Real-time Hand Gesture Recognition for AR Interaction

Project Overview:

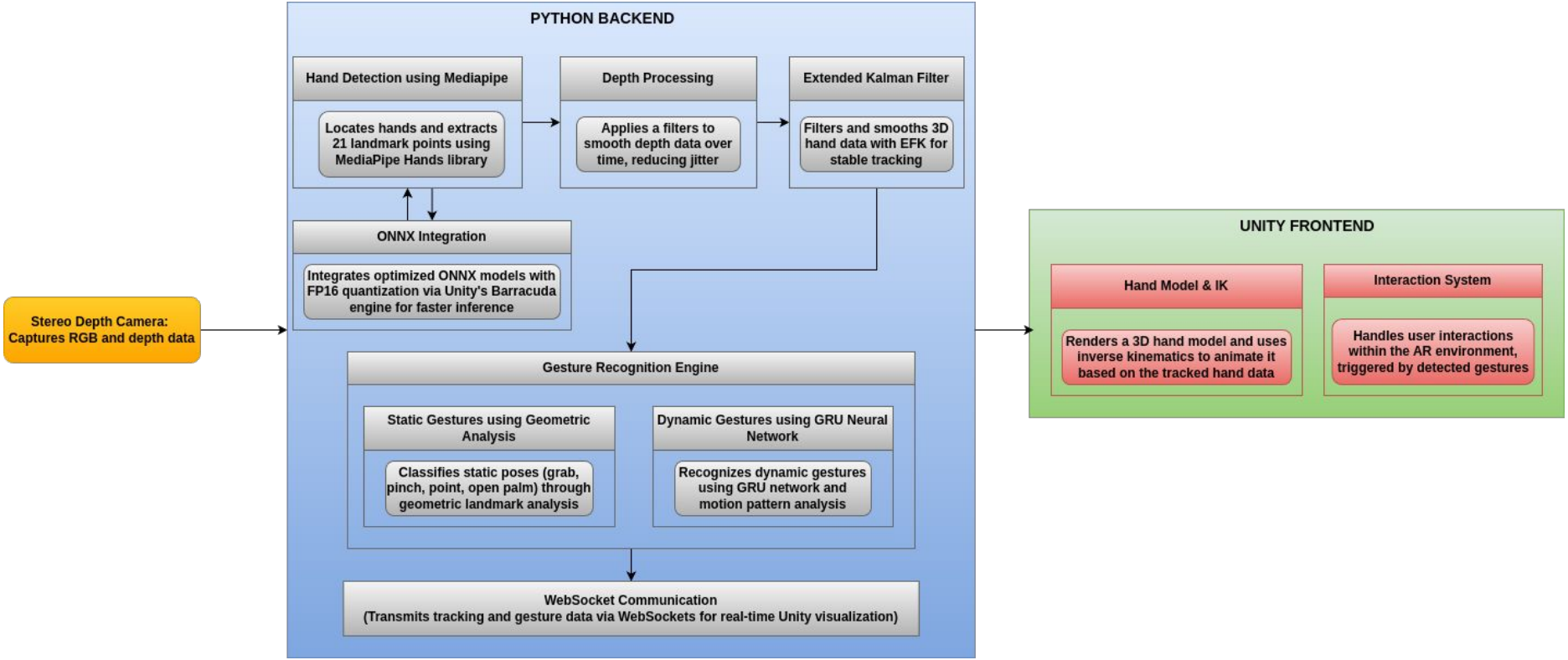
A comprehensive hand tracking and gesture recognition system built for augmented reality applications in automotive training. This system combines advanced computer vision techniques with deep learning to enable intuitive interaction with virtual HVAC components. The architecture integrates Extended Kalman Filtering for precise 3D hand tracking (<7.5mm accuracy), geometric analysis for static gesture recognition (97% accuracy), and a custom GRU neural network for dynamic gesture detection (<30ms latency). Features a Python backend for processing and a Unity frontend for visualization, connected via WebSockets for real-time, low-latency performance at 30+ FPS with optimized ONNX models.

GitHub Repository: VirtuHand

Key Technologies and Skills Used:

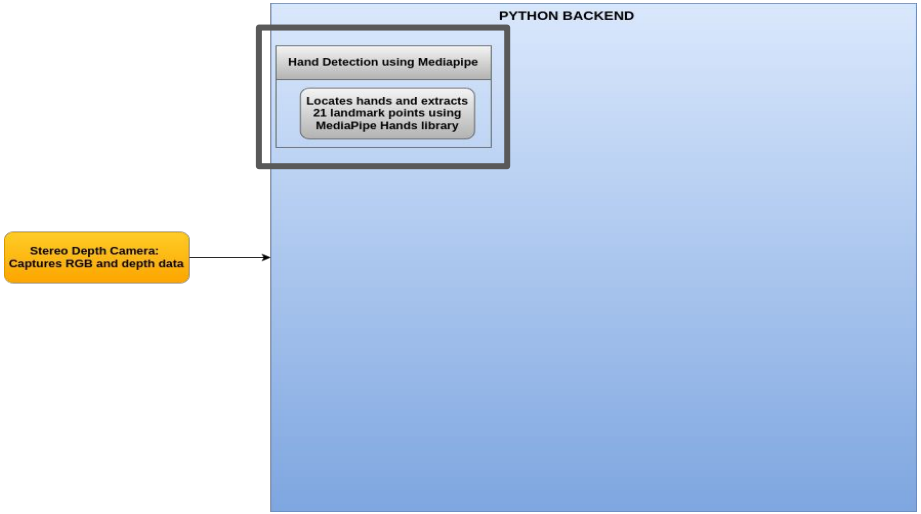
Languages & Frameworks: Python, PyTorch, ONNX, scikit-learn, MediaPipe, Websockets, OpenCV, NumPy
Machine Learning: 3D Tracking, Landmark Detection, Depth Sensing, Geometric Analysis, Kalman Filtering
Deep Learning: Recurrent Neural Networks (GRU), Model Optimization (ONNX), Inference (Unity Barracuda)
Augmented Reality (AR): Unity Development, 3D Interaction Design

Pipeline:

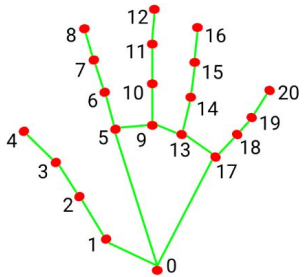


System Architecture and Data Flow

Hand Detection and Landmark Extraction with MediaPipe

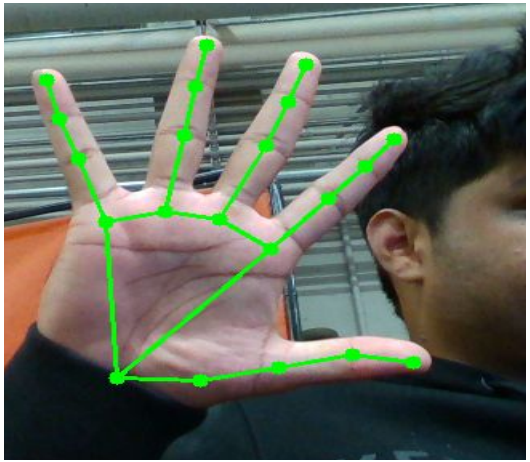


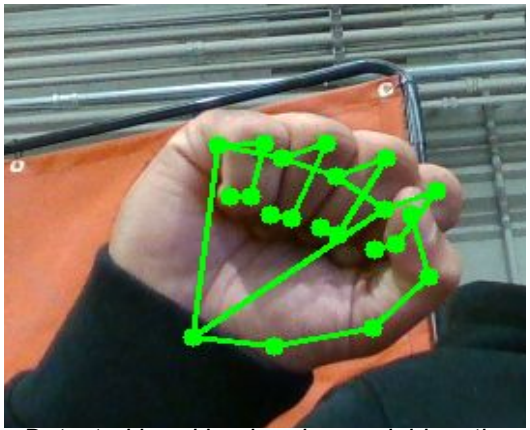
System Architecture and Data Flow



0. WRIST
1. THUMB_CMC
2. THUMB_MCP
3. THUMB_IP
4. THUMB_TIP
5. INDEX_FINGER_MCP
6. INDEX_FINGER_PIP
7. INDEX_FINGER_DIP
8. INDEX_FINGER_TIP
9. MIDDLE_FINGER_MCP
10. MIDDLE_FINGER_PIP
11. MIDDLE_FINGER_DIP
12. MIDDLE_FINGER_TIP
13. RING_FINGER_MCP
14. RING_FINGER_PIP
15. RING_FINGER_DIP
16. RING_FINGER_TIP
17. PINKY_MCP
18. PINKY_PIP
19. PINKY_DIP
20. PINKY_TIP

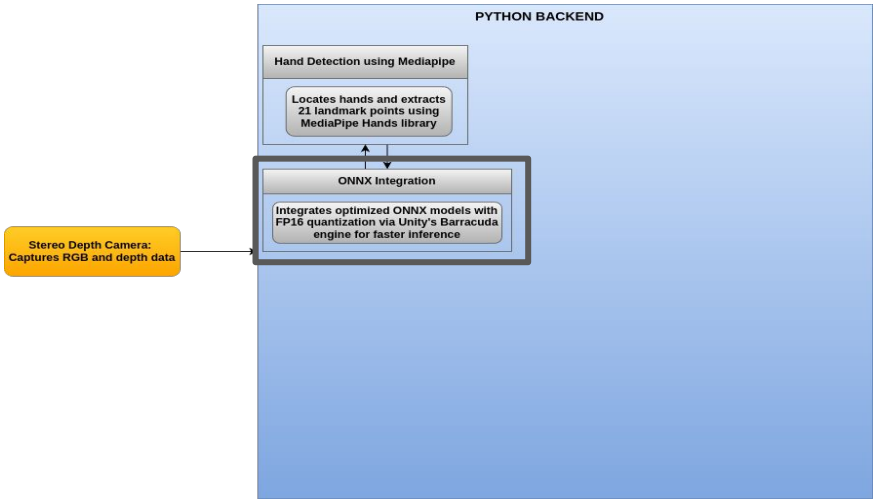
Uses Google's MediaPipe Hands library to detect hand presence and location in the RGB image



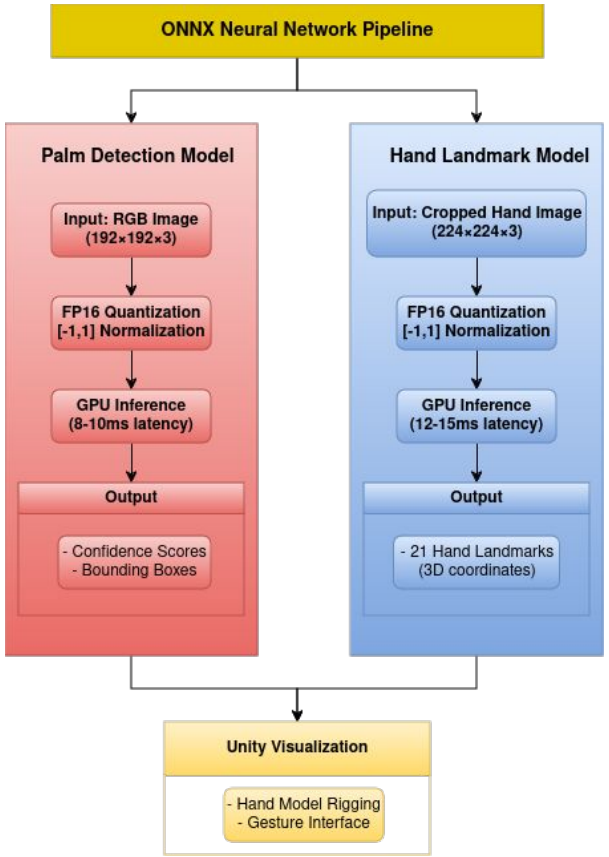


Detected hand landmarks overlaid on the original image in real-time

Neural Network Optimization with ONNX



System Architecture and Data Flow

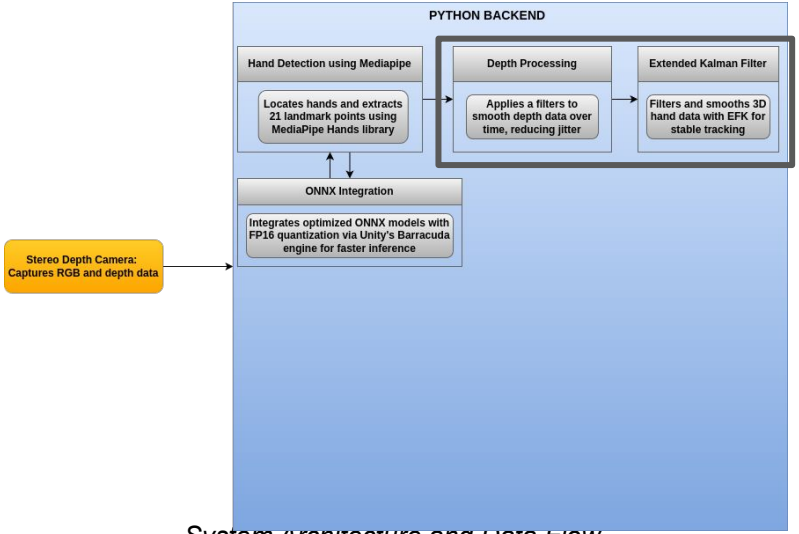


Two-stage ONNX pipeline

To improve performance and reduce reliance on external libraries, an experimental pipeline was developed using ONNX (Open Neural Network Exchange) models for hand detection and landmark extraction. This approach leverages the Unity Barracuda engine for GPU-accelerated inference.

1. **Performance Optimization**
   - FP16 quantization reducing model size by 50%
   - 33% faster inference compared to MediaPipe

3D Hand Tracking with Extended Kalman Filter

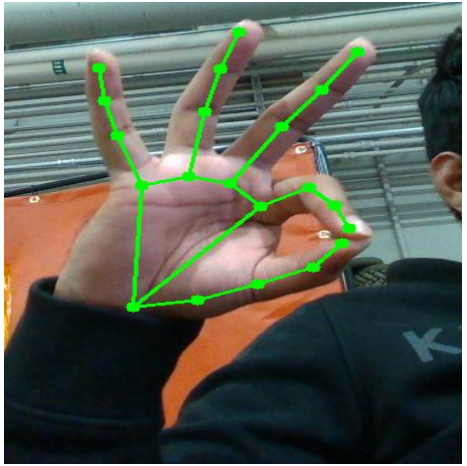


System Architecture and Data Flow

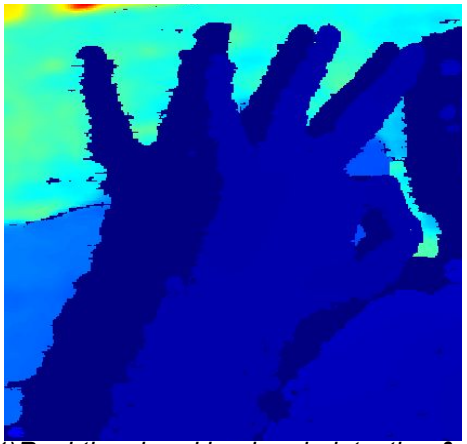
- Combines MediaPipe landmarks with RealSense depth data and applies Extended Kalman Filtering to achieve smooth 3D tracking at 30Hz, with robustness to occlusions and jitter reduction.
- Enables precise velocity estimation and maintains tracking during fast hand movements

Multi-stage depth filtering for 3D hand tracking

(1)

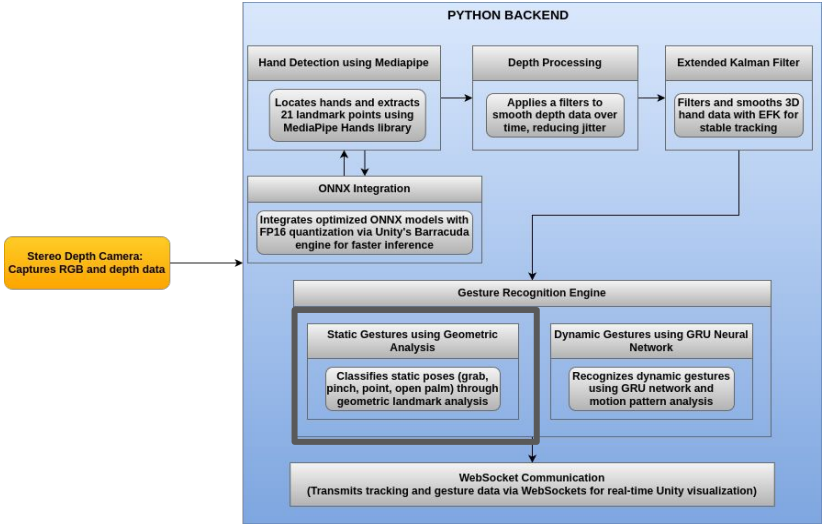


(2)



(1)Real-time hand landmark detection &
(2) Visualization of depth map using Filter

Static Gesture Recognition

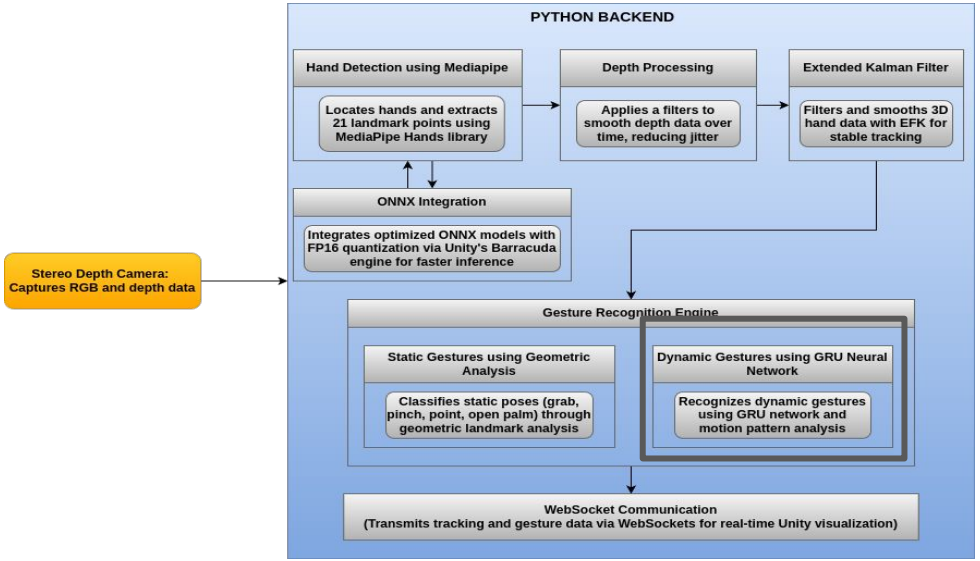


System Architecture and Data Flow

Real-time detection of OPEN_PALM, PINCH, GRAB, and POINT gestures

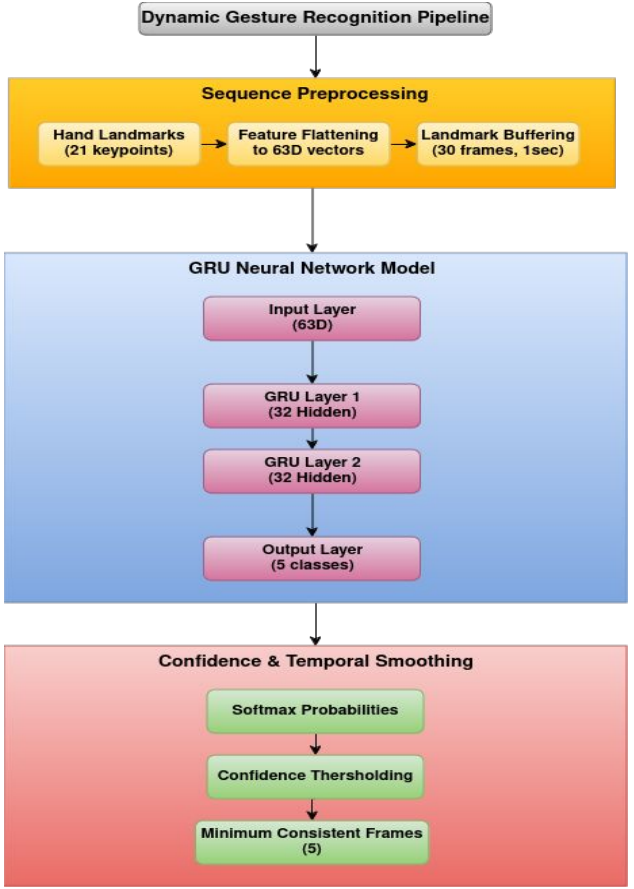
Gesture	Description	Detection Logic
Open Palm	All fingers extended, hand open.	All fingertips are further from wrist than their corresponding base joints. Thumb is extended.
Pinch	Thumb and index finger close together.	Distance between thumb tip and index fingertip below threshold with one finger near thumb
Grab	Fingers curled inwards towards the palm.	All fingertips are closer to wrist than their corresponding base joints. Thumb is also curled.
Point	Index finger extended, others closed.	Index fingertip is extended (from wrist than its base). All other fingertips are curled

Dynamic Gesture Recognition



System Architecture and Data Flow

- A GRU network is a type of recurrent neural network (RNN) that is well-suited for processing sequential data, such as the time series of hand landmark positions.
- The GRU model was trained on a custom dataset of dynamic gestures.
- Input: The model takes a sequence of 30 frames of hand landmark data (21 landmarks x 3 coordinates = 63 input features).
- Output: The model predicts the probability of each supported dynamic gesture.
- Architecture : Input size is 63, hidden layer is 32 and 2 layers.

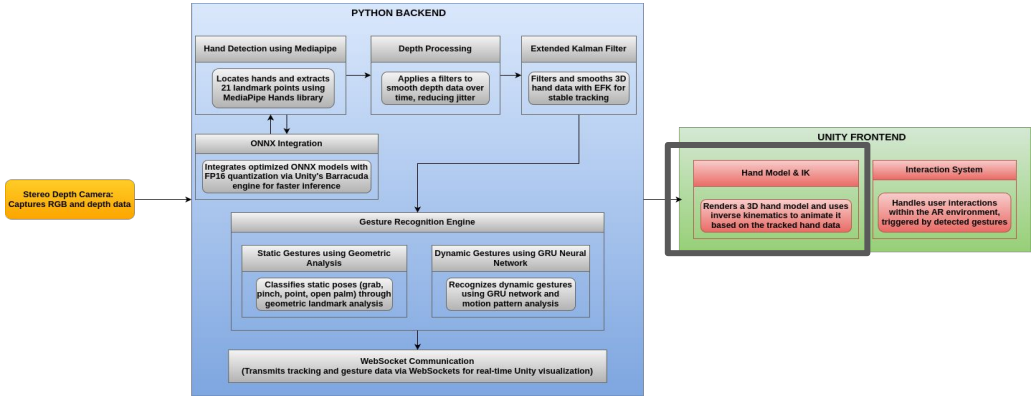


GRU Architecture Diagram

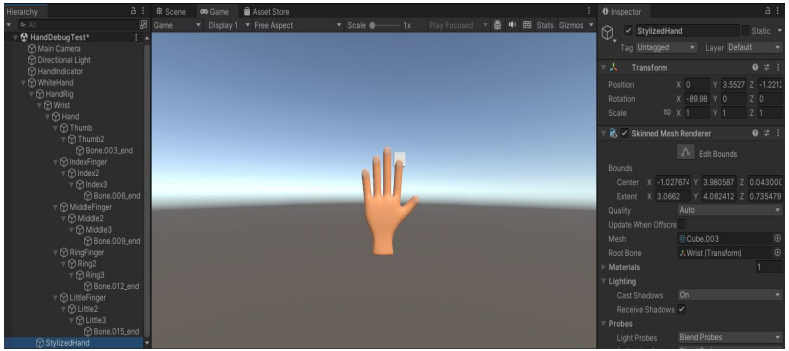
Dynamic Gesture Recognition

Real-time detection of dynamic gestures (SWIPE_LEFT, SWIPE_RIGHT, CIRCLE) using the GRU model and motion pattern analysis

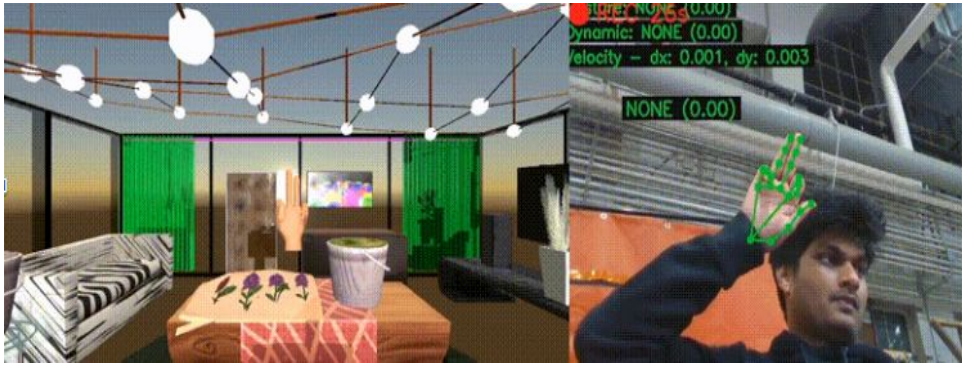
Unity Frontend: Hand Rigging and Interaction



System Architecture and Data Flow

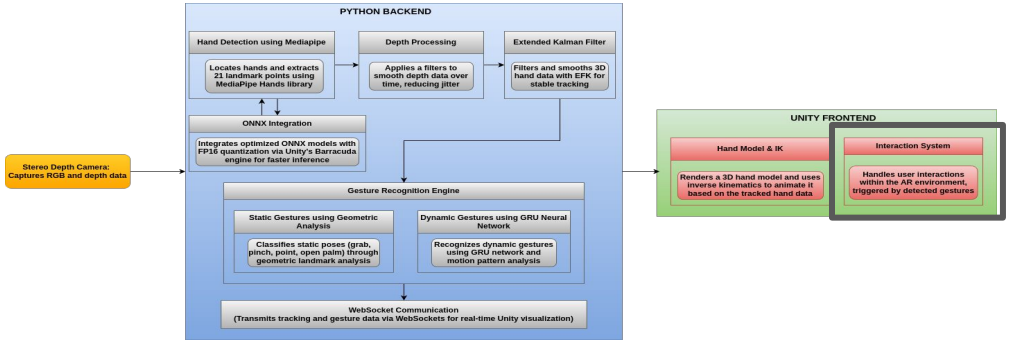


A rigged 3D hand model within the Unity editor

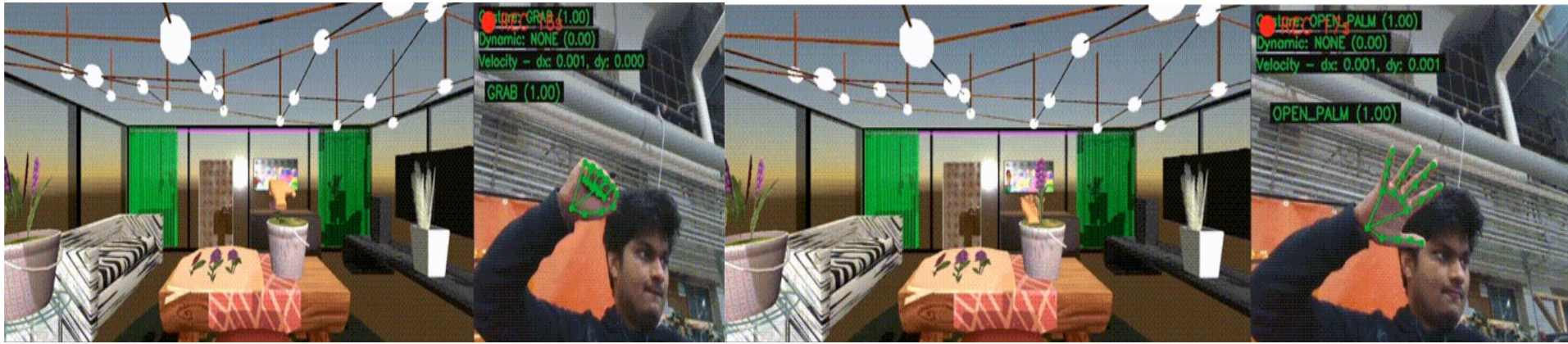


Real-time hand rigging in Unity. The 3D hand model accurately mirrors the user's hand movements and gestures.

Unity Frontend: Real-time AR Implementation & Testing



System Architecture and Data Flow



Sequence demonstrating the virtual flower arrangement demo. The user can grab, move, and place flowers using hand gestures