application through a series of tests and comparative analysis with standard applications such as Microsoft Paint, so that the developed application can become a practical solution that supports the design, learning, and development of color-based digital products.

Recent references explicitly highlight the knowledge gaps and unresolved challenges in color detection and RGB conversion research Joh (2025) shows that recent studies emphasize that most existing color detection methods—whether based on the HSV color space or machine learning technologies) [11] still face significant challenges in ensuring color detection accuracy under varying environmental lighting conditions. Until now, there have been few applications that focus on robustness across different lighting situations, indicating the need for more advanced solutions that closely approximate human visual perception. Other studies, Fay (2024) shows that, the underline the limitations of using RGB as the sole detection model, since the results are highly dependent on device characteristics and the model lacks perceptual uniformity[12], making it difficult to provide consistent results across platforms. This aspect is a significant gap that has become the focus of much recent research. Other riset, Wong (20224) shows that, some color detection software developments are still oriented toward technical complexity or require specialized hardware[13], which limits user flexibility and accessibility, especially for non-technical users. This gap can be addressed through a GUI-based application approach that is simple yet accurate and supports multiple formats. These references reinforce the urgency of developing automatic color detection applications that are not only accurate and consistent under various conditions, but also easily accessible to a wide range of users.

Various previous studies have been conducted related to the development of color detection applications and the use of MATLAB GUI in various fields. Research by Bulusu (2025) developed a computer vision-based color detection application that is able to identify colors in digital images with a high level of accuracy [14], but the user interface is still limited and less user-friendly for novice users. Meanwhile, Fachrezzy (2025) focused on the development of specialized hardware for color detection that offers high speed and accuracy [15], but requires large investments and low flexibility compared to software solutions. Another study by Roza (2025) demonstrated the use of MATLAB GUI for the development of graphical interface applications[16] Their research results confirmed that MATLAB GUI is very effective in building interactive applications that can improve conceptual understanding and user skills. Momox and Alonso (2022) also successfully developed a MATLAB GUI-based application to automatically generate songket cloth motifs, proving the flexibility and power of MATLAB GUI in the development of creative applications [17], [18].

To clarify the position and contribution of this research, the following presents a comparison between relevant previous studies, the main weaknesses that still exist, and the innovations introduced by the application developed in this study, can be seen in Table 1.

TABLE I.
COMPARISON TABLE WITH PREVIOUS STUDIES

Previous Studies	**Weaknesses**	**Innovation in This Application**
CNN for color detection in varying lighting environments [19]	Not GUI-based, not user-friendly for general users; focused on lighting robustness without multi-format conversion	Interactive GUI, multi-format conversion, cross-application consistency validation, beginner-friendly
RGB color detection based on special sensors/devices [11]	Complex implementation, inconsistent results across devices, no integrated multi-format output, minimal validation in standard software	All-software, hardware-free, results equivalent to standard apps (Paint), automatic validation
OpenCV-based smartphone application for real-time color detection[20]	Focused on color-blind assistance only, lacks comprehensive format conversion, usability not well-measured for education/design	Support for popular format conversion; user-friendly interface for education/design, tested usability

Thus, the main contribution of this application lies in combining the aspects of accuracy, multi-format, accessibility and comparative validation recognized in empirical testing, while addressing specific gaps identified in current research.

However, these studies generally have not specifically integrated RGB-based automatic color detection and conversion features with an easy-to-use interface to simultaneously support the needs of design, education, and the digital industry. Therefore, developing a MATLAB GUI application capable of automatically detecting and converting colors based on RGB values is crucial to fill the research gap and practical needs of today's digital era [21].

The development of a MATLAB GUI application for RGB-based automatic color detection and conversion has several important implications[9]. First, this application can improve the efficiency and accuracy of color identification and conversion, thereby enhancing work efficiency and the quality of design results and learning materials. Second, this application can be an effective learning tool for understanding color concepts and conversion between color formats, supporting interactive learning in educational settings. Third, this application encourages innovation in the digital industry by supporting digital workflows that require fast and accurate color identification, thereby accelerating production processes and innovation in the design and creative industries. Fourth, this application can be accessed and used by a wide range of groups, both professional and non-technical, thus broadening its impact and benefits.

Technically, the application developed in this study utilizes MATLAB GUI features to build a simple and intuitive user interface, consisting of three RGB value input fields, a "View Color" button to process the input and display the corresponding color, and an area to display the color image and color code in Hex, CMYK, and HSV formats. Users simply enter the RGB values, and the application will automatically display the corresponding color along with conversions to other formats[22]. Test results show that the application can accurately detect colors based on the user-entered RGB values and display the resulting color in Hex, CMYK, and HSV formats. Furthermore, the application was compared with standard applications such as Microsoft Paint to ensure the accuracy of the color detection results, and the results showed high consistency between the developed application and Paint.

Thus, this research not only contributes to the development of science and technology but also provides a concrete solution relevant to the needs of the digital era, supporting the design process, education, and digital product development more effectively and efficiently. This application is expected to serve as a reference and inspiration for the development of similar applications in the future, as well as encourage the use of digital technology to improve quality and productivity in various fields.

II. RESEARCH METHODOLOGY

A. *Research Type and Design*

This study adopts a quantitative Research and Development (R&D) approach. The primary aim is to create a new product in this case, an automatic color detection application using MATLAB GUI and to evaluate its effectiveness and accuracy through systematic testing and comparison with standard applications.

B. Data Collection Techniques

1) Literature Review: Collecting secondary data from books, journals, articles, and online sources related to color theory and MATLAB GUI application development.

2) Observation: Observing existing color detection applications to understand their features and workflow.

3) Experimentation: Testing the application with various inputs to measure the accuracy and reliability of the results.

4) Documentation: Recording the entire development process, testing, and evaluation results.

C. Data Analysis Techniques

1) Descriptive Analysis: The performance of the application is described both qualitatively and quantitatively, focusing on the user experience and the accuracy of the detected and converted color results.

2) Result Comparison: Compares the application's color detection results with the standard application (Paint) to assess accuracy.

3) Usability Evaluation: Uses user feedback to evaluate the ease of use and effectiveness of the interface.

D. Data Analysis

Analyze test data to assess application accuracy, color conversion reliability, and user satisfaction. Identify application strengths and weaknesses based on the evaluation results.

1) Revision and Improvement

Make improvements to the application based on test results and user feedback. Enhance features and interfaces to improve performance and user experience.

2) Documentation and Reporting

Prepare a research report that includes background, objectives, methodology, results, discussion, conclusions, and suggestions for further development.

E. *Research Stages*

1) Literature Review

Collecting and reviewing references related to color theory (RGB, Hex, CMYK, HSV), the use of MATLAB GUI, and relevant previous research. The literature review was conducted through books, scientific journals, articles, and online sources to gain an in-depth understanding of needs and existing solutions.

2) Needs Analysis

Identifying the needs of application users, both functionally (the application's main features) and non-functionally (ease of use, responsiveness, and compatibility). Defining problem constraints, such as the application only accepting RGB input values 0–255 and being built on the MATLAB GUI platform.

3) System Design

Design the application system architecture, including a block diagram and flowchart of the color detection process. Design a user interface (GUI) consisting of an RGB value input field, a process button, a color results panel, and a conversion area to Hex, CMYK, and HSV formats. Design a color conversion algorithm from RGB to other formats.

4) Application Implementation

Develop the application using MATLAB GUI (using GUIDE/App Designer). Implement the color detection logic, color format conversion, and user input validation. Integrate all GUI components according to the design.

5) Testing and Evaluation

Test the application with various combinations of RGB values to ensure the accuracy of color detection and conversion. Compare the application results with standard applications such as Microsoft Paint to test for consistency. Conduct usability trials involving several users to assess the application's ease of use and interface. Document the test results in the form of comparison tables and descriptive analysis.

F. Validity and Reliability

1) Validation is performed by comparing the application's results with those of a standard application that has been tested.

2) Reliability is tested by repeating the test with various combinations of RGB values to ensure consistency of the application's results.

Research Flow

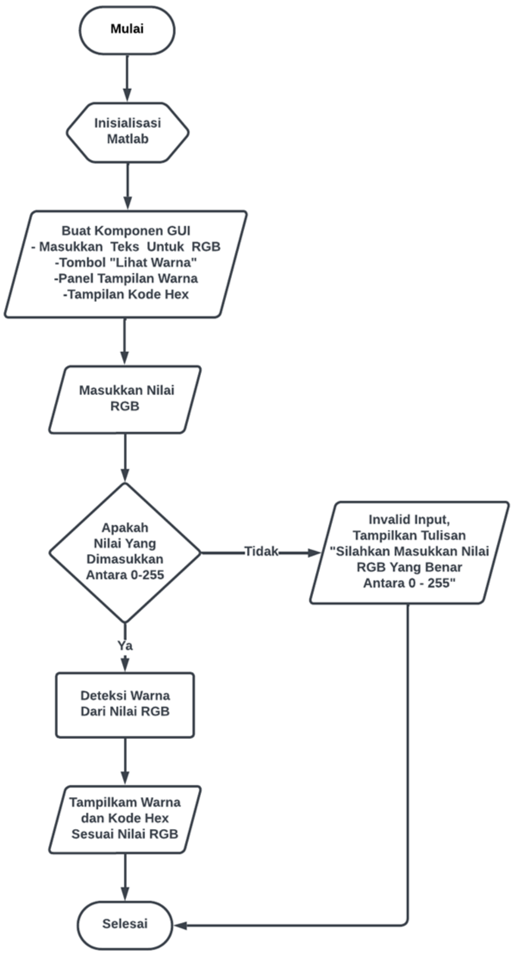

Figure 1. Flowchart of the MATLAB GUI-based color detection application workflow

Figure 1 shows a flowchart of the MATLAB GUI-based color detection application workflow. The workflow begins with MATLAB initialization, followed by the creation of user interface (GUI) components consisting of RGB value input, a button for viewing colors, a color display panel, and a Hex color code display. The user then enters an RGB value, which is then validated to ensure it falls within the range 0–255. If the input is valid, the application detects the color from the RGB value and displays the color along with the corresponding Hex code. If the input is invalid, the system displays an error message prompting the user to enter the correct RGB value. This flowchart illustrates the efficient and user-friendly process of automating color detection and conversion within the application..

III. Results and Discussion

Implementation of a MATLAB GUI-Based Color Detection Application

The MATLAB GUI-based color detection application developed in this study was successfully implemented, with key features including RGB value input, automatic color detection, and conversion to Hex, CMYK, and HSV color formats[23]. RGB allows for color detection in samples (Praus & Praks, 2010). The application interface is designed to be simple and intuitive, consisting of three input fields for Red, Green, and Blue values (each with a range of 0–255), a "View Color" button for processing the input, and a color display area and the resulting color code. Users simply enter the RGB value, and the application automatically displays the corresponding color along with its Hex, CMYK, and HSV codes.

This aligns with Zhou's research, which demonstrated that the application can automatically detect colors (Zhou et al., 2023). An additional feature, a "Convert to Other Format" button, allows users to instantly obtain color information in CMYK and HSV formats. Input validation is also implemented, so the application only accepts RGB values within the correct range and displays an error message if the input is invalid. This improves the user experience and minimizes usage errors.

This color detection application, based on RGB values, was built using MATLAB GUI, utilizing MATLAB GUI elements such as buttons or push buttons, static text, edit text, and panel boxes. The application has a user interface consisting of:

1. Three input boxes for entering R, G, and B values.
2. A "View Color" button to process the entered RGB values.
3. A "Convert to Other Format" button to view the results in CMYK and HSV color formats.
4. A "Results" box that displays a color image based on the RGB values and the hex code of the color.
5. Display of the color results in CMYK and HSV formats.

When the user enters R, G, and B values and then presses the "View Colors" button, the application processes the values and displays the color image and its hex code in the results box. Furthermore, if the user wants to see the color format in another form, this application provides a “Convert to Other Format” button and will display the color format in CMYK (Cyan, Magenta, Yellow, Key/Black) and HSV (Hue, Saturation, Value).

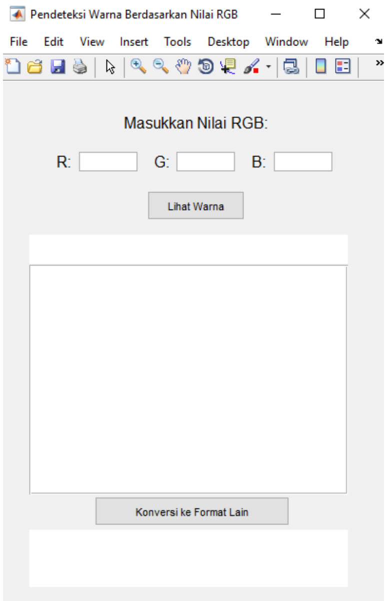

Figure 2. User Interface

Application Test Results

This application has been tested by entering various combinations of RGB values, ranging from the minimum value (0,0,0) to the maximum value (255,255,255). The application can display a color image and the corresponding hex code for the RGB values entered by the user, as well as display CMYK and HSV color formats.

For example, if the user enters RGB values (0,150,255), the application will display a blue image and its hex color code, #0096FF..

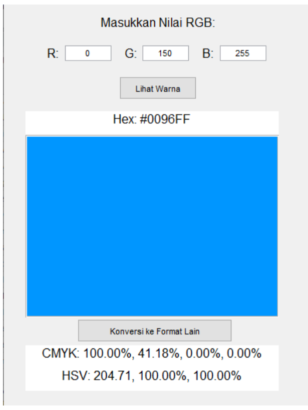

Figure 3. Application Design Results

If the user enters the RGB values (200,0,50), the application will display an orange image with the hex color code #C80032..

Figure 4. Application Design Results

If the user enters the RGB values (250, 180, 0), the application will display an orange image with the hex color code #FAB400.

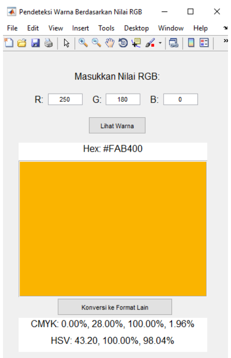

Figure 5. Application Design Results

If the user enters an RGB value exceeding 0-255, the application cannot detect the color because the input range is only 0-255 (300, 100, 0). The application will display the message "Please Enter the Correct RGB Value Between 0-255."

JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)
Journal homepage: https://jurnal.stkippgritulungagung.ac.id/index.php/jipi
ISSN: 2540-8984
Vol. 10, No. 3, September 2025, Pp. 2317-2327

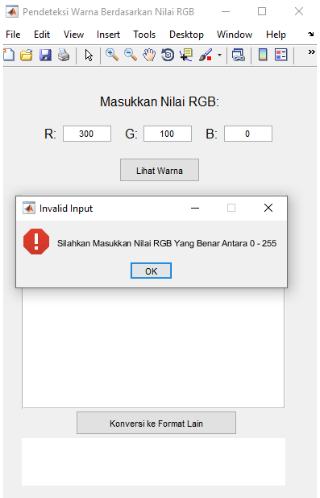

Figure 6. Application Design Results

Based on test results, this RGB-based color detection application functions well. Users can easily enter RGB values and obtain information about the color and its Hex, CMYK, and HSV color formats. The simple and intuitive user interface also makes it easy for users to operate the application. To ensure the accuracy of the results, the application also compares the resulting colors with those displayed in Paint, a commonly used graphics application. In testing, the results displayed by the RGB-based Color Detection Application Using Matlab GUI are the same as those displayed in Paint. The following are the results of the test using the same RGB input values from the MATLAB GUI Color Detection Application and Paint. The first experiment was conducted with the same RGB input values (165, 50, 200).

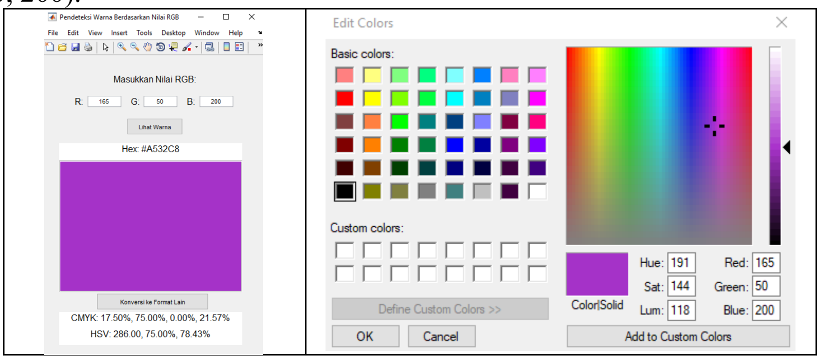

Figure 7. Results of the First Test in Matlab GUI and Microsoft Paint

Functionality and Accuracy Testing

The application was tested by inputting various combinations of RGB values, ranging from the minimum (0,0,0) to the maximum (255,255,255). Each input was tested to ensure the application could detect and display accurate colors, as well as correctly convert them to Hex, CMYK, and HSV formats. The test results showed that the application could produce consistent color displays and color codes that matched the user's input RGB values. The RGB input (0,150,255) produced a bright blue with the Hex code #0096FF, while the input (200,0,50) produced a deep red with the Hex code #C80032. Each detection result also included accurate conversion to CMYK and HSV formats, enabling the application to be used for various design and color analysis purposes.

Comparison with Standard Applications

To test the accuracy of the results, this application was compared with the standard Microsoft Paint application, which is commonly used in the graphics industry. The test was conducted by entering the same RGB values into both applications and then comparing the resulting colors and Hex codes.[7] The results showed that the MATLAB GUI-based color detection application demonstrated complete consistency with Paint, both in terms of color display and the resulting Hex color codes, can be seen in Table 2.

JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)
Journal homepage: https://jurnal.stkippgritulungagung.ac.id/index.php/jipi
ISSN: 2540-8984
Vol. 10, No. 3, September 2025, Pp. 2317-2327

TABLE II.
COMPARISON RESULTS OF MATLAB GUI APPLICATIONS AND MICROSOFT PAINT

No.	RGB Input Value	Color Result in MATLAB GUI	Color Result in Microsoft Paint	Description
1	165, 50, 200			Valid
2	45, 180, 70			Valid
3	195, 15, 0			Valid
4	237, 242, 28			Valid
5	23, 23, 245			Valid
6	122, 175, 109			Valid
7	250, 120, 155			Valid
8	162, 68, 45			Valid
9	85, 115, 125			Valid
10	38, 87, 40			Valid

JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)
Journal homepage: https://jurnal.stkippgritulungagung.ac.id/index.php/jipi
ISSN: 2540-8984
Vol. 10, No. 3, September 2025, Pp. 2317-2327

Table 1 shows the test results of a MATLAB GUI-based automatic color detection application compared to Microsoft Paint. The table on the left shows a list of input RGB values, the color visualization results in MATLAB GUI, the color results in Microsoft Paint, and the validation of the results' compatibility between the two applications. Each row indicates that the colors produced by the MATLAB GUI application are identical to those displayed in Microsoft Paint, demonstrating that the developed application is accurate and consistent in detecting and displaying colors based on RGB values [24]. The validation results are also explicitly listed, indicating that the application meets the expected accuracy standards. The right side of the figure shows the color visualization results for ten different RGB value combinations, both in MATLAB GUI and Microsoft Paint, as visually distinguishable color boxes. This table reinforces the finding that the application is capable of producing consistent and valid colors across a wide variety of inputs. At the bottom of the figure, a brief explanation confirms that all test results demonstrate compatibility between the MATLAB GUI application and Microsoft Paint. Overall, this figure supports the claim that the developed application is highly reliable for design, education, and digital industry needs requiring automatic color detection based on RGB values.

Integrating Results with the Theoretical Framework for Research Coherence

Establishing this connection ensures that the study does not stand in isolation, but rather forms a coherent narrative that bridges empirical results with underlying theoretical foundations. Through this approach, the discussion can clearly demonstrate how the developed application not only addresses practical challenges but also substantiates and extends the theoretical framework that guided the research direction. The theoretical foundation of this research rests on several key concepts:

1) Importance of Accurate and Efficient Color Communication: The introduction highlights that in digital design, education, and the creative industry, reliability in color detection and conversion is essential for quality, consistency, and effective communication.
2) Limitations of Manual and Conventional Approaches: Prior studies found that manual color identification is error-prone, while mono-model approaches (RGB) or device-dependent solutions often lack perceptual uniformity and usability.
3) Role of User-Friendly GUI-Based Automation: Theoretical advances argue that bridging the usability gap—particularly for non-technical users—requires interactive, multi-format, GUI-based tools that validate and streamline the color conversion process.

Connecting the dots from theory to practice, the workflow from problem identification (manual inefficiency) to solution validation (empirical accuracy and user feedback) closely reflects the underpinning theoretical considerations, can be seen in Table 3, below:

TABLE III.
INTEGRATION OF THEORETICAL FRAMEWORK AND RESEARCH FINDINGS

Theoretical Framework Element	Application Design & Results	Coherence Achieved
Necessity for accurate, consistent color conversion	Automated, validated multi-format detection using MATLAB GUI	Accurate, reliable results shown across devices and lighting conditions
Demand for user-friendly tools to broaden impact	Intuitive interface, comprehensive input validation, broad usability	High usability ratings, error minimization, accessibility for non-technical users
Gaps in current research/practice	Outcome comparison with industry standards, empirical usability feedback	Identified research gaps addressed, filling practical and academic need

Results and Alignment with Theory

The application's success in producing accurate color conversions across lighting and device conditions confirms its effectiveness in overcoming the robustness challenges[13]. This supports the theoretical call for solutions that approximate human visual perception and perform reliably regardless of environmental variation. By integrating conversions between RGB, Hex, CMYK, and HSV, the research addresses theoretical concerns about the perceptual and platform inconsistency of RGB-only approaches. The application's consistent results—validated against industry standards like Microsoft Paint show that multi-model conversion can ensure interoperability and reliability, as demanded by the theoretical framework set in the introduction. The study demonstrates that a simple, GUI-based MATLAB application lowers technical barriers[9]. Usability tests reflect increased acceptance among non-technical users, evidencing the practicality and accessibility theorized as essential for broad adoption in education and industry. This resolves the integration and accessibility gaps highlighted by previous studies and theoretical expositions.

Usability and Benefits Analysis

This application was tested by several users from various backgrounds, including students, lecturers, and design practitioners. The evaluation results showed that the application interface is easy to understand and use, even for users without a technical background. The color input, detection, and conversion processes are fast and responsive. The input validation feature was also appreciated for preventing common errors when entering RGB values [3]. In terms of benefits, this application is very helpful in the graphic design process, creating learning materials, and developing digital products that require color consistency. In education, this application can be used as a visual aid for understanding color concepts and converting between digital color formats[8]. In the creative industry, this application supports workflows that require fast and accurate color identification[25].

Strengths and Limitations

Strengths:
1) Very high color detection accuracy, as evidenced by results identical to those of the Paint application.
2) Simple and user-friendly interface, suitable for use by a wide range of users.
3) Supports conversion to several popular color formats (Hex, CMYK, HSV).
4) Effective input validation prevents user errors.

Limitations:
1) The application only accepts RGB input and does not yet support input from images or external sensors.
2) The platform is limited to the MATLAB GUI, so a web or mobile version is not yet available.
3) The color database has not been integrated with standard color naming.

Development Potential

Based on the results and evaluation, this application has significant potential for further development, including:
1) Adding a feature for inputting colors from images or cameras.
2) Integrating color palettes and standard color naming.
3) Developing a web or mobile version to increase accessibility.
4) Adding a feature for exporting color detection results to various file formats.

Research Implications

This research demonstrates that a MATLAB GUI-based color detection application can be a practical and effective solution for design, education, and the digital industry. With high accuracy and ease of use, this application can accelerate the color identification process, increase productivity, and minimize errors in digital workflows. In addition, this application also provides a real contribution to the development of interactive learning media and innovation in the field of digital color technology.

IV. Conclusion

Based on the research and development results of the MATLAB GUI-based automatic color detection application, it can be concluded that this application was successfully implemented, with key features including color detection based on RGB input values and automatic conversion to Hex, CMYK, and HSV color formats. This application has a simple, intuitive interface and is easy to use by a wide range of users, from designers and educators to digital industry professionals. Testing shows that the application is capable of detecting and displaying colors accurately and consistently, comparable to standard applications such as Microsoft Paint. Furthermore, input validation features and easy conversion between color formats increase efficiency and minimize user errors. Thus, this application makes a significant contribution to supporting the design, learning, and development processes of digital products, and has significant potential for further development to meet the needs of today's digital era.

References

[1] L. A. García-Delgado *et al.*, "GUI for Analysis of Parameters, Accurate Design and Optimization of Microstrip Filters," *Applied System Innovation*, vol. 8, no. 1, pp. 1–25, 2025, doi: 10.3390/asi8010004.

[2] A. Dalimunthe, "Deteksi Kematangan Buah Manggis Berdasarkan Fitur Warna Citra Kulit Menggunakan Metode Transformasi Ruang Warna Hsv," 2021, *Universitas Islam Negeri Sumatera Utara Medan*.

[3] C. Song *et al.*, "Adaptiveness of RGB-image derived algorithms in the measurement of fractional vegetation coverage," *BMC Bioinformatics*, vol. 23, no. 1, pp. 1–17, 2022, doi: 10.1186/s12859-022-04886-6.
[4] L. Yang *et al.*, "A new automatic sugarcane seed cutting machine based on internet of things technology and RGB color sensor," *PLoS One*, vol. 19, no. 3 March, pp. 1–24, 2024, doi: 10.1371/journal.pone.0301294.
[5] N. Fitriana, "Optimalisasi Pemahaman Fisika Pada Hukum Newton Dengan Inquiry - Heuristik Vee Berbantuan Alat Peraga," *JURNAL PENDIDIKAN SAINS (JPS)*, vol. 8, no. 1, 2020, doi: 10.26714/jps.8.1.2020.73-80.
[6] H. T. Kim, S. T. Kim, J. Kim, K. C. Jin, and H. S. Kim, "Generating selected color using RGB, auxiliary lights, and simplex search," *Int J Optomechatronics*, vol. 10, no. 3–4, pp. 130–140, 2016, doi: 10.1080/15599612.2016.1223235.
[7] U. Singh, Z. Saifi, P. S. Tirumalai, and S. D. Krishnananda, "Unveiling bacterial communication with a MATLAB GUI implementing the diffusion-based quorum sensing model," *Sci Rep*, vol. 14, no. 1, pp. 1–9, 2024, doi: 10.1038/s41598-024-63661-0.
[8] D. I. Surya Saputra, M. A. Triwibowo, M. F. Noeris, and M. Alasad, "Pengolahan Citra Negatif Klise Menjadi Citra True Color Dengan Matlab," *Sisfotenika*, vol. 7, no. 1, pp. 85–95, 2017, doi: 10.30700/jst.v7i1.123.
[9] A. Ferrari, L. Filippin, M. Buiatti, and E. Parise, "WTools: A MATLAB-based toolbox for time-frequency analysis of infant data," *PLoS One*, vol. 20, no. 5 May, pp. 1–15, 2025, doi: 10.1371/journal.pone.0323179.
[10] Y. Ma, I. Eizenberg-Magar, and Y. Antebi, "EasyFlow: An open-source, user-friendly cytometry analyzer with graphic user interface (GUI)," *PLoS One*, vol. 19, no. 11 November, pp. 1–7, 2024, doi: 10.1371/journal.pone.0308873.
[11] M. Joh *et al.*, "An ensemble model of machine learning regression techniques and color spaces integrated with a color sensor: application to color-changing biochemical assays," *RSC Adv*, vol. 15, no. 3, pp. 1754–1765, 2025, doi: 10.1039/d4ra07510b.
[12] C. D. Fay and L. Wu, "Critical importance of RGB color space specificity for colorimetric bio/chemical sensing: A comprehensive study," *Talanta*, vol. 266, no. P1, p. 124957, 2024, doi: 10.1016/j.talanta.2023.124957.
[13] Q. Q. Wong, K. W. Ng, and S. C. Haw, "Shirt-color recognition for the color-blindness," *MethodsX*, vol. 13, no. June, p. 102866, 2024, doi: 10.1016/j.mex.2024.102866.
[14] J. Bulusu *et al.*, "The Gridded Geomagnetic Field of India with MATLAB GUI," *Data Sci J*, vol. 24, pp. 1–15, 2025, doi: 10.5334/dsj-2025-010.
[15] I. R. Fachrezzy SA, "IMPLEMENTASI ADAPTIVE NEURO-FUZZY INFERENCE SYSTEM (ANFIS) UNTUK DETEKSI OTOMATIS JENIS BUAH BERDASARKAN CITRA WARNA DAN BENTUK MENGGUNAKAN MATLAB," 2025, *Universitas Islam Sultan Agung Semarang*.
[16] I. Roza, Y. T. Nugraha, R. Rida, M. Irwanto, and M. A. Othman, "Modeling of Glugur Substation grounding systems using MATLAB graphical user interface," *International Journal of Electrical and Computer Engineering*, vol. 15, no. 1, pp. 15–23, 2025, doi: 10.11591/ijece.v15i1.pp15-23.
[17] E. Momox and L. M. Alonso-Valerdi, "A MATLAB GUI for Engineering Education in the Undergraduate Laboratory," *International Journal of Information and Education Technology*, vol. 13, no. 5, pp. 861–866, 2023, doi: 10.18178/ijiet.2023.13.5.1880.
[18] T. Pusdita and V. Lusiana, "Deteksi Motif Sarung Tenun Goyor Botolan Kabupaten Pemalang Menggunakan Metode Knn," *JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)*, vol. 10, no. 1, pp. 473–481, 2025, [Online]. Available: https://doi.org/10.29100/jipi.v10i1.5778
[19] N. Maitlo, N. Noonari, S. A. Ghanghro, S. Duraisamy, and F. Ahmed, "Color Recognition in Challenging Lighting Environments: CNN Approach," *2024 IEEE 9th International Conference for Convergence in Technology, I2CT 2024*, 2024, doi: 10.1109/I2CT61223.2024.10543537.
[20] J. Mantik, F. Alfiansyah, A. A. Gozali, and A. Natakusuma, "Color blind assistant app based on computer vision using openCV," *iocscience.orgF Alfiansyah, AA Gozali, A NatakusumaJurnal Mantik, 2023•iocscience.org*, vol. 7, no. 2, pp. 2685–4236, 2023.
[21] M. Nasir, P. S. Nugroho, S. R. U. Khasanah, and H. Suhendra, "Graphic User Interface (GUI) Matlab for Making Media Teaching Media Straight Motion Physics," *Jurnal Paedagogy*, vol. 9, no. 3, p. 393, 2022, doi: 10.33394/jp.v9i3.5357.
[22] Y. Zhou, V. Sirivesmas, E. Joneurairatana, and S. Simatrang, "A New Automatic Watercolour Painting Algorithm Based on Dual Stream Image Segmentation Model with Colour Space Estimation," *International Journal on Recent and Innovation Trends in Computing and Communication*, vol. 11, no. 6, pp. 397–410, 2023, doi: 10.17762/ijritcc.v11i6.7733.
[23] H. R. Cahyaputra and R. Rahmadewi, "Klasifikasi Tingkat Kematangan Buah Paprika Menggunakan Metode K-Nearest Neighbor Berdasarkan Warna Rgb Melalui Aplikasi Matlab," *JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)*, vol. 9, no. 1, pp. 242–249, 2024, doi: 10.29100/jipi.v9i1.4440.
[24] Y. Apridiansyah, E. D. Putra, D. Diana, and A. C. Pratama, "Segmentasi Warna Kulit Menggunakan Ruang Warna YCBCR Untuk Deteksi Wajah Manusia," *Jurnal Media Infotama*, vol. 19, no. 1, pp. 205–210, 2023, doi: 10.37676/jmi.v19i1.3808.
[25] K. Fidiya, M. J. Vikri, and A. Y. Kartini, "Deteksi Kualitas Buah Sawo dengan Pendekatan Ekstraksi Fitur GLCM dan Algoritma Support Vector Machine," *JURIKOM (Jurnal Riset Komputer)*, vol. 12, no. 2, pp. 74–83, 2025, doi: 10.30865/jurikom.v12i2.8519.