UNIVERSITÀ
DI TRENTO

Department of Information Engineering and Computer Science

Bachelor's Degree in
Computer, Communication and Electronic Engineering

Final Dissertation

TinyML-Based Voice Recognition on Syntiant NDP101

From Keyword Spotting to Speaker Verification

Supervisor
Kasim Sinan Yildirim

Student
Matteo Gottardelli

Academic year 2024/2025

Contents

Abstract

List of reference explanation:

- Edge Impulse:
  1. Edge Impulse Explaining how to manage Syntiant TinyML Deployment: [5]
  2. Keyword Spotting on Syntiant Stop/Go Example by Edge Impulse: [6]
  3. Dataset Edge Impulse Background Noise or General Words from Google Dataset: [7]
  4. Firmware Syntiant TinyML Github Repository: [9]
  5. Processing Blocks Edge Impulse Code Processing (No Deployment Model Conversion code): [8]
- Tools:
  1. Node Js Repository Download Source: [1]
- Documentation:
  1. ESP32 Documentation: [3]
  2. Syntiant Tutorial Edge Impulse: [16]
- Syntiant Specifics:
  1. TinyML 2021 Summit Presentation Syntiant NDP120 -¿ Model Processing: [18]
  2. Data of Syntiant NDP101: [12]
  3. Code Experimental Implementation on NDP101: [11]
  4. The Intelligence of Things enabled by Syntiant's TinyML board analyzing performances: [2]
  5. NDP101 General Usage: [15]
  6. Description Syntiant Audio Block Processing: [4]
  7. Hardware NDP101 Properties: [15]
- Theory:
  1. PDM Microphone Module Explanation: [10]
  2. MFE Block Process Explained: [17]
  3. TinyML Neural Network Training: [13]
- SV Model:
  1. Deep Neural Network Model for Speaker Identification: [19]
  2. D-vector Extractor implementing TinySV: [14]

1 Introduction to Syntiant NDP101

1.1 NDP101 Definition and TinyML Concept

• NDP101 what it performs?
• What do make NDP101 different from the other voice IoT devices?
• Evolution during time?
• TinyML concept?

1.2 Project Objectives

• KWS model Deploying (Edge Impulse)
• Extend functionality to SV to be compatible with device (Python)
• Optimize neural network to be deployed on the device and verifying on simulation (Syntiant NDP101 and C-code)

1.3 Constraints of NDP101 device

• 4-bit weight constraints (589k parameter limit 294,5 kB)
• SDK restrictions (NDA)
• Multi-model deployment

2 Background & Related Work

2.1 NDP101 Architecture

core, memory, power, neural network structure (dnn constraints), connectors

2.2 Audio Processing

mel filterbank (pre-emphasis, framing, fft, mel scaling), mfe vs mfcc

2.3 Keyword Spotting Approaches

Classification, softmax

2.4 Speaker Verification Approaches

text-dependent vs independent sv, general methods not suitable for tinyml (gmm-ubm, i-vectors), d-vector concept, tinysv approach

3 Methodology

3.1 Hardware Pipeline

• Structure of the system (1 ESP32 + 2 Syntiant NDP101)
-¿ SPI communication NDP101 + Arduino MKRZero
-¿ SPI communication between ESP32 + NDP101

3.2 Software Pipeline

3.2.1 Signal Capture

Simulation (PortAudio), Real (Direct PDM sampling on NDP101) [Output: 15488]

3.2.2 MFE Block Generation

Simulation (Block explaining, validation with edge impulse generated spectrogram), real (Feature Extractor) [Output: 1600]

3.2.3 KWS Model

Input (1600) → FC256 (ReLU) → FC256 (ReLU) → FC256 (ReLU) → Output (4, Softmax)
Simulation (Weight extraction, target word, dataset), Real (model uploading and classification method)

3.2.4 SV Model

Input (40x40) → Conv+BatchNorm layers → 256-dimensional d-vector
Simulation (model-size, classification purpose, truncation explaination, verification methods (best-matching and mean-cosine)), Real (custom logic for verification)

Enrollment phase

Only for real logic, in simulation is not performed (various iteration of the process) [how to verify this? Model SV impossibility of uploading because of int-4 impossible deployment]

4 Implementation

4.1 KWS Model

• Edge Impulse Integration - Model training and validation
• C implementation, direct MFE computation on device

4.2 SV Model

• D-vector extraction - C port of Python CNN model
• Database design for sample saving
• Cosine similarity EER

4.2.1 Quantization of SV Model

• Float32 → Int8: Performance metrics showing minimal accuracy loss
• Int8 → Int4: Challenges in maintaining model quality
• Performance Comparison:
- Original CNN model vs. distilled DNN
- Full-precision vs. quantized implementations

5 Results

5.1 KWS Performance

Analysis using simulation data (confusion matrix)

5.2 SV Performance

Comparison between float32, int8 and int4 (int4 impossible to obtain good results using basic PQT, difficulties in deploying good QAT training, not true support by tensorflow and the fake quantization technique did not work)

5.3 System Integration

- Overall Power Consumption (ESP32 + 2 Syntiant NDP101)
- Memory Usage

6 Discussion

6.1 Technical Limitations

- SDK Barriers (NDA limitations)
- Hardware Constraints (DNN-only architecture limiting SV capabilities)
- Quantization Challenges (Maintaining accuracy at 4-bit precision)

6.2 Design Trade-offs

- Accuracy vs Power (Best-matching vs mean-cosine approaches)
- Security vs User experience (Enrollment samples)
- Model Size vs Complexity (Feature extraction capabilities)

6.3 Comparison with State-of-Art

- TinySV Addition

7 Conclusion & Future Work

7.1 Key Contributions

• MFE/DNN Code - Full C implementation for Simulation in reproducing the behavior (the model is not quantized, so the behavior is ideal)
• Speaker Verification Distillated
• Quantization - Systematic approach to 8-bit quantization, not 4-bit (too experimental)

7.2 Future Directions

• If able to access to Syntiant, can deploy a compatible binary SV model
• Integration with existing system, that according to my system response will act as consequence
• Adaptive Training (The code is ready, but can't be tested properly)

Bibliography

[1] Node js repository. deb.nodesource.com. Last Access 06/05/2025.

[2] Will McDonald Atul Gupta Mallik Moturi Alireza Yousefi, Luiz Franca-Neto and David Garrett. The intelligence of things enabled by syntiant's tinyml board. Last Access 06/05/2025.

[3] Espressif. Esp32 documentation. https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_ 2025. Last Access 06/05/2025.

[4] Edge Impulse. Audio syntiant. https://docs.edgeimpulse.com/docs/edge-impulse-studio/processing-blocks/audio-syntiant. Last Access 06/05/2025.

[5] Edge Impulse. Edge impulse syntiant tinyml deploying. https://docs.edgeimpulse.com/docs/edge-ai-hardware/mcu-+-ai-accelerators/syntiant-tinyml-board. Last Access 06/05/2025.

[6] Edge Impulse. Keyword spotting on syntiant stop/go example by edge impulse. Last Access 06/05/2025.

[7] Edge Impulse. Audio classification - keyword spotting. https://studio.edgeimpulse.com/public/499022/latest, 2024. Apache 2.0, Last Access 06/05/2025.

[8] Edge Impulse. Edge impulse block processing repository. https://github.com/edgeimpulse/processing-blocks, 2025. Last Access 06/05/2025.

[9] Edge Impulse. Firmware syntiant tinyml github repository. https://github.com/edgeimpulse/firmware-syntiant-tinyml/tree/master, 2025. Last Access 06/05/2025.

[10] Thomas Kite. Understanding pdm digital audio. https://users.ece.utexas.edu/ bevans/courses/rtdsp/lectures/10_Data_Conversion/AP_Understanding_PDM_Digital_Audio.pdf, 2012. Last Access 06/05/2025.

[11] Christian Neuhaus. Code experimental implementation on ndp101. https://github.com/happychriss/Goodwatch_NDP101_SpeeechRecognition/tree/master/include, 2021. Last Access 06/05/2025.

[12] Christian Neuhaus. Ndp101 experimental implementation on ndp101. https://44-2.de/syntiant-ndp-101-always-on-low-power-speech-recognition/, 2022. No other resources that corroborates data listed, Last Access 06/05/2025.

[13] Eugeniu Ostrovan. Tinyml on-device neural network training. https://www.politesi.polimi.it/retrieve/e9f5774c-592d-4741-a6d4-82f5e501630e/TinyML_on_device_neural_network_trainingLast Access 06/05/2025.

[14] Massimo Pavan, Gioele Mombelli, Francesco Sinacori, and Manuel Roveri. Tinysv: Speaker verification in tinyml with on-device learning. 2025. Last Access 06/05/2025.

[15] Syntiant. General description of ndp101 device. https://www.syntiant.com/ndp101#data_sheet. Last Access 06/05/2025.

[16] Syntiant. Syntiant tutorial edge impulse. https://www.eetree.cn/wiki/_media/syntiant_tinyml_tutorial_mac 2021. Last Access 06/05/2025.

[17] Rishabh Tak, Dharmesh Agrawal, and Hemant Patil. *Novel Phase Encoded Mel Filterbank Energies for Environmental Sound Classification*. 11 2017.

[18] TinyML. Tinyml 2021 summit presentation syntiant ndp120. https://cms.tinyml.org/wp-content/uploads/summit2021/tinyMLSummit2021d1_Awards_Syntiant.pdf, 2021. Last Access 06/05/2025.

[19] Feng Ye and Jun Yang. A deep neural network model for speaker identification. *Applied Sciences*, 11(8), 2021. Last Access 06/05/2025.

Acknowledgements

*