Agricultural Research
https://doi.org/10.1007/s40003-024-00788-6

FULL-LENGTH RESEARCH ARTICLE

N-AquaRAM: A Cost-Efficient Deep Learning Accelerator for Real-Time Aquaponic Monitoring

Ali Siddique [1] · Muhammad Azhar Iqbal[2] · Jingqi Sun [3] · Xu Zhang [2] · Mang I. Vai [4] · Sunbal Siddique[5]

Received: 22 March 2024 / Accepted: 30 August 2024

Abstract
Aquaponics is an emerging area of agricultural sciences that combines aquaculture and hydroponics in a symbiotic way to increase crop production. Though it offers a lot of advantages over traditional techniques, including chemical-free and soil-less farming, its commercial application suffers from some problems such as the lack of experienced manpower. To operate a stable smart aquaponic system, it is critical to estimate the fish size properly. In this context, the use of dedicated hardware for real-time aquaponic monitoring can greatly resolve the issue of inexperienced handlers. In this article, we present a complete methodology to train a deep neural network to perform fish size estimation in real time. To achieve high accuracy, a novel implementation of swish function is presented. This novel version is far more hardware efficient than the original one, while being extremely accurate. Moreover, we present a deep learning accelerator that can classify 40 million fish samples in a second. The dedicated real-time system is about 1600 times faster than the one based on general-purpose computers. The proposed neuromorphic accelerator consumes about 2600 slice registers on a low-end model of Virtex 6 FPGA series.

Keywords Aquaculture · Deep learning accelerator · Field programmable gate arrays (FPGAs) · Fish size estimation · Giga operations per second (GOPS) · Smart aquaponics

Introduction

The rapid increase in human population and the associated shortage of food necessitates the development of advanced agricultural techniques and solutions. Aquaponics is a recently-developed agricultural technique that combines aquaculture and hydroponics in order to resolve the issue of food crisis. In an aquaponic system, fish excrete their waste that is transformed into nutrients by nitrifying bacteria, which in turn are readily absorbed by plants. This symbiotic relationship is shown in Fig. 1. A typical smart aquaponic system (SAS) takes input from sensors which are responsible for collecting data from the external environment. These sensors are generally responsible for monitoring the pH level, the dissolved oxygen, and a lot of other parameters taken in and out of the system [34].

Compared with other farming techniques, aquaponics is considered relatively harmless, since it limits the use of dangerous chemicals [38]. Moreover, aquaponics promotes soil-less culture and resolves the issue of water scarcity to a great extent [5, 38]. This is because aquaponic systems consume only 2%-10% of the water consumed by traditional agricultural systems [5, 38]. Despite all these advantages, only 31% of the aquaponic solutions have been found to be commercially viable due to poor management and inexperienced manpower/handlers [38]. For monitoring and controlling the nutrients in an aquaponic system, deep learning has proven its mettle. Deep learning (DL)

✉ Muhammad Azhar Iqbal
m.a.iqbal1@leeds.ac.uk

[1] State Key Lab of Analog and Mixed Signal VLSI, University of Macau, Taipa 999078, Macau

[2] Faculty of Engineering and Physical Sciences, University of Leeds, Leeds LS2-9JT, United Kingdom

[3] Department of Computer Science, Beijing Jiaotong University, Weihai 264003, China

[4] Department of Electrical and Computer Engineering, Faculty of Science and Technology, University of Macau, Taipa 999078, Macau

[5] Department of Computer Science, University of Agriculture, Faisalabad 38000, Pakistan

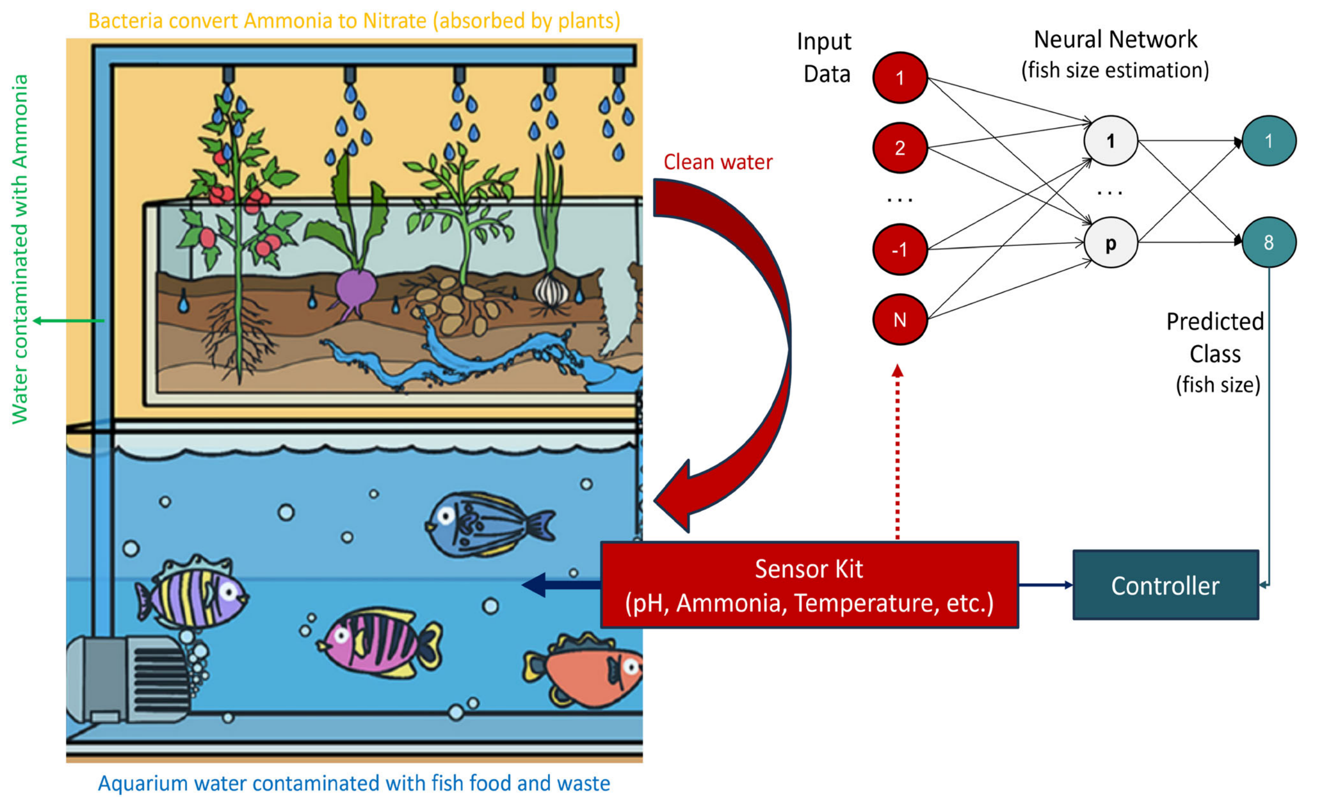
Bacteria convert Ammonia to Nitrate (absorbed by plants)
Water contaminated with Ammonia
Clean water
Aquarium water contaminated with fish food and waste
Input Data
Neural Network (fish size estimation)
Predicted Class (fish size)
Sensor Kit (pH, Ammonia, Temperature, etc.)
Controller

Fig. 1 A typical smart aquaponic system using deep learning for feedback [34]

can predict various parameters such as fish size and water quality at an early stage, and can help in automatic system adjustment and upgradation. For example, if a fish is smaller than the expected size, it can be provided with more food and healthy nutrition. In this context, various deep learning models have been proposed to predict water quality parameters, fish classification, fish size estimation, feeding decisions, etc. [10, 11, 19, 38, and 42]. Since, nutrient deficiency can seriously affect the growth of plants and fishes, it is of critical importance to provide the system with essential nutrients and supplements. To get a better understanding of how different elements/supplements such as potassium and iron can contribute to the overall improvement in aquaculture, the reader is referred to [6, 7, 9, 12, 21, and 22].

Aquaponic systems sometimes involve an enormously large number of environmental parameters that have to be monitored and controlled all the time, which makes such systems quite difficult to be handled by inexperienced humans. This issue of poor management can greatly be resolved if smart automation techniques are adopted. Such techniques and systems reduce the need for a huge manpower and allow better management at a commercial scale. Smart systems (SSs) in the context of aquaponic systems refer to small but intelligent electronic devices capable of performing a huge number of complex operations such as sensing, monitoring, and control in a minimal amount of time [38]. This type of automated decision making can greatly assist in overcoming challenges associated with the lack of experienced manpower.

Though commendable work has been carried out in the development of smart deep learning systems for aquaponics, a major problem surrounding all these systems is that they are based on software, microcontrollers, or big computers based on Von-Neumann architecture. Since, software tools and Von Neumann architectures typically follow a serial model of execution, their speed is extremely slow, which limits their use in large-scale commercial aquaponic systems. For example, in commercial next-generation aquaponic systems, hundreds of parameters have to be monitored, controlled, and maneuvered simultaneously in order to achieve a balance in the system and to provide appropriate amount of nutrients to the plants and fishes. As a result, software-based systems would typically fail to appropriately do all these tasks in real time (at a high speed). Moreover, software tools have to be run on big computers and machines which occupy a large area and are difficult to manage. This is where the role of dedicated hardware based on field programmable gate arrays (FPGAs) and application-specific integrated circuits (ASICs) comes into play. Dedicated hardware systems have a much smaller footprint and consume lower power than microcontrollers and computers. As shown in Sect. 5, the proposed system is 1600 times faster than a typical CPU-based SAS. FPGAs and ASICs offer greater speed and parallelism than central processing units (CPUs), since they contain highly parallel data computing units.

Another big problem in most modern studies is the lack of available data. Most datasets have a few hundred samples for system evaluation, which is insufficient to obtain reliable results. For example, the work in [5] uses only 211 samples for performance evaluation.

Keeping all these issues in mind, we propose a novel, high-speed and small-footprint smart aquaponic system based on deep learning for fish size estimation. The system can monitor and predict the size of fishes in real time; it can classify a given input sample into one of the eight classes defined with respect to weight. The throughput of the system is 40 million samples per second, i.e., the system can classify 40 million samples in a second.

Materials and methods

This paper presents a novel low-cost, high-throughput, hardware-based aquaponic monitoring engine (which uses the proposed algorithm), capable of classifying 40 million fish samples a second based on fish sizes. Classification results can be leveraged by experts in the field of agriculture to make further decisions when needed. The proposed design has been described in Verilog language at the register-transfer level (RTL). The main contributions of this work are mentioned below. A summary of these contributions is given in Fig. 2.

1. To provide a complete methodology to develop an aquaponic monitoring system that uses deep learning to indicate the appropriate fish weight category. If the actual fish weight falls outside the estimated weight range, it means the aquaponic ecosystem needs appropriate modification. The system uses 175,000 samples to train/test the system, which ensure its operation and stability. This sample size is way bigger than the sizes used in most modern studies.
2. Proposal of a novel form of Swish neuron. The proposed 'P-Swish' neuron is not only morphologically similar to *swish*, but also yields a similar level of accuracy. Moreover, the proposed P-Swish neuron is way more hardware efficient than other swish implementations, while being extremely accurate. In fact, the accuracy of P-Swish is better than Swish by about 0.4% (under the specified conditions) on CIFAR-10 dataset.
3. In order to provide greater flexibility to the user, the system reconfigurable neural layers. These layers allow multiple types of neurons and the user can choose any of those in order to achieve high classification accuracy. The edge computer achieves around 99.6% accuracy, which is higher than that achieved by any other modern DL-based aquaponic system.
4. A novel, real-time aquaponic monitoring system implemented on a field programmable gate array (FPGA). It is an edge computer capable of predicting fish size (weight) on the basis of input parameters. The proposed edge computer can predict 8 classes (based on fish weight) using the given data. The throughput of the system is 40 million samples a second. The throughput is about 1600 times higher than a typical CPU-based software system, which makes it suitable for use in commercial or semi-commercial settings.

The rest of this paper is organized as follows. Section 2 presents a review of various modern smart aquaponic systems and neuromorphic accelerators. It also presents the problem definition. Section 3 presents the training methodology for fish size estimation. Section 4 presents the proposed smart edge computer, i.e., N-AquaRAM. The

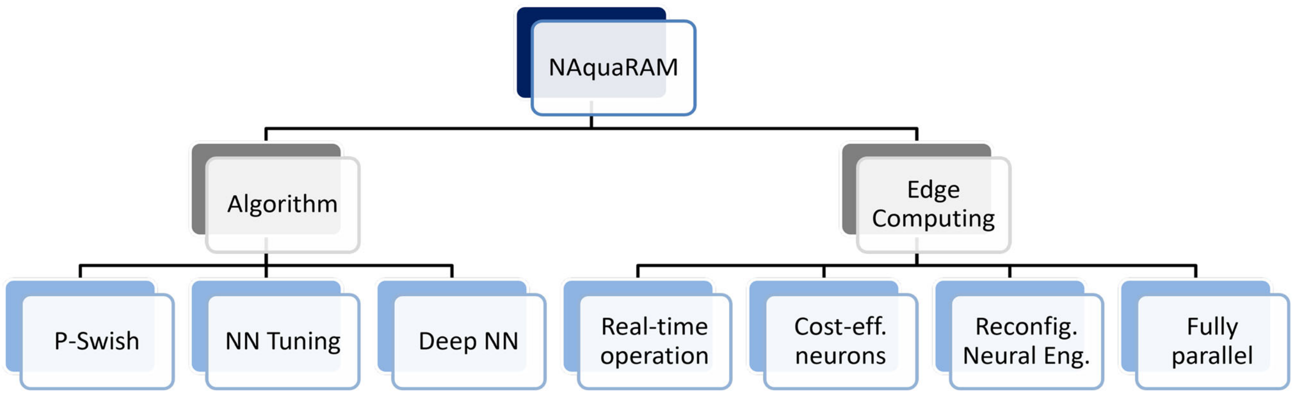

Fig. 2 Proposed system: features and components

results are shown in Sect. 5. Finally, Sect. 6 concludes the work.

Related Work and Problem Definition

Most (in fact, all) of the smart aquaponic systems (SASs) are built using large computers and/or microcontrollers (MCs) [38]. No electronic system or dedicated hardware chip (built using field programmable gate arrays or application-specific integrated circuits) has been designed keeping aquaponics in view. This is due to the fact that the development of SASs is a relatively new area of research. Since, MCs are not very powerful and are not designed for a specific application, they are extremely slow [3, 26]. Not only that, they consume a lot of power and are extremely expensive when manufactured in bulk quantity. In order to build cost-effective dedicated SASs, it is important to focus on both algorithms and customized hardware systems [32, 36]. The efficiency of a hardware system depends critically on the algorithm, as shown in [32, 33, and 36]. This is why we focus on devising hardware-friendly algorithms and schemes.

Since, the proposed scheme N-AquaRAM deals with both algorithm and architecture, we divide this section into two major parts. The first subsection describes various smart aquaponic systems and the second subsection discusses various DL accelerators. After extensive literature review, we conclude that none of the accelerators is specifically built for aquaponics.

Deep Learning in Smart Aquaponic Monitoring Systems

Due to their excellent generalization ability, various researchers have used deep learning to develop smart aquaponic modules. Some of the many applications of DL to solving aquaponic problems are: fish size estimation, fish detection, prediction of water quality, making feeding decisions, and plant disease detection. The dynamics of fish length distribution is a key input for understanding the fish population dynamics and taking informed management decisions on exploited stocks. Nevertheless, the length of landed fish is still (mostly) made by hand. As a result, length estimation is precise at fish level, but due to the inherent high costs of manual sampling, the sample size tends to be small.

In [19], for example, the researchers use multiple cameras and apply convolutional neural networks (CNNs) to estimate the length of pond fish. The accuracy they achieve is around 93%. The estimation of fish length and weight is important in order to properly manage aquaponic systems and to model stock trends. Similarly, in [13], the authors make a comparison between various popular CNN topologies and models such as ResNet50 and VGG19 to estimate the mass of Pintado real fingerlings. They conclude that ResNet50 performs the best for the application and the dataset under consideration.

Some researchers have experimented with videos as well. The authors in [11] present a framework for the automatic detection of fish in underwater videos. The accuracy they achieve is around 95.47% using ResNet-50 model. The authors in [11] propose a system that can detect moving live fish in open aquatic environments with about 87.44% accuracy.

The diagnosis of fish diseases is also very important for developing a healthy aquaponic ecosystem. If a fish is found to be unhealthy, it can either be removed from the system or can be provided with better nutrients, else it would spoil the whole system. In this context, the authors in [10] propose a system that can diagnose white and red spots in a fish with about 94.44% accuracy. In [1], the authors present an image-based machine learning technique to detect diseased fishes in aquaculture. A survey on using intelligent techniques to diagnose fish diseases is presented in [16].

DL can also be used to predict and eventually control various chemicals and nutrients, moving in and out of the system. For example, the concentration of oxygen in aquaponic systems is predicted by the authors using intelligent schemes in [29]. The optimal level of oxygen should always be present in the system for healthy operation and the early prediction of oxygen levels in the system can ensure a stable ecosystem. Other parameters such as the pH level, ammonia, and temperature may also be predicted or manipulated using deep neural networks (DNNs) for optimal aquaponic operation. Smart aquaponic systems may also be used to make appropriate feeding decisions for fishes and/or plants [42].

Neuromorphic Accelerators

In [30], the authors present a hardware design that predicts multiple types of epileptic seizures with 95.14% accuracy. Similarly, a simple neural network having 4-5 synapses is designed by researchers in [31]. The network has Gaussian neurons. The authors, however, do not test their system on any dataset. A system for digit classification is presented in [32]. The system is quite efficient, since it uses ReLU/identity function at all the layers. The system presented in [33] is able to predict cancer with more than 98.23% accuracy. The system can classify more than 63 million samples in one second. The learning engine presented in [36] can train a neural network for any type of application.

In [34], though the authors present a dedicated hardware system for spike-based smart aquaponic monitoring, the system can handle only eight input features. The system in [35] implements the Tempotron learning rule for SNNs. The design in [35] generally takes about four input features that are then mapped to 48 Gaussian Receptive Fields (GRFs) for making the prediction. Whether, the network can take tens of input features–as is required in case of smart aquaponics–is yet to be seen.

In [15], the authors compare GPU- and FPGA-based implementations for weed classification tasks. As per results, the FPGA implementation is about 2.86× faster and about seven× more power efficient than GPU implementation. For efficient hardware processing, they downsample their input data and use binary weights. Moreover, they make input images clearer by changing contrast and brightness. Their system is about 98.83% accurate.

In [17], the authors present two novel methods for weight initialization (WI) and batch normalization (BN) in complex binarized neural networks (BNNs). However, the system is not optimized for agriculture/aquaponics. In [18], the authors present a network pruning scheme to meet computational demands and achieve up to 26.4× compression as compared to the best available network with a minimal loss of accuracy. Similarly, to reduce memory and computational requirements, the authors in [25] use ternary values for both weights and activations. However, the systems in [18, 25] have been designed and tested for image classification tasks only.

Problem Definition

Most of the smart aquaponic systems are based on software only and are implemented on a general-purpose computers or microcontrollers, which is why they have a high implementation cost and low speed. Moreover, they use small datasets or achieve low accuracy. Therefore, the goal is to design a smart aquaponic monitoring system that is cheap and highly accurate.

Keeping all these issues in view, we first present a complete deep learning scheme that predicts fish size with 99.6% accuracy. We then present a high-speed, cost-efficient hardware design that can be implemented on a dedicated hardware platform (FPGA/ASIC). To achieve high accuracy, the proposed design has reconfigurable network layers that allow multiple types of neurons. This system will assist professionals in making critical feeding decisions and in maintaining a stable SAS.

Proposed Aquaponic Monitoring Scheme

We develop and synthesize a neural network for estimating fish weight. The fish weight estimation results can then be sent to a controller that makes feeding decisions. The network has four layers: one input layer, two hidden layers, and an output layer. The input layer has been normalized according to the procedure described in [33].

Neuronal Models

Here, we present the **proposed swish** implementation, i.e., P-Swish. We also give details of an existing neuronal model *thresholded ReLU* (T-ReLU), which is a flexible implementation of the ReLU neuron [40]. The details are given below.

P-Swish

P-Swish is morphologically similar to the Swish function presented in [28]. According to multiple experiments, Swish performs better than ReLU [28]. However, Swish is extremely costly to compute, since it contains a lot of complex terms. The derivative of Swish is even more complex, which makes it unsuitable for hardware implementations. The proposed Swish implementation *P-Swish* can be implemented using an adder, a shifter, and a multiplier. P-Swish is expressed mathematically in Eq. 1, and is shown visually in Fig. 3. The derivative of P-Swish is given in Eq. 2.

$$A_j = \begin{cases} Z_j & Z_j \geq 2 \\ Z_j\left[max[0, (0.25Z_j + 0.5)]\right] & otherwise \end{cases} \tag{1}$$

$$\frac{\partial A_j^{\text{P-Swish}}}{\partial Z_j} = \begin{cases} 0.5(Z_j + 1) & -2 < Z_j < 2 \\ 1 & Z_j \geq 2 \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

T-ReLU

The T-ReLU model is a modified form of the *Flexible ReLU* (F-ReLU) function. F-ReLU is presented in [27]. The F-ReLU model is shown in Eq. 3.

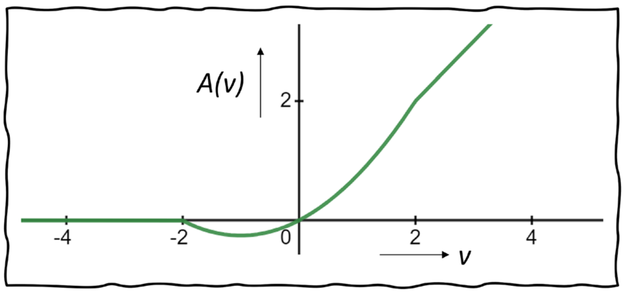

Fig. 3 Proposed swish (P-swish)

$$A_j = \begin{cases} Z_j + b_l & Z_j \geq 0 \\ b_l & Z_j \leq 0 \end{cases} \tag{3}$$

As can be seen in Eq. 3, the neuron produces a valid, nonzero value most of the time, regardless of the condition. As per reports, if a neuron remains active all the time, it can consume a large amount of hardware energy [23].

T-ReLU is a function that can overcome some of the shortcomings of F-ReLU. T-ReLU is given in Eq. 4. It can be seen from Eq. 4 that the neuron produces a non-zero value only if the weighted sum crosses a threshold. This asynchronous behavior can be used to put various neurons to sleep mode and save energy. The T-ReLU function can achieve high accuracy, since it is quite flexible and can change its axis if the parameter α is maneuvered. Moreover, the T-ReLU neuron, just like ReLU, requires only a comparator for activation. Therefore, T-ReLU is as hardware efficient as ReLU. The T-ReLU neuron is shown in Fig. 4b.

$$A_j = \begin{cases} Z_j & Z_j \geq \alpha \\ 0 & Z_j \leq \alpha \end{cases} \tag{4}$$

Network Structure

The complete network has four layers: one for inputs, one for outputs, and two hidden layers for nonlinear computations. In this article, the i^{th} weight and input are denoted by $\boldsymbol{W_i}$ and $\boldsymbol{X_i}$, respectively. The subscript j is for the j^{th} postsynaptic neuron. The letter b represents the bias, and the letter Z represents the weighted sum added to a bias. The un-activated value of a neuron j is represented by Z_j.

$$Z_j = \sum_i (W_i \cdot X_i) + b_j \tag{5}$$

Hidden Layer 1 (HL1)

The incoming weighted sum Z_1 is passed through an actuator. For the first hidden layer, T-ReLU is used for activation.

$$A_1 = \begin{cases} Z_1 & Z_1 \geq \alpha \\ 0 & Z_1 \leq \alpha \end{cases} \tag{6}$$

Hidden Layer 2 (HL2)

The Layer 1 activation vector is then passed as input to Layer 2 in order to obtain the weighted sum Z_2, as shown in Eq. 7. For HL2, there are four actuators available: identity, linear sigmoid (LiSi), P-Swish, and T-ReLU. The customized LiSi is defined in Eq. 8; the definition has been borrowed from [36]. The hard sigmoid defined in [39] is not used, since it involves a multiplier. To allow multiple neurons in a layer can greatly improve accuracy, since one type of neurons cannot be suitable for all types of input data.

$$Z_2 = \sum_i (W_i \cdot A_1) + b_2 \tag{7}$$

$$A_j = \begin{cases} 1 & Z_j \geq 2 \\ max[0, (0.25Z_j + 0.5)] & otherwise \end{cases} \tag{8}$$

Output Layer

The output layer has two types of actuators: identity and HW-efficient sigmoid (HES), proposed in [43]. This is because the use of sigmoid can be beneficial in case binary classification is to be performed or the input data distribution suits sigmoid. The HES neurons in the output layer are activated according to Eq. 9. The complete process is shown in the form of a flowchart in Fig. 5.

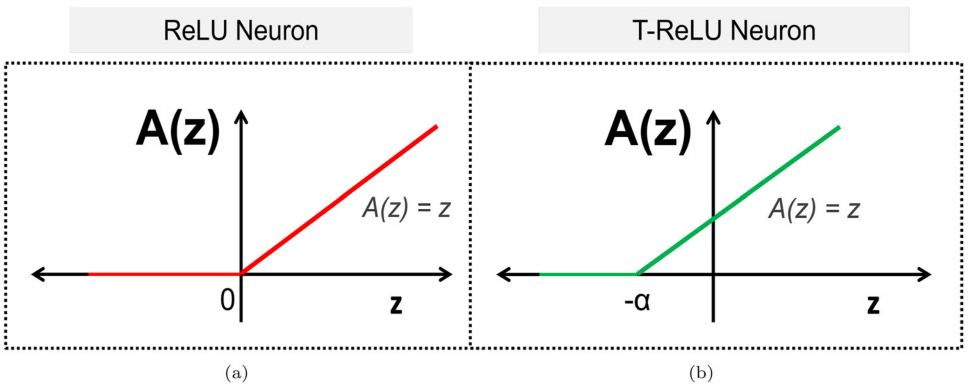

Fig. 4 Various ReLU functions **a** Original ReLU **b** Thresholded ReLU (T-ReLU)

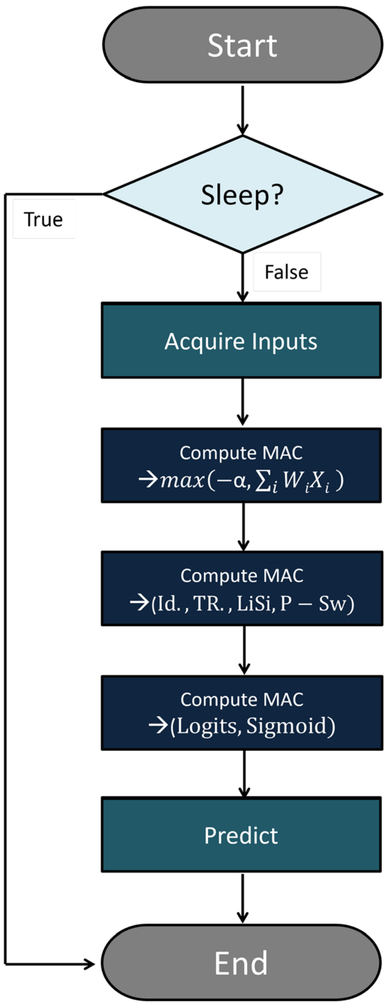
Start
Sleep?
True
False
Acquire Inputs
Compute MAC $\rightarrow max(-\alpha, \sum_i W_i X_i)$
Compute MAC $\rightarrow$(Id., TR., LiSi, P – Sw)
Compute MAC $\rightarrow$(Logits, Sigmoid)
Predict
End

Fig. 5 Flowchart of the operation

$$A_3 = \begin{cases} 1 & Z_3 > 2 \\ \left[Z_3 - 0.25(Z_3^2)\right]0.5 + 0.5 & 0 \le Z_3 \le 2 \\ \left[Z_3 + 0.25(Z_3^2)\right]0.5 + 0.5 & 0 \le Z_3 \le 2 \\ 0 & Z_3 < -2 \end{cases} \quad (9)$$

Proposed Real-Time Aquaponic Monitor

The proposed hardware system can handle up to 30 input features coming from various sensors such as pH sensor and oxygen sensor. The system is fully parallel and can predict eight levels of weight (based on the input data) in a single clock cycle. The system consists of an input layer, two reconfigurable hidden layer aquaponic computers, a reconfigurable output layer computer, and a predictor. There are three weight memories, one for each layer. The top level diagram of the complete system is shown in Fig. 6. It is to be kept in mind that the dataset under consideration has only eight features, but the hardware design can accommodate up to 30 features. This is to allow the use of more features for better accuracy and more intelligent design in future.

Structure of Memory Banks

Every layer in the design has a weight memory, divided into banks. A memory bank in the weight memory stores all weights corresponding to a neuron in the immediate layer. Every memory element in a memory bank is multiplied by a neuronal value coming from the preceding layer.

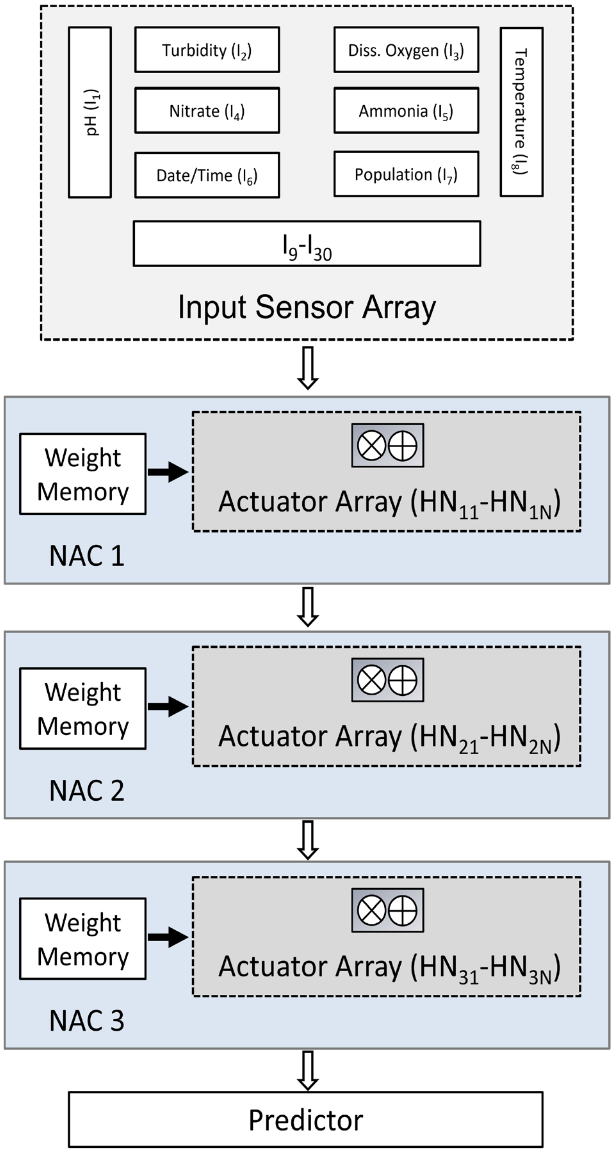

Fig. 6 Complete structure of the N-AquaRAM

In Fig. 7, the i^{th} element of the preceding layer is denoted by A_{0i}, where the subscript i is an integer value. Similarly, the i^{th} neuron in the next layer is represented by HN_i.

NeuroAqua Computers (NACs) - Hidden Layer 1

The full-resolution inputs coming from the preceding layer are multiplied by the corresponding weights. In case of first hidden layer, inputs are normalized values of the features obtained from external sensors.

The products are then summed up using an adder tree and the final sum is applied as input to the actuator, which is responsible for activating a neuron. The structure of a hardware T-ReLU is shown in Fig. 8. In Fig. 8, the comparator 'CMP' is responsible for comparing the incoming voltage value against a threshold. If the voltage is greater than the threshold, the voltage sum is passed on to the subsequent NAC, else zero voltage is passed.

NeuroAqua Computers - Hidden Layer 2

The actuated values coming from the first layer are multiplied by the corresponding weights. In case of first hidden layer, inputs are normalized values of the features obtained from external sensors.

The products are then summed up using an adder tree and the final sum is applied as input to the reconfigurable actuator (RA), whose structure is shown in Fig. 9. The activated values are then sent to the subsequent NAC. The selection of a particular neuron is made by the *Neuron Selector* (NS), input by the user. The NS is a 2-bit input used to select a neuron among the four available options: identity function, proposed swish (P-Swish), ReLU, and the linear sigmoid (LiSi).

Since, the distribution of data varies from application to application, the use of RA makes the system more flexible. For example, in some cases, the use of ReLU results in a lot of dead neurons, since it completely cancels out the negative input region [20]. In such cases, the use of a swish-

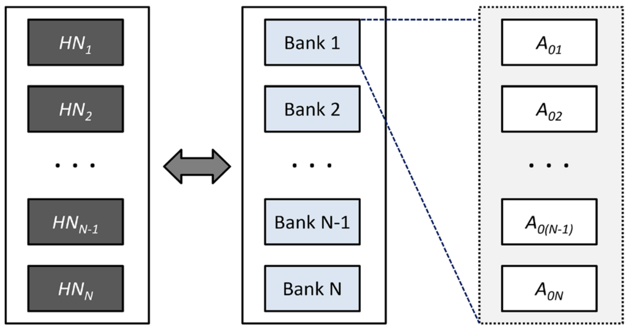

Fig. 7 Structure of memory banks

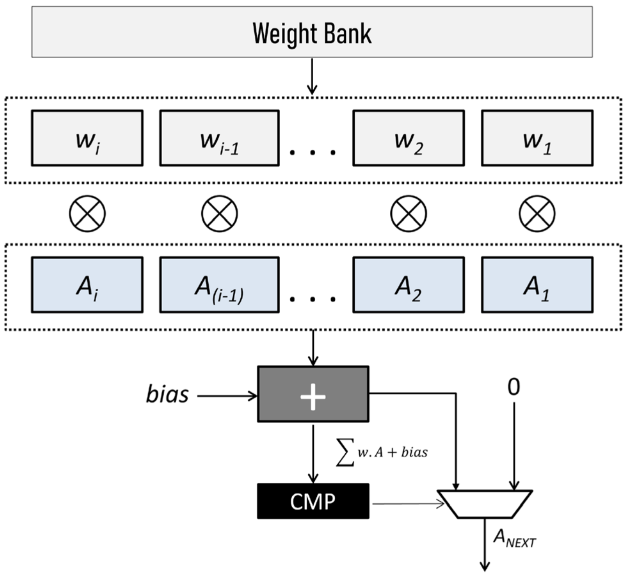

Fig. 8 Internal structure of an HL1 (T-ReLU) actuator

like function can help keeps neurons from becoming completely dead.

Since the reconfigurable actuator is quite complex, it is not used in the first layer of the proposed engine. This is because its use in the early layers increases critical path delay and makes it difficult for the subsequent layers to handle heavy amounts of data. The production of heavy data in the first layer means that the subsequent layers should have high data bandwidth, otherwise they would not be able to handle anything and data would be lost. The loss of data implies an unstable system and degradation of accuracy.

Interestingly, the data produced by LiSi can be used to create P-Swish and no special hardware is required as such. The output produced by LiSi is simply multiplied by the incoming voltage to create P-Swish. This reusability of data and components makes the proposed system more hardware-friendly. This type of reusability is not possible if the neurons proposed in [43] are used: the authors in [43] proposed two different models for Swish and Sigmoid and hence, require many extra components to be realized on the same chip.

Output-Layer NeuroAqua Computers

The structure of output-layer NACs is shown in Fig. 10. The full-resolution inputs coming from the preceding layer are multiplied by the weights corresponding to output neurons. There are two types of actuators available in the output layer: *identity* actuators (IAs) and *sigmoidal actuators* (SAs). Sigmoidal actuators are suitable for binary classification and the IAs are used for multi-class

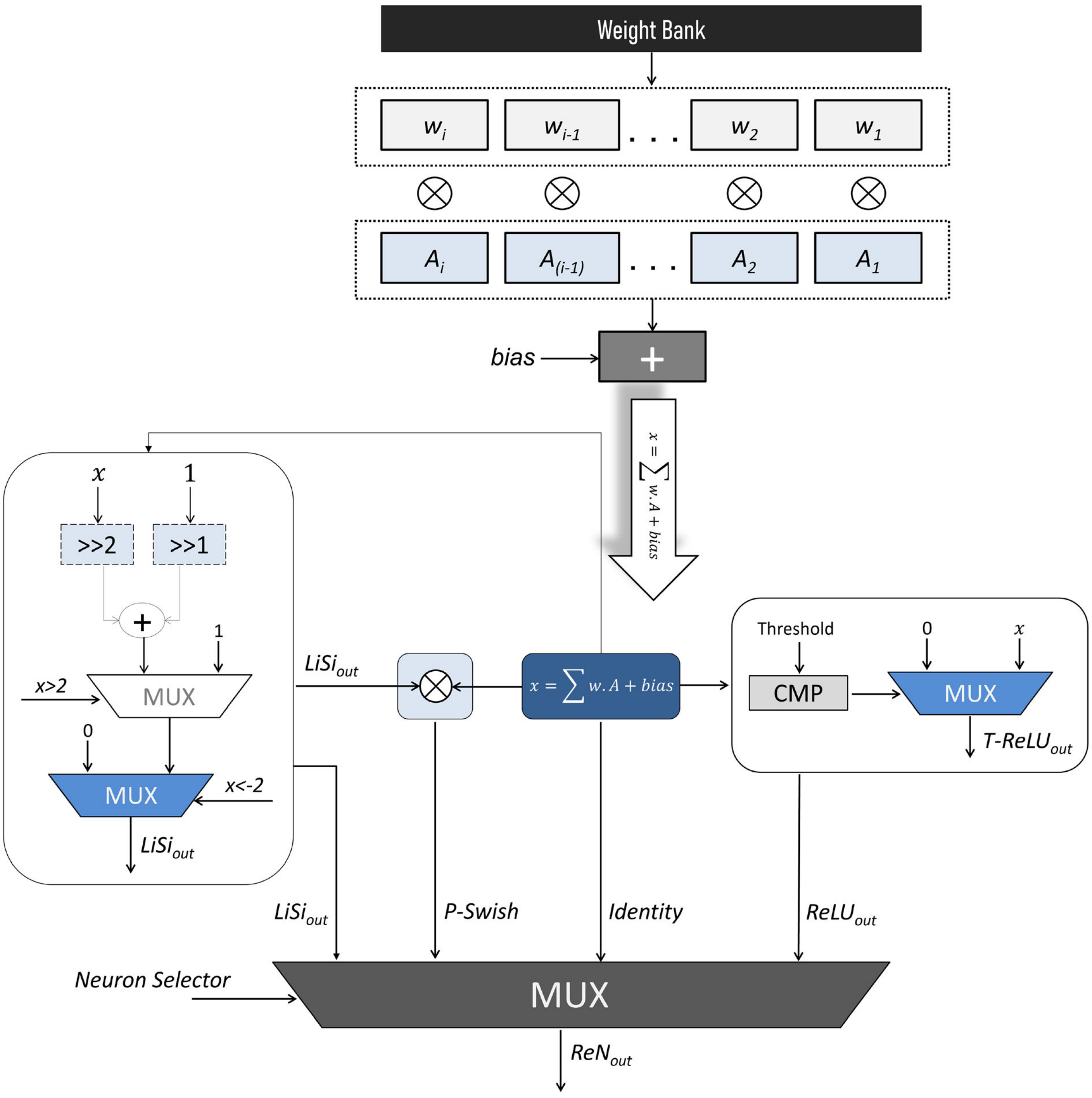

Fig. 9 Internal structure of a reconfigurable actuator (RA)

classification. The un-activated (non-softmax) voltage values at the output layer are termed *logits*.

The softmax function is not used in the proposed system, since it is very expensive to compute [43]. The unactuated logits—coming directly out of the output—are enough to obtain high accuracy [32]. Logits can be used even with cross entropy function. All one has to do is enable the option $from\_logits = True$ in Tensorflow (Python) [32]. Moreover, softmax is beneficial only if it is to be combined with cross entropy function or if it is required to obtain/visualize losses at the output. It does not have any impact on accuracy or backpropagation as such.

Predictor

The predictor is responsible for indicating the classified output. It does so by comparing values coming at the output neurons. The neuron that produces the maximum value corresponds to the classified output.

Results

In this section, we compare the proposed work with other contemporary works in terms of hardware efficiency and algorithmic accuracy. We also mention all the conditions

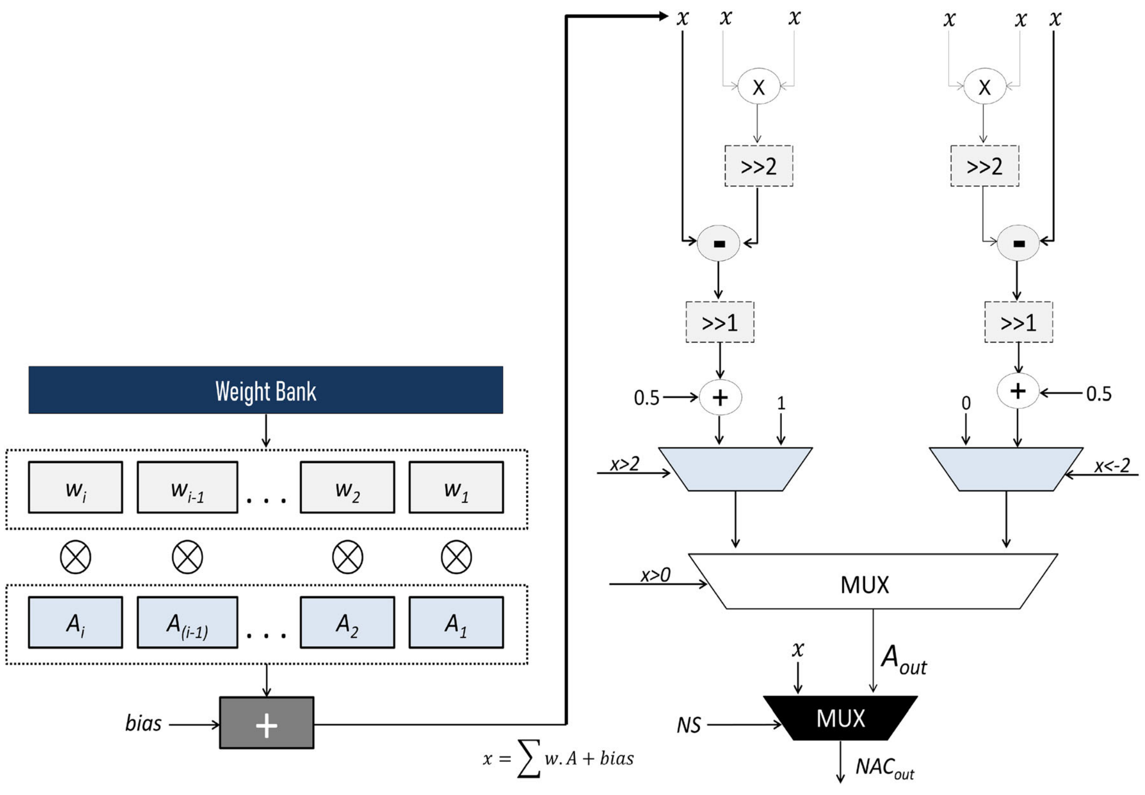

Fig. 10 Internal structure of the output layer (Reconfigurable) actuators

under which the evaluation was carried out. The dedicated hardware system is compared with the CPU-based system as well in terms of speed.

Benchmarks and Test Conditions

The network has been trained using Python on a personal computer. The inference is carried out on a Virtex 6 FPGA. The system is described in Verilog and uses 7 bits for weights and biases. The system classifies a given input sample into one of the eight categories, based on fish size. Every category corresponds to a certain range. If the estimated weight is extremely small, the first output neuron should be activated. Similarly, if the expected weight is extremely large, the last output neuron should be activated. The same goes for other weight sizes. All these ranges are shown in Table 1.

The dataset that has been used for performance evaluation of the N-AquaRAM is available from [4]. The complete dataset consists of 12 files (more than 1 million samples). However, we use only one file, which is enough for a fair evaluation. The data file used in this study consists of 175,000 samples, out of which 150,000 have been used for training. The remaining samples have been used for testing. For evaluation of the proposed P-Swish neuron, CIFAR-10 is also used. CIFAR-10 is a dataset that uses ten output classes, where every class corresponds to a unique object (car, bike, etc.) [14]. The convolutional neural network (CNN) used for CIFAR-10 has the following topology: 4C3-4C3-2P2-8C3-DO$_{0.2}$-8C3-2P2-16C3-DO$_{0.2}$-16C3-2P2-DO$_{0.2}$-FC$_{128}$-DO$_{0.2}$-FC$_{10}$.

Table 1 Weight range corresponding to size

Label (size)	Weight range (g)
Smallest	3.36–50.88
Smaller	52.34–99.40
Small	113.80–142.20
Medium	198.44
Large	207.85–236.08
Larger	254.90–295.90
Largest	297.00–328.75
Heaviest	357.80–394.66

Table 2 presents the test conditions and hyper-parameter values used for evaluating the aquaponic system. For hyper-parameter tuning, we use *grid search* [45].

Table 2 Hyper-parameter values and test conditions

Parameter	Value
Learning rate (η)	Default (0.001)
Batch size	Default (32)
Optimizer	Adam
Loss function	Cross entropy
Output coding	One hot
Test samples	20%
#Epochs	20

Evaluation of the Proposed Neuron Models

The T-ReLU function is 99.59% accurate, whereas the original ReLU function is 99.54% accurate. T-ReLU clearly performs better than the original ReLU on the dataset under consideration. Not only that, the T-ReLU is as HW-efficient as ReLU, since both require only a comparator for operation.

Now, we compare P-Swish with the original swish. Both swish and P-Swish exhibit the same level of accuracy, i.e., 99.6% on the aquaponic dataset. On CIFAR-10, P-Swish is about 0.4% more accurate than the original Swish under the given conditions. Though we do not make big claims, we may safely conclude that the proposed P-Swish function is almost as accurate as the original Swish function.

However, P-Swish is far more HW-efficient than the original swish, as shown in Table 3. The original form of swish contains a lot of exponential terms and divisions, both in the forward and the backward pass. Since, the proposed P-Swish performs well and does not contain any complex term in the backward pass, it is quite suitable even for on-chip learning.

Performance Evaluation and Comparisons

The proposed system yields an accuracy of 99.6%. A comparison of N-AquaRAM with various modern DL-based aquaponic monitoring schemes in terms of accuracy is shown in Table 4.

The proposed system is evaluated on both CPU and FPGA (Virtex 6). The FPGA implementation is about 1600 times faster than the CPU implementation. The results obtained from hardware evaluation are shown in Table 5. An absolutely fair comparison seems impossible, since the platforms, datasets, and underlying test conditions are different. However, it is evident that the proposed system is comparable to other modern works. The maximum clock frequency at which the proposed hardware engine can operate is around 40 MHz. Since, the system is fully parallel and requires only one clock cycle to infer a sample, the throughput of the system is around 40 million samples per second. The proposed system consists of 215 synapses. One synaptic operation consists of an addition and a multiplication. If we take into account the biases and neuronal activations, the system (running at 40 MHz) can carry out more than 18 giga operations per second (GOPS).

The work in [8] uses a small (toy) dataset with 25 binary input pixels and one neuron for binary (X and O) classification; two samples are used for training. The authors do not mention the system throughput explicitly. However, it is safe to assume that the maximum TP is far less than 1.9×10^6 samples per second. This is because the maximum operating frequency of the system is around 189 MHz and the time period requires to compute a sample is 100 ms. Moreover, the discretization step is 0.001. The work in [30] predicts epilepsy; it uses a small number of features and three output classes. No dataset is used in [31]; the authors just demonstrate the efficiency of hardware radial basis function. Though the work in [32] achieves high

Table 3 Hardware efficiency comparison between various swish implementations

	Iter	#Add	#Mult	#Divi	#Exp
Swish	FW	1	1	1	1
H. Swish	FW	1	3	0	0
P-Swish	FW	**1**	**1**	**0**	**0**
Swish	BW	3	3	1	1
H. Swish	BW	1	1	0	0
P-Swish	BW	1	0	0	0

Iter., iteration; FW, forward; Add., addition; Mult., multiplication; Divi., division; Exp., exponentials

Table 4 Accuracy comparisons–smart aquaponic systems

	Accuracy	Application
[10]	94.44%	Fish disease det
[13]	67.08%	Fingerl. size est
[42]	95%	Feeding int. est
[37]	96.50%	Plant det
[24]	97.80%	Fish length est
[44]	92.60%	Plant det
[44]	98.70%	Plant det
[2]	87%	Fish size est
Prop.	**99.60%**	**Fish size est.**

Est., estimation; Det., detection; Int., intensity

Table 5 Hardware cost and throughput comparisons

System	Application	Topology	Accuracy	Regs.	LuTs	DSPs	Platform	TP ($\times 10^6$)
[8]	Bin. C. (X/O)	25-5-1	89%	1023	11,339	–	Virtex 6	$<<1.89$
[30]	Epilepsy det	5-12-3	95.14%	114	12,960	116	Cyclone IV	50
[33]	Cancer det	30-5-2	98.23%	983	2,654	234	Virtex 6	63.5
[41]	None	5-5-2	–	1,898	3,124	154	Virtex 5	–
[31]	None	–	–	790	1,195	14	Spartan 3	10
[32]	Digit class	64-20-10	94.28%	4,677	30,654	0	Virtex 6	93.2
Proposed	Aquaponics	30-5-5-8	99.6%	2,612	33,740	0	Virtex 6	40

Regs., registers; LuTs, look-up tables; DSPs, digital signal processors; TP, throughput; Det., detection; Bin., binary

throughput, the accuracy is around 94.28%, which is lower than that of the proposed system. In fact, the system has not been tested for any aquaponic model. The system in [33] has been designed for classifying cancerous samples.

Though the systems in [30, 31, 33, and 41] consume only a small number of registers and look-up tables, they all require a large number of DSP48 elements that come prefabricated on an FPGA. Moreover, these are not as accurate as the proposed system. The proposed design does not use any DSP48 element and consumes only a small amount of resources.

Conclusions

This article presents a methodology to train smart aquaponic monitoring system using deep learning algorithms in order to achieve high accuracy, and a hardware-based smart aquaponic system that has a very low cost and is quite speedy. The work presents a novel neuron model P-Swish that is not only accurate but hardware efficient also. Such dedicated hardware devices occupy an extremely small amount of area when compared with typical general-purpose computers. This feature facilitates the deployment of cheap and low-power devices in digital agriculture sector.

The system achieves 99.6% accuracy in just a few training iterations. Moreover, this is one of the first works on smart aquaponics that uses more than 100,000 samples for training and testing. The hardware engine can process about 40 million samples per second (SPS), while consuming only a small amount of hardware resources, making it suitable for high-end Aquaponics 4.0 industrial applications. Lastly, the system presented in this article is just to give an idea of how useful the dedicated hardware devices are for smart aquaponics. In future, we intend to design a hardware system that can predict multiple fish/plant features. We may also use P-Swish along with convolutional neural networks to process complex image data.

Author Contributions Ali Siddique and Muhammad Azhar Iqbal contributed to the conceptualization, experiment design, and original draft preparation; Jingqi Sun and Sunbal Siddique were involved in the formal analysis and preparing figures/tables; Xu Zhang and Mang I. Vai involved in writing-review and editing. All authors discussed the results and approved the final version of this manuscript.

Declarations

Conflict of interest The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

1. Ahmed MS, Aurpa TT, Azad MAK (2022) Fish disease detection using image based machine learning technique in aquaculture. J King Saud Univ Comput Inf Sci 34:5170–5182
2. Álvarez-Ellacuría A, Palmer M, Catalán IA, Lisani J-L (2020) Image-based, unsupervised estimation of fish size from commercial landings using deep learning. ICES J Marine Sci 77:1330–1339
3. Azghadi MR, Lammie C, Eshraghian JK, Payvand M, Donati E, Linares-Barranco B, Indiveri G (2020) Hardware implementation of deep network accelerators towards healthcare and biomedical applications. IEEE Trans Biomed Circuits Syst 14:1138–1159
4. Collins Udanor (). Sensor Based Aquaponics Fish Pond Datasets. https://www.kaggle.com/datasets/ogbuokiriblessing/sensor-based-aquaponics-fish-pond-datasets?resource=download
5. Dhal SB, Jungbluth K, Lin R, Sabahi SP, Bagavathiannan M, Braga-Neto U, Kalafatis S (2022) A machine-learning-based iot

system for optimizing nutrient supply in commercial aquaponic operations. Sensors 22:3510
6. Farooq A, Verma AK, Hittinahalli CM, Harika N, Pai M (2023) Iron supplementation in aquaculture wastewater and its effect on the growth of spinach and pangasius in nutrient film technique based aquaponics. Agri Water Manag 277:108126
7. Farooq A, Verma AK, Hittinahalli CM, Varghese T, Pathak MS (2023) Iron supplementation in aquaculture wastewater and its impact on osmoregulatory, haematological, blood biochemical, and stress responses of pangasius with spinach in nutrient film technique based aquaponics. Aquaculture 567:739250
8. Farsa EZ, Ahmadi A, Maleki MA, Gholami M, Rad HN (2019) A low-cost high-speed neuromorphic hardware based on spiking neural network. IEEE Trans Circuits Syst II: Express Briefs 66:1582–1586
9. Harika N, Verma AK, Krishnani KK, Hittinahalli CM, Reddy R, Pai M (2024) Supplementation of potassium in aquaculture wastewater and its effect on growth performance of basil (ocimum basilicumin l) and pangasius (pangasianodon hypophthalmus) in nft-based aquaponics. Scientia Horticult 323:112521
10. Hasan N, Ibrahim S, Aqilah Azlan A (2022) Fish diseases detection using convolutional neural network (cnn). Int J Nonlinear Anal App 13:1977–1984
11. Jalal A, Salman A, Mian A, Shortis M, Shafait F (2020) Fish detection and species classification in underwater environments using deep learning with temporal information. Ecol Inf 57:101088
12. John VC, Verma AK, Krishnani KK, Chandrakant M, Bharti VS, Varghese T (2022) Optimization of potassium (k+) supplementation for growth enhancement of spinacia oleracea l. and pangasianodon hypophthalmus (sauvage, 1878) in an aquaponic system. Agri Water Manag 261:107339
13. Junior ADSO, Sant'Ana DA, Pache MCB, Garcia V, de Moares Weber VA, Astolfi G, de Lima Weber F, Menezes GV, Menezes GK, Albuquerque PLF et al (2021) Fingerlings mass estimation: A comparison between deep and shallow learning algorithms. Smart Agri Technol 1:100020
14. Krizhevsky A (2009) Learning multiple layers of features from tiny images. Technical Report
15. Lammie C, Olsen A, Carrick T, Azghadi MR (2019) Low-power and high-speed deep fpga inference engines for weed classification at the edge. IEEE Access 7:51171–51184
16. Li D, Li X, Wang Q, Hao Y (2022) Advanced techniques for the intelligent diagnosis of fish diseases: a review. Animals 12:2938
17. Li Y, Geng T, Li A, Yu H (2021) Bcnn: binary complex neural network. Microproc Microsyst 87:104359
18. Loni M, Sinaei S, Zoljodi A, Daneshtalab M, Sjödin M (2020) Deepmaker: a multi-objective optimization framework for deep neural networks in embedded systems. Microproc Microsyst 73:102989
19. Lu H, Ma X (2020) Hybrid decision tree-based machine learning models for short-term water quality prediction. Chemosphere 249:126169
20. Lu L (2020) Dying relu and initialization: theory and numerical examples. Commun Comput Phys 28:1671–1706
21. Meena LL, Verma AK, Bharti VS, Nayak SK, Chandrakant M, Haridas H, Reang D, Javed H, John VC (2022) Effect of foliar application of potassium with aquaculture wastewater on the growth of okra (abelmoschus esculentus) and pangasianodon hypophthalmus in recirculating aquaponic system. Scientia Horticult 302:111161
22. Meena LL, Verma AK, Krishnani KK, Hittinahalli CM, Haridas H, John VC (2023) Combined foliar application effect of iron and potassium on growth of okra and striped catfish using media bed based aquaponics. Aquaculture 569:739398
23. Merolla PA, Arthur JV, Alvarez-Icaza R, Cassidy AS, Sawada J, Akopyan F, Jackson BL, Imam N, Guo C, Nakamura Y et al (2014) A million spiking-neuron integrated circuit with a scalable communication network and interface. Science 345:668–673
24. Monkman GG, Hyder K, Kaiser MJ, Vidal FP (2019) Using machine vision to estimate fish length from images using regional convolutional neural networks. Methods Ecol Evol 10:2045–2056
25. Nazari N, Loni M, Salehi ME, Daneshtalab M, Sjodin M (2019) Tot-net: An endeavor toward optimizing ternary neural networks. In: 2019 22nd Euromicro Conference on Digital System Design (DSD) (pp 305–312). IEEE
26. Ortega-Zamorano F, Jerez JM, Urda Muñoz D, Luque-Baena RM, Franco L (2016) Efficient implementation of the backpropagation algorithm in fpgas and microcontrollers. IEEE Trans Neural Netw Learn Syst 27:1840–1850. https://doi.org/10.1109/TNNLS.2015.2460991
27. Qiu S, Xu X, Cai B (2018) Frelu: flexible rectified linear units for improving convolutional neural networks. In: 2018 24th international conference on pattern recognition (icpr) (pp 1223–1228). IEEE
28. Ramachandran P, Zoph B, Le QV (2017) Searching for activation functions. arXiv preprint arXiv:1710.05941
29. Ren Q, Zhang L, Wei Y, Li D (2018) A method for predicting dissolved oxygen in aquaculture water in an aquaponics system. Comput Elect Agri 151:384–391
30. Sarić R, Jokić D, Beganović N, Pokvić LG, Badnjević A (2020) Fpga-based real-time epileptic seizure classification using artificial neural network. Biomed Signal Proc Control 62:102106
31. Shymkovych V, Telenyk S, Kravets P (2021) Hardware implementation of radial-basis neural networks with gaussian activation functions on fpga. Neural Computing and Applications, pp 1–13
32. Siddique A, Iqbal MA, Aleem M, Islam MA (2023a) A 218 gops neural network accelerator based on a novel cost-efficient surrogate gradient scheme for pattern classification. Microprocessors and Microsystems, p 104831
33. Siddique A, Iqbal MA, Aleem M, Lin JC-W (2022) A high-performance, hardware-based deep learning system for disease diagnosis. PeerJ Comput Sci 8:e1034
34. Siddique A, Sun J, Hou KJ, Vai MI, Pun SH, Iqbal MA (2023) Spikoponic: a low-cost spiking neuromorphic computer for smart aquaponics. Agriculture 13:2057
35. Siddique A, Vai MI, Pun SH (2023) A low-cost, high-throughput neuromorphic computer for online snn learning. Cluster Computing, pp 1–18
36. Siddique A, Vai MI, Pun SH (2023) A low cost neuromorphic learning engine based on a high performance supervised snn learning algorithm. Sci Rep 13:6280
37. Taha MF, Abdalla A, ElMasry G, Gouda M, Zhou L, Zhao N, Liang N, Niu Z, Hassanein A, Al-Rejaie S et al (2022) Using deep convolutional neural network for image-based diagnosis of nutrient deficiencies in plants grown in aquaponics. Chemosensors 10:45
38. Taha MF, ElMasry G, Gouda M, Zhou L, Liang N, Abdalla A, Rousseau D, Qiu Z (2022) Recent advances of smart systems and internet of things (iot) for aquaponics automation: A comprehensive overview. Chemosensors 10:303
39. TensorFlow (a). TensorFlow Hard Sigmoid. https://www.tensorflow.org/api_docs/python/tf/keras/activations/hard_sigmoid
40. TensorFlow (b). Thresholded Rectified Linear Unit (T-ReLU). https://www.tensorflow.org/api_docs/python/tf/keras/layers/ThresholdedReLU
41. Tiwari V, Khare N (2015) Hardware implementation of neural network with sigmoidal activation functions using cordic. Microproc Microsyst 39:373–381

42. Ubina N, Cheng S-C, Chang C-C, Chen H-Y (2021) Evaluating fish feeding intensity in aquaculture with convolutional neural networks. Aquacult Eng 94:102178
43. Wuraola A, Patel N, Nguang SK (2021) Efficient activation functions for embedded inference engines. Neurocomputing 442:73–88
44. Yadav A, Thakur U, Saxena R, Pal V, Bhateja V, Lin JC-W (2022) Afd-net: apple foliar disease multi classification using deep learning on plant pathology dataset. Plant Soil 477:595–611
45. Zheng A (2015) Evaluating machine learning models: a beginner's guide to key concepts and pitfalls (2015)