УДК 621.382

DOI: 10.20535/2523-4455.mea.265406

Аналіз особливостей використання ресурсів мікроконтролера для розпізнавання мовлення

Рижова[f] А. Р., ORCID 0000-0003-3278-8448
Оникієнко[s] Ю. О., к.т.н. доц., ORCID 0000-0001-7508-8391,
Кафедра акустичних та мультимедійних електронних систем ames.kpi.ua
Національний технічний університет України
"Київський політехнічний інститут імені Ігоря Сікорського" ROR 00syn5v21
Київ, Україна

***Анотація*—В роботі виконано аналіз використання обчислювальних ресурсів мікроконтролера для машинного навчання та розпізнавання голосу. Поставлено експеримент для визначення залежності часу розпізнавання ключового слова, об'єму використаної оперативної пам'яті та пам'яті програм в залежності від кількості мел-частотних кепстральних коефіцієнтів та типу згорткової нейронної мережі. Для проведення експерименту використано плату розробки Arduino Nano 33 BLE Sense. Модель нейронної мережі створено та треновано на програмній платформі Edge Impulse. В результаті аналізу встановлено, що пам'яті 32-х бітного мікроконтролера достатньо для обчислень та використання нейронної мережі. Однак час класифікації ключового слова складає приблизно 0,3 с, відповідно розпізнавання довгих фраз може зайняти декілька секунд, що не завжди є прийнятним.**

Ключові слова — мікроконтролери; мел-частотні кепстральні коефіцієнти; згорткові нейронні мережі; розпізнавання голосу.

I. Вступ

Розпізнавання голосу та мови з використанням вбудованих та мобільних систем дозволяє суттєво розширити область використання нейронних мереж та машинного навчання. В останні роки отримали розвиток системи розпізнавання на мікроконтролерах, що дозволило значно розширити функціональні можливості мікроконтролерних систем.

Процес розпізнавання будується на основі аналізу енергетичного спектру звукового сигналу і може бути виконано різними методами аналізу спектру сигналу в залежності від подальшої обробки голосової інформації. В системах надійного визначення мовця на основі сталої фрази найбільш популярним є використання мел-частотних кепстральних коефіцієнтів — MFCC (Mel Frequency Cepstral Coefficients) та лінійних прогностичних кепстральних коефіцієнтів — LPCC (Linear Predictive Cepstral Coefficients) коефіцієнтів. MFCC — це коефіцієнти, обчислені на нелінійній частотній шкалі на основі відомого слухового сприйняття людини, тоді як LPCC — це коефіцієнти, які представляють людську слухову систему на основі лінійного передбачення. LPCC у порівнянні з MFCC забезпечує дещо більшу точність при автентифікації мовця. Однак MFCC потребує значно менше часу для прийняття рішення [1], що є дуже важливою властивістю саме для вбудованих систем з обмеженими обчислювальними ресурсами. Також метод MFCC достатньо просто імплементується у вигляді програмного коду і постійно удосконалюється [2, 3].

Для класифікації голосу або мови у вбудованих системах використовують гаусівську модель суміші (Gauss Mixture Model) [4], приховану модель Маркова (Hidden Markov Model) [5], штучні нейронні мережі (Artificial Neural Network) [6] та інші. Нейронні мережі, завдяки гнучкості архітектури та простоті програмної реалізації, дуже популярні саме у вбудованих системах, а тому постійно удосконалюються [7].

Вбудовані системи для розпізнавання мови умовно можна розділити на три групи: системи з використанням FPGA (Field-Programmable Gate Array), системи на основі процесора або мікропроцесора і системи на мікроконтролерах. FPGA системи мають високу швидкість обробки інформації, але обмежені в зміні налаштувань [8]. Мікропроцесорні системи мають великий об'єм пам'яті і, як правило, операційну систему. Реалізовують такі системи розпізнавання як на потужних і дорогих платах типу Nvidia Jetson Tegra K1 [9], так і на більш дешевих платах Raspberry PI [10, 11] Відповідно максимальну швидкість розпізнавання забезпечують спеціалізовані плати вартістю від декількох сотень, або навіть тисяч доларів. Мікрокомп'ютери Raspberry PI з власною операційною системою дешевше, але з хорошим функціоналом також коштують до декількох сотень доларів.

Особливий інтерес для використання в IT-індустрії та системах Інтернету речей представляють вбудовані системи розпізнавання мови на мікроконтролерах, як правило 32-х бітних. Головними перевагами таких систем є розвинена периферія та невелика вартість. Недоліками є обмежений об'єм пам'яті та

DOI: 10.20535/2523-4455.mea.265406

нижчі робочі частоти ядра мікроконтролера, що зменшує швидкість обробки даних у порівнянні з системами на мікропроцесорах. Але для застосувань, де використання розпізнавання мови є однією з функцій системи, використання мікроконтролерів є цілком прийнятне.

В роботі [12] запропоновано систему для голосового виклику медичного працівника побудовану на мікроконтролері STM32. Рішення в системі приймається на порівнянні значень MFC коефіцієнтів, отриманих з голосового сигналу, зі значеннями, які зберігаються в пам'яті мікроконтролера. Розпізнавання голосу реалізовано у складі домофонної системи на 32-х бітному мікроконтролері, де верифікація голосу виконується за допомогою гаусівської моделі суміші [13]. В системі для розпізнавання голосу, побудованій на мікроконтролері STM32, класифікація голосу, виконується з використанням DTW (Dynamic Time Warping) після отримання значень LPCC [14].

В останні роки зростає популярність систем розпізнавання на мікроконтролерах, в яких для класифікації використовуються нейронні мережі, зокрема згорткові (Convolutional Neural Network). Це в першу чергу пояснюється наявністю безкоштовних програмних платформ (фреймворків) з широким функціоналом для створення та навчання нейронних мереж. Прикладами таких платформ є Tensorflow [15] та Keras [16], які використовуються не тільки для розпізнавання мови, а й зображень та інформації з сенсорів [17, 18].

Таким чином, використання нейронних мереж для розпізнавання інформації і, зокрема, голосу розширює функціональні можливості вбудованих систем на мікроконтролерах при необхідності врахування обмеженості їх ресурсів, що використовуються. Метою даної роботи є аналіз впливу параметрів системи обробки голосового сигналу та архітектури нейронної мережі на об'єм використаних ресурсів мікроконтролера. Для здійснення аналізу необхідно:

- Створити базу даних з зразків ключового слова, зразків інших слів та голосів та зразків шуму. Оцінити імовірність розпізнавання ключового слова серед інших слів та шумів;
- Встановити залежності об'єму задіяної пам'яті мікроконтролера та часу прийняття рішення від кількості MFC коефіцієнтів;
- Встановити залежності об'єму задіяної пам'яті мікроконтролера та часу прийняття рішення від типу згорткової нейронної мережі.

Далі розглянуто передумови проведення експерименту та власне його результати.

II. Теоретичні засади роботи

Коефіцієнти мел-частотного кепстру (MFCC) широко використовуються в автоматичному розпізнаванні мови та мовця. MFCC — це представлення, визначене як реальний кепстр віконного короткочасного сигналу, отриманого з швидкого перетворення Фур'є цього сигналу. Відмінність від справжнього кепстру полягає в тому, що використовується нелінійна частотна шкала, яка апроксимує властивості слухової системи, що підвищує якість розпізнавання мови. Мел шкала пов'язує відчуту частоту або висоту чистого тону з його фактично виміряною частотою. Люди набагато краще розрізняють невеликі зміни висоти на низьких частотах, ніж на високих. Завдяки використанню цієї шкали Мел функції точніше відповідають тому, що чують люди. Мел шкала в залежності від частоти f має наступний вигляд [19]:

$$B(f) = 1125\ln(1 + f/700) \tag{1}$$

Зворотне перетворення від Мел шкали b до частоти має наступний вигляд:

$$B^{-1}(b) = 700(e^{b/1125} - 1) \tag{2}$$

Для дискретного перетворення Фур'є з N відліків

$$X_a[k] = \sum_{n=0}^{N-1} x[n] e^{-j2\pi nk/N}, \quad 0 \le k < N \tag{3}$$

можна визначити масив із M фільтрів (m = 1,2, , M), де фільтр m є фільтром трикутної форми, заданим за формулою:

$$H_m^{`}[k] = \begin{cases} 0, & k < f[m-1]; \\ \dfrac{(k - f[m-1])}{(f[m] - f[m-1])}, & f[m-1] \le k \le f[m]; \\ \dfrac{(f[m+1] - k)}{(f[m+1] - f[m])}, & f[m] \le k \le f[m+1]; \\ 0, & k > f[m+1]. \end{cases} \tag{4}$$

При застосуванні таких фільтрів обчислюється середній спектр навколо кожної центральної частоти зі збільшенням смуги пропускання (рис.1).

Граничні точки $f[m]$ рівномірно розташовані на Мел шкалі:

$$f[m] = \left(\frac{N}{F_s}\right) B^{-1}(B(f_l) + m\frac{B(f_h) - B(f_l)}{M+1}) \tag{5}$$

де f_l та f_h– можна визначити як найнижчу та найвищу частоти групи фільтрів у герцах, F_s – частота дискретизації в герцах. Логарифм енергії на виході кожного фільтра можна обчислити як:

$$S[m] = \ln\left[\sum_{k=0}^{N-1} |X_a[k]|^2 H_m[k]\right], \ 0 \le m < M \tag{6}$$

Тоді мел частотний кепстр є дискретним косинусним перетворенням M виходів фільтрів:

DOI: 10.20535/2523-4455.mea.265406

$$c[n] = \sum_{m=0}^{M-1} S[m]\cos(\pi n(m+1/2)/M)\,,\ 0 \le n < M \quad (7)$$

де M змінюється для різних реалізацій від 24 до 40. Для розпізнавання мови зазвичай використовуються тільки перші 13 коефіцієнтів кепстру [19].

Таким чином, в результаті спектральної обробки звукового зразка тривалістю 1 с отримано його спектрограму, яка представляє собою матрицю розміром $n \times m$, де n – число MFC коефіцієнтів, m – кількість часових фреймів. Як правило, довжину фрейму вибирають рівною 20 мс. Зображення спектрограми ключового слова, яка містить 13 MFC коефіцієнтів довжиною 50 часових фреймів наведено на рис. 2.

Для класифікації зразків мови в роботі використано згорткову нейронну мережу. Типова згорткова мережа, наприклад LeNet-5, складається з трьох типів шарів, а саме: згорткового, об'єднаного і повнозв'язаного. Згортковий шар націлений на вивчення представлень характеристик вхідних даних. Як показано на рисунку 3, шар згортки складається з кількох ядер згортки, які використовуються для обчислення різних карт особливостей. Зокрема, кожен нейрон карти ознак з'єднаний із областю сусідніх нейронів у попередньому шарі. Таке сусідство називається рецептивним полем нейрона в попередньому шарі. Нову карту особливостей можна отримати, спершу згорнувши вхідні дані за допомогою навченого ядра, а потім застосувавши поелементну нелінійну функцію активації до згорнутих результатів. Повні карти функцій отримують за допомогою кількох різних ядер. Математично значення параметра в місці (i, j) на k-й карті об'єктів l-го шару, $z^l_{i,j,k}$, обчислюється за формулою:

$$z^l_{i,j,k} = \left(w^l_k\right)^T x^l_{i,j} + b^l_k \quad (8)$$

де w^l_k і b^l_k є ваговим вектором і зміщенням k-го фільтра l-го шару відповідно, а $x^l_{i,j}$ є вхідним фрагментом з центром у розміщені (i, j) l-го шару. Важливо, що ядро w^l_k, яке генерує карту функцій $z^l_{i,j,k}$, є спільним. Такий механізм розподілу ваги має кілька переваг, зокрема він зменшує складність моделі та полегшує навчання нейронної мережі [20].

Одновимірні згорткові нейронні мережі (Convolutional Neural Network, CNN) широко використовуються у додатках обробки різноманітних сигналів та мають ряд суттєвих переваг над звичайними CNN глибокого навчання. Перш за все, компактні 1D CNN можна ефективно навчити з обмеженим набором даних 1D-сигналів, тоді як CNN глибокого навчання зазвичай потребують масивів даних великого розміру. 1D CNN можна безпосередньо застосовувати до необробленого сигналу (наприклад, струму, напруги, вібрації тощо), не вимагаючи будь-яких попередніх чи пост-обробок, таких як, виділення ознак, зменшення розмірності, зменшення шумів тощо. Крім того, завдяки простоті та компактності конфігурації таких адаптивних одновимірних CNN, які виконують лише лінійні одновимірні згортки (скалярні множення та додавання), можлива недорога апаратна реалізація в реальному часі [21]. На рис. 4 показаний приклад простої одновимірної архітектури CNN.

Кожен із етапів згортки на рис.3 показує набір згорткових фільтрів, які можна навчати, за якими слідує операція узагальнення ознак, виділених згортковими фільтрами (Pooling). Фільтри призначені для виділення функцій високого рівня (наприклад, таких як краї та криві на зображенні) із наданого вхідного сигналу шляхом згортання набору вагових коефіцієнтів із вхідними даними та застосування нелінійної функції активації. Вихідні дані потім подаються в операцію об'єднання, яка зменшує просторовий розмір функцій, виділених згортковими фільтрами, одночасно підкреслюючи домінуючі особливості, отримані кожним фільтром. У міру проходження вхідних даних через етапи згортки (зліва направо на рис. 3) мережа вивчає більше специфічних для даного сигналу особливостей [22].

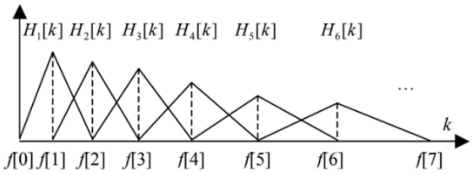

Рис. 1 Трикутні фільтри, які використовуються для обчислення мель-кепстра за допомогою рівняння (4)

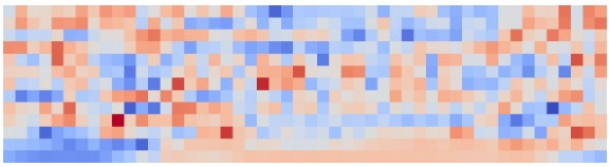

Рис. 2 Спектрограма ключового слова

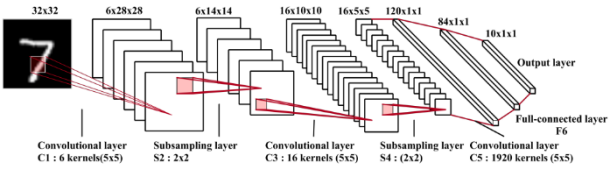

Рис. 3 Архітектура згорткової мережі на прикладі LeNet-5 [20].

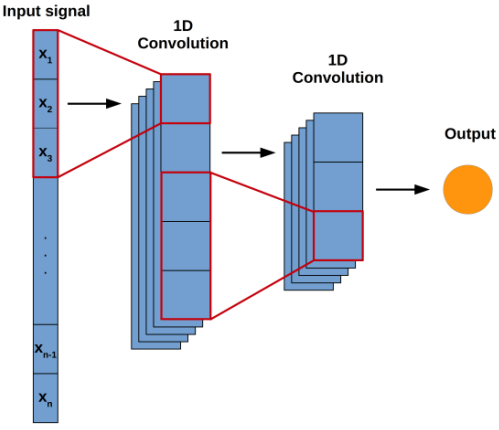

Рис. 4 Архітектура простої 1D згорткової нейромережі [22]

DOI: 10.20535/2523-4455.mea.265406

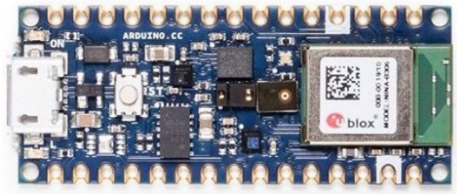

Рис. 5 Плата Arduino Nano 33 BLE Sense

III. Опис експерименту

Експериментальні дослідження виконувались у наступному порядку. Спочатку на комп'ютері створено модель нейронної мережі, виконано її тренування та завантажено в пам'ять мікроконтролера. Далі перевірено роботу моделі при різних значеннях її параметрів. Для створення моделі використано програмну платформу Edge Impulse [23], яка забезпечує машинне навчання на вбудованих пристроях для сенсорів, аудіо та комп'ютерного бачення з можливістю масштабування моделі нейронної мережі під вибране апаратне забезпечення. Такий підхід дає змогу виконувати оптимізоване машинне навчання вбудованих систем, починаючи від мікроконтролерів і закінчуючи центральними процесорами та спеціальними прискорювачами штучного інтелекту. Робота Edge Impulse грунтується на фреймворку Keras, який є набором функцій глибокого навчання для створення застосунків, написаних на мові програмування Python, для взаємодії з платформою машинного навчання TensorFlow. Edge Impulse дає можливість використовувати достатньо велику кількість налаштувань нейронної мережі, а також забезпечує індикацію використання ресурсів процесора та орієнтовний час виконання задачі. Достатньо тільки вибрати необхідний тип процесора. Також на платформі Edge Impulse можна виконувати попередню обробку аудіоданих. В результаті можна завантажити код програми з тренованою моделлю мережі до обраного мікроконтролера.

Після створення та навчання моделі нейронної мережі згенеровано програмний код під вибрану апаратну платформу. Для проведення експерименту вибрано плату розробки Arduino Nano 33 BLE Sense [24]. Плата оснащена 32-розрядним ARM® Cortex™-M4 мікроконтролером nRF52840, що працює на частоті 64 МГц. Плата має декілька датчиків, а саме: 9-осьовий інерційний вимірювальний пристрій, датчики температури, тиску, вологості, світла, кольору та жестів. Для введення аудіо сигналів доступний мікрофон. Основні електричні характеристики плати наступні:

- Частота тактового генератора – 64МГц
- Об'єм пам'яті програм (FLASH) – 1МБ
- Об'єм оперативної пам'яті (SRAM) – 256КБ
- АЦП – 12 біт, 200 000 значень в секунду

Зовнішній вигляд плати Arduino Nano 33 BLE Sense наведено на рис.5.

Для проведення експерименту створено три групи даних з назвами "hello", "unknown", "noise" і ключовим словом «hello». Група "hello" містить 94 зразки слова «привіт» англійською мовою, вимовлених жіночим голосом. Група "unknown" містить 167 зразків інших слів, вимовлених як жіночим, так і чоловічим голосами. Група " noise" містить 166 зразків шумів та випадкових звуків. Згідно з рекомендації Edge Impulse 80% зразків з кожної з груп даних було використано для тренування моделі нейронної мережі, відповідно 20% зразків для перевірки. В результаті платформою Edge Impulse згенеровано код програми, який містить початкові налаштування, алгоритми обробки звуку, функції обчислення MFCC та модель 1D згорткової нейронної мережі. Перед завантаженням в мікроконтролер код доопрацьовано для підвищення стабільності процесу розпізнавання: додано можливість гнучкої синхронізації початку вимовляння ключового слова з початком інтервалу його он-лайн обробки.

В першій частині експерименту досліджено вплив кількості MFC коефіцієнтів на якість розпізнавання голосової команди і, відповідно, об'єм пам'яті програми (FLASH) та оперативної пам'яті (RAM), які використані для зберігання коду 1D згорткової нейронної мережі та роботи з даними. Можна припустити, що чим більше коефіцієнтів має бути обчислено, а потім оброблено мережею в процесі розпізнавання, тим більше має бути задіяно пам'яті і тим більше буде час розпізнавання звукового зразка. Тому для аналізу використання пам'яті мікроконтролера будувались і використовувались моделі 1D згорткової мережі для наступного числа коефіцієнтів: 12, 13, 15, 17. Значення реальних затрат FLASH та RAM пам'яті отримано після компіляції програми в середовищі Arduino. Значення витраченого часу обчислювалось мікроконтролером и виводилось в термінальне вікно середовища розробки Arduino разом зі значеннями точності визначення голосу.

В другій частині експерименту досліджувалась залежність якості розпізнавання голосової команди від типу вибраної згорткової нейронної мережі, а саме 1D CNN або 2D CNN для 12 та 13 MFC коефіцієнтів.

IV. Аналіз результатів експерименту

Сутність розпізнавання звукових зразків мікроконтролером полягає в тому, що класифікатор нейронної мережі на основі аналізу зразків ключових слів, отриманих з мікрофона, надає кількісну оцінку імовірності приналежності ключового слова до певної групи даних. Загальну імовірність того, що зразок належить до заданого набору груп даних (в даному випадку їх три) задано як 1,0. Відповідно, програмою оцінювалась імовірність приналежності зразка до кожної з груп. Якщо імовірність або точність визначення приналежності ключового слова "hello" до однойменної групи зразків складала більше 0,5, то приймалось рішення, що слово визначено правильно.

В таблиці 1 наведені значення точності визначення ключового слова для 20-ти спроб 1D згорткової мережі, середнє значення точності, часу обробки звукового зразка, об'єму використаної оперативної пам'яті (в байтах) та об'єму використаної пам'яті програм (в байтах) в залежності від кількості MFC

DOI: 10.20535/2523-4455.mea.265406

коефіцієнтів. Також в таблиці наведено оціночні значення часу обробки зразка голосу, часу розпізнавання, об'ємів необхідної FLASH та RAM пам'яті, розраховані Edge Impulse для вибраного типу мікроконтролера або мікропроцесора.

Аналіз результатів показує, що зі збільшенням кількості MFC коефіцієнтів з 12 до 17, а відповідно і точності розпізнавання ключового слова, об'єм пам'яті програм, зайнятої кодом, зростає на 480 байт (менше 1%). Для мікроконтролера nRF52840 це не є суттєвим збільшенням. Об'єм використаної оперативної пам'яті в процесі експерименту не змінювався. Хоч час обчислення точності визначення кодового слова збільшився всього на 14 мс (менше 5%) зі збільшенням кількості MFC коефіцієнтів, проте процедура обчислення є достатньо тривалою (приблизно 0,3 с) в порівнянні з довжиною звукового зразка в 1 с. Це може бути певним обмеженням при обробці звукового сигналу 32-х бітними мікроконтролерами. Для аналізу фраз або речень необхідно використовувати більш потужні мікроконтролери або мікропроцесори.

На рисунку 6 наведено залежність точності визначення ключового слова від кількості MFC коефіцієнтів. Як видно з рисунка 6, точність визначення ключового слова значно зростає коли кількість коефіцієнтів складає 13 і більше, що добре корелюється з загальноприйнятою для використання кількістю (в межах 10-20). Порівнюючи оціночні значення параметрів, розрахованих Edge Impulse, з отриманими після компіляції та в ході експерименту можна зробити висновок, що розрахунковий час майже вдвічі перевищує отриманий експериментально. Витрати FLASH та RAM пам'яті майже не міняються при зміні кількості MFC коефіцієнтів. П'ятикратне перевищення абсолютного значення реальних затрат пам'яті над розрахованими можна пояснити тим Edge Impulse не враховує розмір коду завантажувача програми.

Результати другої частини експерименту, а саме, порівняння точності визначення ключового слова в залежності від типу згорткової нейромережі (1D або 2D) для 12 та 13 MFC коефіцієнтів представлені у таблиці 2.

Результати порівняння показують перевагу 2D мережі у точності визначення ключового слова як для 12, так і для 13 MFC коефіцієнтів. Особливо це помітно для випадку з 12-ма коефіцієнтами, де точність підвищилась з 0,7 до 0,97. Однак при цьому об'єм використаної FLASH пам'яті збільшився на 5%. Об'єм використаної RAM пам'яті у випадку 2D мережі дещо зменшився. Час обробки зразка голосу для обох типів мереж є практично однаковим.

Рис. 6 Залежність точності визначення ключового слова від кількості MFC коефіцієнтів

Таблиця 1 Залежність точності розпізнавання від кількості MFCC для 1D згорткової мережі

Номер експерименту	Кількість коефіцієнтів			
	12	*13*	*15*	*17*
1	0,93915	0,94531	0,81707	0,99974
2	0,67989	0,98438	0,97485	0,99783
3	0,65780	0,87109	0,97858	0,99925
4	0,74887	0,99609	0,99259	0,99946
5	0,49337	0,92188	0,98968	0,90597
6	0,92425	0,96484	0,96796	0,99969
7	0,59760	0,99609	0,89386	0,99986
8	0,54474	0,99609	0,99286	0,99933
9	0,51489	0,99609	0,99861	0,99971
10	0,50484	0,99609	0,93233	0,99997
11	0,75275	0,97656	0,98724	0,99933
12	0,59564	0,89062	0,99282	0,99992
13	0,63440	0,99219	0,92486	0,9021
14	0,82102	0,99609	0,9768	0,99769
15	0,92104	0,70312	0,96122	0,99934
16	0,78673	0,99609	0,99618	0,99983
17	0,85399	0,99609	0,98708	0,99497
18	0,56973	0,99609	0,98177	0,99625
19	0,82023	0,98438	0,99645	0,99998
20	0,91910	0,99609	0,98598	0,99998
Середнє значення	0,71400	0,95976	0,96644	0,98951
FLASH, байт	171272	171368	171560	171752
RAM, байт	52216	52216	52216	52216
Час, мс	283	283	290	297
Оціночний час обробки звуку, мс	429	437	452	465
Оціночна RAM для MFCC, кбайт	23	23	24	24
Оціночний час розпізнавання, мс	41	13	20	21
Оціночна RAM для CNN, кбайт	5,0	5,0	5,3	5,5
Оціночний об'єм FLASH, кбайт	34,5	34,7	34,5	34,6

Таблиця 2 Порівняння точності 1D та 2D нейромереж

Номер експерименту	Кількість коефіцієнтів			
	12		*13*	
	1D	*2D*	*1D*	*2D*
1	0,93915	0,9983	0,94531	0,96578
2	0,67989	0,9484	0,98438	0,99747
3	0,65780	0,9967	0,87109	0,96232
4	0,74887	0,8787	0,99609	0,96722
5	0,49337	0,9844	0,92188	0,99725
6	0,92425	0,9060	0,96484	0,99704
7	0,59760	0,9978	0,99609	0,99951
8	0,54474	0,9698	0,99609	0,86226
9	0,51489	0,9382	0,99609	0,91124
10	0,50484	0,9847	0,99609	0,99942
11	0,75275	0,9956	0,97656	0,99837
12	0,59564	0,9547	0,89062	0,99813
13	0,63440	0,9550	0,99219	0,97548
14	0,82102	0,9928	0,99609	0,99986
15	0,92104	0,9831	0,70312	0,99999
16	0,78673	0,9755	0,99609	0,95702
17	0,85399	0,9674	0,99609	0,98774
18	0,56973	0,9990	0,99609	0,99995
19	0,82023	0,9349	0,98438	0,99997
20	0,91910	0,9983	0,99609	0,91524
Середнє значення	0,71400	0,9680	0,95976	0,974563
FLASH, байт	171272	180456	171368	180456
RAM, байт	52216	51552	52216	51552
Час, мс	283	281	283	281

DOI: 10.20535/2523-4455.mea.265406

Висновки

За результатами експериментальних досліджень можна констатувати той факт, що обчислювальних ресурсів 32-х бітних мікроконтролерів цілком достатньо для розпізнавання голосових команд з можливістю попередньої цифрової обробки звукового сигналу, зокрема, використання мел-частотних кепстральних коефіцієнтів. Вибір кількості коефіцієнтів не впливає значним чином на об'єм задіяної FLASH та RAM пам'яті мікроконтролера nRF52840.

Використання для розпізнавання зразка голосу одновимірної згорткової нейромережі у проведеному експерименті забезпечує економію приблизно 5% пам'яті. Якість розпізнавання ключового слова при кількості MFC коефіцієнтів 12 складає приблизно 0,7. Для 17-ти MFC коефіцієнтів якість розпізнавання становить вже 0,97. Таким чином, 1D згорткової нейромережі мають певні переваги у мікроконтролерних застосунках для обробки та розпізнавання голосу.

Обмеженням розглянутого варіанту розпізнавання голосу на мікроконтролері є достатньо довгий час обробки звукового зразка (приблизно 0,3 с) при тривалості самого зразка в 1 с, що можна пояснити достатньо низькою тактовою частотою в 64 МГц. Збільшення тактової частоти дозволить зменшити час обчислень.

Перелік посилань

[1]. S. Misra, T. Das, P. Saha, U. Baruah and R. H. Laskar, "Comparison of MFCC and LPCC for a fixed phrase speaker verification system, time complexity and failure analysis," 2015 International Conference on Circuits, Power and Computing Technologies [ICCPCT-2015], 2015, pp. 1-4, **DOI:** 10.1109/ICCPCT.2015.7159307.

[2]. Zheng, F., Zhang, G. & Song, Z. "Comparison of different implementations of MFCC", J. Computer Science & Technology 16, 2001, pp.582–589, **DOI:** 10.1007/BF02943243.

[3]. Md Sahidullah, G. Saha, "Design, analysis and experimental evaluation of block based transformation in MFCC computation for speaker recognition," Speech Communication, Volume 54, Issue 4, May 2012, pp. 543-565, **DOI:** 10.1016/j.specom.2011.11.004.

[4]. O. Cheng, W. Abdulla and Z. Salcic, "Hardware–Software Codesign of Automatic Speech Recognition System for Embedded Real-Time Applications," in IEEE Transactions on Industrial Electronics, vol. 58, no. 3, pp. 850-859, March 2011, **DOI:** 10.1109/TIE.2009.2022520.

[5]. F. Barkani, H. Satori, M. Hamidi, O. Zealouk and N. Laaidi, "Amazigh Speech Recognition Embedded System," 2020 1st International Conference on Innovative Research in Applied Science, Engineering and Technology (IRASET), 2020, pp. 1-5, **DOI:** 10.1109/IRASET48871.2020.9092014.

[6]. A. G. Howard M. Zhu B. Chen D. Kalenichenko W. Wang T. Weyand et al. "Mobilenets: Efficient convolutional neural networks for mobile vision applications" arXiv preprint arXiv 17 Apr 2017, pp. 1-9, **DOI:** 10.48550/arXiv.1704.04861

[7]. D. Sinha and M. El-Sharkawy, "Ultra-thin MobileNet," 2020 10th Annual Computing and Communication Workshop and Conference (CCWC), 2020, pp. 0234-0240, **DOI:** 10.1109/CCWC47524.2020.9031228.

[8]. Y. -C. Ling, H. -H. Chin, H. -I. Wu and R. -S. Tsay, "Designing A Compact Convolutional Neural Network Processor on Embedded FPGAs," 2020 IEEE Global Conference on Artificial Intelligence and Internet of Things (GCAIoT), 2020, pp. 1-7, **DOI:** 10.1109/GCAIoT51063.2020.9345903.

[9]. S. M. A. H. Jafri, A. Hemani and L. Intesa, "SPEED: Open-Source Framework to Accelerate Speech Recognition on Embedded GPUs," 2017 Euromicro Conference on Digital System Design (DSD), 2017, pp. 94-101, **DOI:** 10.1109/DSD.2017.89.

[10]. F. Raffaeli and S. Awad, "Portable low-cost platform for embedded speech analysis and synthesis," 2016 12th International Computer Engineering Conference (ICENCO), 2016, pp. 117-122, **DOI:** 10.1109/ICENCO.2016.7856455.

[11]. A. P. Pant, K. -R. Wu and Y. -C. Tseng, "Speak to Action: Offline and Hybrid Language Recognition on Embedded Board for Smart Control System," 2020 International Computer Symposium (ICS), 2020, pp. 85-90, **DOI:** 10.1109/ICS51289.2020.00026.

[12]. F. Sutton, R. Da Forno, R. Lim, M. Zimmerling and L. Thiele, "Demonstration abstract: Automatic speech recognition for resource-constrained embedded systems," IPSN-14 Proceedings of the 13th International Symposium on Information Processing in Sensor Networks, 2014, pp. 323-324, **DOI:** 10.1109/IPSN.2014.6846784.

[13]. I. Kramberger, M. Grasic and T. Rotovnik, "Door phone embedded system for voice based user identification and verification platform," in IEEE Transactions on Consumer Electronics, vol. 57, no. 3, pp. 1212-1217, August 2011, **DOI:** 10.1109/TCE.2011.6018876.

[14]. Q. Qu and L. Li, "Realization of embedded speech recognition module based on STM32," 2011 11th International Symposium on Communications & Information Technologies (ISCIT), 2011, pp. 73-77, **DOI:** 10.1109/ISCIT.2011.6092186.

[15]. "TensorFlow", TensorFlow.org **URL:** https://www.tensorflow.org/ (**access data** 05.06.2022)

[16]. "Keras: The Python deep learning API", Keras: the Python deep learning API. **URL:** https://keras.io/ (**access data** 04.06.2022).

[17]. C. M. J. Galangque and S. A. Guirnaldo, "Speech Recognition Engine using ConvNet for the development of a Voice Command Controller for Fixed Wing Unmanned Aerial Vehicle (UAV)," 2019 12th International Conference on Information & Communication Technology and System (ICTS), 2019, pp. 93-97, **DOI:** 10.1109/ICTS.2019.8850961.

[18]. J. Dudak, M. Kebisek, G. Gaspar and P. Fabo, "Implementation of machine learning algorithm in embedded devices," 2020 19th International Conference on Mechatronics - Mechatronika (ME), 2020, pp. 1-6, **DOI:** 10.1109/ME49197.2020.9286705.

[19]. X. Huang, A. Acero, H.-W. Hon, R. Reddy, "Spoken Language Processing - A Guide to Theory, Algorithm, and System Development", Prentice Hall, 2001, 965pp.

[20]. Gu, J., et al., "Recent advances in convolutional neural networks", Pattern Recognition, 2017, 77: pp. 354-377, **DOI:** 10.48550/arXiv.1512.07108.

[21]. S. Kiranyaz, T. Ince, O. Abdeljaber, O. Avci and M. Gabbouj, "1-D Convolutional Neural Networks for Signal Processing Applications," ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2019, pp. 8360-8364, **DOI:** 10.1109/ICASSP.2019.8682194.

[22]. A. Shenfield, M. Howarth. "A Novel Deep Learning Model for the Detection and Identification of Rolling Element-Bearing Faults" Sensors 2020, 20, 5112. **DOI:** 10.3390/s20185112.

[23]. Edge impulse, edgeimpulse.com, URL: https://www.edgeimpulse.com/ (**access data** 05.06.2022).

[24]. Arduino Nano 33 BLE, store.arduino.cc, URL: https://store.arduino.cc/products/arduino-nano-33-ble (**access data** 05.06.2022).

Надійшла до редакції 15 червня 2022 року
Прийнята до друку 21 серпня 2022 року

UDC 621.382

DOI: 10.20535/2523-4455.mea.265406

Analysis of the Microcontroller Resources Using Specifics for Speech Recognition

A. R. Ryzhova[f], ORCID 0000-0003-3278-8448,
Yu. O. Onykiienko[s], PhD Assoc.Prof., ORCID 0000-0001-7508-8391,
Department of Acoustic and Multimedia Electronic Systems, ames.kpi.ua
National Technical University of Ukraine «Igor Sikorsky Kyiv Polytechnic Institute» ROR 00syn5v21
Kyiv, Ukraine

***Abstract*—The use of neural networks for information recognition, in particular, voice, expands the functional capabilities of embedded systems on microcontrollers. But it is necessary to take into account the limitations of the microcontroller resources. The purpose of the work is to analyze the impact of voice processing parameters and neural network architecture on the degree of microcontroller resources usage. To do this, a database of samples of the keyword, samples of other words and voices, and samples of noise are created, the probability of recognizing the keyword among other words and noises is evaluated, the dependence of the amount of memory used on the microcontroller and the decision-making time on the number MFC coefficients is established, the dependence of the amount of used memory of the microcontroller and the decision-making time on the type of convolutional neural network is established also.**

During the experiment, the Arduino Nano 33 BLE Sense development board was used. The neural network model was built and trained on the Edge Impulse software platform. To conduct the experiment, three groups of data with the names "hello", "unknown", "noise" were created. The group "hello" contains 94 examples of the word "hello" in English, spoken by a female voice. The "unknown" group contains 167 examples of other words pronounced by both female and male voices. The "noise" group contains 166 samples of noise and random sounds. According to Edge Impulse's recommendation, 80% of the samples from each of the data groups were used to train the neural network model, and 20% of the samples were used for testing.

Analysis of the results shows that with an increase in the number of MFC coefficients and, accordingly, the accuracy of keyword recognition, the amount of program memory occupied by the code increases by 480 bytes (less than 1%). For the nRF52840 microcontroller, this is not a significant increase. The amount of RAM used during the experiment did not change. Although the calculation time of the accuracy of the code word definition increased by only 14 ms (less than 5%) with the increase in the number of MFC coefficients, the calculation procedure is quite long (approximately 0.3 s) compared to the sound sample length of 1 s. This can be a certain limitation when processing a sound signal with 32-bit microcontrollers. To analyze phrases or sentences, it is necessary to use more powerful microcontrollers or microprocessors.

Based on the results of experimental research, it can be stated that the computing resources of 32-bit microcontrollers are quite sufficient for recognizing voice commands with the possibility of pre-digital processing of the sound signal, in particular, the use of low-frequency cepstral coefficients. The selection of the number of coefficients does not significantly affect the amount of used FLASH and RAM memory of the nRF52840 microcontroller. The comparison results show the superiority of the 2D network in the accuracy of the keyword definition for both 12 and 13 MFC coefficients. The use of a one-dimensional convolutional neural network for voice sample recognition in the conducted experiment provides memory savings of approximately 5%. The quality of keyword recognition with the number of MFC coefficients of 12 is approximately 0.7. For 17 MFC coefficients, the recognition quality is already 0.97. The amount of RAM used in the case of the 2D network has decreased slightly. Voice sample processing time for both types of networks is practically the same. Thus, 1D convolutional neural networks have certain advantages in microcontroller applications for voice processing and recognition. The limitation of voice recognition on the microcontroller is the sufficiently long processing time of the sound sample (approximately 0.3 s) with the duration of the sample itself being 1 s, this can be explained by a sufficiently low clock frequency of 64 MHz. Increasing the clock frequency will reduce the calculation time.

Keywords — microcontrollers; mel-frequency cepstral coefficients; convolutional neural networks; voice recognition.