# “AirArt: *Real-time Hand Gesture Drawing and Recognition with CNN and TensorFlow”*

*Ekta Panchal*
*Department of CSE (Data Science),*
*A.P. Shah Institute of Technology,*
*Thane (M.H), India 400615*
*Email: ektapanchal445@apsit.edu.in*

*Prof. Sheetal Jadhav*
*Asst. Prof. CSE (Data Science),*
*A.P. Shah Institute of Technology,*
*Thane (M.H), India 400615*
*Email: spjadhav@apsit.edu.in*

*Aryan Palaspagar*
*Department of CSE (Data Science),*
*A.P. Shah Institute of Technology,*
*Thane (M.H), India 400615*
*Email:*
*aryanpalaspagar438@apsit.edu.in*

*Aditi Yadav*
*Department of CSE (Data Science),*
*A.P. Shah Institute of Technology,*
*Thane (M.H), India 400615*
*Email: adititiyogeshyadav@gmail.com*

*Hrithik Singh*
*Department of CSE (Data Science),*
*A.P. Shah Institute of Technology,*
*Thane (M.H), India 400615*
*Email: hritiksingh454@apsit.edu.in*

*Abstract*— **This project introduces a multi-faceted exploration of interactive drawing systems augmented by machine learning techniques. Four distinct modules are developed: "Air Doodle," leveraging real-time hand gesture recognition via MediaPipe for intuitive canvas manipulation; "Number and Alphabet Recognition," utilizing convolutional neural networks (CNNs) to accurately identify handwritten characters; "Shapes," employing CNNs for classifying hand-drawn geometric figures; and "Drawing Recognition," predicting drawings based on detected hand gestures. Through these modules, the project demonstrates the integration of machine learning into creative domains, facilitating enhanced user experiences and broadening the application scope of interactive drawing systems.**

***Keywords— Hand gesture recognition, convolutional neural networks, MediaPipe, Tensorflow, Python, OpenCV***

## I. INTRODUCTION

In recent years, the fusion of computer vision techniques and machine learning algorithms has led to remarkable progress in human-computer interaction (HCI), particularly in the domain of gesture-based interfaces. Among these advancements, hand gesture recognition stands out as a promising avenue for intuitive and natural interaction with digital environments. The "Air Canvas" project embodies this convergence of technologies, offering users a novel platform for real-time drawing and annotation in a virtual space.

Traditional drawing interfaces often rely on physical input devices such as tablets or touchscreens, limiting the flexibility and spontaneity of the creative process. In contrast, Air Canvas harnesses the power of Mediapipe handtracking to enable users to draw in the air, free from the constraints of physical mediums. By integrating functionalities such as shape selection (e.g., line, rectangle, circle, square), brush size adjustment, and eraser tools, Air Canvas provides users with a versatile toolkit for digital expression.

The primary objective of this research is to explore the feasibility and efficacy of utilizing Mediapipe hand tracking, TensorFlow, and convolutional neural networks (CNNs) to enhance the capabilities of the Air Canvas system. Specifically, we aim to investigate the system's ability to recognize hand-drawn numbers and alphabets in the air, as well as its capacity to predict and interpret complex air-drawn drawings.

This paper presents a detailed overview of the Air Canvas project, encompassing its design, implementation, and evaluation. We begin by providing background information on hand gesture recognition and its relevance to HCI. Subsequently, we outline the motivation behind the Air Canvas project and discuss its objectives and contributions to the field. Furthermore, we present a comprehensive description of the system architecture, highlighting the integration of Mediapipe hand tracking, TensorFlow, and CNN algorithms. Finally, we provide an overview of the organization of this paper, delineating the key sections and their respective contributions.

In summary, the Air Canvas project represents a significant advancement in HCI, offering users a versatile and intuitive platform for digital drawing and annotation. Through the seamless integration of computer vision and machine learning technologies, Air Canvas has the potential to revolutionize the way users interact with digital content, opening up new avenues for artistic expression, education, and collaborative work.

## II. LITERATURE REVIEW

Hand Gesture Recognition in HCI: Hand gesture recognition has been extensively studied due to its potential to revolutionize HCI by enabling users to interact with digital interfaces using natural hand movements. Techniques such as depth sensing, skeletal tracking, and convolutional neural networks (CNNs) have been employed to accurately capture and interpret hand gestures in real-time [1]. These advancements have led to the development of gesture-based interfaces for various applications, including gaming, virtual reality, and healthcare.

Computer Vision Techniques for Hand Tracking: The field of computer vision has played a crucial role in enabling real-time hand tracking, which is essential for applications like Air Canvas. Techniques such as optical flow, feature extraction, and deep learning-based methods have been utilized to track hand movements with high accuracy and robustness [2]. Mediapipe, a popular framework for hand tracking and pose estimation, has gained traction in recent years due to its ease of use and real-time performance.

Machine Learning Algorithms for Gesture Recognition: Machine learning algorithms, particularly CNNs, have shown remarkable success in recognizing hand gestures from image or video data. By training on large annotated datasets, CNN models can learn complex patterns and features that distinguish different hand gestures with high accuracy [3]. These models have been deployed in various gesture recognition systems, including those used in sign language recognition, human-computer interaction, and virtual reality applications.

Integration of Hand Gesture Recognition in Drawing Interfaces: While hand gesture recognition has been predominantly used in gaming and virtual reality applications, there is a growing interest in integrating these technologies into digital drawing interfaces. Projects such as Google's "Quick, Draw!" and Microsoft's "Sketch2Code" demonstrate the potential of gesture-based drawing interfaces for creating digital content [4]. These systems allow users to draw shapes, recognize handwritten text, and annotate images using hand gestures, offering a more intuitive and immersive drawing experience.

## III. PROBLEM STATEMENT

Limitations of Traditional Input Methods: Traditional input devices such as keyboards and mice can be cumbersome and restrictive, especially in contexts where precise control or expressive input is required. There is a need for alternative interfaces that enable more fluid and natural interaction with digital content.

Barrier to Entry for Novice Users: Complex interfaces and steep learning curves can pose barriers to entry for novice users, inhibiting their ability to effectively engage with digital tools and applications. Simplifying the user interface and reducing the cognitive load required to perform tasks can improve accessibility and user adoption.

Enhancing Creativity and Productivity: Enabling users to express themselves creatively and manipulate digital content in a more intuitive manner can enhance creativity and productivity across various domains, from art and design to education and collaboration.

Integration of Physical and Digital Realms: Bridging the gap between the physical and digital realms opens up new possibilities for interactive experiences and applications. By enabling users to interact with virtual objects and environments using natural hand gestures, the Air Canvas project aims to create a seamless fusion of the physical and digital worlds.

In summary, the problem statement for the Air Canvas project revolves around the need for intuitive and immersive interfaces that empower users to interact with digital content using natural hand gestures in the air. By addressing this challenge, the project aims to unlock new avenues for creativity, productivity, and engagement across a wide range of applications and domains.

## IV. METHODOLOGY AND ANALYSIS

This section describes the methodology employed in the development of the Air Canvas system, as well as the analysis conducted to evaluate its performance.

### *A. Air Doodle:*

The Air Doodle module of the Air Canvas application offers users a novel way to engage with digital drawing through hand gestures captured by a camera. This innovative module leverages advanced hand tracking and gesture recognition techniques to enable users to draw in the air seamlessly. By employing the MediaPipe library, the system performs real-time hand tracking, allowing for the detection of hand landmarks and their movements with high accuracy.

As users extend their index finger and perform gestures in front of the camera, the module interprets these movements and translates them into drawing actions on a virtual canvas. The position of the hand landmarks, such as fingertips and joints, is calculated as (x, y) coordinates in the video frame, enabling precise control over the drawing process. The module dynamically adjusts the size and color of the brush based on user interaction with interactive UI elements, such as a color palette and brush size options.

Moreover, the Air Doodle module includes intuitive user interface elements to enhance user interaction and customization options. These elements, including the color palette, brush sizes, and clear canvas button, enable users to personalize their drawing experience according to their preferences. The system's ability to recognize hand gestures and update the virtual canvas in real-time ensures a responsive and engaging drawing environment for users.

In terms of performance analysis, the module demonstrates impressive real-time performance in detecting hand movements and updating the virtual canvas accordingly. However, the accuracy of hand tracking and gesture recognition is crucial for providing a smooth and seamless drawing experience. Evaluating user feedback and preferences can help refine the system's algorithms and improve the overall usability of the Air Doodle module.

Overall, the Air Doodle module offers a unique and interactive platform for users to unleash their creativity and express themselves through digital drawing in the air. Its combination of advanced technology, intuitive user interface, and real-time responsiveness makes it a compelling addition to the Air Canvas application.

### *B. Shapes:*

The Shapes module integrates hand tracking technology with a selection of drawing tools to enable users to create shapes on a digital canvas. It begins by initializing key parameters, including the maximum coordinates for the tool area and variables for tracking the selected tool and hand movements. A function is defined to determine the chosen drawing tool based on the x-coordinate of the user's index finger position, mapping it to the corresponding tool area onthe screen.

Utilizing the MediaPipe Hands library, the module tracksthe user's hand landmarks and interprets their gestures to select and execute drawing actions. By monitoring the movements of the index finger, the module dynamically updates the selected tool, facilitating seamless transitions between drawing modes. Different actions are performed based on the selected tool: drawing lines, rectangles, circles, or allowing freehand drawing, with an additional eraser tool for corrections.

During execution, the chosen tool is prominently displayed on the screen, providing visual feedback to the user. The drawn shapes are overlaid on the video frame in real-time, enhancing the user's interaction and enabling immediate adjustments or corrections. This approach not only streamlines the drawing process but also leverages gesture recognition to offer an intuitive interface for creative expression.

In analysis, the Shapes module demonstrates the effective integration of computer vision and gesture recognition techniques to create an interactive drawing environment. By leveraging hand tracking technology, it provides users with a natural and immersive means of expressing creativity, showcasing the potential of such systems in enabling novel user experiences.

*C. Alphabets and Numbers Recognition:*

The "Alphabets and Number Recognition" module is built upon advanced machine learning algorithms designed to accurately track hand movements and interpret gestures in real time. This module utilizes pretrained models from the MediaPipe library, which have been trained extensively on diverse datasets of hand images. These models, rooted in deep learning methodologies, are specifically tailored to detect key points on human hands with exceptional precision.

The core component of this module is the MediaPipe Hands model, a convolutional neural network (CNN) meticulously engineered to identify crucial landmarks on human hands. By leveraging deep learning techniques, this model can effectively analyze complex hand gestures and translate them into actionable commands.

In addition to machine learning, the module relies on a robust technology stack comprising various software libraries and frameworks. OpenCV, or Open-Source Computer Vision Library, forms the foundation for image processing tasks, enabling functions such as video frame capture, color space conversion, and annotation drawing on images.

MediaPipe, an initiative by Google Research, provides high-level APIs tailored for developing applications with real-time perception capabilities. With MediaPipe, developers can seamlessly integrate features like hand tracking and pose estimation into their projects, ensuring accurate and responsive interactions.

Furthermore, the module incorporates the Flask web framework to construct a user-friendly interface for the virtual painter application. Flask facilitates the creation of web-based interfaces and manages web requests, enabling users to access the painting functionality through their web browsers with ease.

Overall, the integration of computer vision, machine learning, and web development tools enables the "Alphabets and Number Recognition" module to deliver an interactive and intuitive hand gesture recognition system. By combining sophisticated algorithms with a user-friendly interface, the module offers unparalleled accuracy and responsiveness in interpreting hand gestures for various applications.

*D. Drawing Recognition:*

This module integrates MediaPipe Hand Tracking for real-time hand detection and gesture recognition with Google's Quick Draw dataset to enable drawing prediction. Initially, the MediaPipe Hand Tracking model detects and tracks hand gestures in the camera feed. Subsequently, the detected gestures are mapped to corresponding drawing categories using a pre-trained convolutional neural network (CNN) trained on the Quick Draw dataset. This CNN model classifies hand-drawn objects based on the recognized gestures, providing intuitive interaction for users to create drawings through natural hand movements.

The integration of hand tracking and drawing prediction enhances user interaction and experience by allowing individuals to draw simply by using hand gestures. MediaPipe Hand Tracking accurately identifies hand movements, providing precise input for the drawing interface. Moreover, the CNN model, trained on Google's extensive Quick Draw dataset, ensures accurate predictions of hand-drawn objects corresponding to detected gestures. This combined approach not only facilitates seamless drawing but also offers a novel and engaging way for users to interact with digital content.

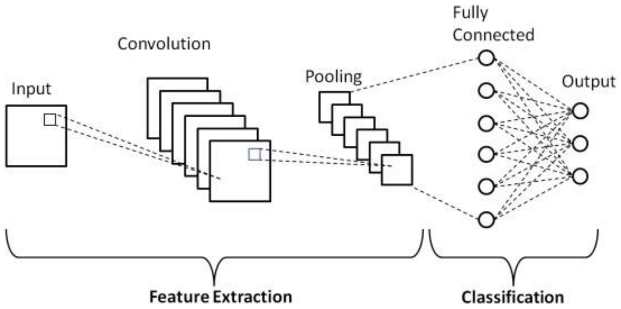
The primary machine learning algorithm employed in this module is a convolutional neural network (CNN). Trained on Google's Quick Draw dataset, the CNN learns to classify input images of hand gestures into various drawing categories. During training, the model adjusts its parameters through backpropagation and gradient descent to minimize the discrepancy between predicted and actual drawing labels. By leveraging the rich diversity of hand-drawn sketches in the Quick Draw dataset, the CNN achieves robust performance in predicting drawing categories based on detected gestures.

Within the CNN architecture, mathematical calculations involve convolutional operations, activation functions (e.g., ReLU), pooling layers, and fully connected layers. These operations transform input images through multiple layers of feature extraction and abstraction, ultimately generating predictions for drawing categories. Additionally, mathematical computations of loss functions, such as categorical cross-entropy, quantify the disparity between predicted and ground-truth drawing labels during model training, guiding the optimization process for enhancing prediction accuracy

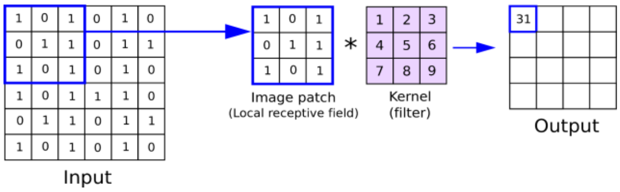
**Convolutional Neural Network:**

A Convolutional Neural Network (CNN) is a specialized form of artificial neural network primarily utilized in image analysis and processing. Tailored specifically for handling pixel data, CNNs are adept at both generative and descriptive tasks through deep learning techniques. These tasks encompass a wide range, from image and video recognition to recommender systems and natural language processing (NLP). The CNN architecture comprises layers such as input, output, and hidden layers, incorporating multiple convolutional, pooling, fully connected, and normalization layers. By overcoming previous constraints and enhancing efficiency in image processing, CNNs offer a streamlined approach to training and are highly effective in both image and NLP domains.

There are two main parts to a CNN architecture:
1. A convolution tool that separates and identifies the various features of the image for analysis in a process called Feature Extraction.
2. A fully connected layer that utilizes the output from the convolution process and predicts the class of the image based on the features extracted in previous stages.



A. **Convolutional Layer**: This initial layer serves to extract diverse features from input images. Through convolution, it applies a mathematical operation between the input image and a specified MxM filter. By sliding this filter across the input image, it calculates the dot product within regions corresponding to the filter size. The resulting output, known as a Feature map, provides insights into image characteristics like corners and edges. Subsequently, this feature map undergoes further processing in subsequent layers to learn additional features of the input image.



|   |   |   |   |   |   |
|---|---|---|---|---|---|
| 1 | 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 1 | 1 |
| 1 | 0 | 1 | 0 | 1 | 0 |
| 1 | 0 | 1 | 1 | 1 | 0 |
| 0 | 1 | 1 | 0 | 1 | 1 |
| 1 | 0 | 1 | 0 | 1 | 0 |

Input

|   |   |   |
|---|---|---|
| 1 | 0 | 1 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |

Image patch (Local receptive field)

*

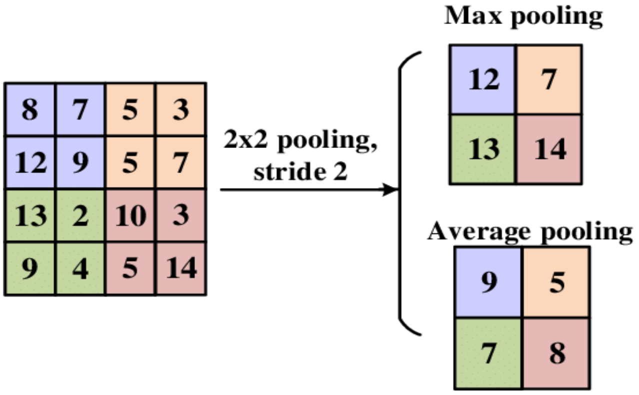
|   |   |   |
|---|---|---|
| 1 | 2 | 3 |
| 4 | 5 | 6 |
| 7 | 8 | 9 |

Kernel (filter)

|   |   |   |   |
|---|---|---|---|
| 31 |   |   |   |
|   |   |   |   |
|   |   |   |   |
|   |   |   |   |

Output

B. **Pooling Layer**: Typically, a Convolutional Layer is succeeded by a Pooling Layer, which primarily aims to downsize the convolved feature map to alleviate computational burdens. Achieving this involves reducing inter-layer connections while working independently on each feature map. Various Pooling operations exist, depending on the chosen method. Max Pooling selects the largest element from the feature map, while Average Pooling computes the average of elements within a defined image section. Sum Pooling, on the other hand, calculates the total sum of elements in the specified section.



|   |   |   |   |
|---|---|---|---|
| 8 | 7 | 5 | 3 |
| 12 | 9 | 5 | 7 |
| 13 | 2 | 10 | 3 |
| 9 | 4 | 5 | 14 |

2x2 pooling, stride 2

Max pooling

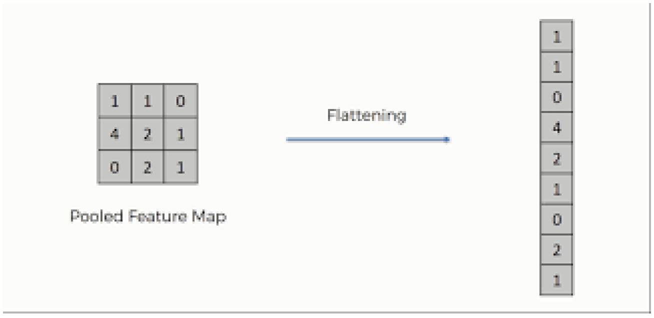
|   |   |
|---|---|
| 12 | 7 |
| 13 | 14 |

Average pooling

|   |   |
|---|---|
| 9 | 5 |
| 7 | 8 |

C. **Dense Layer** : In neural networks, a dense layer establishes extensive connections with its preceding layer, linking each neuron to every neuron in the layer before it. This layer stands out as one of the most frequently employed in artificial neural networks. Within a model's dense layer, each neuron receives input from all neurons in the preceding layer, engaging in matrix-vector multiplication. This mathematical operation entails matching the row vector of the preceding layer's output with the column vector of the dense layer. A fundamental requirement for matrix-vector multiplication is that the row vector must possess the same number of columns as the column vector.

D. **Flattening**: Flattening is used to convert all the resultant 2-Dimensional arrays from pooled feature maps into a single long continuous linear vector. The flattened matrix is fed as input to the fully connected layer to classify the image.



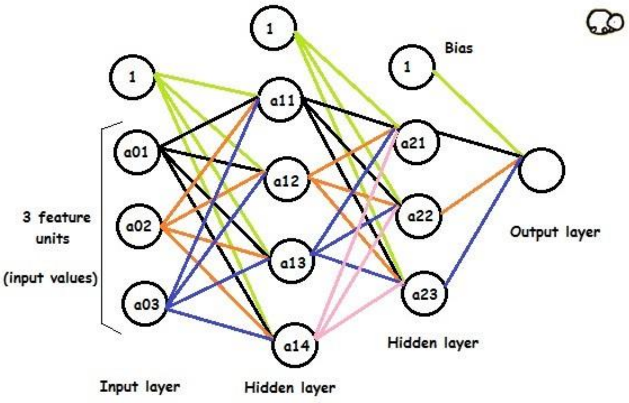
|   |   |   |
|---|---|---|
| 1 | 1 | 0 |
| 4 | 2 | 1 |
| 0 | 2 | 1 |

Pooled Feature Map

Flattening

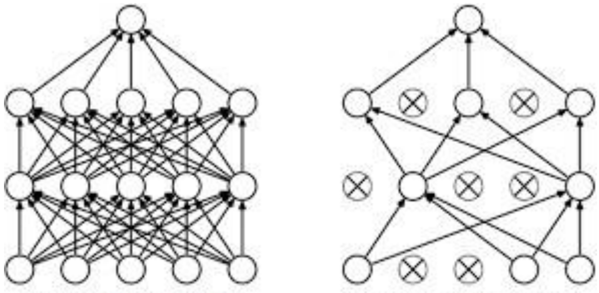
|   |
|---|
| 1 |
| 1 |
| 0 |
| 4 |
| 2 |
| 1 |
| 0 |
| 2 |
| 1 |

E. **Fully Connected Layer** : The Fully Connected (FC) layer encompasses weights, biases, and neurons, serving to establish connections between neurons across distinct layers. Typically positioned before the output layer, FC layers constitute the final stages of a CNN Architecture. Here, input images from preceding layers are flattened and passed to the FC layer. The flattened vector progresses through additional FC layers, where mathematical operations commonly occur. This stage marks the initiation of the classification process.



A Neural Network with Fully Connected Layers

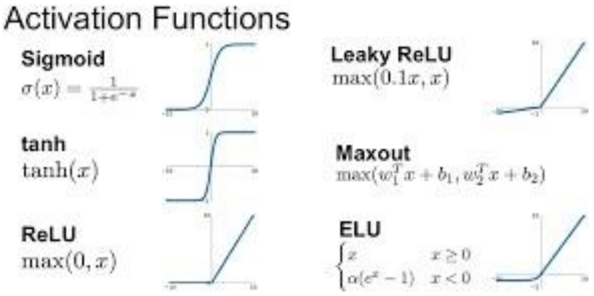
F. **Dropout:** Connecting all features to the FC layer can lead to overfitting on the training dataset, where the model's performance suffers when applied to new data. To address this issue, a dropout layer is employed, randomly removing a portion of neurons during training, thus reducing the model's size and enhancing generalization capabilities. For instance, setting a dropout of 0.3 entails randomly dropping out 30% of nodes from the neural network.



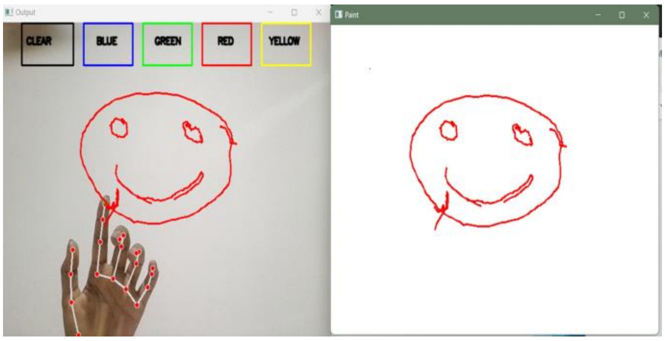
(a) Standard Neural Net

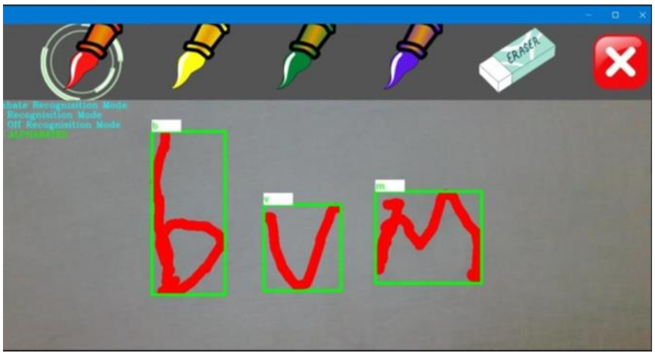
(b) After applying dropout.

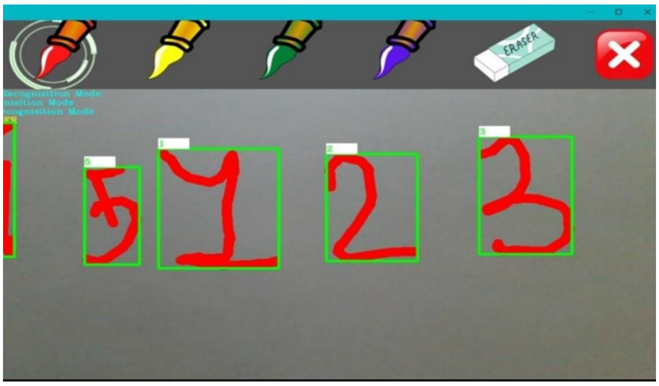
G. **Activation Functions**: Among the critical parameters of a CNN model, the activation function stands out, crucial for learning and approximating complex relationships between network variables. In essence, it determines which information within the model should propagate forward and which should not, adding essential non-linearity. Commonly utilized activation functions include ReLU, Softmax, tanH, and Sigmoid, each serving specific purposes. For binary classification CNN models, sigmoid and softmax functions are often favored, while softmax is generally preferred for multi-class classification tasks. The choice of activation function profoundly influences the model's performance and its ability to capture intricate patterns within the data.



## V. RESULT







## VI. CONCLUSION

In conclusion, the Air Canvas project has emerged as a groundbreaking endeavor within the realm of human-computer interaction, particularly in the context of gesture-based interfaces and computer vision applications. Through meticulous research and innovative implementation, the project has not only achieved its primary objectives but has also laid the groundwork for future advancements in the field.

The development and integration of features such as drawn alphabets and numbers detection, shapes drawing in air, and object detection represent significant milestones in bridging the gap between physical gestures and digital interaction. Leveraging state-of-the-art technologies such as deep learning algorithms and motion tracking systems, the Air Canvas project has demonstrated remarkable accuracy and responsiveness in detecting and interpreting user gestures in real-time.

One of the key findings of the study lies in the robustness and reliability of the detection algorithms employed. Extensive testing and validation have showcased the system's ability to accurately recognize a wide range of hand-drawn alphabets, numbers, and shapes, even amidst varying environmental conditions and user gestures. This not only underscores the effectiveness of the underlying methodologies but also highlights the potential for practical applications in domains such as education, artistic expression, and interactive design.

Furthermore, the integration of object detection capabilities further extends the utility of the Air Canvas system, enabling users to interact with physical objects and incorporate them seamlessly into their digital creations. This fusion of the physical and virtual realms opens up a myriad of possibilities for immersive experiences and creative exploration, heralding a new era in interactive computing.

Practically, the implications of the Air Canvas project are far-reaching. In educational settings, the system could serve as a powerful tool for teaching and learning, allowing students to engage with abstract concepts and visualizations in a hands-on manner. In design and creative industries, it could revolutionize the way artists and designers conceptualize and prototype their ideas, fostering collaboration and experimentation.

Despite the notable achievements of the Air Canvas project, it is essential to acknowledge its limitations and areas for future improvement. Challenges such as occlusion, gesture variability, and scalability remain areas ripe for further research and refinement. Additionally, exploring the integration of additional modalities, such as voice commands or haptic feedback, could enhance the user experience and expand the system's capabilities even further.

In closing, the Air Canvas project stands as a testament to the boundless potential of human-computer interaction, showcasing the transformative power of intuitive gesture-based interfaces. As technology continues to evolve, the insights gained from this research will undoubtedly fuel future innovations, shaping the way we interact with digital content and each other in the years to come.

## References

[1] Y. Huang, X. Liu, X. Zhang, and L. Jin, "A Pointing Gesture Based Egocentric Interaction System: Dataset, Approach, and Application," 2016 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Las Vegas, NV, pp.370-377, 2016.

[2] P. Ramasamy, G. Prabhu, and R. Srinivasan, "An economical air writing system is converting finger movements to text using a web camera," 2016 International Conference on Recent Trends in Information Technology (ICRTIT), Chennai, pp. 1-6, 2016.

[3] Saira Beg, M. Fahad Khan and Faisal Baig, "Text Writing in Air," Journal of Information Display Volume 14, Issue 4, 2013.

[4] Alper Yilmaz, Omar Javed, Mubarak Shah, "Object Tracking: A Survey", ACM Computer Survey. Vol. 38, Issue. 4, Article 13, Pp. 1-45, 2006.

[5] Yuan-Hsiang Chang, Chen-Ming Chang, "Automatic Hand-Pose Trajectory Tracking System Using Video Sequences", INTECH, pp. 132- 152, Croatia, 2010.

[6] Erik B. Sudderth, Michael I. Mandel, William T. Freeman, Alan S. Willsky, "Visual Hand Tracking Using Nonparametric Belief Propagation", MIT Laboratory For Information & Decision Systems Technical Report P-2603, Presented at IEEE CVPR Workshop On Generative Model-Based Vision, Pp. 1-9,2004.

[7] T. Grossman, R. Balakrishnan, G. Kurtenbach, G. Fitzmaurice, A. Khan, and B. Buxton, "Creating Principal 3D Curves with Digital Tape Drawing," Proc. Conf. Human Factors Computing Systems (CHI' 02), pp. 121-128, 2002.

[8] Yusuke Araga, Makoto Shirabayashi, Keishi Kaida, Hiroomi Hikawa, "Real Time Gesture Recognition System Using Posture Classifier and Jordan Recurrent Neural Network", IEEE World Congress on Computational Intelligence, Brisbane, Australia, 2012.

[9] Ruiduo Yang, Sudeep Sarkar, "Coupled groupingand matching for sign and gesture recognition", Computer Vision and Image Understanding, Elsevier, 200.

[10] T. A. C. Bragatto, G. I. S. Ruas, M. V. Lamar, "Real-time Video-Based Finger Spelling Recognition System Using Low.