IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 3, March 2024

DOI: 10.17148/IJARCCE.2024.133131

AirInk Studio: A Virtual Drawing Model

Prof. Archana Dirgule[1], Shreyas Shinde[2], Amey Ashtankar[3], Mandar Terkhedkar[4], Vedant Ingale[5]

Assistant Professor, Computer Department, Sinhgad College of Engineering, Pune, India[1]

Student, Computer Department, Sinhgad College of Engineering, Pune, India[2-5]

Abstract: In the realm of online education and artistry, the limitations of conventional mice for digital drawing and illustration have posed challenges for educators, students, and artists. The impracticality of a mouse hinders the fluidity and precision crucial for effective teaching and learning, particularly in visually dependent subjects. Additionally, the high cost of specialized drawing tablets has restricted access, limiting creative expression in the digital realm. The "AirInk Studio" project addresses these challenges by introducing an innovative desktop application. Powered by Python, OpenCV, Mediapipe, and Tkinter, it utilizes computer vision and hand tracking for a seamless drawing experience. Boasting diverse brush styles, an undo feature, and an extensive color palette, the project caters to varied artistic preferences. Not only does it redefine online education, but it also empowers artists with an affordable and versatile digital canvas, democratizing creativity in the virtual space. Furthermore, the project expands its capabilities with features like drawing shapes, including circles, rectangles, and lines, as well as a text box feature for annotations and labels. The integration of a chat web application, developed with React.js and Firebase, enables real-time collaboration and connection among users. Moreover, a community showcase web app, leveraging React.js and Firebase, provides users with a platform to share and exhibit their creations, fostering a vibrant digital art community. Together, these enhancements elevate the "AirInk Studio" project, enriching the digital art experience and promoting collaboration and creativity among users.

Keywords: Python, OpenCV, Mediapipe, Tkinter, React.js, Firebase, Computer vision and Hand tracking..

I. INTRODUCTION

The "AirInk Studio" project emerges at the intersection of digital artistry and online education, responding to the longstanding challenges faced by educators, students, and artists alike. In the realm of digital drawing and illustration, conventional input devices like mice have proven inadequate, impeding the fluidity and precision essential for effective teaching and learning, particularly within visually dependent subjects. Furthermore, the prohibitive cost of specialized drawing tablets has served as a barrier, limiting access to the tools necessary for unleashing creative expression in the digital domain. Recognizing these barriers, the "AirInk Studio" project sets out to revolutionize the digital art landscape by introducing an innovative desktop application.

Powered by a sophisticated blend of Python, OpenCV, Mediapipe, and Tkinter technologies, this application harnesses the power of computer vision and hand tracking to provide users with a seamless drawing experience. Offering a diverse array of brush styles, an intuitive undo feature, and an extensive color palette, the project caters to a wide spectrum of artistic preferences. Beyond its utility as a creative tool, the "AirInk Studio" project aims to redefine online education by providing educators and students with a comprehensive digital canvas for engaging visual learning experiences. Through its ground breaking features and user-centric design, the project endeavours to democratize creativity in the virtual space, empowering individuals to express themselves artistically and facilitating the seamless integration of digital artistry into educational settings.

II. RELATED WORK

The research paper on "Text recognition by air drawing" presents insights that closely align with the goals of the "AirInk Studio" project. By exploring methods for recognizing text drawn in the air, the paper offers valuable techniques that can be integrated into the application. This integration would enable users to incorporate text into their digital artwork or educational content, enhancing the platform's versatility. Additionally, the paper's methodologies could inspire improvements to the precision and fluidity of the hand tracking module, making the application suitable for educational presentations, online tutorials, and collaborative work. Overall, leveraging the findings of this research paper has the potential to enhance the capabilities of the "AirInk Studio" project, advancing its mission to reshape online education and foster creativity in the virtual space [1].

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 3, March 2024

DOI: 10.17148/IJARCCE.2024.133131

A review paper on hand gesture recognition systems offers valuable insights for the "AirInk Studio" project, which leverages computer vision and hand tracking to enhance digital drawing and image creation. By delving into the latest advancements and challenges in hand gesture recognition, the review provides essential knowledge on cutting-edge methods and algorithms. This understanding is crucial for refining the precision and effectiveness of the project's hand tracking module, ensuring a seamless drawing experience for users. Additionally, the review may explore diverse applications of hand gesture recognition beyond art and education, inspiring innovative use cases and integrations for the "AirInk Studio" platform. Incorporating insights from the review can help the project evolve into a versatile tool, serving a broader audience and democratizing creativity in the virtual space [2].

The research paper "Hand Gesture Recognition Algorithms for Minimizing Computational Cost" holds significant relevance for the "AirInk Studio" project, aimed at providing efficient hand tracking for digital drawing and image creation. This paper likely offers techniques to optimize computational efficiency in real-time applications like "AirInk Studio," ensuring smooth performance across various hardware. By studying this paper, the project team can gain valuable strategies for reducing computational load while maintaining accuracy, enhancing accessibility to a wider audience. Moreover, the paper's insights may inspire the implementation of a more sustainable hand tracking system, extending usability to lower-end devices and improving overall user experience. Ultimately, this research can guide the project towards a more inclusive and accessible digital drawing tool, aligned with the goal of democratizing creativity in the virtual space [3].

III. METHODOLOGY

A. System Overview

The AirInk Studio system offers an innovative virtual drawing space that combines hand tracking technology with an intuitive interface, enabling real-time digital art creation. Users can draw, erase, and shape with a variety of brushes and features, including text overlay and image import. Enhanced by a color palette and eraser tool, the system promotes creativity and precision. Beyond drawing, AirInk Studio connects users through a Showcase Community and Chat feature, built with React.js and Firebase, fostering a collaborative and interactive artistic community.

- User Interface

The UI is engineered for user-friendly navigation and interaction, offering a suite of drawing tools, text overlay options, image import capabilities, and a dynamic color palette. It integrates the Chat app and Community Showcase app, providing platforms for users to engage and share their creations.

Hand Tracking Module: This module utilizes the computer's webcam and the OpenCV library to detect and track user hand movements, enabling intuitive control over the drawing interface and seamless interaction with the Chat app and Community Showcase app.

Canvas Rendering: The rendering engine is responsible for processing user inputs and hand gestures, updating the digital canvas in real time with drawings, shapes, text annotations, and imported images. It supports various brush styles and image manipulation features, enhancing the creative experience.

- Application Logic

Gesture Recognition: Employing OpenCV, this component interprets hand gestures, translating them into commands for tool selection, drawing, and interface navigation.

Image Processing: Handles real-time image processing tasks, including rendering user drawings, managing layer operations, and integrating imported images onto the canvas.

Community Interaction: Integrates with React.js and Firebase to support features like Showcase Community and Chat, enabling users to share their creations and communicate within the platform.

B. System Architecture

The user's hand movements are captured and processed for gesture recognition, enabling interaction with the UI for tool selection and canvas manipulation.

Drawing actions and tool selections are processed in real time, with the canvas reflecting changes immediately to provide an interactive drawing experience.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 3, March 2024

DOI: 10.17148/IJARCCE.2024.133131

The application supports community engagement and user collaboration through the Showcase Community and Chat features, enhancing the user experience beyond the drawing capabilities.

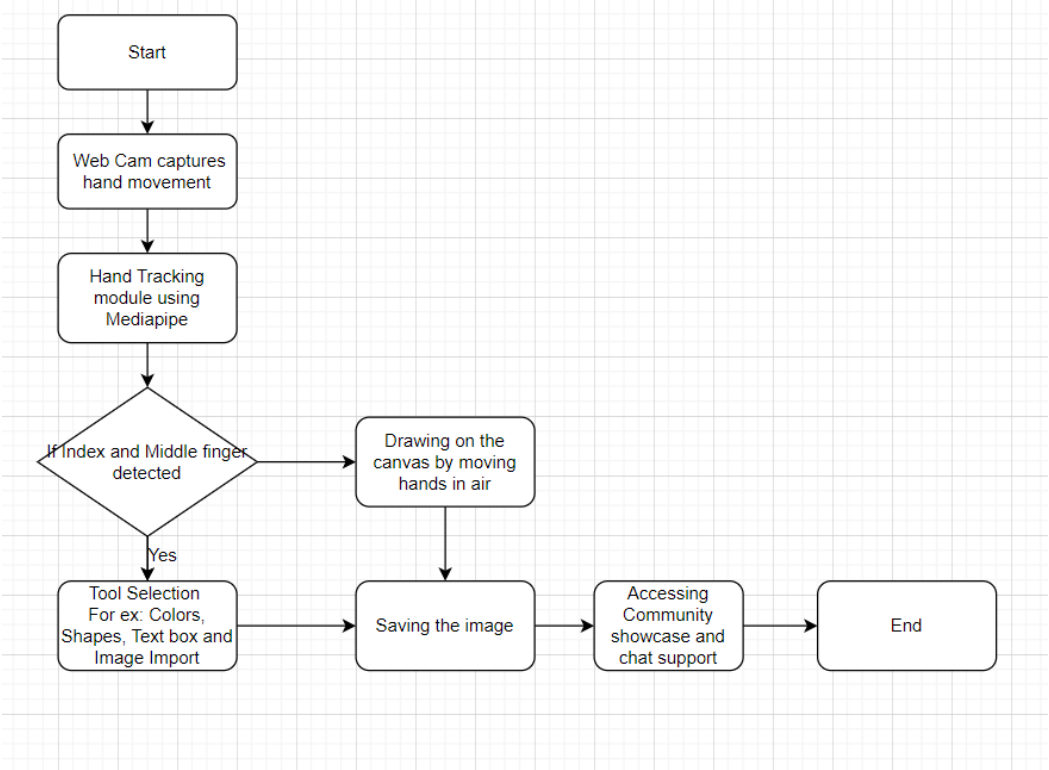

Fig. 1 System Architecture Diagram

C. Algorithm

Begin

Initialize webcam and hand tracking module
Initialize drawing canvas
Initialize default brush settings
Initialize color palette and trackbar controls

While the application is running:
Capture frame from webcam
Detect and track hand gestures using hand tracking module

If hand gesture is recognized as selection mode:
Update user interface based on hand position (e.g., select brush, color, or tool)

If drawing mode is active:
If line drawing mode is selected:
Draw line based on hand position and trackbar settings
Else if rectangle drawing mode is selected:
Draw rectangle based on hand position and trackbar settings
Else if circle drawing mode is selected:
Draw circle based on hand position and trackbar settings
Else:
Draw on canvas based on hand position and selected brush settings

If text mode is active:
Prompt user to input text
Display entered text on canvas at the position indicated by hand

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 3, March 2024

DOI: 10.17148/IJARCCE.2024.133131

If image import mode is active:
Load selected image onto canvas at the position indicated by hand

If undo action is triggered:
Revert the last action on the canvas

If save action is triggered:
Save the current canvas as an image file

Update the display to show the latest canvas state

End While
End

IV. RESULT AND DISCUSSION

A. Implementation and Result

The implementation of AirInk Studio encompasses the integration of a hand-tracking module with a dynamic user interface, enabling real-time drawing and editing on a virtual canvas. Utilizing the webcam, the system detects user hand gestures to select tools, draw shapes, input text, and manipulate images. Customizable brush styles and color selections enhance user creativity, while additional features like undo functionality and image import/export offer flexibility in the creative process. The result is a responsive and intuitive drawing environment where users can freely express their artistic vision. The addition of community interaction through the Showcase Community and Chat with Others enhances user engagement, making AirInk Studio not just a drawing tool but a platform for collaborative creativity and sharing.

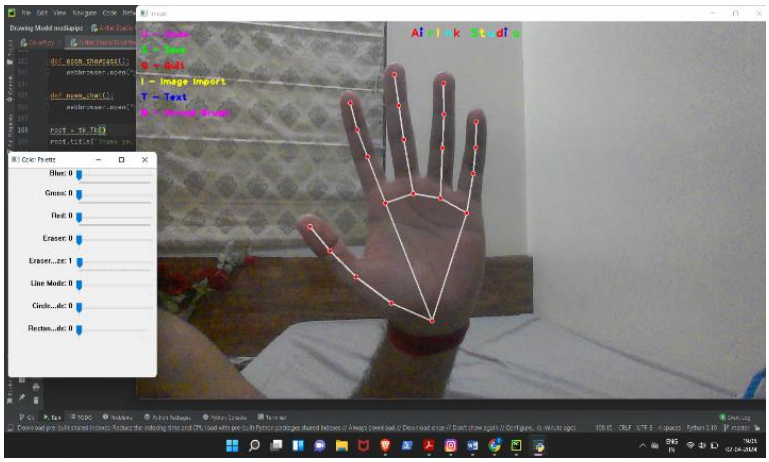

Fig. 2 Hand Detection using Mediapipe

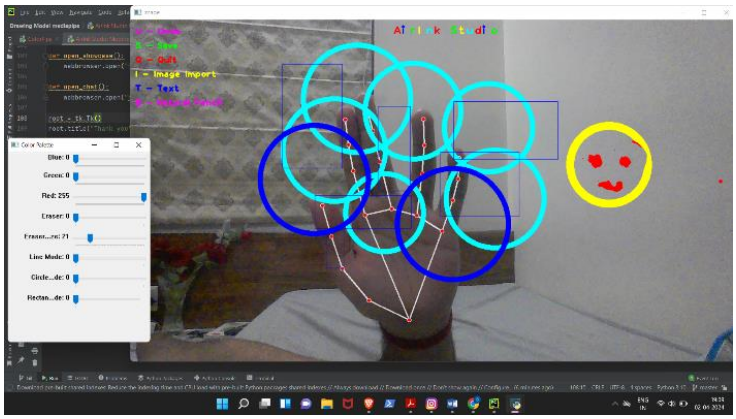

Fig. 3 Drawing on the Canvas

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 3, March 2024

DOI: 10.17148/IJARCCE.2024.133131

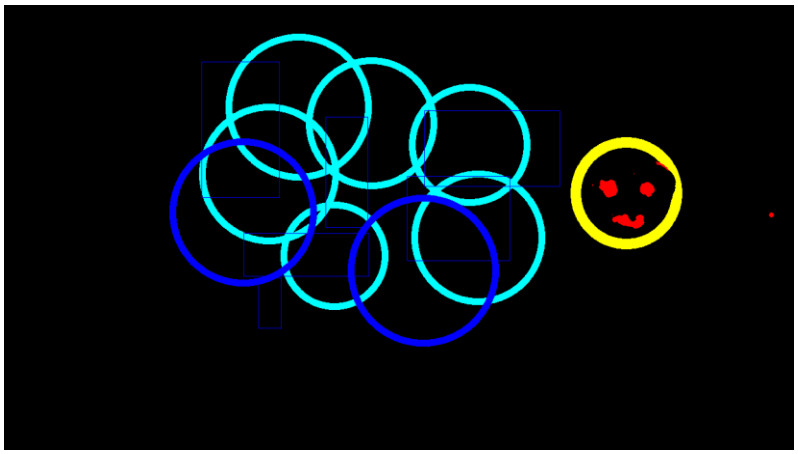

Fig. 4 Final Result

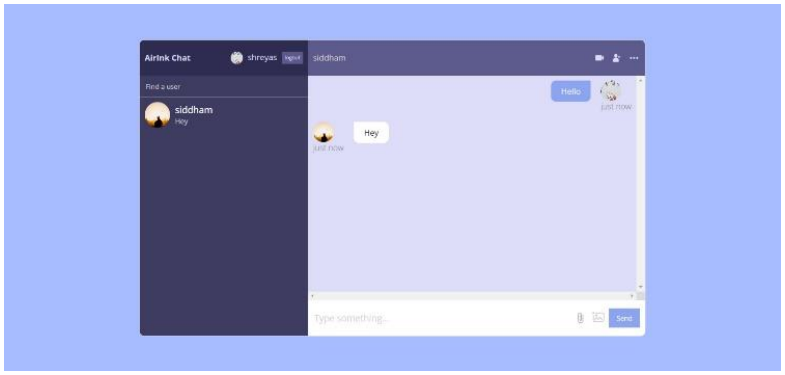

Fig. 5 Chat Feature

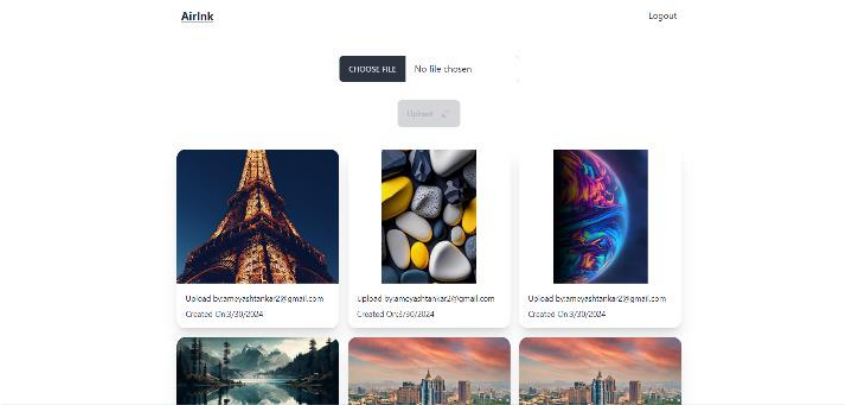

Fig. 6 Community Showcase

B. Test Cases table

To ensure the robustness and usability of the AirInk Studio application, a series of test cases were developed and executed. These test cases are designed to rigorously evaluate the application's core functionalities, including hand detection accuracy, drawing capabilities, color and shape tool effectiveness, text input functionality, undo feature reliability, image import accuracy, and the integration of community showcase and chat features.

The objective is to validate that the application behaves as expected under various scenarios, providing a seamless and intuitive user experience. This testing phase is crucial for identifying and rectifying any potential issues, thereby enhancing the overall quality and performance of the application.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 3, March 2024

DOI: 10.17148/IJARCCE.2024.133131

Test ID	Test Case	TEST SCENARIO	TEST INPUT	EXPECTED OUTPUT	ACTUAL OUTPUT	TEST RESULT
1	Hand detection test	Placing hand in front of web camera	Hand in front of web camera	Detecting hand correctly with Mediapipe hand points.	Detecting hand correctly with Mediapipe hand points	Passed
2	Drawing test	Drawing using index finger	Index finger detection and moving it in front of web camera	Drawing getting rendered on the canvas if color is selected.	Drawing getting rendered on the canvas if color is selected.	Passed
3	Tool window accessibility	Tool window must be accessible on the left side of the canvas to select various tools	Access tools inside the tools window by using mouse	Selection of tool visible on the canvas	Selection of tool visible on the canvas	Passed
4	Color Selection	Selecting suitable color from tools window	Selecting color from tools window or mixing the color with other color using the mouse	Correct and selected color should be rendered on the screen as an output	Correct and selected color rendered on the screen as an output	Passed
5	Shape drawing	Selecting type of shape from the tools window to draw on the canvas	Selecting a specific type of shape from the tools window using a mouse	The selected shape should get rendered while drawing on the canvas	The selected shape got rendered while drawing on the canvas	Passed
6	Text and Image import	Text and Image tool can be accessible using ‘T’ and ‘I’ keyboard keys	Pressing ‘T’ or ‘I’ keyboard key for text or image import feature	The given text or image imported should be rendered on the canvas	The given text or image imported rendered on the canvas	Passed
7	Save option	The canvas can be downloaded by pressing ‘S’ key on the keyboard	Pressing ‘S’ key on the keyboard	The canvas should be downloaded at user defined location	Canvas downloaded at user defined location	

Table. 1 Test Cases

V. SYSTEM REQUIREMENTS

The technical feasibility of our project largely depends on the compatibility of hardware and software. In this context, we have successfully identified and analysed the necessary components:

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 3, March 2024
DOI: 10.17148/IJARCCE.2024.133131

Hardware Requirements:
• Webcam: We require a webcam for hand gesture recognition, and our project has been tested on standard webcams, making it highly compatible with a wide range of setups.
• Computer: The application runs on standard personal computers without any specialized hardware requirements. This enhances accessibility and usability.

Software Requirements:
• Operating System: The application is developed for Windows, MacOS, and Linux environments, ensuring compatibility across major operating systems.
• OpenCV: We use the OpenCV library for image processing and hand tracking. This open-source framework is widely adopted and supports various platforms.
• Python: Our project is implemented in Python, a versatile and accessible programming language with extensive libraries.
• Mediapipe: We have used Mediapipe library to implement Hand Detection module.
• React.js: We have used React.js framework to implement chat and community showcase.
• Firebase: Firebase is utilized in the chat and community showcase feature to provide real-time authentication capability.

VI. CONCLUSION

The AirInk Studio project successfully demonstrates the integration of hand tracking technology with an interactive drawing interface, enhancing the creative process in a digital environment. The application's robust feature set, including real-time drawing, shape creation, text input, image import, and unique community interaction capabilities, underscores its potential as a versatile tool for artists, designers, and casual users alike. By leveraging advanced hand tracking and computer vision techniques, AirInk Studio offers an intuitive and accessible platform for digital artistry, showcasing the possibilities of combining traditional art methods with cutting-edge technology. This project not only exemplifies innovation in human-computer interaction but also sets a foundation for future developments in the realm of virtual art creation.

ACKNOWLEDGMENT

We have great pleasure in presenting the research paper on “AirInk Studio: A Virtual Drawing Model”. We take this opportunity to express my sincere thanks towards the staff of Sinhgad College of Engineering, Computer Department, Pune, for providing the technical guidelines, and the suggestions regarding the line of this work. We would specially like to thank Prof. Archana Dirgule from Sinhgad College of Engineering, Computer Department, for her encouragement and guidance.

REFERENCES

[1]. Jay Patel, Umang Mehta, Kevin Panchal, Dev Taylo, Devam Zanzmera, “Text Recognition by Air Drawing” IEEE 2021.
[2]. Norani Mohamed, Mumtaz Begum Mustafa, Nazean Jomhari “A Review of the Hand Gesture Recognition System: Current Progress and Future Directions” IEEE 2021.
[3]. Roman Golovanov. Dmirtry Vorotnev, Darina Kalina “Hand Gesture Recognition Algorithms for Minimizing Computational Cost” IEEE 2020
[4]. Sachin Agrwal, Agnishrota Chakraborty, M. Rajalakshmi, “Real-Time Hand Gesture Recognition System Using Mediapipe and LSTM” International Journal of Research Publication and Reviews.
[5]. Feliz Zhan “Hand Gesture Recognition with Convolution Neural Network” IEEE 2020.
[6]. Hamzah Luqman “An Efficient Two-Stream Network for Isolated Sign Language Recognition Using Accumulative Video Motion” IEEE 2022.
[7]. Munir Oudah, Ali Al-Naji, Javaan Chahl “Hand Gesture Recognition Based on Computer Vision: A Review of Techniques” Journal of Imaging 2020.
[8]. Indriani, Moh.Harris, Alo Suryaperdana Agoes “Applying Hand Gesture Recognition for User Guide Application Using Mediapipe” International Seminar of Science and Applied Technology.
[9]. Ramadan TH. Hasan, Amira Bibo Sallow “Face Detection and Recognition Using OpenCV” Journal of Soft Computing and Data Mining 2021.
[10]. Shubham Mishra, Versha Verma, Dr. Nikhat Akhtar, Shivam Chaturvedi, Dr. Yusuf Perwej “An Intelligent Motion Detection” International Journal of Scientific Research in Science, Engineering and Technology 2022.