Virtual reality enabled activity recognition for immersive computing

Sudeep Babasaheb Kakade, Aleeza Khan, Ryan Naini, Samhita Shriram, and Andrew Chan

Introducing The Team

Presenter

Sudeep Kakade

Aleeza Khan

Ryan Naini

Samhita Shriram

Andrew Chan

Mentor

Changming Li

Honglu Li

Zejun Xu

Dirk Catpo Risco

Advisor

Prof. Yingying (Jennifer) Chen

Motivation

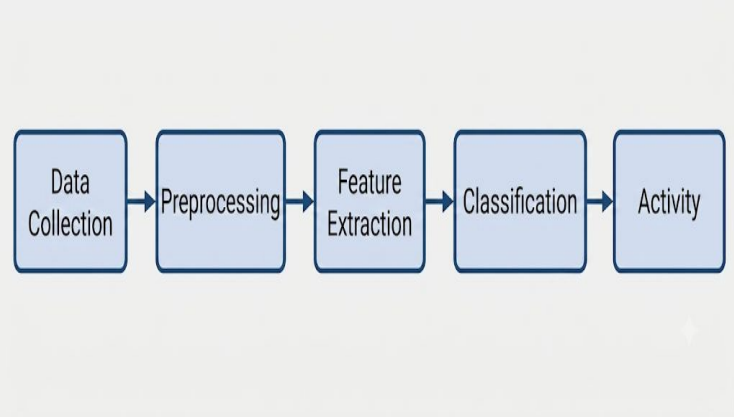
- Meta Quest VR headsets used in gaming, fitness, healthcare — 20M+ units sold
- Built-in IMU sensors capture motion data at 72Hz — always active, no permission needed
- **Problem:** Current activity recognition requires extra wearables or cameras
- **Our goal:** Recognize activities using only the headset's existing sensors

Project Goal & Flow

Goal: Recognize human activities using only Meta Quest built-in sensors — no extra hardware needed

Project Flow:

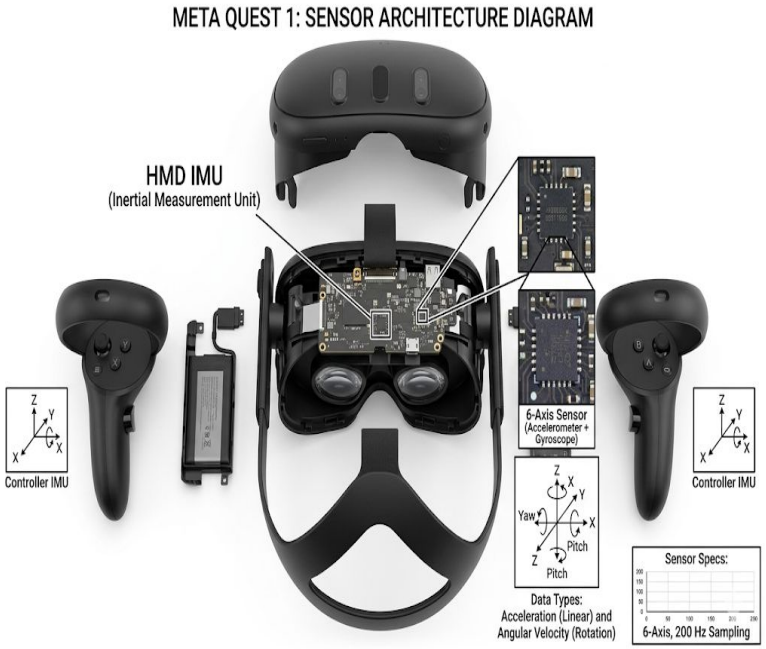
- Build complete pipeline: data collection → preprocessing → feature extraction → classification
- Implement signal preprocessing
- Compare ML classifiers (KNN, SVM, Random Forest, CNN)
- Explore LLM-based classification with zero training data following Penetrative AI concept



Sensor Data

What each sensor physically measures:

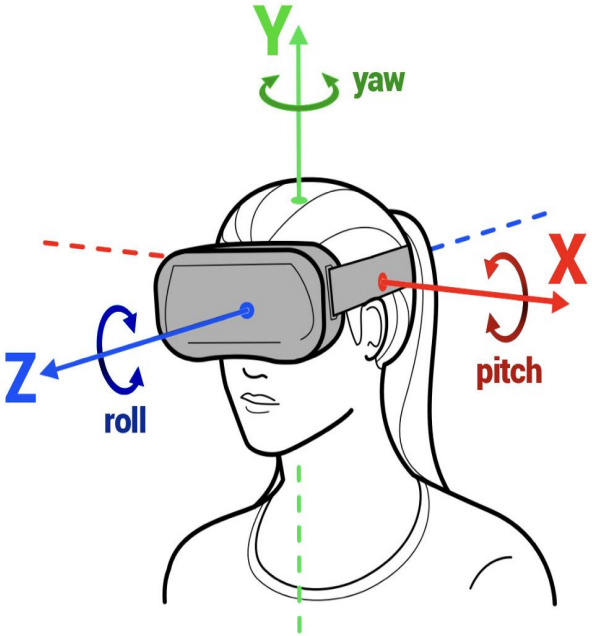
- **Accelerometer** → captures LINEAR_ACCEL — detects body movement forces like lunging forward, squatting down, arm raising
- **Gyroscope** → captures ANGULAR_VEL — detects rotational motion like head turning left/right, nodding up/down



VR Sensors & IMU

Meta Quest HMD contains an Inertial Measurement Unit (IMU) with accelerometer and gyroscope, providing 4 sensor outputs at 72Hz:

- **LINEAR_ACCEL (x,y,z)** — measures translational forces (how fast body moves)
- **ANGULAR_VEL (x,y,z)** — measures rotational speed (how fast head turns)
- **LINEAR_VEL (x,y,z)** — measures translational speed (body movement direction)
- **ANGULAR_ACCEL (x,y,z)** — measures rotational acceleration (how quickly rotation changes)



Data Collection Module

- Built custom C++ app using Android Studio + Meta VrApi SDK
- Extracted raw sensor data via ADB logcat at 72Hz
- Filtered and parsed using PowerShell + Python scripts
- Collected 50 reps per activity with 3 second rest period

11 Activities Captured

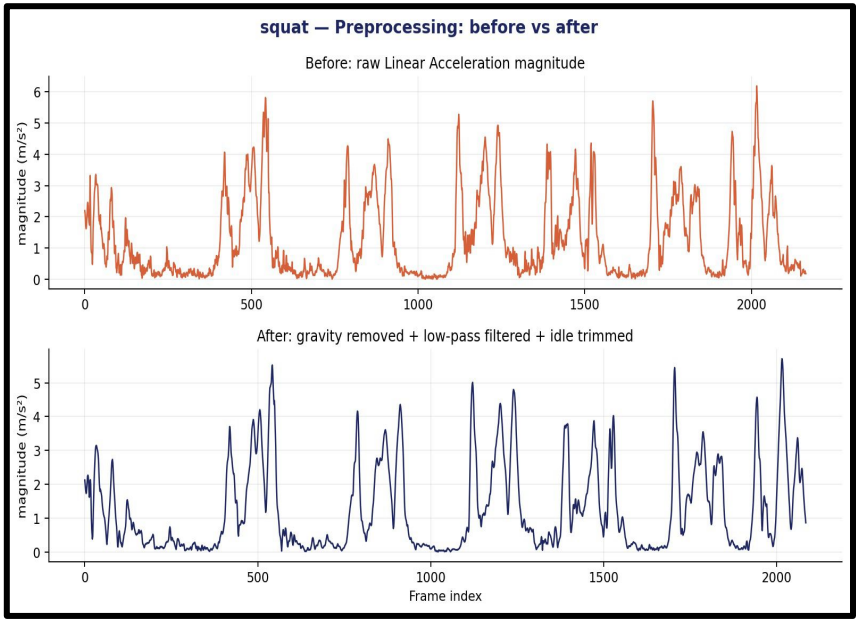
Activity	Device	Windows
Lunge	HMD	21,083
Bicep Curl	CTRL	19,452
Head Rotation	HMD	34,436
Squat	HMD	21,303
Nodding	HMD	24,127
Side Raise	CTRL	22,811
Arm Swing	CTRL	18,835
Neck Stretch	HMD	21,070
Tree Pose	CTRL	19266
Jumping Jacks	HMD	20,494
Forward Fold	HMD	24,523
Total		**247,400**

Data Preprocessing Module

Raw Sensor Data → Gravity Removal → Low-Pass Filter → Idle Frame Trimming → Sliding Window (40 frames, stride=1)

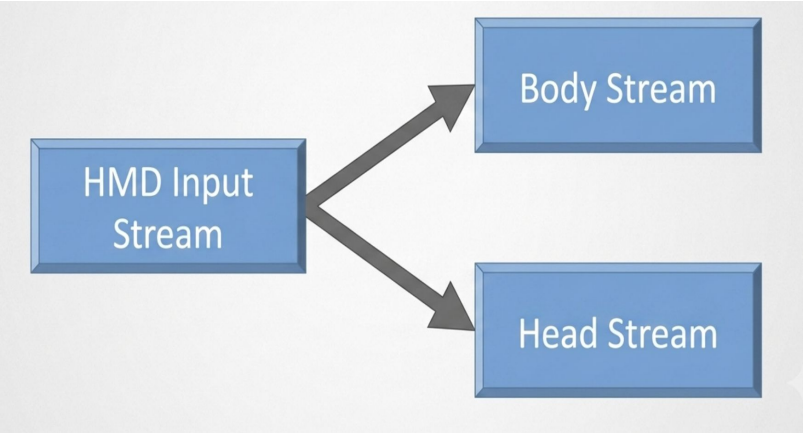
Filters Used:

- **Gravity removal** — subtract mean acceleration
- **Low-pass filter** — 5-frame moving average
- **Idle trimming** — remove standing-still frames (15% threshold)
- **Sliding window** — 40 frames ≈ 0.55 seconds per window



Two-Stream Design

- **Body Stream** — Linear Velocity (x,y,z) — captures translational motion
- **Head Stream** — Angular Velocity + Angular Acceleration (x,y,z each) — captures rotational motion
- Processing separately preserves physical distinction between body actions and head gestures



PCA Feature Extraction + Top 5 Features

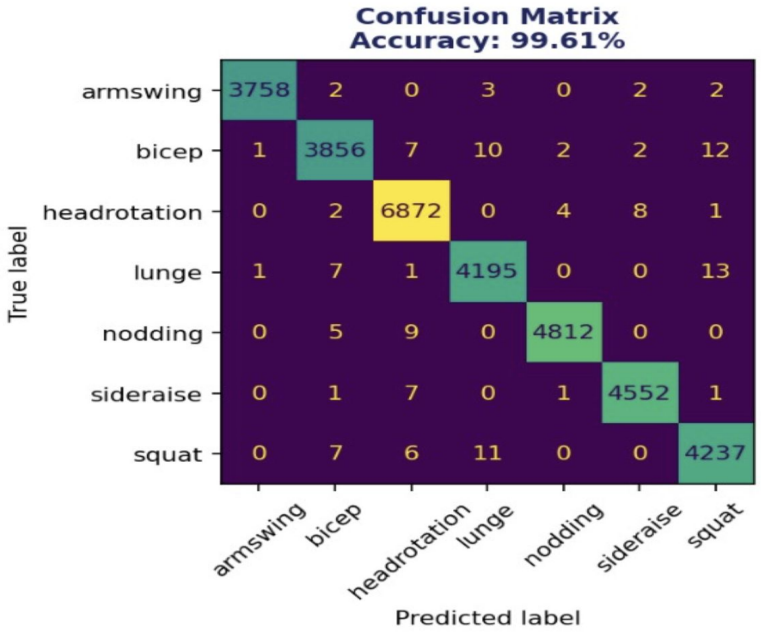
Feature Extraction:

- 5 statistical features (mean, std, min, max, energy) × 12 sensor channels = 60 features per window
- Two-stream PCA: Body (2 components) + Head (3 components) = 5 PCA components
- Accuracy: 98.70%

Rank	Feature	Individual Accuracy
1	ANGULAR_ACCEL_z_min	95.77%
2	ANGULAR_ACCEL_x_max	95.67%
3	ANGULAR_ACCEL_x_min	95.61%
4	ANGULAR_ACCEL_z_max	95.31%
5	ANGULAR_ACCEL_y_max	95.08%

Classifier Comparison

Classifier	Accuracy	Train Time	Predict Time
Two-Stream CNN/MLP	99.61%	103s	0.13s
Random Forest (5 PCA)	93.91%	50.57s	0.05s
KNN k=3 (5 PCA)	95.43%	0.35s	0.18s
KNN k=5 (5 PCA)	93.14%	0.25s	0.26s
SVM RBF (5 PCA)	67.07%	373s	3.76s



Two-Stream CNN achieves best accuracy (99.61%) by learning body and head patterns separately

Prompt Engineering

3 prompting strategies tested with Gemini LLM:

- **Zero-shot** — background info + sensor data, no examples
- **Few-shot** — 1 labeled example per activity before unknown window
- **Chain of thought** — step-by-step reasoning through sensor physics

Please answer with:

1. Your predicted activity
2. Confidence level (low/medium/high)
3. Which sensor features most influenced your decision
4. Which activities you ruled out and why

Tested 5 times per activity (55 total tests)
Result: 100% accuracy for all 11 activities across all 3 strategies

LLM vs Trained Classifiers

Classifier	Accuracy	Training Data Needed
Two-Stream CNN/MLP	99.61%	247,400 labeled windows
KNN k=3 (5 PCA)	95.43%	247,400 labeled windows
Gemini LLM	100%	**Zero training data**

- Gemini LLM outperforms trained classifiers with zero labeled training data
- Validates Penetrative AI — LLMs comprehend physical sensor data using built-in world knowledge
- LLM provides interpretable reasoning for each prediction

Key Contributions

- Complete VR activity recognition using only built-in HMD sensors
- 247,400 labeled windows across 11 activities
- 99.61% accuracy with two-stream CNN/MLP
- Angular Acceleration min/max identified as most discriminative features
- 100% LLM accuracy with zero training data — validates Penetrative AI concept

Future Work

- Expand to more activities and multiple users for generalization
- Implement real-time activity recognition on live VR sensor stream
- Deploy on-device classification for standalone VR headset
- Explore fine-tuning LLM with sensor-specific training for even more complex activities