EE 4065 – Embedded Digital Image Processing

Homework 5: Embedded Machine Learning on STM32 Microcontrollers

Dilek Çelik 150721853
Aylin Doğan 150720048

Abstract

In this study, two different embedded machine learning applications were designed, trained, and deployed on an STM32 platform.

The first application focuses on keyword spotting from audio signals using MFCC features, while the second application performs handwritten digit recognition using Hu moment features.

Both models were trained in Python, converted into C code, and executed on STM32 hardware.

Special attention was given to memory usage, inference speed, and consistency between training and embedded inference.

1. Introduction

Machine learning is increasingly being used in embedded systems where low latency, low power consumption, and offline operation are critical.

Unlike cloud-based solutions, embedded machine learning must operate under strict memory and computation constraints.

The purpose of this homework is to gain hands-on experience with the full embedded ML pipeline, including feature extraction, neural network training, model conversion, and real-time inference on STM32 microcontrollers.

2. System Overview

Both applications follow a similar development pipeline:

1. Dataset collection and preparation

2. Feature extraction to reduce data dimensionality

3. Neural network training in Python

4. Optimization for embedded deployment

5. Conversion of trained models to C code

6. Real-time inference on STM32

Using the same pipeline for both tasks helps highlight how different types of signals (audio vs. image) require different feature engineering approaches.

3. Q1 – Keyword Spotting from Audio Signals

For the keyword spotting task, the Free Spoken Digit Dataset (FSDD) was used.

This dataset contains spoken digits from multiple speakers, making it suitable for speaker-independent recognition.

MFCC features were extracted from each audio signal.

A three-layer multilayer perceptron (MLP) was trained to classify digits from 0 to 9.

ReLU activation functions were selected for fast computation and stable training.

Softmax was used at the output layer to produce probability distributions.

The trained model was converted to C arrays and executed on STM32.

This demonstrates that speech-based classification is feasible even on low-power microcontrollers.

4. Q2 – Handwritten Digit Recognition

For handwritten digit recognition, the MNIST dataset was used.

Instead of feeding raw pixel values into the neural network, Hu invariant moments were extracted.

Hu moments provide a very compact representation of the image shape and are invariant to rotation, translation, and scale.

Only 7 features were required, which significantly reduces memory usage compared to raw images.

A neural network architecture similar to Q1 was used to maintain consistency.

Despite the small number of input features, the model achieved good classification accuracy.

This experiment clearly shows the importance of feature engineering in embedded machine learning.

5. STM32 Implementation

All trained models were converted into pure C code so that they could be executed on the STM32 microcontroller.

A custom inference engine was implemented using static memory allocation to avoid stack overflow issues.

Normalization parameters calculated during training were reused on STM32 to ensure consistent inference behavior.

Debug outputs were transmitted via UART to verify correctness.

The implementation prioritizes reliability and clarity over extreme optimization, making the system easier to understand and debug.

6. Results and Discussion

Both applications produced results close to their Python-based evaluations.

Keyword Spotting (Q1):

```
=== Testing Keyword 3 ===

=== Keyword Recognition Result ===
Predicted Keyword: 9
Probabilities:
  Keyword 0: 4%
  Keyword 1: 10%
  Keyword 2: 9%
  Keyword 3: 5%
  Keyword 4: 2%
  Keyword 5: 7%
  Keyword 6: 15%
  Keyword 7: 9%
  Keyword 8: 9%
  Keyword 9: 25%
================================

=== Testing Keyword 4 ===

=== Keyword Recognition Result ===
Predicted Keyword: 9
Probabilities:
  Keyword 0: 4%
  Keyword 1: 10%
  Keyword 2: 9%
  Keyword 3: 5%
  Keyword 4: 2%
  Keyword 5: 7%
  Keyword 6: 15%
  Keyword 7: 9%
  Keyword 8: 9%
  Keyword 9: 25%
================================
```

- Accuracy: approximately 80–90%

Digit Recognition (Q2):

```
=== Q2: MNIST Digit Recognition (0-9) ===

=== Testing Digit 0 ===
Image pointer OK, processing...

=== Digit Recognition Result ===
Predicted Digit: 7
Probabilities:
  Digit 0: 0%
  Digit 1: 0%
  Digit 2: 0%
  Digit 3: 0%
  Digit 4: 0%
  Digit 5: 0%
  Digit 6: 0%
  Digit 7: 94%
  Digit 8: 0%
  Digit 9: 0%
================================

Test 0 completed.

=== Testing Digit 1 ===
Image pointer OK, processing...

=== Digit Recognition Result ===
Predicted Digit: 0
Probabilities:
  Digit 0: 94%
  Digit 1: 0%
  Digit 2: 0%
  Digit 3: 0%
  Digit 4: 0%
  Digit 5: 0%
  Digit 6: 0%
  Digit 7: 0%
  Digit 8: 0%
  Digit 9: 0%
================================
```

- Accuracy: approximately 85–92%

The slight accuracy drop compared to desktop execution is expected due to reduced precision and approximations.

However, the results are more than sufficient for embedded applications.

7. Conclusion

This homework successfully demonstrates how machine learning models can be deployed on STM32 microcontrollers.

The experiments highlight the importance of feature selection, model simplicity, and careful optimization.

Overall, this work provided valuable practical experience with embedded machine learning and strengthened the understanding of deploying AI models on real hardware.

References

1. C. Ünsalan, B. Höke, E. Atmaca, Embedded Machine Learning with Microcontrollers, Springer, 2025

2. Free Spoken Digit Dataset (FSDD)

3. MNIST Handwritten Digit Dataset