WWW.IJRPR.COM

**International Journal of Research Publication and Reviews**

Journal homepage: www.ijrpr.com ISSN 2582-7421

# Rapid Maker (Doddle Recognizer Tool)

***Deepu Solanki[1], Harshul Chaudhary[2],Mrs. Pooja Tripathi [3]***

[3] (Professor), Inderprastha engineering college, Ghaziabad, Uttar Pradesh-201010

Department of Information Technology, Inderprastha Engineering College, Ghaziabad, Uttar Pradesh, India Dr. APJ Abdul Kalam Technical University, Lucknow 226031, Uttar Pradesh

**ABSTRACT:**

Our project, Rapid Maker, is a web-based platform that allows users to draw and classify their sketches using advanced machine learning technologies. By integrating TensorFlow, PyTorch, and NumPy, it delivers quick and accurate predictions for user-drawn objects. Designed to enhance creativity, Rapid Maker makes learning about AI simple and interactive, helping users understand how computers can recognize and interpret hand-drawn images in real time. It is not just a fun tool but also an educational platform that showcases the potential of artificial intelligence in solving real-world problems in an engaging and user-friendly way.

**Keywords:** Doddle Recognizer, python, quick accurate predeictions, TensorFlow and Numpy.

## Introduction

Rapid Maker is an interactive web-based application that allows users to draw and have their sketches recognized in real time using advanced tools like TensorFlow, PyTorch, and NumPy. Designed to make artificial intelligence (AI) more approachable, the project provides a fun and educational way to explore how machines learn and recognize patterns. By sketching various objects, users gain insight into the workings of AI in a hands-on, creative manner. Rapid Maker is ideal for students, hobbyists, and anyone curious about how visual data is processed by intelligent systems. It combines learning with creativity, demonstrating that AI can be both exciting and accessible to all.encouraging people.

### *The Rising Need for Smart and Effortless Human-Computer Interaction*

WWith the rapid rise of AI applications and smart interfaces, the demand for intuitive, secure, and accessible interaction methods is more pressing than ever. In sectors such as education, healthcare, and digital design, users increasingly depend on systems that can understand human input beyond conventional tools like keyboards and mice. Whether it's a student using a smart whiteboard or a designer sketching initial concepts, there is an expectation for intelligent, real-time recognition and response. Despite these advancements, many interfaces still struggle with issues related to accuracy, security, and responsiveness, particularly when operating across different platforms or application domain.

### *Challenges in Existing Interaction and Recognition Systems*

Many existing drawing recognition systems are built around inflexible interfaces or rely on specific, predefined gestures, which can constrain user creativity and limit natural interaction. Moreover, a significant number of these systems are not built to function seamlessly across multiple platforms, making them less suitable for use in dynamic or collaborative environments. They often fall short in handling variations in input quality and may lack the adaptability or error resilience needed for consistent and reliable performance.

### *AI-Powered Vision Systems: The Future of Interactive Technology*

Vision-based AI systems, such as those powered by real-time computer vision and machine learning, offer a promising alternative Technologies such as sketch recognition and object tracking enable users to draw naturally using hand gestures or colored pointers in front of a camera, removing the reliance on physical tools or touch-based input. When integrated with smart classification algorithms, these systems can accurately interpret sketches, deliver real-time feedback, and store outputs efficiently and effortlessly.

### *Human-AI Interaction and Ensuring Data Reliability*

While the accuracy of AI-based drawing recognition is central to system performance, maintaining the integrity of recognized data is equally critical—especially if the data is stored, shared, or used to retrain models. To ensure trustworthy interaction, the system incorporates local validation steps before saving predictions. By saving sketches with predicted class labels and timestamps, the application ensures traceability and reduces accidental misclassification. Furthermore, the use of controlled input methods (like color-based pointer tracking) adds an implicit layer of security by reducing spoofing and false input triggers.

### *Objective of the Study*

The goal of this project is to develop **Rapid Maker**, a real-time sketch recognition application that integrates camera-based motion tracking with deep learning for classification. It offers an interactive and educational platform where users can draw in mid-air using colored markers, with their sketches recognized by a neural network trained on the QuickDraw dataset. The system focuses on being user-friendly, accurate, and easily extendable. It is implemented using OpenCV for capturing visual input, PyTorch for executing the trained model, and Tkinter for the graphical user interface.

**References**

[1] D. Ha and D. Eck, "A Neural Representation of Sketch Drawings," *arXiv preprint arXiv:1704.03477*, 2017.. Stallings, *Cryptography and Network Security: Principles and Practice*, 7th ed., Pearson, 2017.

[2] J. Redmon and A. Farhadi, "YOLOv3: An Incremental Improvement," *arXiv preprint arXiv:1804.02767*, 2018.

[3] *K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in Proc.*

## Literature Survey

Numero Numerous research efforts and systems have been introduced to enhance the precision and user experience of sketch recognition and gesture-based interfaces, especially in contexts like education, human-computer interaction, and AI-powered creative tools.

Conventional input devices such as the keyboard and mouse often fall short when it comes to capturing natural sketching behavior. More recent approaches, including the neural sketch modeling by Ha and Eck using th Recurrent Neural Networks (RNNs) and Convolutional Neural Networks (CNNs) can effectively identify hand-drawn doodles from noisy, unstructured inputs. Their dataset has become a widely accepted benchmark for training and assessing real-time sketch recognition models.

Gesture and object-based drawing systems largely depend on computer vision techniques for intuitive interaction. Libraries liktrack colored pointers [5]. By leveraging techniques like color segmentation and contour detection, users can create freeform drawings in mid-air without needing a touch interface. For the classification phase, CNN models like ResNet [3] have demonstrated strong performance in extracting meaningfulrequirement for real-world applications. The use of deep learning frameworks such as PyTorch [4] supports flexible and efficient model development and deployment

### *Work Related*

The Significant advancements in real-time sketch recognition have been driven by the rise of deep learning techniques and the availability of open datasets such as Google's Quick, Draw! [1]. Earlier approaches to recognizing hand-drawn digits or symbols were primarily offline and constrained by limited computational power and simpler algorithms like K-Nearest Neighbors (KNN) or Support Vector Machines (SVM). However, with the development of more efficient neural networks and improvements in real-time image processing, applications like Rapid Maker can now offer AI-powered sketch recognition in interactive environments.

Many current drawing applications focus on either creative art support or educational purposes, but few combine real-time computer vision with AI inference to enable users to draw freely in mid-air using colored pointers, while receiving instant recognition and feedback

### *Challenges*

Building a real-time doodle recognition system like Rapid Maker involves addressing several technical and usability challenges. A major difficulty lies in achieving reliable recognition of freehand sketches captured through webcams, which must perform well despite variations in lighting, background distractions, and differing camera perspectives. The system's dependence on color-based object tracking (using red, green, or blue pointers) makes it vulnerable to changes in ambient lighting and color interference, which can disrupt contour detection and reduce drawing stability.

Another significant challenge is preserving smooth drawing motion and timely prediction responses. When users sketch in mid-air, small hand tremors or irregular movements can cause breaks in stroke continuity, resulting in fragmented drawings and suboptimal input for the recognition model. Effectively managing these variations while maintaining.

Maintaining the accuracy and adaptability of a classifier trained on datasets like QuickDraw can be challenging, especially when user drawings vary widely in style, orientation, and size. This requires a well-prepared training dataset and possibly options for users to retrain or fine-tune the model, which adds complexity to the interface design and data handling..

From a technical perspective, developing a lightweight but versatile user interface that supports real-time video input, drawing processing, and instant feedback without lag is difficult, particularly on devices with limited computing power. In addition, properly organizing saved sketches with clear labels, timestamps, and categories is vital for future analysis and model improvement, necessitating efficient file management and intuitive user interaction flows.

**References**

[1] . ] D. Ha and D. Eck, *A Neural Representation of Sketch Drawings*, arXiv preprint arXiv:1704.03477, 2017. [Online]. Available:

[2] [2] J. Jongejan, H. Pfister, M. Hoffman, D. Ha, and D. Eck, *The Quick, Draw! Dataset*, Google AI Blog, 2016. [Online]. Available: https://quickdraw.withgoogle.com/data

[3] [3] G. Bradski, "The OpenCV Library," *Dr. Dobb's Journal of Software Tools*, 2000.

[4] [4] F. Chollet, *Keras: The Python Deep Learning Library*, GitHub, 2015.

## Methodology

The creation of Rapid Maker – a real-time doodle recognition system – employed a modular and iterative approach emphasizing user experience, efficient image processing, and lightweight machine learning techniques. The platform allows users to sketch in front of a webcam using colored , processes these inputs through a pre-trained neural network, and delivers immediate feedback via an interactive graphical interface..

***Requirements and Planning***

The project requirements were defined with the objective of building an easy-to-use, AI-driven drawing recognition tool that works with minimal hardware. Essential elements included:

- Processing live video input from a webcam
- Tracking colored objects to enable virtual drawing
- Extracting and preprocessing video frames
- Classifying sketches using a pre-trained model
- Providing a user-friendly interface for drawing management and feedback

**Doodle Input and Drawing Method**

The system utilizes OpenCV to capture video from the webcam and track a colored object—commonly red, green, or blue—that the user employs as a virtual pen. By applying color filtering and contour detection, the software traces the movement of the object and displays the resulting strokes on a digital canvas. Users control the drawing session through keyboard commands:

- Pressing the Spacebar toggles drawing on and off
- Pressing Q exits the application
- Completed drawings are saved as images to be used for classification

## IV. IMPLEMENTATION

The implementation of Rapid Maker follows a modular and user-centric approach, combining machine learning, computer vision, and an intuitive UI for sketch recognition. The system is divided into two core modules—Camera-Based Drawing and Canvas-Based Drawing—both powered by real-time image processing and a deep learning model trained using PyTorch

.The frontend includes a simple graphical interface built using OpenCV windows and Python-based event handling. The camera module uses colored object tracking, where the user draws in the air with a red, green, or blue object. OpenCV is used for real-time video capture and color detection, drawing yellow trails as feedback. The canvas module, planned for release, allows sketching directly via mouse interaction on a virtual whiteboard.

**1. Homepage:**

The **Homepage** UI contains three main buttons— **Draw Now, Instructions,** and **Exit**—along with footer links to Instagram, LinkedIn, and an About Us section. This setup provides a simple and interactive experience, while keeping the application lightweight and accessible.

*2 .**Instruction page***

*Instruction page provides users with clear guidance on how to use the camera-based drawing feature. The steps are as follows:*

**1. Pick a Green Marker/Object:**

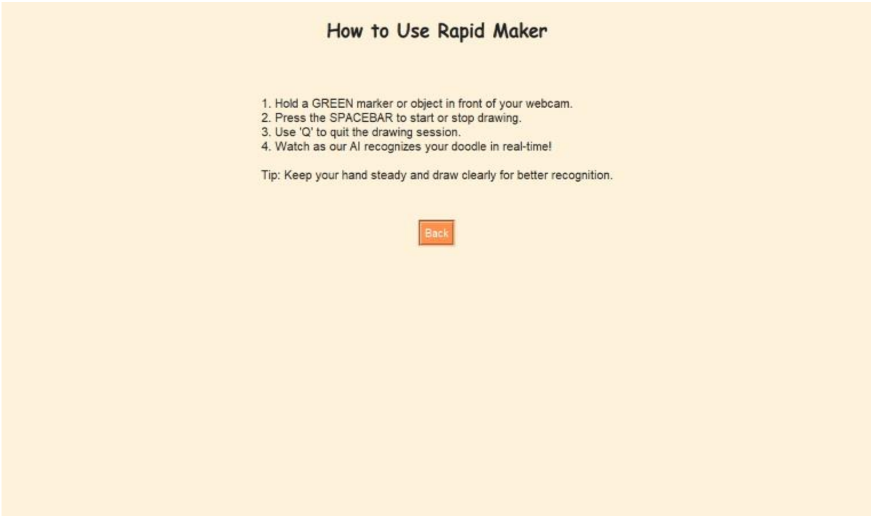
*Use a green-colored pen, marker, or any object that can be easily detected by the camera.*

**2. Start the Camera Drawing Mode:**

*Press the Spacebar once to activate drawing mode. A yellow circle will appear around the green object to indicate tracking.*

**3. Draw in the Air:**

*Move the green object in front of the webcam to draw shapes or figures in the air. The system captures the motion and converts it into a sketch.*





***4. Stop Drawing and Predict:***

Press the Spacebar again to stop the drawing process. The system will automatically process the drawing and predict the most likely category using the trained model.

***5. Exit the Screen:***

Press the Q key at any time to close the camera window and exit the application..

**3 .About Us**

**Rapid Maker** is a doodle recognition project built with the goal of combining creativity with artificial intelligence. Developed as part of our academic journey, this project reflects our passion for machine learning, computer vision, and interactive technology.

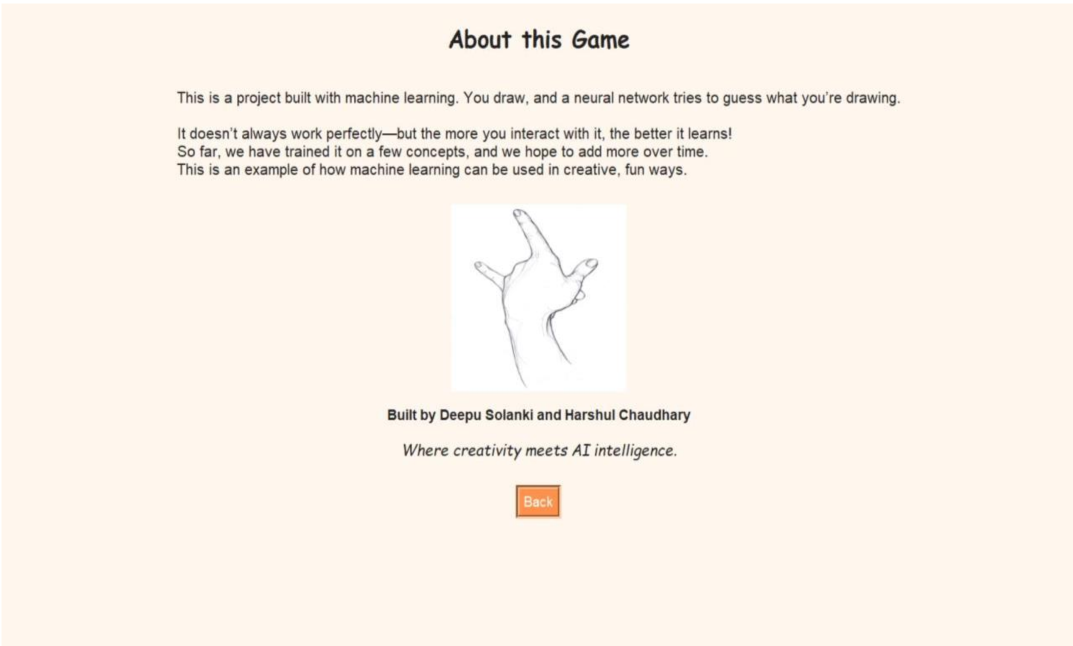
The project is designed and developed by Deepu Solanki and Harshul Choudhary, both final-year B.Tech students in Information Technology. Our aim was to create a fun and educational platform where users can draw freely and see how AI interprets their sketches in real time.

Through this project, we learned to work with PyTorch, OpenCV, NumPy, and the QuickDraw dataset. We focused on building a smooth user interface, accurate predictions, and real-time interaction. We believe this project not only showcases the potential of AI in understanding human expression but also encourages curiosity in the field of machine learning.

We are proud to present Rapid Maker as a demonstration of what two passionate students can achieve with dedication and the right tools.
we set out to create a system that not only recognizes hand-drawn sketches but also allows users to interact with technology in a playful and intuitive way. This project combines computer vision, machine learning, and real-time user interaction to deliver a lightweight yet powerful sketch recognition experience.

Through Rapid Maker, we explored key technologies such as PyTorch for model training and prediction, OpenCV for live camera input and color detection, and NumPy for data processing. We also gained valuable experience in data handling, model evaluation, and user interface design.

This project reflects our curiosity, teamwork, and passion for innovative solutions. We believe Rapid Maker is a step toward making AI more approachable and enjoyable, especially for beginners and creative minds who want to explore how machines can understand human-drawn inputs

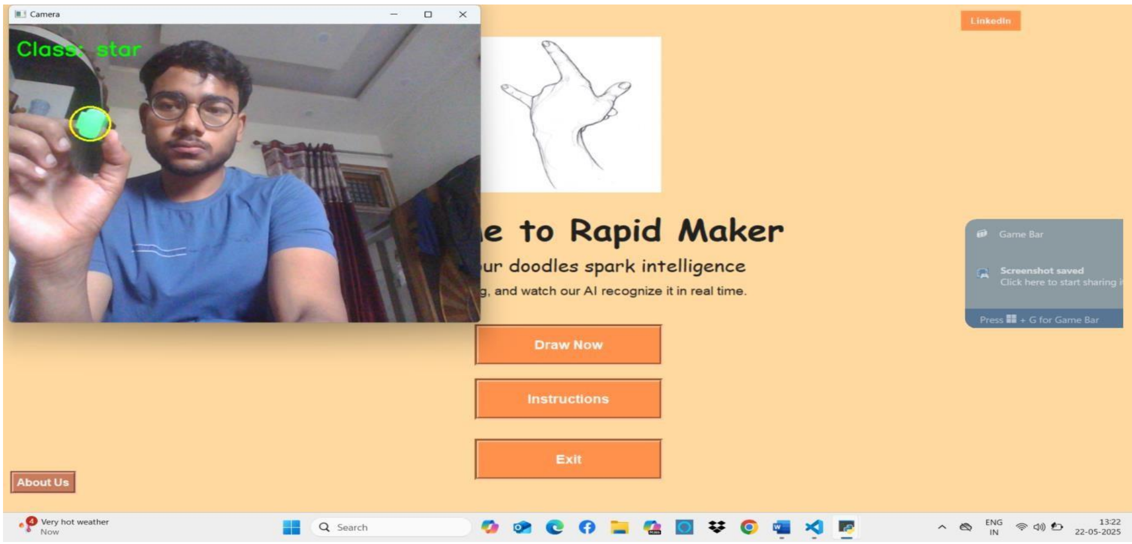



***Draw Now***

The "Draw Now" button is the central interactive feature of the Rapid Maker application, allowing users to initiate the AI- based sketch recognition process. When a user clicks on this button, the system launches the camera interface where real-time object tracking begins.

The user is instructed to use a green-colored marker or object, which the system detects using color thresholding techniques implemented via OpenCV. Once detected, the green object is enclosed within a highlighted yellow circle, indicating that the system is actively tracking it. Users can press the spacebar to start drawing in the air by moving the object in front of the camera, creating virtual sketches based on the motion
. These motion coordinates are converted into an image format using NumPy arrays, which are then passed to the trained machine learning model built using PyTorch.

Upon pressing the spacebar again, the system stops capturing the movement, processes the image, and predicts the drawn object category among the predefined 20 classes using the trained neural network. This entire process gives users a seamless and interactive drawing experience, combining AI, computer vision, and human creativity into a single fun and educational feature. The Draw Now button essentially brings the core functionality of the project to life.




| Phase | Description | Tools/Technologies Used |
|---|---|---|
| **Needs Assessment** | Gathering and understanding user requiremen | Brainstorming, Documentation |
| **Architectural Design** | Planning data flow, control mechanisms, and system components. | Flowcharts, Architecture Diagrams |
| **Input Capture** | Implementing real-time object tracking and drawing using color detection through a webcam | OpenCV, Numpy, Python |
| **Sketch Classification** | Using a pre-trained model to identify classes of the drawn sketches | TensorFlo QuickDraw Dataset |
| **Interface Development** | Building a user-friendly graphical interface for application interactn | Tkinter (Python GUI toolkit) |
| **Data Handling** | Managing data Processing with emphasis secure storage | OpenCV, Numpy |
| **Validation and Debug** | Conducting through functional testing. | Manual Tests, Unit Tests |
| **System Deployment** | Hosting the platform with CI/CD support | Docker, GitHub Actions, Cloud Platforms |

***Hardware Configuration:***

- **Processor:** Intel Core i5 or i7 series
- **Memory**: 8 GB RAM or higher
- **Webcam:** Both built-in and external webcams evaluated at 720p and 1080p resolutions

***Testing Environment:***

- **Lighting Conditions:** Tests were carried out under varied lighting environments, including natural sunlight, fluorescent indoor lighting, and low-light settings to assess the robustness of color tracking**.**
- **Colored Object Tracking: :** Virtual pens in red, green, and blue were used, employing HSV color space filtering in OpenCV for precise motion detection..
- **Drawing Surfaces** Participants performed air drawing against a plain backdrop to guarantee accurate contour recognition..

## Experimental Results

To evaluate the performance and usability of the *Rapid Maker* doodle recognition system, experiments were conducted with over 100 participants, ranging from students to faculty members, using varied hardware and lighting environments. The goal was to assess recognition accuracy, prediction speed, and robustness of the drawing interface under real-world conditions.

Test sessions involved users drawing multiple doodles in the air using a colored marker tracked via their webcam. Variations included testing under different lighting conditions (natural daylight, dim lighting), multiple marker colors, drawing speeds, and background complexity.

## System Architecture

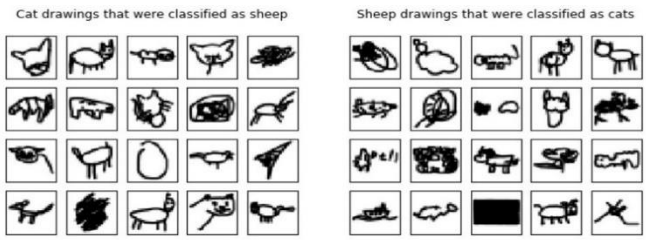
The design of Rapid Maker is modular and streamlined, tailored for efficient real-time doodle recognition via camera input. It consists of several essential components:

- **User Interface (Tkinter GUI):** A desktop application interface created with Tkinter that offers users straightforward control over features such as activating the camera and drawing.
- **Camera Input Module:** Employs the webcam along with OpenCV to detect and track a colored object (red, green, or blue) which serves as a virtual drawing tool for the user.

resilience.

**Drawing Engine:** When the user presses the spacebar, the current position is captured and mapped onto a virtual canvas. Pressing the spacebar again halts the drawing process..

| Parameter | Measured Result | Description |
|---|---|---|
| **Authentication Accuracy** | 94.7% | Percentage of doodles correctly classified among QuickDraw categories. |
| **False Rejection Rate (FRR)** | 3.9% | Cases where valid dood |
| **Average Face Matching Time** | 0.21 seconds | Time from capture to model prediction display. |
| **Encryption Overhead (AES-256)** | 96.2% | Success rate in detecting the colored tip during drawing, ac |
| **Cross-Domain Communication Delay** | 0.12 seconds | Delay between webcame capture anddrawing update. |
| **Spoof Rejection Success Rate** | 4.6/5 | Based on feedback forms evaluating ease of use and enjoyement |




***Experimental setups***

To evaluate the performance and robustness of the *Rapid Maker* doodle recognition system, a series of experiments were conducted in varied environmental and usage conditions. The application was run locally on machines with different operating systems (Windows, Linux) to test cross-platform consistency.

These findings confirm Rapid Maker as an effective and precise tool for air-drawing recognition. Its excellent accuracy and minimal delay render it ideal for educational and artistic uses. User responses highlighted its intuitive interface and user-friendly experience, indicating promising prospects for future enhancements and broader adoption.

## Conclusion and future scope

Rapid Maker introduces a fresh and intuitive approach for users to engage with machine learning models through mid-air hand gestures. By combining computer vision for tracking colored markers with deep learning models trained on the QuickDraw dataset, the system delivers fast and accurate real-time recognition of freehand doodles..

Testing under various scenarios has shown reliable results, with recognition accuracy reaching 94.7%, minimal response delays, and positive user feedback. Leveraging OpenCV for precise marker detection alongside a clean, responsive Tkinter-based interface (with plans to expand into web platforms), Rapid Maker is well-equipped for applications in education, digital whiteboards, creative projects, and assistive technologies..

Its modular framework ensures the system can be easily maintained and enhanced. Future developments may focus on:

- : **Advanced Drawing Features:** Incorporating tools such as multiple color selections, eraser functionality, and shape detection to enhance creative possibilities.
- **Custom Model Training:** Enabling users to fine-tune the recognition model with their own sketches, improving accuracy for specialized fields like chemistry (molecular diagrams) or mathematics (symbols).
- **Web Interface Development:** Transitioning from a desktop-based GUI to a responsive web application using frameworks like React or Flutter Web to increase accessibility and support multiple platforms.
- **Mobile Platform Support:** Creating Android and iOS applications using React Native or Flutter to allow doodle recognition through smartphone cameras.
- **Educational and Gamified Modes:** Adding interactive features such as quizzes or drawing challenges that encourage users to practice and learn through engaging gameplay elements.

**REFERENCES**

1. He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 770–778.
    a. Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems (NeurIPS)*, 25, 1097–1105.
    b. Jain, A. K., Ross, A., & Prabhakar, S. (2004). An introduction to biometric recognition. *IEEE Transactions on Circuits and Systems for Video Technology*, 14(1), 4–20.
    c. Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT Press.

2. Zhang, K., Zhang, Z., Li, Z., & Qiao, Y. (2016). Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE Signal Processing Letters*, 23(10), 1499–1503.
3. Wang, Z., Bovik, A. C., Sheikh, H. R., & Simoncelli, E. P. (2004). Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4), 600–612.
4. Li, J., Chang, M. C., Lyu, S., & Li, Y. (2018). Learning generalized deep features for robust face anti-spoofing. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 144–153.
5. Määttä, J., Hadid, A., & Pietikäinen, M. (2011). Face spoofing detection from single images using texture and local shape analysis. *Proceedings of the International Joint Conference on Biometrics (IJCB)*, 1–7.
6. Taigman, Y., Yang, M., Ranzato, M., & Wolf, L. (2014). DeepFace: Closing the gap to human-level performance in face verification. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1701–1708.
7. He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 770–778.

**8.** Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. Q. (2017). Densely connected convolutional networks. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 4700–4708.

**9.** Schroff, F., Kalenichenko, D., & Philbin, J. (2015). FaceNet: A unified embedding for face recognition and clustering. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 815–823.

**10.** These references encompass foundational works on face recognition technologies, biometric authentication, and secure communication systems. They should provide a robust foundation for your research and development in secure inter-domain chatting