UDC 004.855.5

DOI: 10.20535/2523-4455.mea.300851

Comparison of the Efficiency of a Neural Network for Image Recognition on Microcontrollers

R. D. Sharuiev[f], 0009-0007-9644-6865
P. V. Popovych[s] , PhD Assoc.Prof., 0000-0002-1572-3127
National technical university of Ukraine "Igor Sikorsky Kyiv polytechnic institute" 00syn5v21
Kyiv, Ukraine

***Abstract*—The paper is devoted to comparing two popular models of 32-bit microcontrollers for working with neural networks for object recognition. The target devices were the ESP32 and STM32 microcontrollers, on which an artificial neural network was deployed, written using the Python programming language and the TensorFlow library. Micropython was chosen as the operating system for the microcontrollers. The paper compares the performance of the ESP32 and STM32 microcontrollers for object detection using a neural network and their classification. The image recognition time and the percentage of correctly classified objects were compared depending on the number of neuron layers and the number of training epochs within these networks. The article shows that the number of layers and training epochs directly affects the accuracy of object classification in the image. The obtained results show that increasing the number of layers of the neural network increases the overall accuracy of object recognition using the studied neural network, increasing the number of training epochs logarithmically increases the accuracy of recognition and classification within the neural network, but at the same time, increasing the number of neuron layers leads to an increase in the total recognition time. The difference in the obtained results for the accuracy of image recognition of microcontrollers differs within 5%.**

Keywords* — *microcontroller; neural network; epoch; training; classification.

I. Introduction

The modern world is based on the widespread use of electronic devices and systems. With the development of technology, these systems have become digital, that is, they began to contain certain computing blocks, built on microcontrollers or microprocessors. Examples of these devices can be found in your pocket, in your hand, in your car, etc. An example of such a device can be the smart ring Samsung Galaxy Ring [1], announced by Samsung, shown in Figure 1.

These devices have a complex of software and hardware solutions that are responsible for their aspects of work.

From the side of software problem-solving, the following can be highlighted:

- Writing an operating system using ready-made solutions if needed, or writing firmware.
- Choosing a programming language with which the firmware or operating system will be written.
- Writing basic functionality that has to solve certain tasks.
- Further expansion of the system.

The main problem of hardware solution can be identified: the choice of an optimal microcontroller or microprocessor.

The problem of hardware solutions is the most relevant today. The modern world uses the method of software development with the future perspective of product development. This process is called the CI/CD pipeline. This leads to the fact that throughout the entire time of technical product support, its hardware must be powerful enough to support and ensure the normal functioning of the software firmware. In essence, a modern technical product is a platform for the program that will be deployed on it [2].

Fig. 1. Smart ring Samsung Galaxy Ring.

DOI: 10.20535/2523-4455.mea.300851

Understanding this problem, STMicroelectronics [3] has developed a series of microcontrollers based on ARM Cortex [4] cores. Among them, it is worth highlighting the STM32H5 microcontroller [5], presented in Figure 2. This microcontroller was released in 2023. Among its advantages, one can highlight its adaptability to work with neural networks. However, this chip has 2 main disadvantages:

1. Large dimensions.
2. High price.

These disadvantages increase the entry threshold for beginner developers who are developing neural networks for use with microcontrollers [6].

On the other hand, microcontrollers from Broadcom are presented on the market. Among them, one can highlight Broadcom BCM2711 [7] and microcomputers from Raspberry [8], presented in Figure 3. If you compare Broadcom BCM2711 and STM32H5, then Broadcom BCM2711 has smaller dimensions, and a higher price, but gives more opportunities for the developer, and considering the Raspberry platform, for example, Raspberry Pi 4B [9], the developer gets not just a module for development, but a microcomputer.

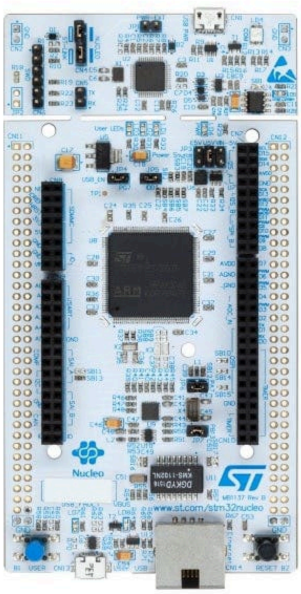

Fig. 2. Nucleo 32 with STM32H5 microcontroller.

Fig. 3. Raspberry Pi 4B with Broadcom BCM2711 microcontroller.

If writing and training a neural network is not a problem, then choosing the optimal microcontroller is a more complex task.

In the paper [10], which presents the application of neural networks together with microcontrollers, there are several drawbacks, namely: small image size, and comparison of the work of a microcontroller with a personal computer. The second drawback is significant because the computer processor and the microcontroller are different in their structure and have different computing power. The paper shows that the performance of the neural network on a personal computer is on average 5% higher than on a microcontroller. At the same time, the speed of object recognition using a neural network on a personal computer is almost 2.5 times faster than on a microcontroller.

In another paper [11], the application of neural networks with microcontrollers is also presented. The main idea of this article is the use of additional external memory to expand the amount of RAM, which is not enough in the microcontroller. The article shows that the use of external memory is feasible and possible, but this approach has certain disadvantages: higher power consumption, which is significant for autonomous devices, rapid wear of the memory chip due to frequent cycles of rewriting information, slowing down the model's work due to the lower data transfer speed between the microcontroller and the memory chip.

This work [12] describes a method for optimizing a convolutional neural network for use with microcontrollers. The results presented in the work look optimistic and promising. The disadvantage of this work is the use of grayscale images and sizes of 32x32 pixels.

In the work [13], the application of TinyML technology for minimizing neural networks for use with microcontrollers is presented. It talks about the advantages of this technology over the classic use of neural networks and describes the principle of this technology. This paper is important for the application of neural networks with microcontrollers, and the technology described in this paper is revolutionary and already has practical applications in work [10]. This work has a drawback: it lacks examples of application on a real microcontroller.

This paper will consider a complex of problems, such as comparing the most accessible and widespread models of 32-bit microcontrollers, the optimal size of the neural network, and the optimal degree of neural network training.

II. EQUIPMENT AND DATA USED IN EXPERIMENT

Considering the specifics of the system, namely working with images, a model was written for the experiment, which deals with the recognition and classification of an object in the image [14] [15]. This task is the most common when working with images.

DOI: 10.20535/2523-4455.mea.300851

For the synthesis of the neural network, a convolutional neural network architecture (CNN) [16] [17] was chosen. This architecture is basic for tasks of classifying objects in images.

Python [18] was chosen as the programming language and its firmware for microcontrollers called Micropython [19]. Considering that Micropython cannot directly work with full-fledged neural network models, and the network has a large weight, it is necessary to optimize the model before using it [13].

Python and its framework, Micropython, suffer from a practical single-threaded limitation. This issue stems from the core of the Python programming language and is constrained by the Global Interpreter Lock (GIL). On one hand, the GIL ensures thread safety by restricting access to memory areas for all threads except the active one. On the other hand, it executes threads sequentially, preventing true multithreading in Python. Consequently, genuine full-fledged multithreading is absent in this programming language [20].

Regarding multiprocessing, it's worth noting that Python inherently multiplies interpreters and applies the GIL to distribute tasks across different cores and switch between them. Micropython inherits these limitations as well. Therefore, when working with models and accessing the same memory area, the significant benefit of multiprocessing and multithreading is lacking; thus, this capability remains untapped.

The tool for synthesizing and training the neural network will be the TensorFlow [21] (TF) framework. The advantage of this framework is the ability to minimize the obtained model for use with microcontrollers and the presence of a minimized version of the framework called TensorFlow Lite [22] (TFL). As a training dataset, a pre-classified set from Stanford University researchers called ImageNet [23] was chosen. It contains 1000 classes of different objects, 1281167 training images, 50000 images for validation, and 100000 test images. Unfortunately, the exact sizes of the images in the ImageNet dataset are not specified, but it was possible to find information in the public domain that the average size of the images is 482x418 pixels. An example of the dataset is shown in Figure 4.

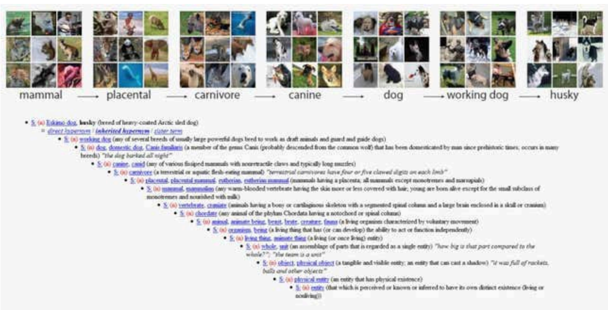

Fig. 4. Example of the ImageNet dataset.

Two of the most popular microcontrollers available on the market were chosen as microcontrollers, namely the ESP32 CAM module [24] and the STM32 Smart module [25]. It should be noted that the ESP32 series microcontroller is more powerful, but STM32 has special software for working with neural networks, which makes them more competitive. The characteristics of the microcontrollers are given in Table 1.

TABLE 1 MAIN CHARACTERISTICS OF TARGET DEVICES

Main characteristics of the device	**Device**	
	STM32	***ESP32***
Number of cores, pcs.	1	2
Amount of RAM, kB	20	520
Processor clock speed, MHz.	72	240

III. EXPERIMENT SETUP

To study the efficiency of the neural network on the selected microcontrollers, the following metrics were introduced:

- Accuracy depends on the epoch.
- Recognition time of a pre-prepared dataset.
- The maximum number of epochs that can be used with the microcontroller.

Fig. 5 Appearance of the ESP32 CAM microcontroller module.

Fig. 6 Appearance of the STM32-Smart microcontroller module.

DOI: 10.20535/2523-4455.mea.300851

To ensure transparency, the neural network for both microcontrollers is unified for the epoch, the experiment is conducted 10 times, and the obtained results are averaged.

The test set for comparing the operation of microcontrollers is taken from the ImageNet dataset. They combine 1000 classes of different objects with 50 images each. In total, the experiment is conducted on the full test data set. The test image set for both microcontrollers is the same.

The operating time of the model is defined as the total time spent on classifying a set of images, averaged over the number of experiments, and is determined by the formula:

$$t_{total} = \frac{\sum t_n}{10},$$

where t_{total} – average time spent by the model on recognizing the test data set, t_n – time spent by the model on 1 experiment, 10 – total number of experiments for each epoch.

The accuracy of the model's work is determined by the formula:

$$N = \frac{p}{10n} \cdot 100\%,$$

where p – total number of correctly classified objects, n – total size of the test sample, 10 – total number of experiments for each epoch.

The number of training epochs directly affects the accuracy of recognition. To study the impact of the number of epochs on recognition accuracy, the following values are fixed, which are given in Table 2.

TABLE 2 EXPERIMENT NUMBER AND NUMBER OF TRAINING EPOCHS

Experiment number	The number of training epochs
1	1
2	5
3	10
4	25
5	50

The number of neuron layers also directly affects the accuracy of recognition. They are limited by the amount of RAM and the frequency of the processor. The more layers of neurons, the heavier the final model and the greater the load on the processor, which is very important for real-time operation. The minimum number of neuron layers is limited to 5 layers, and the maximum number of layers is determined experimentally and is equal to 8. A larger number of layers requires a larger amount of RAM, which the STM32 microcontroller lacks.

IV. ANALYSIS OF OBTAINED RESULTS

After training the model for a different number of epochs and conducting experimental measurements for two models of microcontrollers, the results were obtained, which are entered in Table 3.

From the obtained results, it can be seen that the accuracy of the systems on both microcontrollers depends on the number of layers and the number of training epochs. On average, the difference in experiments is within 5%, which can be considered a statistical error that depends on the operation of the microcontroller and cannot be influenced. On the other hand, the execution time of models on STM32 and ESP32 differs on average by 2 or more times. The reason for this is the difference in the performance of the microcontrollers, namely:

- The number of cores in STM32 is half that of ESP32.
- The amount of RAM in STM32 is 26 times less than in ESP32.
- The operating frequency of the microcontroller in STM32 is 3.3 times less than in ESP32.

Considering that this model operated in a single-threaded mode, we can disregard the difference in the number of cores.

TABLE 3 EXPERIMENTAL RESULTS

Number of layers	Number of training epochs	Obtained results			
		STM32		*ESP32*	
		Time t_{total}, sec	*Accuracy an %*	*Time t_{total}, sec*	*Accuracy an %*
5	1	7.87	4.23	4.55	4.23
	5	8.58	17.18	4.19	18.67
	10	6.25	33.89	4.48	31.02
	25	6.80	46.12	4.48	48.25
	50	8.48	65.06	3.52	63.78
6	1	12.57	8.37	6.14	9.13
	5	12.68	24.58	7.14	22.56
	10	9.45	36.29	7.53	38.24
	25	13.32	49.85	6.83	50.73
	50	13.80	72.35	6.16	71.26
7	1	16.19	17.84	7.96	15.39
	5	13.38	30.37	6.42	31.58
	10	13.77	49.83	6.4	46.89
	25	18.44	66.78	7.09	68.49
	50	13.96	79.36	6.14	81.84
8	1	22.17	23.73	8.51	24.94
	5	23.85	45.78	9.83	42.75
	10	21.97	60.37	11.23	63.67
	25	21.96	85.83	10.73	85.36
	50	24.83	93.75	12.45	94.47

The obtained results indicate that the STM32 microcontroller is unsuitable for use in this system.

However, it should be noted that theoretically these results can be significantly improved if we abandon Micropython and switch to a faster Robot.js firmware or the 'C/C++' programming languages. In this case, the results of the model's work on STM32 and ESP32 will be more similar, and the model's operation time should significantly decrease. This hypothesis will be further investigated, but it is not essential for this work.

The dependence of the model's accuracy on the number of layers can be seen in Table 3. From the obtained data, it can be seen that the more layers of neurons are used, the greater the accuracy of recognition and classification of objects is obtained, but the longer the model execution time is. Figure 7 shows the maximum accuracy results depending on the number of neuron layers.

It should be noted that the maximum accuracy obtained for the maximum number of training epochs for both microcontrollers differ by 5%, which is within the statistical error. Therefore, this shows of the equal capabilities of both microcontrollers for work with neural networks, which are used for the recognition and classification of objects.

Experiments were conducted for each number of layers for a different number of model training epochs. From the obtained data, it can be seen that the more epochs of model training are used, the greater the accuracy of the model's work is obtained. The execution time for different degrees of training (1, 5, 10, 25, 50 epoch) can be considered the same because the difference between the maximum and minimum model operating time does not exceed 5%. It should be noted that the operating time of the model on STM32 is approximately twice as long as on ESP32, which can be explained by the difference in the characteristics of these microcontrollers (Table 3). The graphs of the obtained results are presented in Figures 8-11.

The obtained data clearly illustrate the increase in overall model accuracy with an increase in the number of training epochs, however, the maximum accuracy for 5 layers of neurons (Figure 8) is 65.06% for STM32 and 63.78% for ESP32, and for 8 layers of neurons (Figure 11) the maximum accuracy increases to 93.75% for STM32 and 94.47% for ESP32 respectively. Therefore, the more layers of neurons, the higher the overall accuracy of the model, but the overall operating time of the model increases (for 5 layers 8.48 seconds for STM32 3.52 seconds for ESP32, and 8 layers 24.83 seconds for STM32 and 12.45 seconds for ESP32).

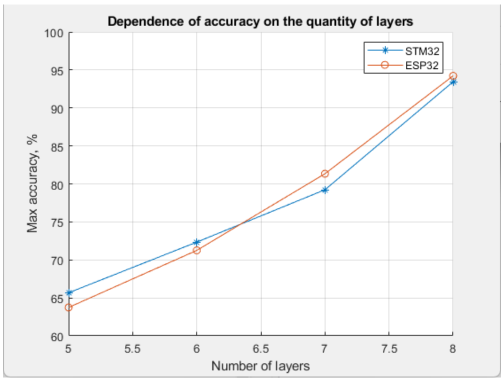

Fig. 7 Graph of accuracy dependence on the number of layers.

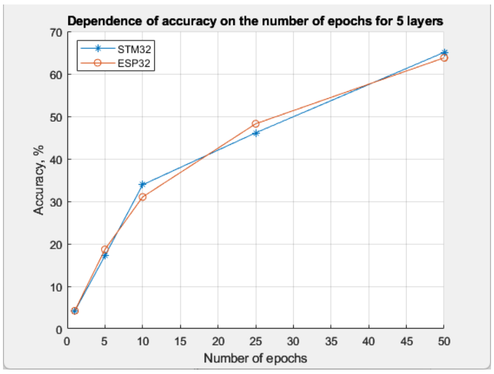

Fig. 8 Graph of recognition accuracy depending on the number of training epochs for 5 neuron layers.

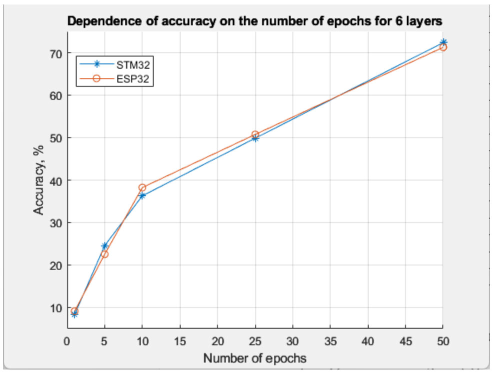

Fig. 9 Graph of recognition accuracy depending on the number of training epochs for 6 neuron layers.

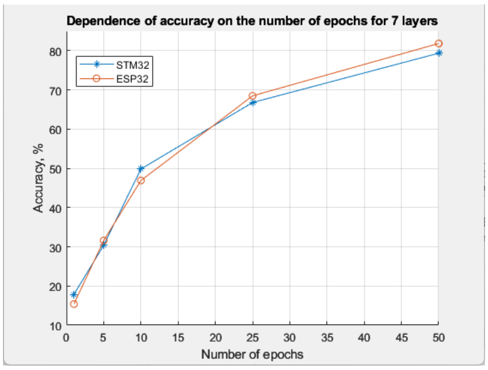

Fig. 10 Graph of recognition accuracy depending on the number of training epochs for 7 neuron layers.

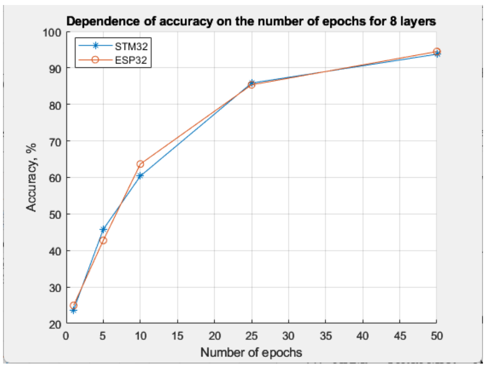

Fig. 11 Graph of recognition accuracy depending on the number of training epochs for 8 neuron layers.

DOI: 10.20535/2523-4455.mea.300851

DOI: 10.20535/2523-4455.mea.300851

This investigation is not possible regarding the memory usage of a microcontroller. In Micropython, there is a module called GC (Garbage Collector), but it only shows the HEAP memory, which is dynamic and indicates the portion occupied by the program. However, it does not provide complete information about the resources being used, and so on.

Conclusions

This article explores the use of neural networks on two popular models of modern microcontrollers. The task of the neural network is to recognize and classify objects in images. The study aimed to compare the accuracy of recognition and the operating time of the model on two microcontroller platforms.

For the experiment, a neural network was written using the Python programming language using the TensorFlow framework for synthesizing and training neural networks. The model was trained using the ImageNet dataset. For model validation, 50,000 images from the ImageNet test dataset were used.

The obtained results show that the accuracy of the model's work depends on the number of training epochs and the number of neuron layers. At the same time, increasing the number of neuron layers in the model leads to a sharp increase in the model's operating time. The results obtained in the work show that the application of neural networks with microcontrollers is possible and appropriate. The difference in accuracy between the platforms under study is within 5%, which can be neglected, but the model's operating time differs more than 2 times.

The main obstacle to use microcontrollers with neural networks is the availability of libraries for running neural networks on low-power devices. For the task of working in real-time, "Micropython" is not suitable due to the large use of microcontroller resources for stable firmware operation, which takes resources from the model. To improve the results of the neural network, it is necessary to switch to lighter and faster firmware, for example, Robot.js, or to write your firmware using the "C/C++" programming language.

References

[1] THE ELEC, "Samsung starts development of 'Galaxy Ring'," THE ELEC, Korea Electronics Industry Media, 19 07 2023. [Online]. Available: https://www.thelec.net/news/articleView.html?idxno=4595. [Accessed 05 02 2024].

[2] Red Hat, "What is a CI/CD pipeline?," Red Hat, Inc., 11 05 2022. [Online]. Available: https://www.redhat.com/en/topics/devops/what-cicd-pipeline. [Accessed 07 02 2024].

[3] STMicroelectronics, "STMicroelectronics website," STMicroelectronics, [Online]. Available: https://www.st.com/content/st_com/en.html. [Accessed 07 02 2024].

[4] ARM, "Cortex-M33," Arm Limited, [Online]. Available: https://www.arm.com/products/silicon-ip-cpu/cortex-m/cortex-m33. [Accessed 12 02 2024].

[5] STMicroelectronics, "STM32H5 Series," STMicroelectronics, [Online]. Available: https://www.st.com/en/microcontrollers-microprocessors/stm32h5-series.html. [Accessed 07 02 2024].

[6] V. Chandra and A. Hareendran, Artificial Intelligence and Machine Learning, Delhi: PHI Learning, 2014.

[7] Alldatasheet, "Broadcom BCM2711," Alldatasheet, [Online]. Available: https://www.alldatasheet.com/datasheet-pdf/pdf/1283902/ETC1/BCM2711.html. [Accessed 07 02 2024].

[8] Raspberry Pi, "Raspberry Pi website," Raspberry, [Online]. Available: https://www.raspberrypi.com/. [Accessed 07 02 2024].

[9] Raspberry Pi, "Raspberry Pi 4 Model B," Raspberry Pi, [Online]. Available: https://www.raspberrypi.com/products/raspberry-pi-4-model-b/. [Accessed 07 02 2024].

[10] R. Sharuiev and P. Popovich, "Research of the Characteristics of a Convolutional Neural Network on the ESP32-CAM Microcontroller," [Online]. **DOI:** 10.20535/2523-4455.mea.277487.

[11] H. Miao and F. X. Lin, "Towards Out-of-core Neural Networks on Microcontrollers," IEEE , 02 01 2023. [Online]. **DOI:** 10.1109/SEC54971.2022.00008.

[12] T. Garbay, P. Dobias, W. Dron, P. Lusich and I. Khalis, "CNN Inference Costs Estimation on Microcontrollers:," Sorbonne Universite, 17 12 2021. [Online]. **DOI:** 10.1109/ICECS53924.2021.9665540.

[13] P. Warden and D. Situnayake, TinyML: Machine Learning with TensorFlow Lite on Arduino and Ultra-Low-Power Microcontrollers, "O'Reilly Media, 2019.

[14] X. Jiang, A. Hadidp, Y. Pang, E. Granger and X. Feng, Deep Learning in Object Detection and Recognition, Texas: Computer Science and Engineering Department, University of Texas at Arlington, 2019.

[15] R. O. Duda, P. E. Hart and D. G. Stork, Pattern Classification, Toronto: A Wiley Interscience publication John Wiley & Sons, 1973.

[16] U. Kern, "Convolutional neural network (CNN)," [Online]. Available: https://ujjwalkarn.me/2016/08/11/intuitive-explanation-convnets/. [Accessed 17 04 2023].

[17] A. H. Habibi and H. E. Jahani, Guide to Convolutional Neural Networks, Spain: Springer International Publishing, 2017. **DOI:** 10.1007/978-3-319-57550-6

[18] Python organization, "Python official webpage," Python.org, [Online]. Available: https://www.python.org/. [Accessed 09 02 2024].

[19] Micropython organization, "Micro Python," [Online]. Available: https://micropython.org/download/esp32/. [Accessed 09 02 2024].

[20] DevCademy Media Inc., "What Is the Python Global Interpreter Lock (GIL)?," [Online]. Available: https://realpython.com/python-gil/. [Accessed 25 07 2024].

[21] Tensorflow organization, "Tensorflow official webpage," Google incorporated, [Online]. Available: https://www.tensorflow.org/?hl=ru. [Accessed 07 02 2024].

[22] Tensorflow organization, "Tensorflow lite compiler," Google incorporated, [Online]. Available: https://www.tensorflow.org/lite?hl=en. [Accessed 07 02 2024].

DOI: 10.20535/2523-4455.mea.300851

[23] Stanford University, "ImageNet website," Stanford Vision Lab, 11 03 2021. [Online]. Available: https://www.image-net.org/. [Accessed 15 02 2024].

[24] AI-Thinker, "ESP-32 CAM datasheet," [Online]. Available: https://docs.ai-thinker.com/_media/esp32/docs/esp32-cam_product_specification_zh.pdf. [Accessed 14 02 2023].

[25] STM32 base org., "STM32 Smart V2.0," [Online]. Available: https://stm32-base.org/boards/STM32F103C8T6-STM32-Smart-V2.0.html. [Accessed 09 02 2024].

[26] B. Dickson, "TinyML is bringing neural networks to microcontrollers," TechTalks, 17 01 2022. [Online]. Available: https://bdtechtalks.com/2022/01/17/mcunetv2-tinyml-deep-learning-microcontrollers/. [Accessed 07 03 2024].

Надійшла до редакції 28 березня 2024 року
Прийнята до друку 04 липня 2024 року

Порівняння ефективності роботи нейронної мережі для розпізнавання зображень на мікроконтролерах

Р. Д. Шаруєв[f], 0009-0007-9644-6865
П. В. Попович[s], PhD, 0000-0002-1572-3127
Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського» 00syn5v21
Київ, Україна

***Анотація*—Статтю присвячено порівнянню двох популярних моделей 32 бітних мікроконтролерів для роботи з нейромережами для розпізнавання об'єктів. Як цільові пристрої використано мікроконтролери ESP32 та STM32, на яких було розгорнуто штучну нейронну мережу, написану за допомогою мови програмування Python та бібліотеки TensorFlow. В якості операційної системи для мікроконтролерів обрано Micropython. У роботі виконано порівняння продуктивності мікроконтролерів ESP32 та STM32 для виявлення об'єктів за допомогою нейронної мережі та їх класифікації. Порівняння проведено за часом розпізнавання зображень та відсотком правильно класифікованих об'єктів в залежності від кількості шарів нейронів та кількості епох навчання в рамках даних мереж. У статті показано, що кількість шарів та епох навчання напряму впливає на точність класифікації об'єктів на зображенні. Отримані результати показують, що збільшення кількості шарів нейронної мережі збільшує загальну точність розпізнавання об'єктів за допомогою вивченої нейронної мережі, збільшення кількості навчальних епох логарифмічно збільшує точність розпізнавання та класифікації в рамках нейромережі, але при цьому збільшення кількості шарів нейронів призводить до збільшення загального часу розпізнавання. Аналіз отриманих даних показав, що різниця точності розпізнавання зображень на мікроконтролерах відрізняється в межах 5%, що не є суттєвим, проте відмінність у затраченому часі в середньому склала 2 рази. Під час проведення експерименту помічено, що максимальна кількість шарів нейромережі є обмеженою до 8 для мікроконтролера STM32 через брак постійної та оперативної пам'яті. Через це обмеження повні можливості мікроконтролера ESP32 не було розкрито, тож теоретично система з використанням ESP32 може бути більш ефективною для задач розпізнавання та класифікації об'єктів на зображеннях. Проведений експеримент показав, що збільшення кількості епох навчання збільшує точність, а кількість шарів моделі впливає на початкове значення точності.**

Ключові слова — мікроконтролер; нейромережа; епоха; навчання; класифікація.