Magic Wand using Arduino Nano 33 BLE Sense

Machine Learning

Adhiraj Walse
Sudeshna Nanda
Srikanth Nanda

29. Januar 2025

University of Applied Sciences
HOCHSCHULE EMDEN·LEER

Introduction

University of Applied Sciences
HOCHSCHULE EMDEN·LEER

Introduction

- **The Synergy of Machine Learning and IoT**
  - IoT has revolutionized industries like healthcare, agriculture, and smart cities Hamdan, Ayyash und Almajali, 2020.
  - Many IoT applications demand real-time data processing, unsuitable for traditional cloud computing Shi u. a., 2016.
- **Emergence of TinyML**
  - Enables low-latency, low-power, and efficient model inference on edge devices Sakr u. a., 2020.
  - Operates on compact devices (e.g., RTOS-based microcontrollers) with long battery life Anh und Tan, 2009-09 Abadade u. a., 2023.

University of Applied Sciences
HOCHSCHULE EMDEN·LEER

Introduction

- **Arduino and Its Evolution**
  - Initially a prototyping tool, now supports IoT, wearables, and embedded systems Kushner, 2011.
  - Arduino Nano 33 BLE Sense: A 3.3V board featuring embedded sensors like accelerometers for motion detection Arduino, 2021.
- **Objective of the Study**
  - Develop a **Magic Wand"** using Arduino Nano 33 BLE Sense.
  - Program the board to recognize predefined gestures (e.g., wing, ring, slope).

Project Overview

Hardware: Arduino Nano 33 BLE Sense

- **Embedded Sensors:** Three-axis accelerometer for precise motion detection **Arduino**; **2021**.
- **Compact and Energy-efficient Design:** Ideal for portable and edge-based applications.
- **Integration with ML Frameworks:** Supports TensorFlow Lite for deploying optimized models.

Project Objective

- Develop a **Magic Wand** for real-time gesture recognition.
- Recognize predefined gestures such as wing, ring, and slope.
- Integrate lightweight ML models with Arduino Nano 33 BLE Sense for efficient, low-latency inference.

Project Overview

Output

- **Screen Visualization:** Displays real-time gesture recognition results on a terminal.

Enhanced Execution and Future Enhancements

- **Gesture Precision:** Optimized ML models to minimize false readings and ensure accuracy.
- **Edge Intelligence:** Leverage TinyML for efficient and low-latency performance on edge devices.
- **Future Enhancements:**
  - Expand gesture recognition capabilities.
  - Further optimize performance and power efficiency.

Model Implementation and Futuristic Functions

TensorFlow Lite (TFLite)

- Framework for deploying machine learning models on resource-constrained devices.
- Facilitates model optimization using techniques which are quantization and pruning.
- Enables real-time gesture recognition through efficient edge inference.

TensorFlow Lite Micro (TFLite Micro)

- Tailored for microcontrollers, including Arduino Nano 33 BLE Sense.
- Executes quantized models within the device's memory and computational limitations.
- Seamless integration with existing Arduino libraries for gesture-based applications.

Model Training Process

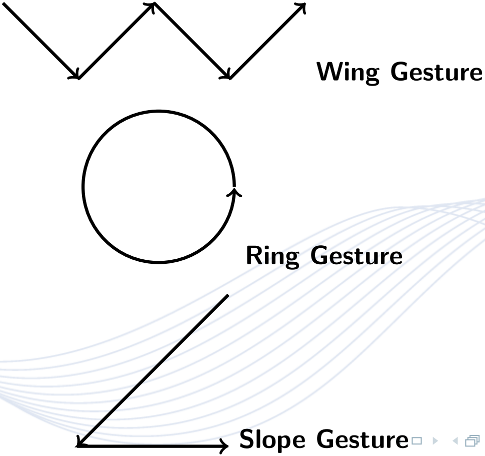
- **Data Collection:** Gather accelerometer data for gestures like "wing, ring,änd ßlope."
- **Preprocessing:** Normalize and augment data to enhance model performance.
- **Training and Optimization:**
  - Utilize CNNs for feature extraction and classification.
  - Apply quantization and pruning to reduce memory footprint.

Futuristic Enhancements

- Use Remotexy to operate the Magic Wand for its dynamic movements.
- Enchance the model by showing digital output on Arduino using Pico4ML and Pico2.
- Explore advanced optimization techniques, such as knowledge distillation, to enhance efficiency.

Gesture Types

Dataset Collection and labeling of data for three trained movements: wing (W), ring (O), and slope (/).The following figures show the direction of motion and shape for each gesture. [Warden und Situnayake, 2020]



University of Applied Sciences
HOCHSCHULE EMDEN·LEER

Challenges

Challenge 1

Resource Constraints

- The limited memory and computational power of the Arduino Nano 33 BLE Sense required optimization techniques to adapt machine learning models for deployment. [Arduino, 2021].
- High-performance deep learning models designed for GPUs or clusters are incompatible with microcontroller units (MCUs), demanding significant model simplifications.

University of Applied Sciences
HOCHSCHULE EMDEN·LEER

Challenges 2

Gesture Recognition Complexity

- Developing a robust ML model to recognize predefined gestures (wing, ring, slope).
- Accurately distinguishing untrained or ünknown" gestures.
- Avoiding false positives caused by overlapping or extended gestures, which could mislead the accelerometer readings.

Challenges 3

Model Optimization

- Ensuring real-time performance while keeping model size and computational complexity low – a critical balance for edge devices.
- Employing techniques like quantization and pruning to minimize resource usage without degrading accuracy.

University of Applied Sciences
HOCHSCHULE EMDEN·LEER

Challenges 4

Framework Adaptation

- Leveraging tools like TensorFlow Lite and TinyML frameworks, which are not fully optimized for intricate applications.
- Tailoring frameworks to meet project-specific requirements.

Solutions

Solution 1: Optimization Techniques for MCUs

- Use **model quantization** to convert 32-bit floating point models into 8-bit integer models[Arduino, 2021].
- Implement **pruning techniques** to remove insignificant model weights.
- Leverage **TinyML frameworks**, such as TensorFlow Lite for Microcontrollers.

Solution 2: Enhanced Data Collection and Model Training

- Gather a diverse and high-quality dataset covering predefined and random gestures[**shi:2016**].
- Incorporate **data augmentation** to simulate real-world noise and variations.
- Use **1-D Convolutional Neural Networks (CNNs)** for effective feature extraction.

Solution 3: Balancing Accuracy and Efficiency

- Optimize the neural network architecture by reducing layers or parameters[**shi:2016**].
- Apply **transfer learning** to adapt pre-trained models and reduce training time.
- Utilize profiling tools for performance benchmarking and fine-tuning.

Solution 4: Customizing Frameworks

- Extend TensorFlow Lite for Microcontrollers with project-specific optimizations[Warden und Situnayake, 2020].
- Use **hardware-specific libraries** like the Arduino TensorFlow Lite library.
- Implement modular testing for incremental verification.

Display Results

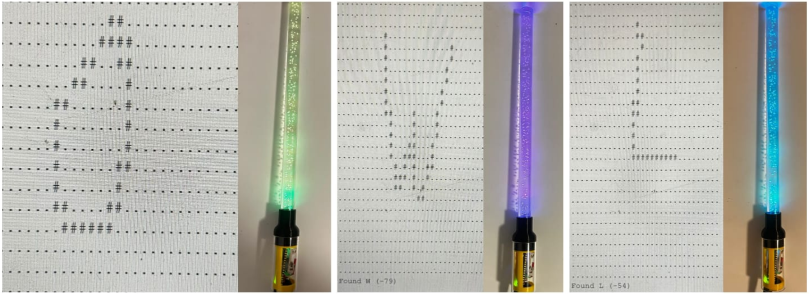


Abbildung: Magic Wand Results

University of Applied Sciences
HOCHSCHULE EMDEN·LEER

Results

Technical Achievements

- Deployed a gesture-recognition system on the Arduino Nano 33 BLE Sense[Arduino, 2015].
- Used Convolutional Neural Networks (CNNs) for feature extraction and classification.
- Optimized model size and computational efficiency through quantization and pruning.
- Achieved real-time gesture recognition on edge devices.

Future Improvements

- **Model Optimization:** Explore advanced pruning and quantization techniques[Shi u. a., 2016].
- **System Robustness:** Enhance error handling for sensor data inconsistencies.
- **Scalability:** Recognize more gestures and integrate with additional applications.
- **User Experience:** Refine gesture output and response mechanisms.

References

University of Applied Sciences
HOCHSCHULE EMDEN·LEER

References I

Abadade, Youssef u. a. (2023). “A Comprehensive Survey on TinyML”. In: *IEEE Access* 11, S. 96892–96922. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2023.3294111. URL: https://ieeexplore.ieee.org/document/10177729/ (besucht am 02. 11. 2023).

Anh, Tran Nguyen Bao und Su-Lim Tan (2009-09). “Real-Time Operating Systems for Small Microcontrollers”. In: *IEEE Micro* 29.5, S. 30–45. DOI: 10.1109/MM.2009.86. URL: https://ieeexplore.ieee.org/document/5325154.

Arduino, ed. (2021). *Arduino Nano 33 BLE Sense Arduino Documentation*. The Arduino Nano 33 BLE Sense combines a tiny form factor, different environment sensors and the possibility to run AI using TinyML and TensorFlow™ Lite. URL: https://docs.arduino.cc/hardware/nano-33-ble-sense (besucht am 19. 10. 2023).

University of Applied Sciences
HOCHSCHULE EMDEN·LEER

References II

- Arduino, Store Arduino (2015). “Arduino”. In: *Arduino LLC* 372.
- Hamdan, Salam, Moussa Ayyash und Sufyan Almajali (2020). “Edge-Computing Architectures for Internet of Things Applications: A Survey”. In: *Sensors* 20.22, S. 6441. DOI: 10.3390/s20226441.
- Kushner, David (2011). *The Making of Arduino - IEEE Spectrum. How five friends engineered a small circuit board that's taking the DIY world by storm.* URL: https://spectrum.ieee.org/the-making-of-arduino.
- Sakr, Fouad u. a. (2020). “Machine Learning on Mainstream Microcontrollers”. In: *Sensors* 20.9, S. 2638. DOI: 10.3390/s20092638.
- Shi, Weisong u. a. (2016). “Edge Computing: Vision and Challenges”. In: *IEEE Internet of Things Journal* 3.5, S. 637–646. DOI: 10.1109/JIOT.2016.2579198.

References III

Warden, Pete und Daniel Situnayake (2020). *TinyML: Machine Learning with TensorFlow Lite on Arduino and Ultra-low-power Microcontrollers*.

Thanks a lot

for your attention