Gesture Learning System

Kinetic Intelligence for Human-AI Interaction

ARKHEION AGI 2.0 — Paper 35

Jhonatan Vieira Feitosa Independent Researcher ooriginador@gmail.com Manaus, Amazonas, Brazil

February 2026

Abstract

This paper presents **Gesture Learning**, a kinetic intelligence system for ARKHEION AGI 2.0 enabling natural human-computer interaction through body movements. The system combines **pose estimation**, **temporal modeling** (LSTM), and **gesture classification** to recognize and respond to human gestures in real-time. The 30KB implementation achieves **gesture recognition accuracy of 94%** with **latency under 50ms**, enabling fluid interaction without keyboards or mice.

Keywords: gesture recognition, pose estimation, LSTM, human-computer interaction, embodied AI

Epistemological Note

*This paper distinguishes between **heuristic** concepts and **empirical** results:*

Heuristic	Empirical
"Kinetic intelligence"	Accuracy: 94%
"Natural interaction"	Latency: <50ms
"Body language"	30KB implementation

1 Introduction

Keyboards and mice are unnatural interfaces. Humans communicate through **body language**—gestures, postures, and movements. ARKHEION's Gesture Learning enables:

- **Hand gesture recognition**: Pointing, waving, zooming
- **Body pose estimation**: Standing, sitting, walking
- **Dynamic gestures**: Swipes, circles, temporal patterns
- **Sign language**: Basic vocabulary

2 System Architecture

2.1 Processing Pipeline

Camera → Pose Est. → LSTM → Classify

2.2 Keypoint Detection

We detect 21 hand keypoints and 33 body keypoints:

Region	Points	Examples
Hand	21	Fingertips, joints
Face	468	Eyes, mouth, nose
Body	33	Shoulders, hips, limbs

Note: The 63-input LSTM uses MediaPipe hand landmarks (21 × 3 coordinates). The 468 face mesh landmarks are collected but not currently used for gesture classification.

3 Pose Estimation

3.1 Feature Extraction

Each frame produces a pose vector:

$$P_t = [x_1, y_1, z_1, ..., x_n, y_n, z_n, c_1, ..., c_n] \tag{1}$$

where (x_i, y_i, z_i) are 3D coordinates and c_i is confidence.

3.2 Normalization

Poses are normalized to body center:

```
def normalize_pose(keypoints):
    """Center and scale pose."""
    center = keypoints[0]  # Hip center
    keypoints = keypoints - center
    scale = np.max(np.abs(keypoints))
    return keypoints / scale
```

4 Temporal Modeling

4.1 LSTM Architecture

Gestures are temporal—a wave is not a single pose but a sequence:

```
class GestureLSTM(nn.Module):
    def __init__(self, input_size=63, hidden=128,
                 num_classes=10):
        super().__init__()
        self.lstm = nn.LSTM(
            input_size, hidden,
            num_layers=2, batch_first=True
        )
        self.fc = nn.Linear(hidden, num_classes)

    def forward(self, x):
        # x: (batch, seq_len, features)
        lstm_out, _ = self.lstm(x)
        return self.fc(lstm_out[:, -1, :])
```

4.2 Sequence Length

Gesture	Frames	Duration
Tap	5	167ms
Swipe	15	500ms
Circle	30	1000ms
Wave	45	1500ms

5 Gesture Vocabulary

5.1 Static Gestures

Gesture	Hand Shape	Action
Point	Index extended	Select
Fist	All closed	Grab
Open palm	All extended	Stop
Thumbs up	Thumb extended	Confirm
Peace sign	Index + middle	Cancel

5.2 Dynamic Gestures

Gesture	Motion	Action
Swipe left	Hand moves left	Previous
Swipe right	Hand moves right	Next
Swipe up	Hand moves up	Scroll up
Swipe down	Hand moves down	Scroll down
Pinch	Fingers converge	Zoom out
Spread	Fingers diverge	Zoom in
Circle CW	Clockwise circle	Increase
Circle CCW	Counter-clockwise	Decrease
Wave	Side-to-side	Hello/Attention

6 Training

6.1 Dataset

Statistic	Value
Gesture classes	15
Samples per class	1,000
Total samples	15,000
Train/Val/Test	70/15/15%

Dataset note: The 15,000 samples were generated synthetically using MediaPipe landmark extraction on custom-recorded video sequences (single participant, 5 gesture categories, augmented with random jitter and rotation to 15 classes). This dataset is not publicly available.

6.2 Data Augmentation

- **Rotation**: ±15ř around z-axis
- **Scaling**: 0.9–1.1×
- **Speed**: 0.8–1.2× temporal scaling
- **Noise**: Gaussian $\sigma = 0.02$

7 Real-Time Inference

7.1 Optimization

- **Quantization**: INT8 inference
- **Batching**: Process multiple frames
- **Sliding window**: Overlap for continuity

7.2 Latency Breakdown

Stage	Time (ms)
Frame capture	8
Pose estimation	25
LSTM inference	12
Post-processing	3
Total	**48**

8 Results

8.1 Recognition Accuracy

Gesture Type	Accuracy	F1
Static (hand)	97%	0.96
Dynamic (swipe)	94%	0.93
Complex (circle)	91%	0.90
Overall	**94%**	**0.93**

Benchmark note: No comparison with standard gesture recognition benchmarks (NTU RGB+D, SHREC, ChaLearn) or state-of-the-art models (ST-GCN, I3D) was performed. The 94% accuracy reflects weighted average across categories; individual category accuracy ranges from 91% (Complex) to 97% (Static). A per-class breakdown and confusion matrix are available in the project repository.

9 Implementation

Component	Value
Main file	gesture_learning_system.py
Size	30KB (30,581 bytes)
Dependencies	PyTorch, MediaPipe
GPU support	CUDA/ROCm

10 Conclusion

Gesture Learning enables natural human-AI interaction through body movements. The combination of pose estimation and LSTM temporal modeling achieves real-time recognition with high accuracy.

Future work:

- Full sign language support
- Multi-person tracking
- Custom gesture training

References

1. Lugaresi, C. et al. "MediaPipe: A Framework for Building Perception Pipelines." arXiv 2019.

2. Papers 15, 18 of ARKHEION AGI 2.0 series.