Methodology for “Air Canvas”

Project by – Darshita Singh (229310009)

Chaitanya Sekra (229310016)

1. Requirement Analysis and Planning:
   1. Define the detailed requirements for the Air Canvas project, including user stories and use cases for each feature:
      1. Color change: Users should be able to select from a variety of colors for drawing.
      2. Clear screen: User should have the ability to clear the canvas to start fresh.
      3. Air drawing: The application should recognize hand gestures and movements to draw on the canvas.
      4. Saving the canvas: User should be able to save the drawing for future reference or sharing.
   2. Plan the development roadmap, including milestones, timelines, and resource allocation.
2. Environment Setup:
   1. Set up the development environment with Python installed along with the required libraries: OpenCV, scikit-learn (for machine learning), GUI framework (e.g., Tkinter, PyQt), networking libraries (e.g., sockets, requests), PIL (Python Imaging Library), and version control system (e.g., Git).
3. User Interface Design:
   1. Design the user interface for the Air Canvas application using the chosen GUI framework.
   2. Implement UI elements for color change, clear screen, and saving functionality.
4. Air Drawing Implementation:
   1. Utilize OpenCV for capturing video frames from the camera.
   2. Implement hand gesture detection and tracking algorithms for air drawing recognition.
   3. Integrate machine learning techniques (e.g., SVM, CNN) for gesture classification and mapping to specific drawing actions.
5. Color Change Feature:
   1. Implement color selection functionality allowing users to choose different drawing colors.
   2. Integrate the selected color with the drawing tool.
6. Clear Screen Functionality:
   1. Develop functionality to clear the canvas upon user request.
   2. Provide a button or gesture recognition mechanism for clearing the screen.
7. Saving the Canvas:
   1. Implement functionality to save the canvas as an image file using PIL.
   2. Allow users to specify the filename and location for saving the canvas.
   3. Integrate networking libraries for sharing or storing the canvas on remote servers if required.