Journal of Advanced Research in Computer Graphics and Multimedia Technology
Volume 11, Print Issue 1 - 2024, Pg. No. 18-23
Peer Reviewed Journal

Review Article

Touch-Free text Entry: Revolutionizing Input Methods Through Fingertip-Based Air Writing

Heena[1], Sandeep Ranjan[2]

[1]PhD Scholar, [2]Professor, CT University, Jagraon, Punjab, India.

INFO

Corresponding Author:
Heena, CT University, Jagraon, Punjab, India.
E-mail Id:
heenajand21@gmail.com
Orcid Id:
https://orcid.org/0000-0002-1729-7504

How to cite this article:
Heena, Ranjan S. Touch-Free text Entry: Revolutionizing Input Methods Through Fingertip-Based Air Writing. *J Adv Res Comp Graph Multim Tech.* 2024; 11(1): 18-23.

Date of Submission: 2024-03-05
Date of Acceptance: 2024-04-12

ABSTRACT

This paper explores the innovative realm of touch-free text entry, specifically focusing on the paradigm shift brought about by fingertip-based air writing. Traditional input methods often require physical contact, limiting user interaction in various scenarios. In response to this challenge, our research delves into the development and evaluation of a novel system that enables users to input text seamlessly through gestures in the air using their fingertips. The objective of this study is to assess the feasibility, accuracy, and user experience of fingertip-based air writing as a touch-free text entry method. We present a detailed description of the underlying technology, including the algorithms and sensors employed in capturing and interpreting fingertip movements. Our methodology involves user trials with diverse participants, evaluating the system's performance across different contexts and linguistic patterns. Results indicate promising potential for fingertip-based air writing as an intuitive and efficient touch-free text entry method. Participants demonstrated quick adaptability to the system, achieving comparable typing speeds to conventional methods. The implications of this research extend beyond conventional touch-based interfaces, offering a transformative approach to human-computer interaction. In conclusion, this paper contributes to the ongoing evolution of input methods by introducing and validating fingertip-based air writing. The findings underscore the viability of touch-free text entry, paving the way for future developments in user-friendly interfaces and expanding the horizons of interactive technology.

Keywords: Touch-Free Input, Air Writing, Fingertip-Based Interaction, Human-Computer Interaction, Text Entry Revolution

Introduction

Designers were all well aware of the fact that writing and art are the most common and effective ways for us to convey our feelings. Writing and printing were once commonplace on rocks, walls, and other surfaces, but with time, these practices evolved. Humans had found new plants, extracted the minerals, and made paper from them. Next, they produced various synthetic paper types. This is too far in this modern era.[28] Individuals use electronic devices such as screens, on which we may write, draw, and express many kinds of artwork, symbols, painting, and other media by hand, by typing, or even just by tapping. However, we are now going a step further with this approach, making it simpler to use and more straightforward to communicate your emotions with a few tiny finger movements.[30] As is well known, the canvas is a unique kind of surface that is utilized for painting. And now try writing or painting anything on a surface that is devoid of any tangible objects. Is it feasible? Yes, air canvas makes this possible. A free-hand digital drawing is called an "air canvas." To convey something, all you need to do is think about it and then move your finger in the appropriate direction.[1] The Air Writing App offers customers a cutting-edge and captivating digital drawing experience by utilizing cutting-edge technologies. Modern elements like gesture detection, Augmented Reality (AR), and collaborative drawing are all incorporated into the project workflow. This cutting-edge technology project adheres to a methodical development and deployment process.

Project Planning and Requirement Gathering

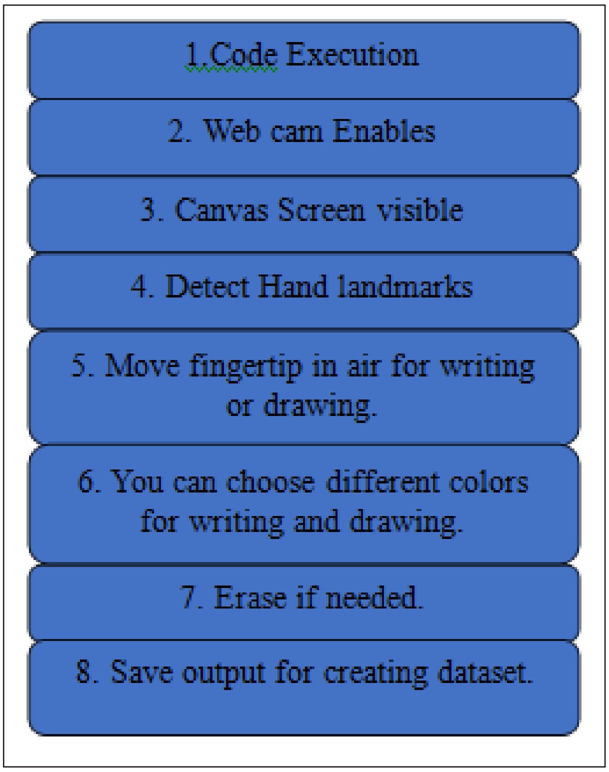
1.Code Execution
2. Web cam Enables
3. Canvas Screen visible
4. Detect Hand landmarks
5. Move fingertip in air for writing or drawing.
6. You can choose different colors for writing and drawing.
7. Erase if needed.
8. Save output for creating dataset.

Figure 1.Representation of the Flow chart of the Project

Core Functionality:

- Capture hand movements using a webcam.
- Track Fingertip positions for drawing.
- Provide a virtual canvas for air writing.
- Allow users to select colors and erase drawings.
- Save the created drawing as images.

Additional Features (Potential):

- Enable text input for written notes.
- Allow image importing for tracing or annotation.

Technology Stack Selection

Python: Renowned for its readability and flexibility, Python is a high-level, flexible programming language.[31] Owing to its vast library and framework ecosystem, it is a well-liked choice for numerous applications, including web development, data analysis, artificial intelligence, and more.[9] Because Python is so simple to use and has so many libraries, it's an excellent language for creating laptop vision apps and prototypes.[11]

Tkinter: The standard GUI (Graphical User Interface) toolkit included with Python is called Tkinter. It offers a collection of tools and frameworks to design graphical user interfaces for desktop programs.[36] Tkinter is a freely available, user-friendly framework for creating interactive and aesthetically pleasing applications.

Here are some key points about Tkinter

1. **Cross-Platform:** Tkinter is cross-platform, because of this feature programs created with it can function without change on a variety of operating systems, such as Windows, macOS, and Linux.
2. **Simple and Lightweight:** Tkinter is renowned for being straightforward and user-friendly. It is a lightweight library that can be easily used by beginners and those who want to quickly construct simple GUI apps because it doesn't require any additional installations or dependencies.[37]
3. **Widgets:** Many GUI elements, or widgets, like buttons, labels, text boxes, check buttons, radio buttons, and more are available with Tkinter. The layout of an application can be changed by arranging and modifying these widgets.
4. **Event-Driven Programming:** Event-driven programming is the paradigm used by Tkinter. Keyboard input and button clicks are examples of actions that cause events that can be handled by particular functions or methods. This makes it possible for developers to produce interactive and responsive apps.[38]
5. **Integration with Python:** Python and Tkinter combine to form a seamless whole. Since it is a component of the standard library, the Python installation comes pre-installed with it. To begin utilizing Tkinter, no extra setups or installations are needed.

Here's a one-line code to install Tkinter in Python programming:

Import tkinter as tk

OpenCV: OpenCV is the name of an open-source computer vision library.[8] The C and C++ library runs on Linux and Windows and provides interfaces for Python, Ruby, MATLAB, and other languages. Access advanced functions in image processing, computer vision, and algebra using the OpenCV library.[18] The OpenCV computer vision library contains techniques for object recognition picture processing.[14] With the Python programming language, real-time computer vision applications can be developed with the OpenCV library. The OpenCV library is used for image and video processing, as well as analytical methods like object and face detection.[35]

MediaPipe: MediaPipe was more of a collection of libraries and tools for various computer vision and machine learning tasks than it was a standalone Python library. Rather, Media Pipe offered pre-trained models and tools that could be incorporated into Python programs.[8] Before utilizing Python to communicate with Media Pipe models and pipelines, you typically had to install the required dependencies.[19] The following are the standard protocols for utilizing Media Pipe in a Python project:

Install the Required Libraries: Usually, installing Media Pipe and its dependencies is required. For this, pip can be used:

pip install mediapipe

Import and utilize the MediaPipe Modules: After installing the library, you can import the necessary modules and utilize them in your Python code.

Import mediapipe as mp

Hand Landmark Detection: Hand landmarks are unique locations or landmarks on a person's hand that are commonly used in computer vision and machine learning applications for tasks like hand tracking and gesture identification.[27] These landmarks, which represent significant regions of the hand's surface such as finger joints and fingertips, aid in the understanding of hand motions and poses.[29] A pre-trained hand-tracking model may be found in the Google product MediaPipe, which uses a camera feed to identify and track these hand landmarks in real-time.[4]

Numpy: NumPy offers data structures that are utilized while utilizing OpenCV with Python. Numerical Python, or NumPy for short, is a library that includes multidimensional array objects along with several array processing techniques.[33] NumPy can be used to conduct logical and mathematical operations on arrays. NumPy is a module for Python. Numerical Python is what it stands for.[34]

Pillow (PIL): A potent image-processing library for Python is the Pillow library, sometimes known as PIL (Python Imaging Library). It has several features for accessing, modifying, and storing different kinds of picture files.

Among the Pillow Library's principal attributes are:

1. **Image Manipulation:** Pillow can do standard image manipulation functions, including scaling, rotating, flipping, and cropping.
2. **Image Enhancement:** It offers instruments to improve the quality of images by performing operations such as sharpening, contrast, brightness, and filtering.
3. **Image Formats:** Pillow is compatible with several image file formats, including widely used ones like JPEG, PNG, GIF, TIFF, and BMP.
4. **Image Drawing:** You can draw text, shapes, and other features on images using the library.
5. **Image Filtering:** Pillow can handle edge recognition, sharpening, and blurring, among other image filtering methods.

To use Pillow, you can install it using the following:

pip install Pillow

User Interface

1. **Camera Frame:** The camera frame is used to capture the hand and finger movements of the user and translate them into desired action.
2. **Paint Frame:** Paint frame is used to draw the captured data by camera frame on the virtual canvas.
3. **Buttons:** Different buttons are used for different functionalities.
4. **Clear:** Clear button is used to clear the screen.
5. **Draw extra:** This option is used when the user wants to draw multiple things.
6. **Erase:** used to erase the content with fingertips.
7. **Color buttons:** Provide different colors for writing and drawing.
8. **Save Image:** used to save the canvas screen as an image (yet to be implemented)
9. **Quit:** Used to quit the application.

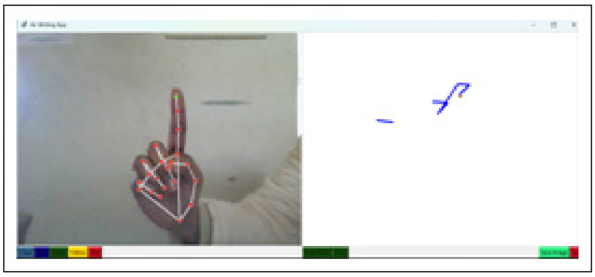

Figure 2.Representation of Air writing App

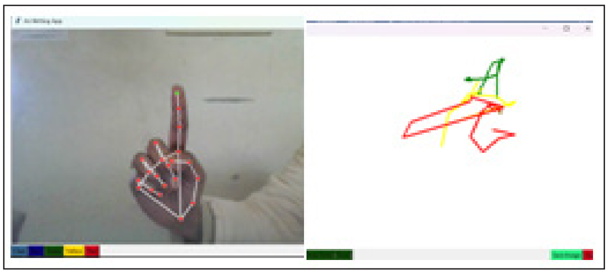

Figure 3.Represents working of different color buttons

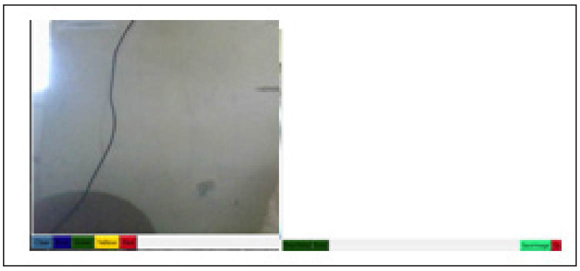

Figure 4.Represents working of clear button

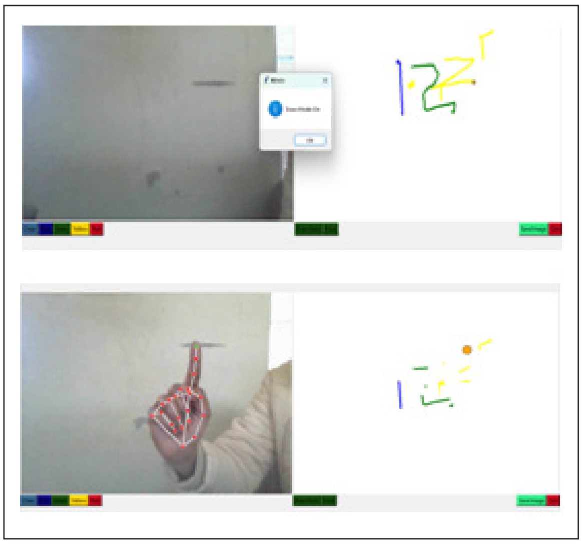

Figure 5.Represents Working of Eraser

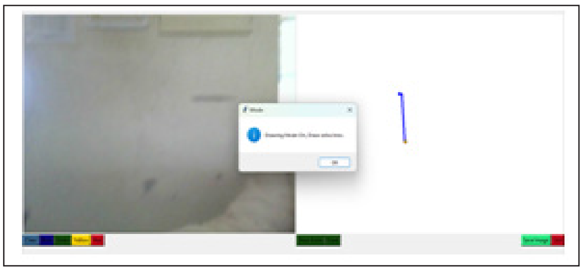

Figure 6.Represents Working of Draw Extra

Development

Main loop: Continuously update the camera view and canvas using root. after(20,update_camera_view).

Update_camera_view() Function: Reads a frame from the webcam.

Detects hands using MediaPipe: If a hand is detected then it tracks the fingertip positions. Draw the landmarks on the frame for visual feedback. Then it determines drawing actions based on finger positions and updates the virtual canvas with corresponding lines or erasures.

Machine Learning Integration

Mediapipe Hands: It's a Pre-Trained machine-learning model for real-time hand tracking and landmark detection .[7]

Key Functions

hands.Hand(): Initializes the hand tracking model.

hands.Process(): Processes a frame to detect hands and landmarks.

mpDraw.draw_landmarks(): Draws landmarks on a frame for visualization.

Conclusion

In conclusion, the development of touch-free text entry systems, especially through fingertip-based air writing, signifies a revolutionary change in input techniques. This cutting-edge technology creates new opportunities for improved accessibility, ease, and hygiene in addition to addressing the drawbacks of conventional input techniques. Touch-free text entry systems facilitate seamless communication and engagement in a variety of scenarios, from routine chores to specialized applications, by utilizing gestures and motion tracking to enable users to connect with devices in more intuitive and natural ways.

Future developments and improvements in touch-free text entry systems have enormous potential to completely transform how we use digital interfaces and gadgets. We should expect even higher accuracy, speed, and adaptability in touch-free text entry systems with continued improvements in sensor technology, machine learning algorithms, and user interface design. Furthermore, the incorporation of these systems into an extensive array of gadgets, such as wearables, tablets, smartphones, and augmented reality devices, will enhance their influence and practicality across various user segments and scenarios.

To accommodate users with a range of skills and preferences, inclusive design approaches are necessary, and privacy concerns and learning curves are only a few of the potential issues that must be addressed when implementing touch-free text entry.

In addition, we continue to work on this project to include other capabilities like cloud-based storage and drawing sharing, as well as collaborative drawing, which synchronizes canvases across several devices via network communication. Possible Enhancements Such as error handling for MediaPipe or webcam problems, and improvements to the user experience (such as smoother drawing, more user-friendly controls, etc.). Touch-free text entry system development and implementation may fully realize the potential of these technologies to improve communication, productivity, and quality of life for people worldwide if privacy protection, accessibility, and user experience are given top priority.

In summary, fingertip-based air writing technology, which enables touch-free text entry, has the potential to fundamentally alter the field of human-computer interaction. It provides an early look at a time when human gestures will be seamlessly translated into digital input, enhancing our digital experiences and opening up previously uncharted territory.

References

1. Li, K., & Zhang, X. (2014, October). A new fingertip detection and tracking algorithm and its application on writing-in-the-air system. In 2014 7th International Congress on Image and Signal Processing (pp. 457-462). IEEE..
2. Wani, A., Joshi, I., Khandve, S., Wagh, V., & Joshi, R. (2021). Evaluating deep learning approaches for covid19 fake news detection. In Combating Online Hostile Posts in Regional Languages during Emergency Situation: First International Workshop, CONSTRAINT 2021, Collocated with AAAI 2021, Virtual Event, February 8, 2021, Revised Selected Papers 1 (pp. 153-163). Springer International Publishing.
3. Li, C., Zhang, R., Liu, Z., Hang, C., & Li, Z. (2017, December). Algorithm of fingertip detection and its improvement based on kinect. In 2017 International Conference on Industrial Informatics-Computing Technology, Intelligent Technology, Industrial Information Integration (ICIICII) (pp. 63-66). IEEE.
4. Gopinath, N., Anuja, J., Anusha, S., & Monisha, V. (2020). A Survey on Hand Gesture Recognition Using Machine Learning. International Research Journal of Engineering and Technology (IRJET).
5. Baig, F., Fahad Khan, M., & Beg, S. (2013). Text writing in the air. Journal of Information Display, 14(4), 137-148.
6. Ramasamy, P., Prabhu, G., & Srinivasan, R. (2016, April). An economical air writing system converting finger movements to text using web camera. In 2016 International Conference on Recent Trends in Information Technology (ICRTIT) (pp. 1-6). IEEE.
7. Joseph, V., Talpade, A., Suvarna, N., & Mendonca, Z. (2018, June). Visual gesture recognition for text writing in air. In 2018 Second International Conference on Intelligent Computing and Control Systems (ICICCS) (pp. 23-26). IEEE.
8. RamachandraH, V., Balaraju, G., Deepika, K., & Sebastian, S. R. (2022, November). Virtual Air Canvas Using OpenCV and Mediapipe. In 2022 International Conference on Futuristic Technologies (INCOFT) (pp. 1-4). IEEE
9. Verma, P. K., & Ojha, P. K. Air Drawing Using Python.
10. C. Science et al., “Air-Writing Alphanumeric Character Recognition,” vol. 10, no. 7, pp. 548–559, 2022.
11. A. Kumar, A. K. Jha, A. Shandilya, S. K. Dubey, and A. Chaitra, “Virtual Air Board Using Python,” Int. Res. J. Mod. Eng. Technol. Sci., no. 05, pp. 3110–3112, 2023, doi: 10.56726/irjmets39154.
12. K. Navya, M. Sowmyah, S. A. Amreen, V. Sravani, Y. G. Devi, and M. P. N. Bhanu, “A Machine Learning Approach for Air Writing Recognition,” Int. J. Res. Appl. Sci. Eng. Technol., vol. 11, no. 4, pp. 3528–3535, 2023, doi: 10.22214/ijraset.2023.50991.
13. Kumar, S., Srivastava, M. D., Singh, U., Yadav, A. S., & Agrawal, A. Motion Canvas.
14. Nethravathi, T. L., Patil, R. L., Bhavana, S., Choudhury, S. R., & Monisha, S. (2022). VIRTUAL PAINTER USING ARTIFICIAL INTELLIGENCE AND OPENCV. International Research Journal of Modernization in Engineering Technology and Science, 4(6), 3617-3620.
15. Dhaigude, S., Bansode, S., Waghmare, S., Varkhad, S., & Suryawanshi, S. (2022). Computer vision based virtual sketch using detection.
16. A. Kr Pandey and M. Tripathi, “Air Writing Using Python (2021-2022),” Int. Res. J. Eng. Technol., no. May, pp. 3009–3013, 2022, [Online]. Available: www.irjet.net.
17. Tripathi, A., Mondal, A. K., Kumar, L., & Prathosh, A. P. (2022). ImAiR: Airwriting recognition framework using image representation of IMU signals. IEEE Sensors Letters, 6(10), 1-4.
18. Saoji, S. U., Dua, N., Choudhary, A. K., & Phogat, B. (2021). Air canvas application using Opencv and numpy in python. IRJET, 8(08).
19. Hsieh, C. H., Lo, Y. S., Chen, J. Y., & Tang, S. K. (2021). Air-writing recognition based on deep convolutional neural networks. IEEE Access, 9, 142827-142836.
20. Alam, M. S., Kwon, K. C., & Kim, N. (2019, September). Trajectory-based air-writing character recognition using convolutional neural network. In 2019 4th International Conference on Control, Robotics and Cybernetics (CRC) (pp. 86-90). IEEE.
21. Sivamani, K., Seetharaman, R., & Nedumaran, D. (2019, January). Design and implementation of gesture recognition and writing text virtually using finger. In 2019 Third International Conference on Inventive Systems and Control (ICISC) (pp. 48-51). IEEE.
22. Puranik, P., Sankhe, T., Singh, A., Vishwakarma, V., & Rane, P. (2019, July). AirNote–Pen it Down!. In 2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT) (pp. 1-7). IEEE.
23. Kurian, C. E., Martin, C., Deepthi, P. J., & Eldhose, K. A. (2018). Air Writing Recognition and Speech Synthesis.
24. Lakshmi, D. U., & Harish, B. (2017, August). A novel air writing recognition system using Raspberry Pi for the control and interaction of digital systems. In 2017 International Conference on Energy, Communication, DataAnalytics and Soft Computing (ICECDS) (pp. 3800-3804). IEEE.
25. Huang, Y., Liu, X., Zhang, X., & Jin, L. (2016). A pointing

gesture based egocentric interaction system: Dataset, approach and application. In Proceedings of the IEEE conference on computer vision and pattern recognition workshops (pp. 16-23).

26. Ye, Z., Zhang, X., Jin, L., Feng, Z., & Xu, S. (2013, July). Finger-writing-in-the-air system using Kinect sensor. In 2013 IEEE international conference on multimedia and expo workshops (ICMEW) (pp. 1-4). IEEE.
27. Ayachi, N., Kejriwal, P., Kane, L., & Khanna, P. (2015, April). Analysis of the hand motion trajectories for recognition of air-drawn symbols. In 2015 Fifth International Conference on Communication Systems and Network Technologies (pp. 505-510). IEEE.
28. Gopinath, N., Anuja, J., Anusha, S., & Monisha, V. (2020). A Survey on Hand Gesture Recognition Using Machine Learning. International Research Journal of Engineering and Technology (IRJET)..
29. Abhishek, B., Krishi, K., Meghana, M., Daaniyaal, M., & Anupama, H. S. (2020). Hand gesture recognition using machine learning algorithms. Computer Science and Information Technologies, 1(3), 116-120.
30. Kumar, P., Saini, R., Behera, S. K., Dogra, D. P., & Roy, P. P. (2017, May). Real-time recognition of sign language gestures and air-writing using leap motion. In 2017 Fifteenth IAPR international conference on machine vision applications (MVA) (pp. 157-160). IEEE. [31] Saoji, S. U., Dua, N., Choudhary, A. K., & Phogat, B. (2021). Air canvas application using Opencv and numpy in python. IRJET, 8(08). [32] Zou, Y., Yang, Q., Han, Y., Wang, D., Cao, J., & Wu, K. (2019, March). AcouDigits: Enabling users to input digits in the air. In 2019 IEEE International Conference on Pervasive Computing and Communications (PerCom (pp. 1-9). IEEE..
31. Usha, V., Arivubrakan, P., & Balaji, K. S. (2024). Webcam painting canvas window application using OpenCV and Numpy. In Artificial Intelligence, Blockchain, Computing and Security Volume 2 (pp. 350-354). CRC Press. [34] Kumar, Subhang, et al. “Motion Canvas.”
32. Saoji, S. U., Dua, N., Choudhary, A. K., & Phogat, B. (2021). Air canvas application using Opencv and numpy in python. IRJET, 8(08).
33. Moore, A. D. (2018). Python GUI Programming with Tkinter: Develop responsive and powerful GUI applications with Tkinter. Packt Publishing Ltd.
34. Akbar, M. G., Witriyono, H., Apridiyansyah, Y., & Abdullah, D. (2023). Implementation Of The Inter Tk Package, Sub-Process And Os In The Network Management Application Development With Python Programming Language. Jurnal Komputer, Informasi dan Teknologi (JKOMITEK), 3(1), 187-196.
35. Zaman, H. U., Mahmood, S., Hossain, S., & Shovon, I. I. (2018, October). Python based portable virtual text reader. In 2018 Fourth International Conference on Advances in Computing, Communication & Automation (ICACCA) (pp. 1-6). IEEE.
36. Kang, Z., Xiao, E., Li, Z., & Wang, L. (2024). Deep Learning Based on ResNet-18 for Classification of Prostate Imaging- Reporting and Data System Category 3 Lesions. Academic Radiology.
37. Rakesh, K., Singh, B. S., Surya, A., & Kaladevi, R. (2024). Image recognition based attendance system with spoof detection. In Artificial Intelligence, Blockchain, Computing and Security Volume 2 (pp. 482-486). CRC Press.
38. Toto, L., Romano, A., Pavan, M., Degl’Innocenti, D., Olivotto, V., Formenti, F., ... & Mastropasqua, R. (2024). A Deep Learning Approach to Hard Exudates Detection and Disorganization of Retinal Inner Layers Identification on OCT images.
39. Ayachi, N., Kejriwal, P., Kane, L., & Khanna, P. (2015, April). Analysis of the hand motion trajectories for recognition of air-drawn symbols. In 2015 Fifth International Conference on Communication Systems and Network Technologies (pp. 505-510). IEEE.
40. Ometov, A., Shubina, V., Klus, L., Skibińska, J., Saafi, S., Pascacio, P., ... & Lohan, E. S. (2021). A survey on wearable technology: History, state-of-the-art and current challenges. Computer Networks, 193, 108074.
41. Zhou, L., Paul, S., Demirkan, H., Yuan, L., Spohrer, J., Zhou, M., & Basu, J. (2021). Intelligence augmentation: Towards building human-machine symbiotic relationship. AIS Transactions on Human-Computer Interaction, 13(2), 243-264.
42. Heena , Ranjan, S. (2023). A Systematic Study Of Various Fingertip Detection Techniques For Air Writing Using Machine Learning. arXiv preprint arXiv:2306.04332.
43. Heena , Ranjan, S. (2023). A Machine learning approach for writing in the air with fingertip using python.
44. Gurav, R. M., & Kadbe, P. K. (2015, May). Real time finger tracking and contour detection for gesture recognition using OpenCV. In 2015 International Conference on Industrial Instrumentation and Control (ICIC) (pp. 974-977). IEEE.