TinyML: Analysis of Xtensa LX6 microprocessor for Neural Network Applications by ESP32 SoC

Md Ziaul Haque Zim
[1] *Department of Computer Science and Engineering*
Daffodil International University
Dhaka, Bangladesh
[2] *Department of Computer Science and Engineering*
Saint-Petersburg Electrotechnical University "LETI"
Saint Petersburg, Russia
ziaul15-1133@diu.edu.bd

***Abstract*—In recent decades, Machine Learning (ML) has become extremely important for many computing applications. The pervasiveness of ultra-low-power embedded devices such as ESP32 or ESP32 Cam with tiny Machine Learning (tinyML) ap-plications will enable the mass proliferation of Artificial Intelligent powered Embedded IoT Devices. In the last few years, the microcontroller device (Espressif ESP32) became powerful enough to be used for small/tiny machine learning (tinyML) tasks. The ease of use of platforms like Arduino IDE, MicroPython and TensorFlow Lite (TF) with tinyML application make it an indispensable topic of research for mobile robotics, modern computer science and electrical engineering. The goal of this paper is to analyze the speed of the Xtensa dual-core 32-bit LX6 microprocessor by running a neural network application. The different number of inputs (9, 36, 144 and 576) inputted through the different number of neurons in neural networks with one and two hidden layers. Xtensa LX6 microprocessor has been analyzed because it comes inside with Espressif ESP32 and ESP32 Cam which are very easy to use, plug and play IoT device. In this paper speed of the Xtensa LX6 microprocessor in feed-forward mode has been analyzed.**

Keywords—TinyML, Xtensa LX6 microprocessor, Machine Learning, Neural Network, Embedded IoT Device, Espressif ESP32 and ESP32 Cam

I. INTRODUCTION

The explosive growth in machine learning and miniaturization of electronics has been created new research opportunities. The thought of eco-friendly energy, which stresses the utilization of electric-driven gear [1, 2], which has been recently in the discussion [3]. Machine Learning already achieved the great success with the multiple core enabled processor, and in recent days researchers thinking to apply small or tiny machine learning application on system on chip (SoC) microcontrollers with less processing power [4].

However, the extensive deployment of the machine learning algorithms will bring a clear rise in the artificial intelligence where a huge amount of processing power will be used due to process small algorithm with high-end multicore processor. Other animals and humans have a brain and a central nervous system to avail the process of information with neural networks. In the different fields, neural network has been successfully implemented, from detecting criminal [5, 6] and medical diagnosis [7, 8] through image classification [9], to autonomous driving [10].

As part of a long-established custom, practice, or belief, the training of neural network is commonly centralized, and for the computational intensity and big memory requirements it is mostly done on cloud servers. An instance of a particular situation the dataset of deep neural networks it could be more than several dozen terabytes for learning [11, 12].

In centralized approach some researcher found disadvantages like data privacy issues [13], between devices and cloud it increased latency and network traffic for that forbid in some determination [14]. It also decreases the reliability, when exchanging of information is important between cloud and device or sensors [15]. And some researcher presents that sometimes the center can be SoC, smartphone and micro data center between source and data [16].

In the most recent decade, many researcher, engineers and professionals have studied on machine learning applications and implement these on microcontrollers, SoC's and embedded IoT devices. Special libraries [17] and hardware development with ML/AI functions and capabilities [18–20]. For the ESP32 and ESP32-S Series SoCs Espressif developed an IoT Development Framework named ESP-IDF [21] and face detection and recognition platform that is currently based on ESP32 chip [22].

In addition, many research works have been done, studied and used ESP32 widely like in embedded IoT devices such as LPG gas leakage [23], home monitoring system [24], solar cell current and its battery power prediction [25] and air pollution detect system [26].

In this paper, the Tensilica Xtensa LX6 microprocessor by Cadence is discussed and its neural network capabilities will be analyzed [27].

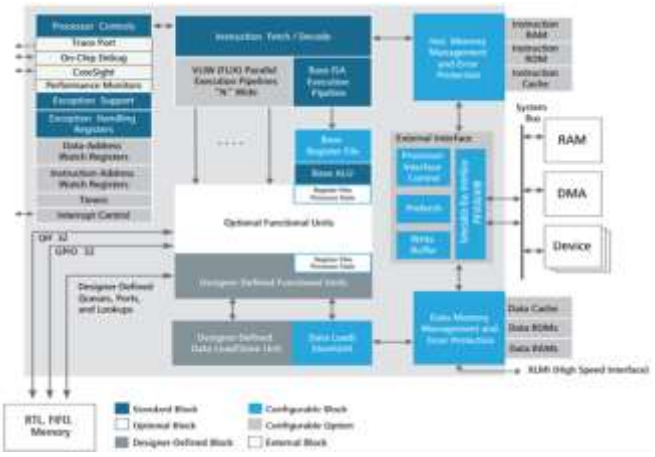

Fig. 1. Xtensa LX6 DPU showing standard, optional, and designer-defined blocks

In Fig. 1, the ESP32 SoC's processor Xtensa LX6 DPU showing standard, optional, and designer-defined blocks. For neural network acceleration, this SoC does not have any special hardware but Espressif has developed and published some ML frameworks and examples for ESP32 chip. So that I choose a ESP32 WROOM IoT development board by Espressif for the neural network acceleration that inside have a Xtensa LX6 microprocessor SoC.

A SoC or a new class of programmable processor that combines high-performance and industry-standard, software-programmable multi-core CPU is called data plane processing units (DPUs). Xtensa LX6 called as DPU. It's highly efficient, small and it have low-power 32-bit base architecture. In Fig. 2 an ESP32 functional block diagram by Espressif can be seen.

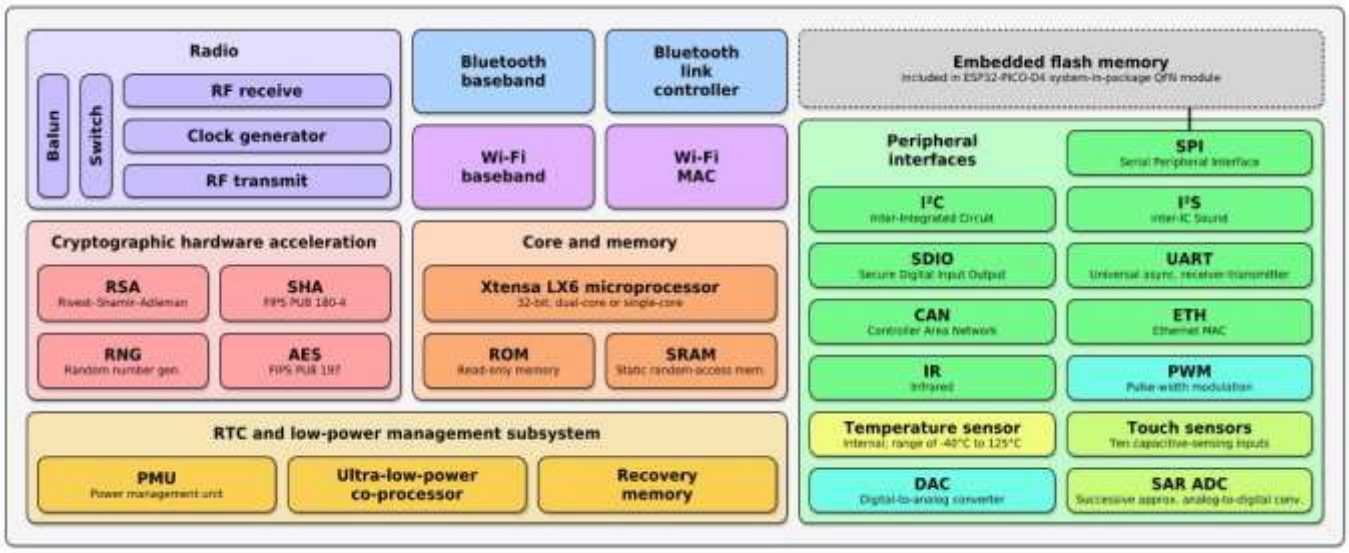

Fig. 2. Espressif ESP32 Functional Block Diagram

The formation of this studies as follows. In section 2, a scheme and some recent studies that deal with ESP32 SoC and machine learning usage on ESP32 will be described. In section 3, some study that deal with neural networks and multiple core typical example or pattern will be described. In the section 4, analysis of neural network feedforward process propagation speed with one and two cores. With the different number of neurons in the hidden layer testing will be done and finally in section 5 result will be analyzed and discussed.

II. The Scheme

For analysis of Xtensa LX6 microprocessor for neural network applications, need to perform few necessary steps. An ESP32 development board is essential tools for this. Arduino IDE with Espressif ESP32 board compatible is important. Using Arduino IDE, a neural network will be developed and will implement into an ESP32 for the analyzing its processing power.

It can be observed that Xtensa LX6 microprocessor or microcontrollers are mostly used for simple with a few neurons and there are no papers regarding machine learning applications that study, being researched or investigating the speed of running neural networks on a microcontroller in the real-world conditions.

In the next, the different approaches analyzed that already studied by the researchers.

A. Machine Learning on the Cloud

By Chand et al. an ultrasonic sensor with ESP32 has been used to predict the behavior and analyzing the status of a person in a room. With the help of sensor, the ESP32 is used for collecting the data and send it to the cloud. A machine learning algorithm is used to analyze data and give decision whether the situation is normal or abnormal situation [28].

By Fernoaga et al an ESP32 camera has been used for take the pictures of electricity/water/gas utility counters and to send them to the cloud for the neural network's recognition [29].

Rosato and Masciadri has make a non-invasive observation system by using an ESP32 and detect the siting people by analyzing the data on a cloud server and Logistic Regression was chosen to tackle the classification of the problem [30].

By Islam et al. a biosensing module has been designed with an Analogue Devices AD8232, ESP32 as microcontroller to send the data to the Linux server with the neural network [31].

By Komarek et al. a project has been presented that aimed to provide an abstraction layer that used MQTT for interfacing nodes with sensors and those particular nodes are used to send data to other nodes and the clouds for analysis [32].

By Zidek et al. optimized input data for wireless IoT devices by accumulation to the multi-value packet before sending them to the cloud service [33].

B. Machine Learning on Xtensa LX6 microprocessor by using ESP32

Very few authors have been studied for ESP32 for some Machine Learning applications.

Kokoulin et al implement a video stream from a public place and detects the presence of a face or silhouette fragment by using ESP32. For reduce high network traffic and computing load of the central face recognition server and got the result of 80 % to 90 % traffic decrease [34].

Espressif Systems has been developed an open-source ESP-WHO framework that is available on GitHub for recognition and detection [22]. The Cascaded Convolutional

Networks model and new mobile architecture – Mobile Net V2 is based on this ESP-WHO framework [35, 36].

C. Machine Learning on Arduino with Arduino IDE

Arduino is an open-source hardware platform and it has many hardware devices that being used for machine learning applications. Some author already implements machine learning applications on Arduino development boards by using Arduino IDE.

C-Mantec neural network algorithm that can add new neurons in the leaning process has been implemented by Ortega-Zamorano et al. [37]. A feed forward neural network with two hidden and one output neuron has been used to control PID controller and DC motor in Arduino Uno based development board, designed and implemented by Rai and Rai [38]. Adhitya et al. used neural networks with two inputs derived from sensor data readings, two hidden layers with seven neurons each and two neurons as output layer [39].

III. Decentralized Neural Networks

In this paper, data propagation speed of single-machine parallelism is tested in neural network feed-forward process. With the different number of neurons in hidden layer Xtensa LX6 microprocessor has been done testing with it's one and two cores that was previously mentioned.

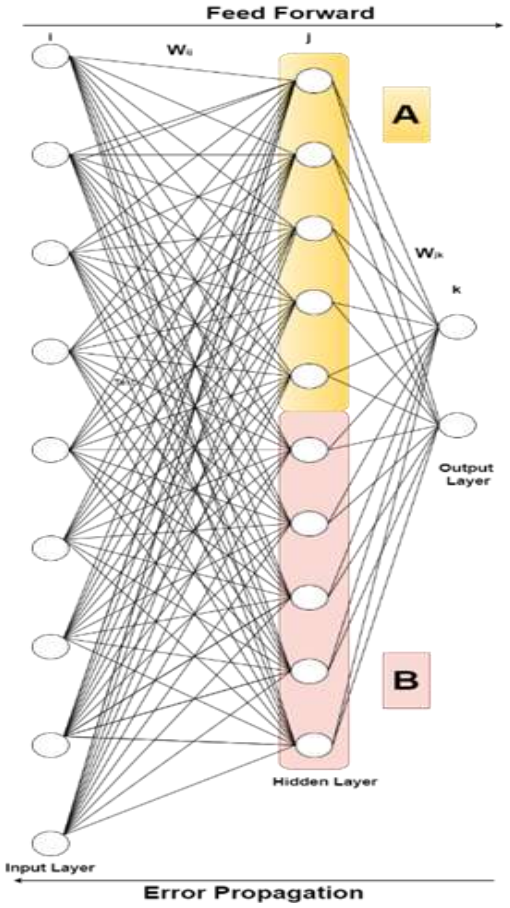

Fig. 3. Task Division of Hidden Layer

Data propagation will be divided into two same-size groups. Three simple method tasks division can be seen in the Fig. 3. For the upper half of neurons hidden layer (A) the first core will calculate values and for the lower half of neurons (B) another core will calculate values.

IV. The Propagation Speed of the Feed Forward Process

A. Hardware Preparation

For testing Xtensa LX6 microprocessor propagation speed of feed-forward neural network process with one or two cores, an ESP32 SoC has been used. In the market, there are various type of ESP32 boards are available and for the testing purpose ESP32-WROOM-32 board has been used that is produced by Espressif. This ESP32 has TLS for certificate-based mutual authentication and communication and trusted boot. Also includes several advanced features like encrypted flash.

B. Software Preparation

Collaboration between Amazon (Amazon Web Services) and Espressif (Espressif Systems Co.) it comes with FreeRTOS – light real-time operating system. Because of opensource there are various ways to program ESP32 boards.

Two main methods for programming ESP32:

a. ESP-IDF (Espressif IoT Development Framework)

b. Arduino IDE, ESP32 Arduino Core

Arduino IDE has some extra facilities compared with ESP-IDF low-level programming. In this paper, for testing Xtensa LX6 microprocessor Arduino IDE is used to compare propagation speed of feedforward neural network process.

C. Measuring the Execution time

There are various ways for measuring the execution time. In this paper micros() simple method is used. Code examples:

```
time_start = micros();
//some code
time_end = micros();
time = time_end – time_start;
```

Arduino IDE receive the values through serial monitor when micros() function returns the number of microseconds [40].

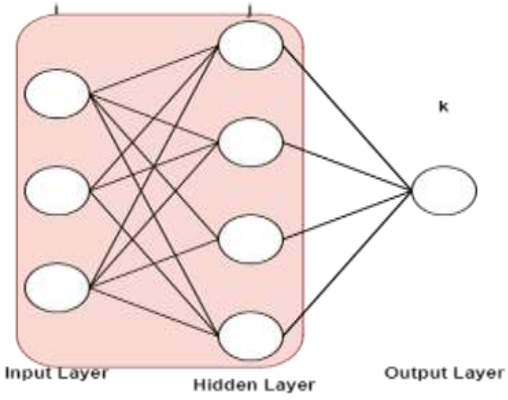

Fig. 4. Pre-activation Layer

In Fig. 4, it can be seen which part is being implemented. Actually, this part is time consuming to measure in neural networks with one hidden layer. For the fact, number of inputs are much higher than outputs in a common neural network. Expressed as following equation,

$$z = b + \sum_{i=1}^{N} a_i w_i \quad (1)$$

whereas, b = Bias, w_i = weights, z = a sum of all multiplications of input values a_i

D. Experiments and Results

Xtensa LX6 microprocessor on ESP32 board have two cores and for the first time only one core is used for measuring the propagation time. Describes in Table 1. The FreeRTOS have xTaskCreatePinnedToCore() function which is capable for setting the number of processor core for task. In ESP32 have only two cores and for the first part of taking measurements with only one core last argument in the function is set to 1, this argument can be "core 0" or "core 1".

TABLE I. RESULTS WITH SINGLE CORE IN XTENSA LX6 MICROPROCESSOR ON ESP32

No	Layer		Operations	Time
	Input	*Hidden*		
1	50	20	1000	383 μs
2		40	2000	681 μs
3		60	3000	966 μs
4		80	4000	1258 μs
5		100	5000	1582 μs
6		120	6000	1830 μs
7		140	7000	2203 μs
8		160	8000	2415 μs
9		180	9000	2815 μs
10		200	10000	3071 μs

Some additional setup and code are needed for measuring the propagation time in Xtensa LX6 microprocessor on ESP32. For single core, all tasks have been done in using only one core. But for using both cores, as we mentioned previously Xtensa LX6 microprocessor have two cores, all tasks have to be divided into two parts.

Table 2 describe about the results of using two cores in Xtensa LX6 microprocessor on ESP32. In this paper, for the simplicity all numbers of neurons in the hidden layer setups are even and they multiples by 20. Half of neuron are calculated in the first core (core 0) and another half is calculated in next core (core 1) and the Fig. 3 gives a crystal-clear idea about this process.

Previously mentioned function xTaskCreatePinnedToCore is called two times in this for selecting the both core setup and first core get the values of first half of neurons in the hidden layers when it called in first time. Assume that, there are 20 neurons in hidden layers and the written function will get values 0 to 9 in the first call and 10 to 19 values will take in second call. A global array is declared for define inputs and weights values, because every core need access on them.

Example Code:

```
xTaskCreatePinnedToCore(
powerTask,    /* Function to implement the task */
"powerTask",          /* Name of the task */
 1000,              /* Stack size in words */
(void*)&twoTasks1,    /* Task input parameter */
    20,               /* Priority of the task */
    NULL,             /* Task handle. */
    0);               /* Core where the task should run */
xTaskCreatePinnedToCore(
    powerTask,        /* Function to implement the task */
    "coreTask",       /* Name of the task */
    1000,             /* Stack size in words */
    (void*)&twoTasks2,    /* Task input parameter */
    20,               /* Priority of the task */
    NULL,             /* Task handle. */
    1);               /* Core where the task should run */
```

TABLE II. RESULTS WITH BOTH CORE IN XTENSA LX6 MICROPROCESSOR ON ESP32

No	Layer		Operations	Time
	Input	*Hidden*		
1	50	20	1000	253 μs
2		40	2000	406 μs
3		60	3000	547 μs
4		80	4000	699 μs
5		100	5000	858 μs
6		120	6000	976 μs
7		140	7000	1159 μs
8		160	8000	1268 μs
9		180	9000	1482 μs
10		200	10000	1599 μs

Elapsed time ration can be seen in Table 3. For the number of multiplication and addition operations on Xtensa LX6 microprocessor cores on ESP32 boards, it is obvious that ration is rising. If increasing number of operations has less than the impact is less.

TABLE III. COMPARISON RESULTS OF XTENSA LX6 MICROPROCESSOR CORES ON ESP32

No	Time		Ratio
	One Core	*Two Core*	
1	383 μs	253 μs	1.51
2	681 μs	406 μs	1.68
3	966 μs	547 μs	1.76
4	1258 μs	699 μs	1.80
5	1582 μs	858 μs	1.84
6	1830 μs	976 μs	1.87
7	2203 μs	1159 μs	1.90
8	2415 μs	1268 μs	1.90
9	2815 μs	1482 μs	1.90
10	3071 μs	1599 μs	1.92

V. DISCUSSION AND CONCLUSION

Xtensa LX6 microprocessor has two cores on ESP32. If speedup the original execution time divided by an enhanced execution time is the main theme of Amdahl's law [41]. In the recent decades, the modern version of Amdahl's law says that enhancing the fraction f of a computation by a speed up S , then we will get [41],

$$S_{speedup_{enhanced}}(f,S) = \frac{1}{(1-f)+\frac{f}{S}} \quad (2)$$

In the Fig. 5, we can see a chart of comparison results of Xtensa LX6 microprocessor cores on ESP32 from the Table 3.

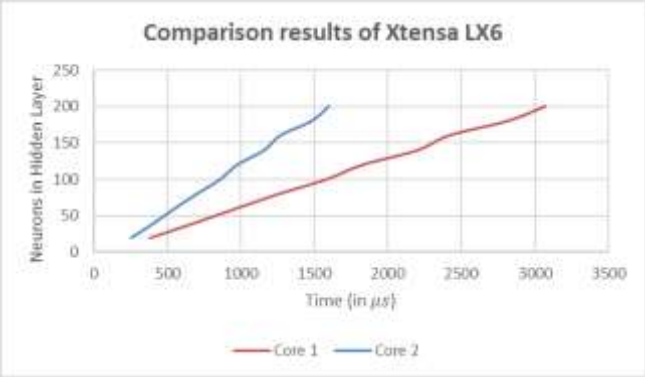

Fig. 5. Comparison results of Xtensa LX6 cores on ESP32

In this study, Xtensa LX6 microprocessor have two cores and from Amdahl's law perspective the result will be analyzed. So that we formulated the Amdahl's law as,

$$S_{latency}(S) = \frac{1}{(1-p)+\frac{p}{S}} \tag{3}$$

whereas, $S_{latency}$ – is speedup of code execution S is instead of value 2 because of Xtensa LX6 microprocessor multiple cores, and p – is proportion of code execution time.

According to this equation (3) in Table 4, the values for p are calculated.

TABLE IV. PROPORTION OF THE CODE EXECUTION TIME

Ratio	Number of Operations	p
1.51	1000	0.68
1.68	2000	0.81
1.76	3000	0.86
1.80	4000	0.89
1.84	5000	0.91
1.87	6000	0.93
1.90	7000	0.95
1.90	8000	0.95
1.90	9000	0.95
1.92	10000	0.96

It is noticeable in the Table 4, that the calculated values of p proportion of code execution time is increased and this is obvious because of fixed tasks for processor cores setup and data transfer between different types of memory have less and less impact on total run time [4].

REFERENCES

[1] J.X. Wang, Y.Z. Li, X.D. Liu, C.Q. Shen, H.S. Zhang, K. Xiong, Recent active thermal management technologies for the development of energy-optimized aerospace vehicles in China, Available online, Chin. J. Aeronaut. (2020), https:// doi.org/10.1016/j.cja.2020.06.021.

[2] B.J. Brelje, J. Martins, Electric, hybrid, and turboelectric fixed-wing aircraft: a review of concepts, models, and design approaches, Prog. Aero. Sci. 104 (2019) 1–19.

[3] A.R. Gnadt, R.L. Speth, J.S. Sabnis, S.R.H. Barrett, Technical and environmental assessment of all-electric 180-passenger commercial aircraft, Prog. Aero. Sci. 105 (2019) 1–30.

[4] Đokić, K., Martinović, M. and Radišić, B., "Neural networks with ESP32-are two heads faster than one?." In Conference on Data Science and Machine Learning Applications (CDMA 2020) (p. 141).

[5] Verma, Harsh, Siddharth Lotia, and Anurag Singh. "Convolutional Neural Network Based Criminal Detection." In 2020 IEEE REGION 10 CONFERENCE (TENCON), pp. 1124-1129. IEEE, 2020.

[6] Karamchandani, Sunil, and Ganesh Shukla. "Face Sketch-Image Recognition for Criminal Detection Using a GAN Architecture." International Conference on Information and Communication Technology for Intelligent Systems. Springer, Singapore, 2020.

[7] Q. K. Al-Shayea, "Artificial neural networks in medical diagnosis," International Journal of Computer Science Issues, vol. 8, pp. 150-154, 2011.

[8] F. Amato, A. López, E. M. Peña-Méndez, P. Vaňhara, A. Hampl and J. Havel, Artificial neural networks in medical diagnosis, Elsevier, 2013.

[9] D. C. Ciresan, U. Meier, J. Masci, L. M. Gambardella and J. Schmidhuber, "Flexible, high performance convolutional neural networks for image classification," 2011.

[10] Van Uytsel, Steven, and Danilo Vasconcellos Vargas. "Challenges for and with Autonomous Vehicles: An Introduction." Autonomous Vehicles. Springer, Singapore, 2021. 1-17.

[11] He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Deep residual learning for image recognition." In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 770-778. 2016.

[12] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M. and Berg, A.C., 2015. Imagenet large scale visual recognition challenge. International journal of computer vision, 115(3), pp.211-252.

[13] Zhao, Jianxin, Richard Mortier, Jon Crowcroft, and Liang Wang. "Privacy-preserving machine learning based data analytics on edge devices." In Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society, pp. 341-346. 2018.

[14] Khelifi, Hakima, Senlin Luo, Boubakr Nour, Akrem Sellami, Hassine Moungla, Syed Hassan Ahmed, and Mohsen Guizani. "Bringing deep learning at the edge of information-centric internet of things." IEEE Communications Letters 23, no. 1 (2018): 52-55.

[15] Wang, Xiaofei, Yiwen Han, Victor CM Leung, Dusit Niyato, Xueqiang Yan, and Xu Chen. "Convergence of edge computing and deep learning: A comprehensive survey." IEEE Communications Surveys & Tutorials 22, no. 2 (2020): 869-904.

[16] Aazam, Mohammad, and Eui-Nam Huh. "Fog computing micro datacenter based dynamic resource estimation and pricing model for IoT." In 2015 IEEE 29th International Conference on Advanced Information Networking and Applications, pp. 687-694. IEEE, 2015.

[17] G. Hegde, N. Ramasamy and N. Kapre, "CaffePresso: an optimized library for deep learning on embedded accelerator-based platforms," 2016.

[18] Y. Zhu, M. Mattina and P. Whatmough, "Mobile machine learning hardware at arm: a systems-on-chip (soc) perspective," arXiv preprint arXiv:1801.06274, 2018.

[19] J. Tang, D. Sun, S. Liu and J.-L. Gaudiot, "Enabling deep learning on IoT devices," Computer, vol. 50, pp. 92-96, 2017.

[20] A. Shawahna, S. M. Sait and A. El-Maleh, "FPGA-Based Accelerators of Deep Learning Networks for Learning and Classification: A Review," IEEE Access, vol. 7, pp. 7823-7859, 2019.

[21] Espressif, "Espressif IoT Development Framework, ESP-IDF" Available: https://docs.espressif.com/projects/esp-idf/en/latest/esp32/ [Accessed on January 3, 2021]

[22] Espressif, "ESP-WHO is a face detection and recognition platform that is currently based on Espressif Systems' ESP32 chip." Available: https://github.com/espressif/esp-who [Accessed on January 3, 2021]

[23] Abdullah, A. H., Sudin, S., Ajit, M. I. M., Saad, F. S. A., Kamaruddin, K., Ghazali, F., ... & Bakar, M. A. A. (2018, August). Development of ESP32-based Wi-Fi electronic nose system for monitoring LPG leakage at gas cylinder refurbish plant. In 2018 International Conference on Computational Approach in Smart Systems Design and Applications (ICASSDA) (pp. 1-5). IEEE.

[24] Babiuch, Marek, and Jiri Postulka. "Smart Home Monitoring System Using ESP32 Microcontrollers." In Internet of Things. IntechOpen, 2020.

[25] Fuadi, Soni Asmaul, and Ario Dwi Prabowo. "Real-Time Measurement of Solar Cell's Current And Voltage to Predict Battery Capacity Using Iot Technology with ESP 32 Microcontroller." Solid State Technology (2020): 8623-8628.

[26] Asra Noorain, F., Jibin Raju, V. Varsha, and H. G. Nanditha. "An IoT Based Approach To Minimize And Monitor Air Pollution Using ESP32 and Blynk Platform."

[27] Cadence Tensilica Xtensa LX6 dataplane processing units (DPUs), https://mirrobo.ru/wp-content/uploads/2016/11/Cadence_Tensillica_Xtensa_LX6_ds.pdf[Accessed on January 3, 2021]

[28] Chand, Gavin, Mustafa Ali, Bashar Barmada, Veronica Liesaputra, and Guillermo Ramirez-Prado. "Tracking a person's behaviour in a smart house." In International Conference on Service-Oriented Computing, pp. 241-252. Springer, Cham, 2018.

[29] FERNOAGA, V. P., STELEA, G. A., BALAN, A., & SANDU, F. (2018, October). OCR-based solution for the integration of legacy and-or non-electric counters in cloud smart grids. In 2018 IEEE 24th International Symposium for Design and Technology in Electronic Packaging (SIITME) (pp. 398-403). IEEE.

[30] Rosato, D., Masciadri, A., Comai, S., & Salice, F. (2018, November). Non-invasive monitoring system to detect sitting people. In Proceedings of the 4th EAI International Conference on Smart Objects and Technologies for Social Good (pp. 261-264).

[31] Chowdhuryy, Md Hafizul Islam, Maliha Sultana, Rajib Ghosh, Jamal Uddin Ahamed, and M. A. I. Mahmood. "AI assisted portable ECG for fast and patient specific diagnosis." In 2018 International Conference on Computer, Communication, Chemical, Material and Electronic Engineering (IC4ME2), pp. 1-4. IEEE, 2018.

[32] Komarek, A., Pavlik, J., Mercl, L., & Sobeslav, V. (2017, September). Hardware layer of ambient intelligence environment implementation. In International Conference on Computational Collective Intelligence (pp. 325-334). Springer, Cham.

[33] Židek, K., Janáčová, D., Piteľ, J., Hošovský, A., & Lazorík, P. (2018, December). Data optimization for communication between wireless IoT devices and Cloud platforms in production process. In MMS 2018: 3rd EAI International Conference on Management of Manufacturing Systems (p. 118). European Alliance for Innovation.

[34] Kokoulin, A. N., Tur, A. I., Yuzhakov, A. A., & Knyazev, A. I. (2019, January). Hierarchical convolutional neural network architecture in distributed facial recognition system. In 2019 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus) (pp. 258-262). IEEE.

[35] Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L. C. (2018). Mobilenetv2: Inverted residuals and linear bottlenecks. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 4510-4520).

[36] Ortega-Zamorano, F., Jerez, J. M., Subirats, J. L., Molina, I., & Franco, L. (2014). Smart sensor/actuator node reprogramming in changing environments using a neural network model. Engineering Applications of Artificial Intelligence, 30, 179-188.

[37] Zhang, K., Zhang, Z., Li, Z., & Qiao, Y. (2016). Joint face detection and alignment using multitask cascaded convolutional networks. IEEE Signal Processing Letters, 23(10), 1499-1503.

[38] Rai, N., & Rai, B. (2013). Neural network based closed loop speed control of DC motor using arduino uno. International Journal of Engineering Trends and Technology, 4(2), 137-140.

[39] Adhitya, R. Y., Ramadhan, M. A., Kautsar, S., Rinanto, N., Sarena, S. T., Munadhif, I., ... & Soeprijanto, A. (2016, November). Comparison methods of Fuzzy Logic Control and Feed Forward Neural Network in automatic operating temperature and humidity control system (Oyster Mushroom Farm House) using microcontroller. In 2016 International Symposium on Electronics and Smart Devices (ISESD) (pp. 168-173). IEEE.

[40] Arduino, "micros()," Arduino, [Online]. Available: https://www.arduino.cc/reference/tr/language/functions/time/micros/. [Accessed February 28, 2021]

[41] Hill, Mark D., and Michael R. Marty. "Amdahl's law in the multicore era." Computer 41, no. 7 (2008): 33-38.

[42] Amdahl, Gene M. "Validity of the single processor approach to achieving large scale computing capabilities." In Proceedings of the April 18-20, 1967, spring joint computer conference, pp. 483-485. 1967.