Chikitsak Samuha's S. S. & L.S. Patkar College of Art & Science, and V.P. Varde College of Commerce & Economics

SDBI | School of Data Science & Business Intelligence

"The Gesture Gallery Experience"

Image and Video Analytics

By

Shruti Parulekar

Ayush Singh

Sujal Patel

<u>Under the Guidance of</u>

Prof. Kalpesh Modi

Acknowledgement

We would like to express our sincere gratitude to everyone who supported and assisted us in completing our project, **"The Gesture Gallery Experience."** This endeavor has been both challenging and rewarding, and we are thankful for the guidance and encouragement we received throughout the process.

First and foremost, we extend our heartfelt appreciation to Kalpesh Sir, our professor of Image and Video Analytics, for his support and guidance throughout our project.

We also wish to acknowledge the contributions of our project team members. The dedication, creativity, and hard work of each individual were instrumental in the success of this project. Every member brought unique ideas and skills, enabling us to navigate the complexities of hand gesture recognition and create an engaging user experience.

Additionally, we are grateful to our friends and peers for their continuous support, constructive feedback, and motivation throughout the project. Their encouragement helped us remain focused and resilient in the face of challenges.

In conclusion, we are deeply thankful to everyone who contributed to the success of this project. This experience has enriched our learning, fostered teamwork, and enabled us to grow as individuals and as a group.

Thank you.

Introduction:

The Gesture Gallery Experience is an innovative interactive virtual art gallery application that leverages advanced hand gesture recognition technology to create an immersive and engaging user experience. In a world where technology continues to transform how we interact with art and culture, this project aims to bridge the gap between traditional art appreciation and modern digital interfaces.

This application allows users to navigate through a curated collection of famous paintings using intuitive hand gestures, providing a seamless way to explore and engage with art. By incorporating features such as detailed information display about each painting—including the title, artist, and year—along with the ability to rate artworks through specific gestures, the Gesture Gallery Experience seeks to enhance the user's connection to the art.

The application is built using Python and employs libraries such as TensorFlow for hand gesture recognition, OpenCV for image processing, and Tkinter for creating a user-friendly graphical interface. Through this project, we explore the intersection of art, technology, and human-computer interaction, aiming to offer a novel platform for both art enthusiasts and casual viewers.

Ultimately, the Gesture Gallery Experience not only showcases the beauty of art but also emphasizes the potential of technology to transform how we interact with cultural artifacts, making art more accessible and engaging for everyone.

Abstract

The Gesture Gallery Experience is an innovative interactive virtual art gallery application that employs hand gesture recognition technology to enhance user engagement with art. The primary objective of this project is to create an intuitive platform that allows users to navigate through a curated collection of famous paintings,

access detailed information, and rate artworks using simple hand gestures.

Developed in Python using the PyCharm environment, the application utilizes several key libraries, including TensorFlow for training the hand-gesture recognition model, OpenCV for image processing and hand movement detection, Tkinter for building the graphical user interface, and PIL (Pillow) for image manipulation. These technologies work in harmony to deliver a seamless user experience that encourages exploration and interaction.

Key findings from user testing indicate a positive reception, with participants expressing appreciation for the ease of navigation and the engaging interface. The ability to interact with art through gestures not only enhances the overall experience but also fosters a deeper appreciation for the artworks.

The Gesture Gallery Experience signifies a meaningful advancement in the intersection of technology and art, showcasing the potential of interactive applications to transform how we engage with cultural artifacts. Its implications extend beyond mere entertainment, suggesting applications in education and digital exhibitions, making art more accessible and engaging for a broader audience.

Real-World Application

This application serves multiple purposes in various domains:

- **Museums and Galleries**: Allows broader accessibility to art collections, enabling visitors from around the world to explore exhibitions.
- **Art Education**: Creates an engaging and interactive learning platform for students to learn about art and artists in a handson manner.

- **Accessibility**: People with disabilities or limited mobility can use gestures to interact with computers, improving accessibility and interaction with digital devices.

Technologies Used

- **Programming Language**: Python
- **Libraries**:
  - **OpenCV**: For real-time hand movement detection and image processing.
  - **MediaPipe**: Hand detection model to recognize hand landmarks.
  - **TensorFlow**: Used to build the gesture recognition model.
  - **Tkinter**: Graphical User Interface (GUI) toolkit.
  - **PIL (Pillow)**: Image manipulation and display within the application.

Hand Gesture Recognition

Hand gesture recognition is at the core of this application, allowing for natural user interaction through the following process:

- **Hand Detection**: Identifies and tracks hand positions using computer vision techniques (OpenCV + MediaPipe).
- **Feature Extraction**: Extracts relevant features from the detected hand pose and movement.
- **Gesture Classification**: Machine learning models classify gestures (e.g., swiping, thumbs-up, palm gestures) for specific actions.
- **Supported Gestures:**

- **Next Painting**: Swipe right.
- **Previous Painting**: Swipe left.
- **Show Details**: Open palm gesture.
- **Hide Details**: OK sign.
- **Rate Painting**: Thumbs up or down.
- **Exit Application**: Closed fist.

System Requirements

- **Operating System**: Windows, MacOS, or Linux
- **Python Version**: 3.7 or above
- **RAM**: Minimum 4GB (8GB recommended for optimal performance)
- **Webcam**: A functional camera is required for gesture detection.
- **Graphics**: Basic GPU support recommended for TensorFlow.

Installation Guide

- **1. Clone the Repository**

Download the project from the repository:

The_Gesture_Gallery_Experience_Repository

- **2: Install Visual Studio Code**

Download and install https://code.visualstudio.com/ for coding, debugging, and project management.

- **3: Install Git**

Install [Git](https://git-scm.com/) for version control. Use Git to clone your repository and manage your code.

- **4: Install Python**

Install [Python](https://www.python.org/downloads/) to ensure compatibility with the libraries you will use for the plant disease detection model.

- **5. Install Dependencies**

Install the required dependencies using pip:

```
pip install -r requirements.txt
```

- **6. Run the Application**

Start the application by executing the main Python file: `python picture.py`

How to Use the Application

Once the application is launched, the webcam activates to track your hand gestures. Below are the gestures and their respective actions:

- **Next Painting**: Perform a swipe right to move to the next painting.
- **Previous Painting**: Swipe left to view the previous painting.
- **Show Details**: Show your open palm to reveal details of the current painting (title, artist, year).
- **Hide Details**: Perform an "OK" sign to hide the information panel.
- **Rate Painting**: Thumbs-up or thumbs-down gesture to like or dislike a painting.
- **Exit Application**: Perform a closed fist gesture to exit the application.

Project Architecture

The project consists of several core modules:

1. **Hand Gesture Recognition Module**: Built using TensorFlow and MediaPipe, this module detects and recognizes hand gestures in real-time.
2. **Image Processing**: OpenCV is used to handle the real-time video feed and track hand movements.
3. **GUI Module**: Tkinter is used to build the user interface, enabling the display of paintings and interaction panels.
4. **Application Logic**: Connects the gesture recognition with the virtual gallery for seamless interaction.

Conclusion

The Gesture Gallery Experience represents a significant step forward in the integration of technology and art appreciation. By utilizing hand gesture recognition, the application offers an intuitive and interactive way for users to explore a diverse collection of paintings, making art more accessible and engaging than ever before.

Throughout the development process, we took reference from a GitHub project on an "**Interactive Virtual Art Gallery Application with Hand Gestures Recognition,**" which served as an inspiration for our implementation. We aimed to optimize the execution time of the application while simplifying the code to ensure clarity and ease of understanding. To achieve this, we added comprehensive comments throughout the code, making it easier for others to follow our thought process and methodologies.

The positive feedback from users during testing phases highlights the effectiveness of this application in facilitating meaningful interactions with art, encouraging users to engage with each piece on a deeper level and fostering a greater appreciation for artistic expression.

As we look ahead, there is immense potential for expanding the Gesture Gallery Experience. Future enhancements could include integrating augmented reality features, expanding the collection of artworks, or incorporating educational components to further enrich the user experience.

Overall, the Gesture Gallery Experience stands as a testament to the power of innovation in reshaping how we experience and interact with art, and we are excited about the possibilities that lie ahead.