Project Report: Touchless HCI for Media Control

Platform: NVIDIA Jetson Nano Developer Kit and VLC Player

Core Technologies: Python 3.6, OpenCV, MediaPipe, Scikit-Learn (SVM), VLC Media Player

1. Objective and Problem Statement

The objective of this project was to engineer a robust, touchless Human-Computer Interaction (HCI) system utilizing an NVIDIA Jetson Nano. The system translates real-time hand gestures captured via a USB webcam (Logitech C920 Pro) into system-level media control commands for VLC Media Player. The primary challenge was achieving ultra-low latency (<200 ms) and high accuracy (>90%) on edge-computing hardware with constrained memory and CPU resources.

2. Methodology & Model Choice

The pipeline is divided into three core stages: Data Collection, Training, and Live Edge Inference.

- **Perception Pipeline (MediaPipe):** Google's MediaPipe Hands framework was selected to extract 21 3D landmarks from a live video feed. MediaPipe was chosen for its highly optimized performance on ARM-based CPUs and Jetson hardware.
- **Data Normalization (Position-Independence):** To prevent the machine learning model from memorizing the spatial location of the hand on the screen, all landmark coordinates were mathematically normalized. By subtracting the root wrist node (x0, y0) from all subsequent joints, the system focuses strictly on the geometric shape of the gesture, vastly improving real-world accuracy.
- **Classification Model (Support Vector Machine):** A Support Vector Machine (SVC with a linear kernel) was trained on a custom dataset of 2,198 samples. SVM was explicitly chosen over deep learning classifiers because of its lightweight memory footprint, making it ideal for the hardware constraints of the Jetson Nano while still delivering high-dimensional accuracy.
- **System-Level Control:** Gesture classifications are mapped to system keyboard inputs using the pyautogui library, which interfaces smoothly with the local VLC instance.

3. Defined Gesture Set and Mapping Table

The system defaults to a secure "Locked" state to prevent unintended inputs and uses a biometric-style handshake (Palm) to initiate control.

Gesture Recognized	Mapped VLC Action	System Execution
PALM	UNLOCK / PLAY / PAUSE	Spacebar
FIST	HARD STOP	s
PINCH	VOLUME UP (+)	Volume Up
POINT	VOLUME DOWN (-)	Volume Down
THUMB LEFT	REWIND (<<)	Left Arrow
THUMB RIGHT	FORWARD (>>)	Right Arrow
NO HAND	AUTO-LOCK / STANDBY	*System Wait*

4. Hardware Utilization & Optimization Techniques

Running MediaPipe inference alongside a video decoder natively exceeds the Jetson Nano's shared memory limit, often triggering Linux Out-Of-Memory (OOM) crashes. The following edge-optimizations were engineered to stabilize the pipeline:

- **Edge Frame Skipping:** The inference loop is programmed to process frame_count. By only executing the heavy AI model on every second frame, CPU throttling is eliminated, allowing VLC to decode video smoothly.

- **Resolution and Memory Management:** The camera feed read resolution was dynamically reduced to 320x240, freeing up sufficient RAM. Furthermore, a 4GB Swapfile was provisioned on the SD card to prevent OOM events.

5. Performance Analysis

The system was tested in controlled lighting at a 50cm operating distance, successfully passing and exceeding all rubric targets:

- **Target: >90% Accuracy** -> **Achieved: 95.45%** (Validated via Scikit-Learn validation splits using position-independent coordinate data).
- **Target: <200 ms Latency** -> **Achieved: ~40-80 ms** (Optimized via Frame Skipping and OpenCV Buffer reduction).
- **Target: Stable at ≥15 FPS** -> **Achieved: 30 FPS Camera Read / 15 FPS Inference** (Ensures fluid interaction without crashing the local media player).

To implement the "Touchless HCI for Media Control" system detailed in your document, you will need the following hardware components and specific specifications for the NVIDIA Jetson Nano platform.

Core Hardware Components

- **NVIDIA Jetson Nano Developer Kit**: The central processing unit for the AI pipeline. You can use either the 4GB version or the 2GB version, though 4GB is recommended for better handling of concurrent neural networks.
- **USB Webcam**: A Logitech C920 Pro is specified in your report for capturing real-time hand gestures.
- **MicroSD Card**: A minimum of 32GB (UHS-1) is required for the operating system and project files; 64GB or more is recommended to avoid storage bottlenecks.
- **Display & Peripherals**: An HDMI or DisplayPort monitor, a USB keyboard, and a mouse are needed for initial setup and running the VLC Media Player interface.

Jetson Nano Technical Specifications

According to the hardware documentation, the device provides the following capabilities utilized by your project:

- **GPU**: 128-core NVIDIA Maxwell architecture, which accelerates the MediaPipe hand tracking and Scikit-Learn inference.
- **CPU**: Quad-core ARM Cortex-A57 MPCore processor.
- **Memory**: 4 GB 64-bit LPDDR4 (25.6 GB/s bandwidth).
- **Connectivity**: Gigabit Ethernet or an optional M.2/USB Wi-Fi adapter for downloading libraries like mediapipe and opencv-python.

6. Learning Outcomes

This project provided profound practical experience in real-time edge computer vision. Key learnings included pipeline optimization for low-latency inference, hardware-specific memory management on the Jetson architecture, and the successful integration of AI perception with system-level control logic.