LLM-based Activity Recognition using VR Headsets

Leon Du & Jerry Huang

Project Advisor: Dr. Yingying (Jennifer) Chen
Mentors: Changming Li, Honglu Li, and Tianfang Zhang

Introducing the Team

Interns

Intern
Leon Du
HS

Intern
Jerry Huang
HS

Mentors & Advisor

Mentor
Changming Li
RU ECE PhD

Mentor
Honglu Li
RU ECE PhD

Mentor
Tianfang Zhang
RU ECE PhD

Advisor
Professor. Yingying (Jennifer) Chen

Motivation

- VR devices have attracted millions of users and facilitate a broad array of emerging VR applications
- Motion sensors imbedded into the Head Mounted Display (HMD) can be used to detect these movements
- LLM shows high generality to interpret rich semantic interactions from input (i.e., text, motion data, etc.)
- LLM eliminate the needs for large scale data collection

Gaming

Shopping

Banking

Education

Objectives

- This project aims to recognize human activities through analyzing motion sensor data with artificial intelligence (AI) technologies

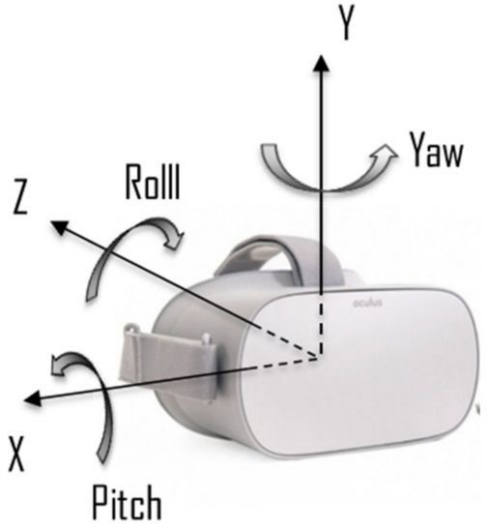
- Collect motion sensor data from VR headsets and extracting useful features from the motion sensor data using Support Vector Machine (SVM)

- Use Large Language Models (LLMs) with motion sensor data input to perform human activity recognition task

Data Collection

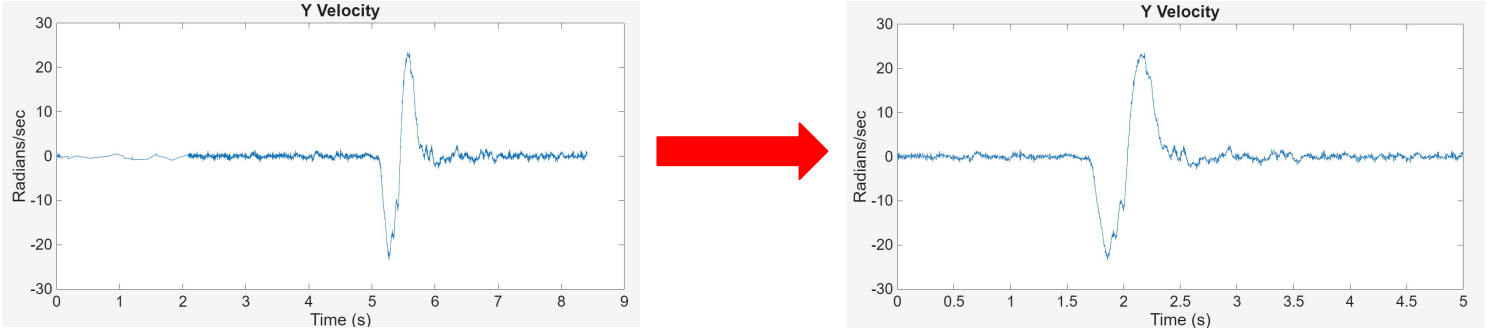
- We used the integrated motion sensors built into the Oculus Quest 1

- The motion sensors provides two streams of data
  - Accelerometer (axis rotation)
  - Gyroscope (axis acceleration)

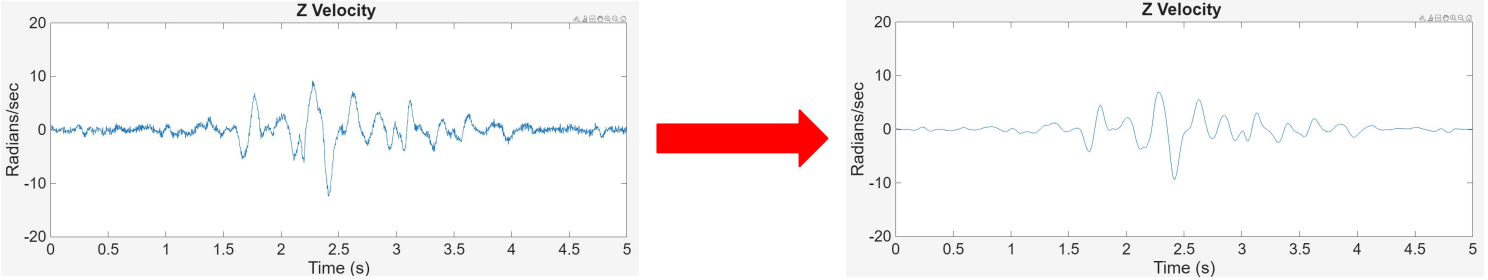


Data Preprocessing

- Segmentation: 5000 effective data samples for each activity

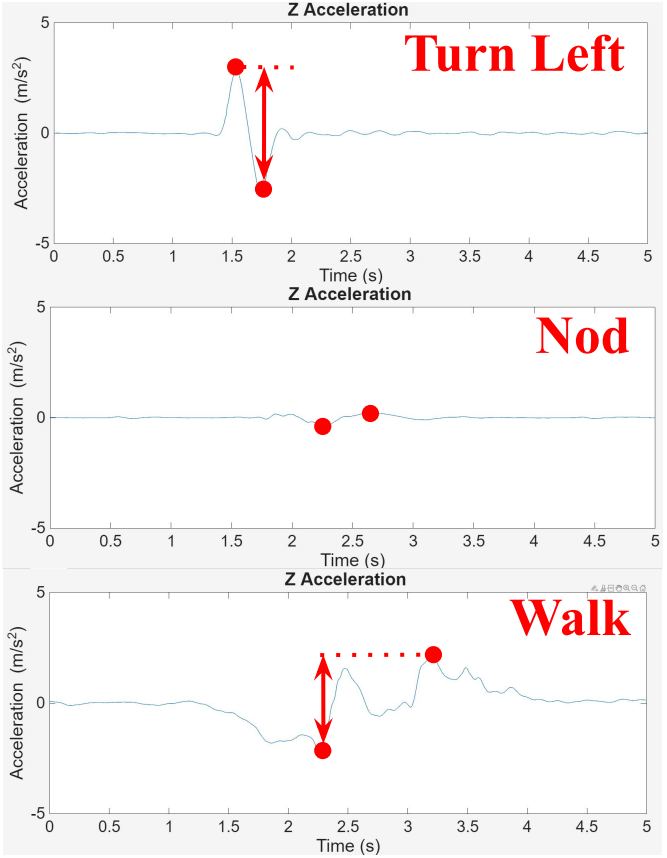


- De-noise and smooth data to generate accurate waveforms



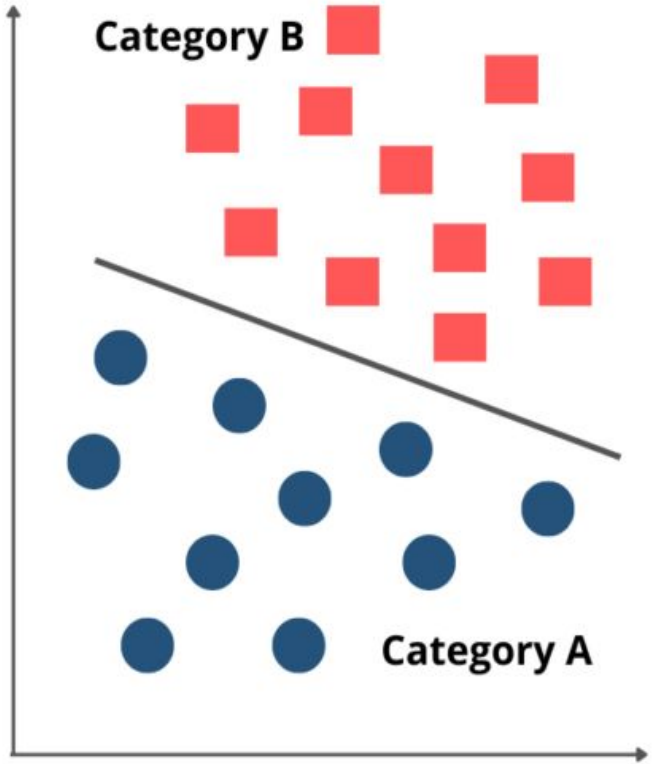
Feature Extraction

- Effective features need to be extracted from the processed motion sensor data for further classification
- Simplify the model input by extracting only the most informative attributes
- Well-selected features can lead to better classification performance



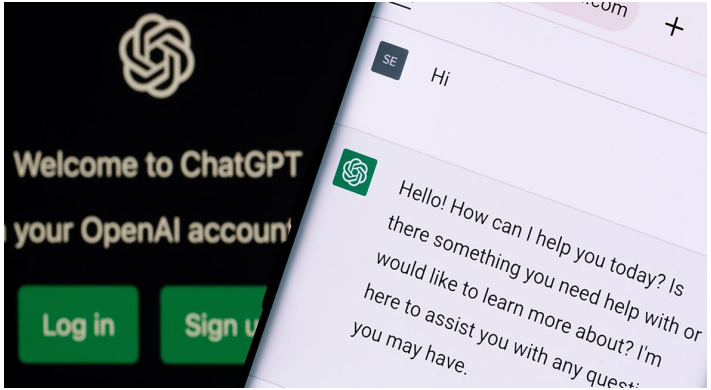
Support Vector Machine (SVM)

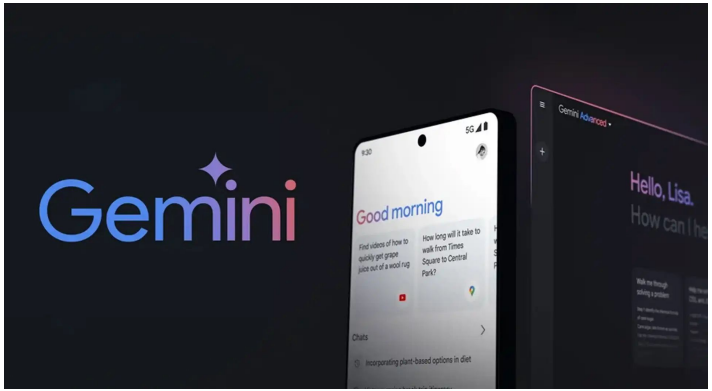
- An effective machine learning algorithm that finds a hyperplane that separates classified data points
- Requires a huge amount of data from users to train the model training for accurate prediction
- We use SVM to identify effective features for classification



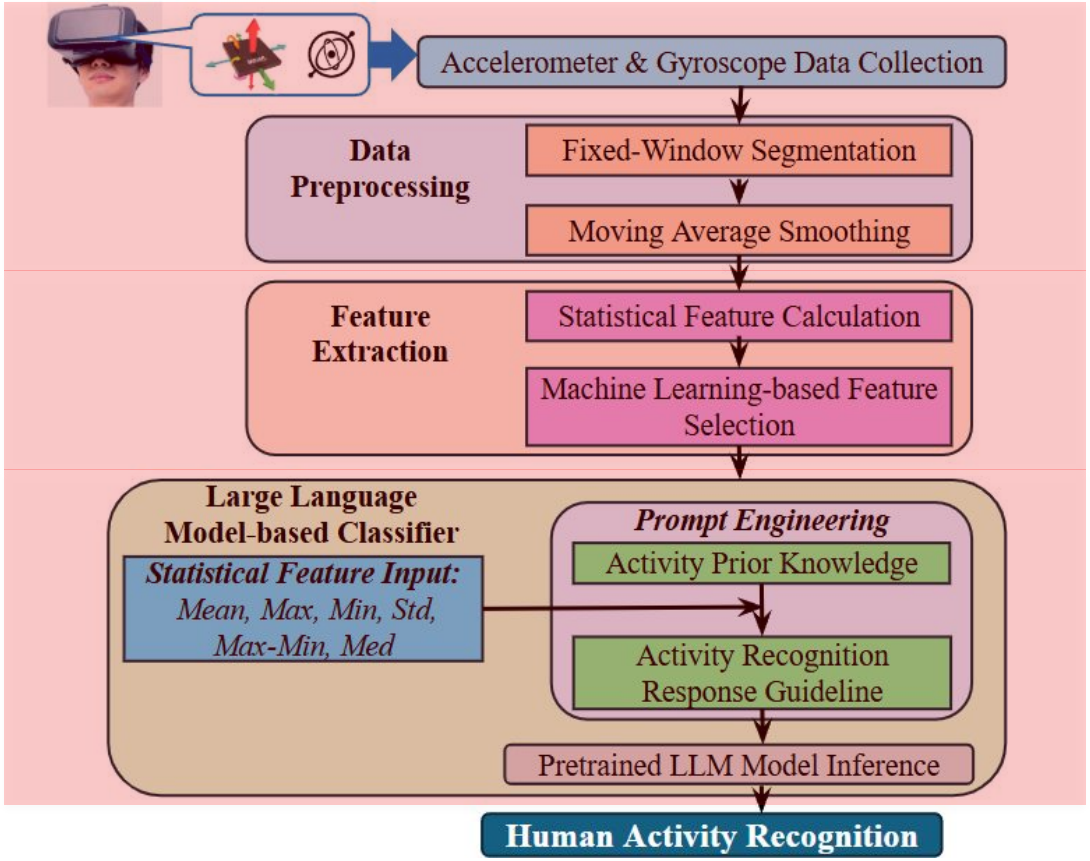
Large Language Model (LLM)

- LLMs can capture complex temporal and semantic relationships in activity data
- Unlike traditional ML models, LLMs can be used as pre-trained model with simple prompts for any new classification tasks
- Requires no training data from users to accurately recognize human activities





System Overview



Experimental Setup

- Using Android Studio, we develop an application to extract data from the IMU sensors on HMD of Meta Quest

```
for (int i = 0; i < NUM_INSTANCES; i++) {
    const int index = scene->CubeRotations[i];

    // Write in order in case the mapped buffer lives on write-combined memory.
    cubeTransforms[i].M[0][0] = rotationMatrices[index].M[0][0];
    cubeTransforms[i].M[0][1] = rotationMatrices[index].M[0][1];
    cubeTransforms[i].M[0][2] = rotationMatrices[index].M[0][2];
    cubeTransforms[i].M[0][3] = rotationMatrices[index].M[0][3];

    cubeTransforms[i].M[1][0] = rotationMatrices[index].M[1][0];
    cubeTransforms[i].M[1][1] = rotationMatrices[index].M[1][1];
    cubeTransforms[i].M[1][2] = rotationMatrices[index].M[1][2];
    cubeTransforms[i].M[1][3] = rotationMatrices[index].M[1][3];

    cubeTransforms[i].M[2][0] = rotationMatrices[index].M[2][0];
    cubeTransforms[i].M[2][1] = rotationMatrices[index].M[2][1];
    cubeTransforms[i].M[2][2] = rotationMatrices[index].M[2][2];
    cubeTransforms[i].M[2][3] = rotationMatrices[index].M[2][3];

    cubeTransforms[i].M[3][0] = scene->CubePositions[i].x;
    cubeTransforms[i].M[3][1] = scene->CubePositions[i].y;
    cubeTransforms[i].M[3][2] = scene->CubePositions[i].z;
    cubeTransforms[i].M[3][3] = 1.0f;
}
```

```
while(1){
    double current_time = GetTimeInSeconds();
    if(current_time-prev<gap){
        continue;
    }

    //ovrTracking2 tracking2 = vrapi_GetPredictedTracking2(appState->Ovr, current_time);
    ovrTracking tracking2 = vrapi_GetPredictedTracking(appState->Ovr, current_time);
    double x_acc = tracking2.HeadPose.LinearAcceleration.x;
    double y_acc = tracking2.HeadPose.LinearAcceleration.y;
    double z_acc = tracking2.HeadPose.LinearAcceleration.z;

    double x_gyro = tracking2.HeadPose.AngularAcceleration.x;
    double y_gyro = tracking2.HeadPose.AngularAcceleration.y;
    double z_gyro = tracking2.HeadPose.AngularAcceleration.z;

    ALOGV("Acceleration %f %f %f %f", current_time, x_acc, y_acc, z_acc);
    ALOGV("Gyroscope %f %f %f %f", current_time, x_gyro, y_gyro, z_gyro);

    prev = current_time;
};
```

Experimental Setup

- We designed 5 activities for evaluation, including three head-related activities and two body-related activities


Turn left


Turn right


Nod

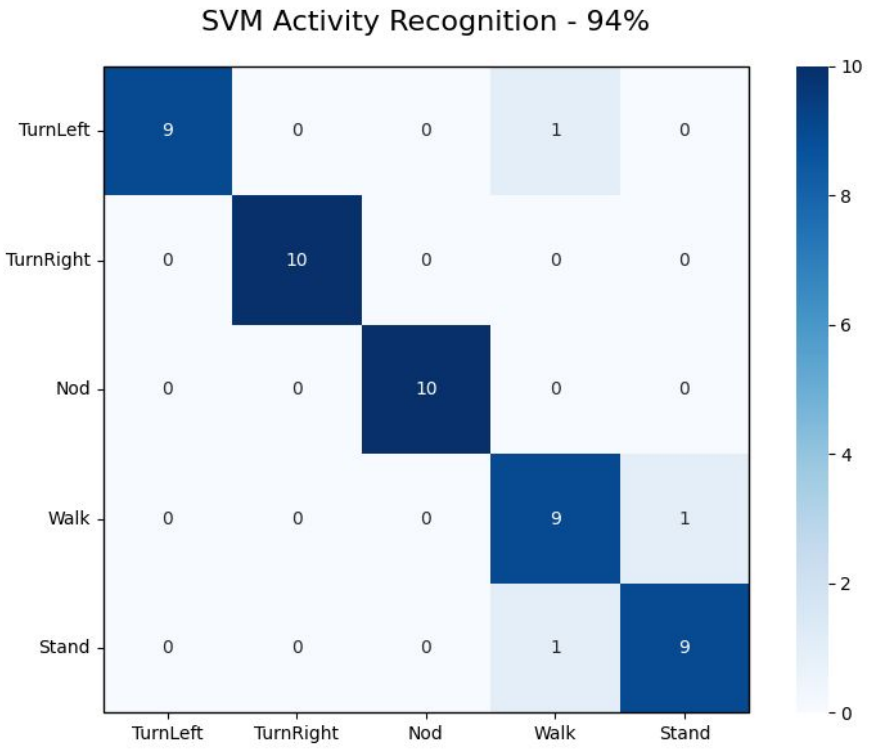

Walk


Stand up

- 50 samples per activity were collected
  - 250 samples total
  - 8:2 Training and Testing Ratio

Activity Recognition Using SVM

- 6 effective statistical features are identified
  - Maximum
  - Minimum
  - Max-to-min
  - Standard deviation
  - Mean
  - Median
- The overall testing accuracy achieves 94%



Prompt Design for LLM

Develop LLM prompt to understand and analyze the motion data

- Explain the goal of the task and data types to be received
- Extract features from the data and provide expert knowledge for how to analyze
- Provide a response template for results

Prompt design example

Y-Axis Maximum Acceleration (from Accelerometer)
- High values → Active vertical movements (e.g., stand up, walk, nod)
- Low values → Minimal vertical movement (e.g., turnleft, turnright)

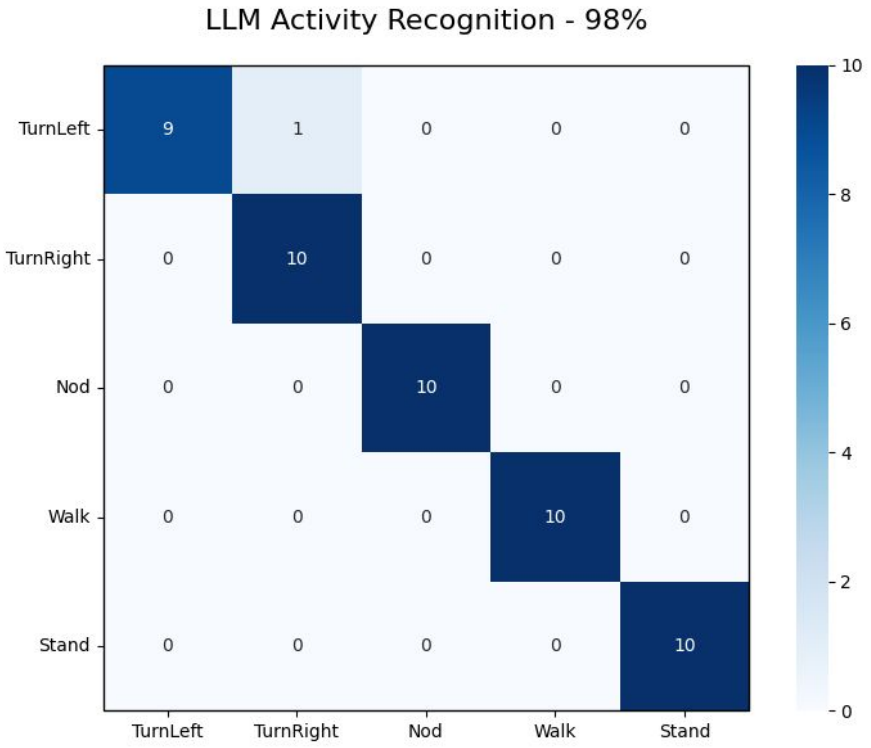
Z-Axis Mean Acceleration (from Accelerometer)
- Far from 0 → Forward/backward motion (e.g., walk, nod, stand up)
- Near 0 → Stationary, head rotation (e.g., turnleft, turnright)

X-Axis Gyroscope Activity (Rotational velocity)
- High variation → Nodding or possibly standing up
- Low variation → Turning or walking

Y-Axis Mean Gyroscope Reading (Rotational velocity)
- Negative → Turning head left
- Positive → Turning head right
- Near 0 → Other motions (nod, stand up, walk)

Activity Recognition Using LLM

- Using our prompt with Gemini 2.5 Pro, we achieved 98% accuracy
- Compared with machine learning models, the LLM achieved higher recognition accuracy



Conclusion and Future Work

- With designed prompts, the LLM achieves an higher accuracy compared to the baseline machine learning model SVM, indicating the potential of activity recognition without training effort from users

- With further prompt fine-tuning, the users could realize recognition from more types of human activities

Thank You for Your Time

Any Questions?