electronics

MDPI

Article

Quantized Semantic Segmentation Deep Architecture for Deployment on an Edge Computing Device for Image Segmentation

Afaroj Ahamad [1,2,3], Chi-Chia Sun [1,2,*] and Wen-Kai Kuo [3]

[1] Smart Machine and Intelligent Manufacturing Research Center, National Formosa University, Huwei 632, Taiwan
[2] Digital System Design Laboratory, Department of Electrical Engineering, National Formosa University, Huwei 632, Taiwan
[3] Department of Electro-Optical Engineering, National Formosa University, Huwei 632, Taiwan
* Correspondence: ccsun@nfu.edu.tw; Tel.: +886-5-6315631

Abstract: In the field of computer vision technology, deep learning of image processing has become an emerging research area. The semantic segmentation of an image is among the utmost essential and significant tasks in image-processing research, offering a wide range of application fields such as autonomous driving systems, medical diagnosis, surveillance security, etc. Thus far, many studies have suggested and developed neural network modules in deep learning. To the best of our knowledge, all existing neural networks for semantic segmentation have large parameter sizes and it is therefore unfeasible to implement those architectures in low-power and memory-limited embedded platforms such as FPGAs. Building an embedded platform with that architecture is possible after reducing the parameter size without affecting the module's architecture. The quantization technique lowers the precision of the neural network parameters while mostly keeping the accuracy. In this paper, we propose a quantization algorithm for a semantic segmentation deep learning architecture, which reduces the parameter size by four to eight times with a negligible accuracy abatement. As long as the parameter size is reduced, the deep learning architecture is improved in terms of required storage, computational speed, and power efficiency.

Keywords: deep neural networks; quantization; fully convolutional network; SegNet; HR-Net; edge device; Taguchi method

check for updates

Citation: Ahamad, A.; Sun, C.-C.; Kuo, W.-K. Quantized Semantic Segmentation Deep Architecture for Deployment on an Edge Computing Device for Image Segmentation. *Electronics* **2022**, *11*, 3561. https://doi.org/10.3390/electronics11213561

Academic Editor: Adam Glowacz

Received: 16 September 2022
Accepted: 28 October 2022
Published: 31 October 2022

1. Introduction

Image segmentation partitions an image into multiple sets. Because of advancements in visual recognition systems, image segmentation has become a significant area in the image-processing field. Image segmentation [1,2] is further categorized into two parts, semantic segmentation and instant segmentation. In this article, we focus on semantic segmentation [3]. Semantic segmentation allows for the labeling of pixels with a set of categorized objects. An instance of semantic segmentation is the inherent classification, detection, and boundary localization of each pixel of an image. There are earlier non-deep learning approaches to semantic segmentation, i.e., texton forest [4] and random-forest-based classifiers [5]. Semantic segmentation also plays an essential role in a wide range of applications [6,7], e.g., in clinical diagnosis [8,9] to examine and understand disease through medical imaging. An autonomous driving system [10,11] provides partial automatic or fully automatic driving tasks. A robot navigation system [12] provides self-localization, path planning, and map interpretation to ensure smooth mass production and also to avoid dangerous situations in unsafe weather and natural disaster conditions. Typically, deep learning techniques have provided more precise, more accurate, and faster semantic segmentation than traditional non-deep learning methods.

The implementation of semantic segmentation neural networks in edge devices is a difficult task, although it is an advantageous objective that would provide real-time power efficiency and portable deep learning architecture for semantic segmentation. Edge devices have limited memory storage and low operational power. Embedded platforms such as FPGAs are more effective than a GPU and CPU in terms of cost, speed, and power efficiency [13]. In this era, FPGAs are rapidly surpassing other embedded platforms such as the GPU [14] as deep learning accelerators in real-time applications of machine learning. However, with high computations and storage restrictions, the implementation of large-parameter-size neural networks in edge devices is not straightforward. Therefore, optimization techniques such as quantization or pruning are used to compress neural network models into suitable parameter sizes. Because of the development of deep learning-based applications and service latency, reliability and cost play important roles. These factors are directly related to the used types of embedded platforms. In 2020, Wang et al. [15] conducted a comprehensive survey of deep learning architecture implementation in an edge device and discussed various application scenarios and fundamental enabling techniques. Furthermore, several [16–18] studies have discussed the benefits of edge computing devices for deep learning applications, although prevailing studies are still flourishing and have limitations in terms of resources.

In computer vision technology, image processing with deep learning architecture is an emerging area of research and has set a benchmark in computer vision research. Prior semantic segmentation network parameters such as weights and activations are floating numbers during training and inference and require large memory storage, as well as high operational power. Therefore, the implementation of floating point semantic segmentation networks in edge devices is not straightforward. The efficient implementation of deep learning architecture in edge devices requires the burdenless computation of neural network inferences such as storage. The two most popular neural network compression techniques are quantization and pruning, which provide a reduction in the size of architectures and hence reduce the required storage and operation power of neural network modules. The pruning technique provides the removal of less-sensitive weights, increases computational speed, and decreases the required storage size, whereas the quantization technique provides a reduction in the precision of the type of data used to compress neural network modules while mostly keeping the accuracy.

In this paper, we propose a quantization algorithm for semantic segmentation architectures, which provides a low-precision deep learning architecture for the semantic segmentation of an image, which could be implemented in edge devices such as FPGAs. Furthermore, we search for a sub-optimal quantized architecture using the Taguchi method.

Our most important contributions are as follows:

1. We propose a quantized semantic segmentation neural network. By using the Taguchi method, we find the suitable quantization bit length, i.e., 8 bits, to maintain accuracy higher than 80%, with reduced required storage.
2. Finally, the proposed architecture deploys an edge device and we discuss its implementation.

This paper is organized as follows. In Section 2, we survey the different studies about semantic segmentation and quantization techniques. Section 3 discusses the proposed quantization technique and briefly discusses the Taguchi method and circuital aspect for sub-optimal QSSN. Section 4 discusses the experimental setup of the proposed algorithm. Section 5 concludes this paper.

2. Background

Semantic segmentation is a subclass of image segmentation, which has a wide range of real-time applications. For semantic segmentation task realization, several prior deep learning architectures have been suggested. Typically, deep learning architectures for a semantic segmentation task have large parameter sizes, which makes them unfeasible for deployment on edge computing devices because of edge device resource restrictions such as memory and operational power. Quantization techniques could allow for the compression and optimization of deep learning architectures. Thus, in this study, we propose a compact deep learning architecture for semantic segmentation.

2.1. Semantic Segmentation Neural Network Architecture

For semantic segmentation, various deep learning architectures have been suggested, which produced optimistic predictions. In 2015, the most favored semantic segmentation architecture was the fully convolutional network (FCN), as suggested by J. Long et al. [19]. As illustrated in Figure 1, an FCN is constructed with locally connected layers with no dense layers such as the convolutional layer, pooling layer, and upsampling layer. Because there are no dense layers, there is a reduction in the computations and parameters. To improve the segmentation details, information is fused from the layers with different strides, i.e., FCN-32, FCN-16, and FCN-8. However, the FCN architecture is not fast enough for real-time inference; it also leaves out the global context information of the image and is complex for a 3D image. In 2016, Vasin et al. [20] suggested an architecture for semantic segmentation based on a recurrent neural network (RNN) termed "ReSeg", which is an extended form of ReNet [21], followed by an upsampling layer and a nonlinear softmax layer.

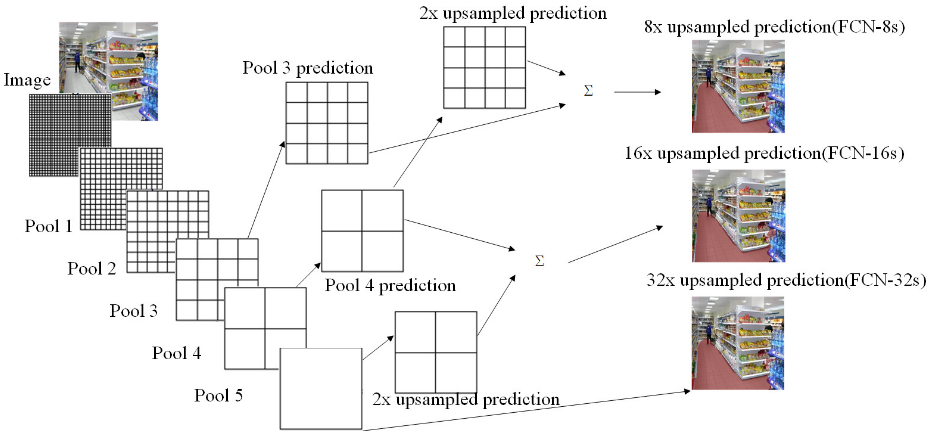

Figure 1. Fully convolutional network layout.

In 2015, Ronneberger et al. [22] suggested a convolutional network based on an FCN for image segmentation termed "U-net". The basic architecture consists of two paths. The first is the analysis path, also known as the encoder/contracting path, which provides the classification information. The second is the synthesis path, also known as the decoder/expansion path, which allows a network to learn localized classification information as well as enhance the resolution of the output. In 2016, Çiçek et al. [23] suggested a 3D U-net based on the basic U-net framework, which provides 3D volumetric segmentation. Unlike a traditional U-net, all 2D operations are replaced with corresponding 3D operations such as convolutional, max pooling, and up-convolutional. The advantage of a 3D U-net over a basic U-net is the faster training process with minimal annotations.

In 2017, Vijay et al. [24] suggested an encoder–decoder architecture termed "SegNet". As illustrated in Figure 2, the architecture consists of an encoder network that corresponds to the decoder networks, followed by a pixel-wise classification layer. Each network consists of 13 convolutional layers. The output of the final decoder is fed to the multi-class softmax classifier, which provides the class expectation value for each pixel independently. The SegNet model size is smaller than the FCN but its inference time is larger than the FCN because of the decoder network. Furthermore, several studies [25–28] have called for semantic pixel-wise labeling. Chen et al. [25], for example, suggested a deep convolutional neural network (DCNN) layer with a fully connected Conditional Random Field (CRF), which improves the localization property. Noh et al. [26] suggested another architecture adopted from VGG-16, which identifies pixel-wise class labels and predicts segmentation masks. Nico et al. [29] suggested speeding up semantic segmentation by applying a histogram of oriented depth (HOD) descriptors.

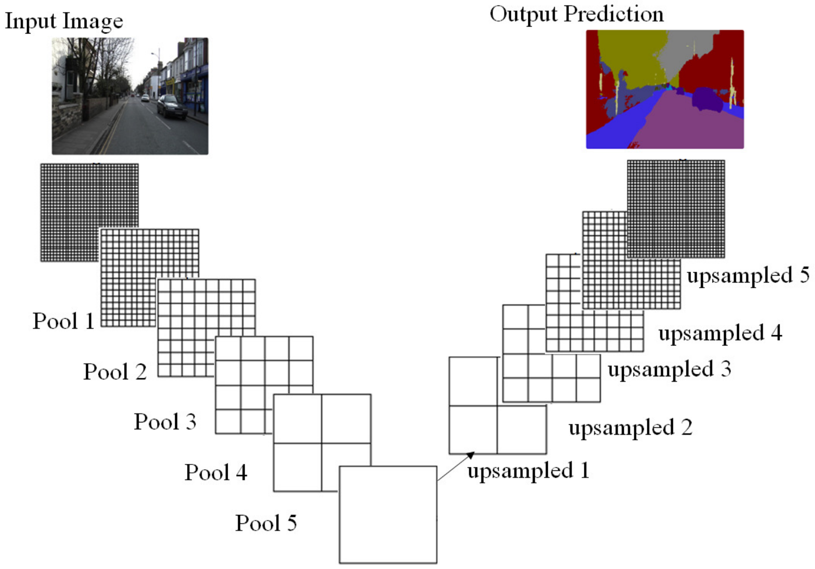

Figure 2. Encoder–decoder architecture for semantic segmentation.

Most existing deep learning architectures first encode the input image into low-resolution representations and extract the image information. Then, they reconstruct the high-resolution representations from the encoded low-resolution representations. Wang et al. suggested [30] an HR-Net architecture that provides high-resolution representation throughout the process. The HR-Net providing high-resolution representation throughout the process has two key characteristics. The first is that all high-to-low convolution streams are connected in parallel instead of in series; the second is the continuous process to and from the information through the resolutions. There is a repetition of multiresolution fusions to boost the high-resolution representations with the help of the low-resolution representations and vice versa. As shown in Figure 3, it is a three-stage architecture, which begins with a high-resolution convolution stream, followed one by one by high-to-low-resolution convolution streams. Each stage consists of several components such as parallel multi-resolution convolutions, repeated multi-resolution fusions, and representation heads. There are three kinds of representation heads illustrated in HR-Net, as shown in Figure 4, i.e., HRNetV1, HRNetV2, and HRNetV2p. In HRNetV1, the representation of the output is from the high-resolution stream. The other three representations are ignored. In HRNetV2, the output is represented as high resolution after rescaling the low-resolution stream along with two more representations. In HRNetV2p, multi-level representations are constructed by downsampling the high-resolution representation output from HRNetV2 to multiple levels. HR-Net provides better accuracy than the FCN and SegNet modules; however, the resolution is higher throughout the process and has a higher memory cast and power efficiency. Therefore, the real-time realization of HR-Net with an edge device is not feasible.

In 2021, Wang et al. [31] suggested an algorithm for a new supervised learning paradigm that provides pixel-wise contrastive semantic segmentation. In 2022, Zhou et al. [32] suggested a prototype view for rethinking semantic segmentation, which provides a representation of each class as a set of non-learnable prototypes, relying solely on the mean features of several training pixels within that class, unlike in the prior method. In 2022, Zhou et al. [33] suggested the volumetric memory network (VMN), which provides segmentation rules for 3D medical images. The VMN involves a memory-augmented network design and quality-assessment module. The memory-augmented network design allows the fast encoding of past segmentation information, whereas the estimation of the segmentation quality is done through the memory-augmented network. Still, image segmentation is a challenging issue because of ample intra-class variations, context variations, and ambiguities originating from occlusions and low image resolutions. Due to the limitations of convolutional filters, the global information in the image may not be fully accessed. Meanwhile, such information is particularly important for segmentation when designing the problem. To overcome this issue, in 2021, Strudel et al. [34] suggested a transformer approach for semantic segmentation, which captures the global context efficiently. In 2022, Hatamizadeh et al. [35] suggested a transformer-based segmentation architecture for 3D medical images, which includes a transformer encoder, providing an efficient model capable of learning long-range dependencies and effectively capturing global contextual representations at multiple scales.

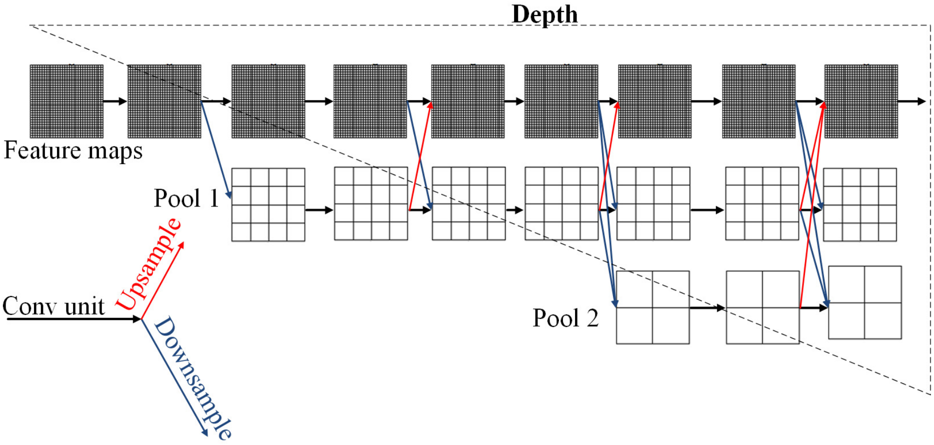

Figure 3. High-resolution network layout.

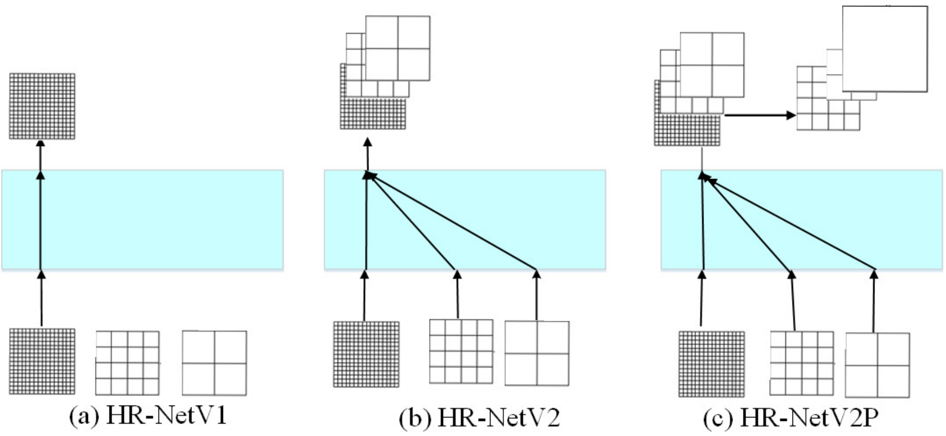

Figure 4. Illustration of head representation of HR-Net.

2.2. Comparison of Semantic Segmentation Architecture

In this section, we summarize all three semantic segmentation methods based on performance. Ideally, multiple aspects of a model should be evaluated, such as quantitative accuracy, speed (inference time), and storage requirements (memory footprint), in real-time applications. Several metric factors are evaluated to measure the performance of semantic segmentation such as pixel accuracy, mean pixel accuracy, mean intersection over union (m-IoU), mean accuracy, etc. Table 1 summarizes the three semantic segmentation methods' performances in terms of the m-IoU and mean accuracy of the different datasets.

In machine learning, FP 32-bit neural network models require a lot of memory, i.e., hundreds of MBs; they also require more operating power, i.e., hundreds of watts. These needs make the FP 32-bit neural networks unstable for implementation in an embedded system such as an FPGA. The issue of parameter size and computation complexity can be reduced by transforming full-precision networks into low-precision networks. The quantization [36] approach is a way to create a clone of the original neural network, which is a low-bit new neural network with acceptable accuracy and a much lower parameter size. In this research paper, we simulate an arbitrary quantized bit of a semantic segmentation deep learning architecture. Further, we apply the Taguchi method to find a suitable case combination that provides better accuracy with limited use of both storage and power.

Table 1. Comparison of semantic segmentation neural networks for different databases.

Ref.	Methodology	Topology	Datasets	Mean Acc.	Mean IoU
[19]	FCN = 8 s	VGG-16	Cityscapes	-	65.3
			PASCAL-Context.	-	35.1
[24]	SegNet	VGG-16	Cityscapes	-	57
			CamVid	71.2	60.1
			SUN-RGBD	44.76	31.84
			Cityscapes	-	57
[30]	HR-Net	HRNet V2-W40	Cityscapes	-	80.2
		HRNet V2-W48		-	81.1
		HRNet V2-W48	PASCAL-Context.	-	54
		HRNet V2-W48	LIP	67.3	55.9

2.3. Quantization Technique in Neural Networks

In computer vision technology, the implementation of a deep learning architecture in an edge-computing device for real-time application in various fields heightens the stipulation of an architecture's optimization. Quantization [37] is a technique that provides a low-precision compact deep learning architecture with negligible altered accuracy. In 2016, Courbariaux et al. [38] suggested using the binary neural network (BNN), which is a 2-bit quantized convolutional neural network that allows for a reduction in the parameter size. Other studies [39–43] suggested using an improved version of Courbaiaux et al.'s BNN by controlling the learning rate, gain factor, and backpropagation of a BNN, as arithmetic operations in BNNs computed by bitwise operations provide an extensive reduction in parameter size and hence ultimately power efficiency. In 2021, Vandersteegen et al. [44] suggested a quantization scheme based on power-of-two quantization scales, which provides 4-bit weights and 8-bit activations. Table 2 shows the accuracy of quantized neural networks and full-precision neural networks. It is obvious that the accuracy of quantized neural networks is higher than 85%, which is more suitable for real-time robotic applications.

Table 2. Accuracy of different methodologies on the CIFAR-10 database.

Ref.	Methodology	Topology	Acc. (%)
[45]	Simonyan et al., Full Precision	FP32	97.09
[38]	Courbaiaux et al., BNN	BNN	89.85
[39]	Rastegri et al., XNOR-Net	BNN	89.83
[40]	Zhou et al., BNN+	DoReFa-Net	87.16
[43]	Darabi et al., BNN+	AlexNet	87.16

However, the semantic segmentation architecture has a larger parameter size than other neural network architectures. Some semantic segmentation architectures have been modified and implemented in edge computing devices such as FPGAs. Olaf R. et al. [22] suggested a modified CNN-based architecture for semantic segmentation termed "U-Net", which can be implemented in an FPGA. In 2019, Vogel et al. [46] suggested an architecture for semantic segmentation for an FPGA, which uses an 8-bit quantization method. In 2019, Shimoda et al. [47] suggested an FCN-based semantic segmentation architecture for an FPGA, which provides an 8-bit quantization network with a filter-wise pruning technique that causes a drastic reduction in parameter size as well as computational complexity. In

2020, Miyami et al. [48] suggested a 3-bit quantized CNN-based architecture for semantic segmentation with 11 TOPs/s at a 300 MHz computational speed.

3. Proposed Quantized Semantic Segmentation Neural Network

In this section, we first discuss the quantization technique for the quantized semantic segmentation neural network (QSSN). Afterward, we introduce the Taguchi method and evaluate the best-quantized bit length for the suboptimal conditions, leading to an improved version of the architecture that provides better accuracy. Finally, the proposed QSSN architecture is implemented in an FPGA accelerator.

3.1. Quantization Technique of Full Semantic Segmentation Neural Network

Quantization is a process in machine learning (ML) that allows for building a similar ML model in which all operations and computations occur at low precision. On the other hand, the parameter size is reduced; hence, the new architecture improves execution performance and efficiency. Since the FPGA is an embedded platform that has limited storage and low-power operations, we should transform an FP 32-bit ML model into an equivalent model that provides a parameter size under the BRAM size and requires low power to operate. The above target can be achieved by applying the model optimization method, that is, quantization.

Quantization Method

In this subsection, we discuss the mathematical equation of the quantization scheme [49]. The weight and activation of each convolutional layer are quantized into b-bit integers, which correspond with their bit representations of the floating point values. Assume that the range of floating point values and b-bit integers are, respectively, $x\epsilon[\alpha, \beta]$ and $x^q\epsilon[\alpha^q, \beta^q]$, then the quantization of x is given as

$$x^q = round\left(\frac{1}{s}x + z\right) \tag{1}$$

where s is defined as the scale factor and z is the zero point. The scale factor and the zero point are represented, respectively, in Equations (2) and (3), which represent the α, β, and b-bits of the quantitation length.

$$s = (2^b - 1)/(\alpha - \beta) \tag{2}$$

$$z = -round(\beta.s) - 2^{b-1} \tag{3}$$

When the real values of the weights and activations fall beyond the defined range, i.e., beyond $[\alpha, \beta]$, then the quantized values also lie outside of $[\alpha^q, \beta^q]$. To resolve this issue of falling beyond the range, the use of the clip function is defined as

$$clip(x, u, v) = \left\{ \begin{array}{cc} u & \text{if } x < u \\ x & \text{if } u \leq x \leq v \\ v & \text{if } x > v \end{array} \right. \tag{4}$$

The quantization of x beyond the defined range is defined as

$$x^q = clip[round\left(\frac{1}{s}x + z\right)] \tag{5}$$

The dequantizer function (x^d) is defined in Equation (6), which computes an approximation of the original input of the real value, i.e., $x \approx x^d$.

$$x^d = \frac{1}{s}\left(x^q - z\right) \tag{6}$$

As shown in Figure 5, the quantization data flow diagram shows the conversion of a high-precision form into a low-precision form. The full-precision 32-bit float inputs into the quantizer, which provides an arbitrary low-precision output with different bit sizes such as 2-, 4-, 8-, or 16-bit. The input float tensors first go to max and min operations; afterward, they are fed to the dequantizer operations. Finally, an arbitrary bit results as the output from the dequantizer. The needless conversion to and from the float is eliminated in the next stage after the conversion of the individual operations. Numerous quantizer and dequantizer operations are required in the presence of consecutive sequences of floating equivalents. At this stage, pattern recognition operations are involved, which allow for the cancellation and removal of each other. This could be applied on a large scale to the models, where all operations have quantized equivalents, and offers a graph where all of the tensor calculations are performed in a quantized bit without having to convert to floating values.

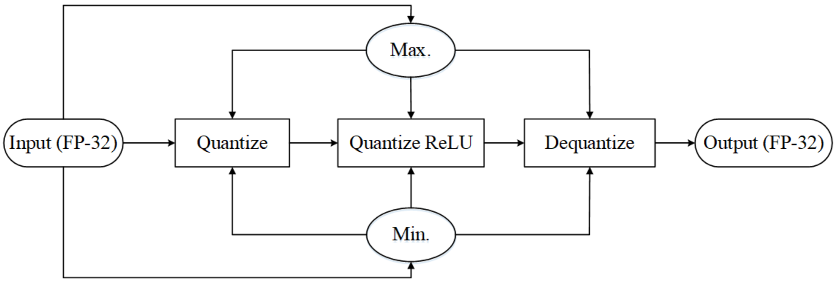

Figure 5. Schematic Flow Diagram: Quantization Algorithm of Deep Learning Architecture.

3.2. Taguchi Method-Based Sub-Optimal Quantized Semantic Segmentation Neural Network

In this subsection, we discuss the proposed quantized semantic segmentation neural network (QSSN) and search for suitable quantized bit lengths for the sub-optimal conditions using the Taguchi method, which is based on an orthogonal array, i.e., it is a statistical approach proposed by Dr. Taguchi while working for the Nippon Telephones and Telegraph Company [50,51]. This method provides an experimental design during the process-design stage and allows for a certain level of quality control for magnificent performance. In the past few years, the Taguchi method has been used to design experiments in several fields, such as engineering, biotechnology, and computer science to achieve the best performance results [52–55]. There are three factors in the Taguchi method that should be appraised while using the experimental design, i.e., the Taguchi loss function, offline quality control, and orthogonal arrays.

Figure 6 shows a schematic flow diagram of the proposed sub-optimal QSSN, which involves quantization, training, and finding suitable quantization bit sizes with a fixed range of accuracy using the Taguchi method. First, the encoder–decoder semantic segmentation model with full precision is quantized with a bit length of b-bits, i.e., 16, 8, 4, or 2. The bit size of the proposed QSSN is assumed as a controlling factor for one input into the Taguchi block. After the quantization process, the resulting architecture, i.e., the QSSN, goes to the training block and evaluates the accuracy. The output of the training block is the assumed accuracy and is used as one input for the comparator. The output of the Taguchi block is the required accuracy and is used as the other input for the comparator. If the output of the comparator is greater than or equal to zero it means that our requirement is satisfied and the bit size is accepted; otherwise, the bit size needs to be altered again.

The forecast matrix of the design experiment of the proposed QSSN architecture is illustrated in Table 3 and shows the eight optimal designs for the experiment, i.e., R1 to R8. The Taguchi-based QSSN considers two factors: the first is storage, which is directly related to the architecture's parameter size; the second is the accuracy of the architecture, which indicates the system's robustness. The table shows the accuracies and storage performances of the four different precisions in this study, i.e., 16-, 8-, 4-, and 2-bit, with two different levels of data pruning, i.e., 96 × 96 and 48 × 48. The first concern of the QSSN is that the accuracy must be higher than 80%; the second concern is the required storage. Hence, R1 has the best and R8 has the worst accuracy. First, we set R1 as the maximum accuracy, then, we locked the minimum accuracy at higher than 80%, i.e., R6. Afterward, between R1 and R6, the accuracy was higher than 80% and was locked as the baseline accuracy, i.e., baseline 1 was R2 and baseline 2 was R5. For baseline 1, R2, the accuracy decreased by only 3.24%, with an 8.65% decrease in the required storage in comparison with the maximum accuracy, R1. Similarly, for baseline 1, R2, the accuracy increased by 1.94%, with only 0.01% extra storage required. Baseline 2, R5, had a lower accuracy of 0.09%; further, the required storage was 3.10% extra in comparison with baseline 1, R2. Thus, baseline 1, R2, was the best case in this study.

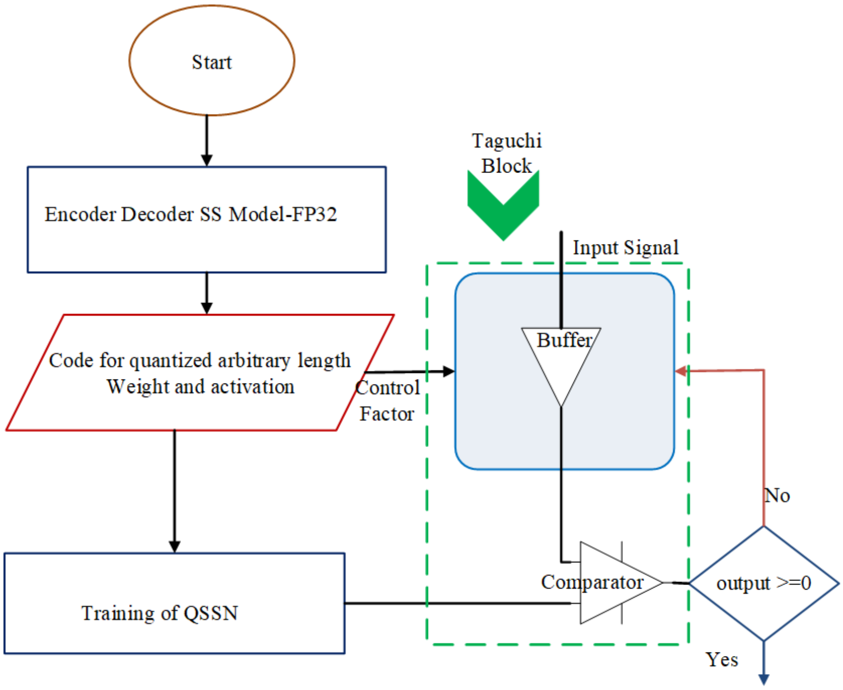

Figure 6. Schematic Flow Diagram of Sub-Optimal QSSN.

Table 3. Design experiment's forecast matrix based on Taguchi method.

S. No.	Precision	Data Pruning	Resources Utilization (%)			Accuracy (%)	Prediction
			DSP	**LTU**	**FF**		
1	***16-bit***	***96 × 96***	***92***	***24.41***	***26.85***	***88.38***	***Maximum***
2	***8-bit***	***96 × 96***	***77***	***21.15***	***18.20***	***85.14***	***Baseline 1***
3	4-bit	96 × 96	64	18.20	16.18	70.23	
4	2-bit	96 × 96	57	14.30	14.16	60.14	
5	***16-bit***	***48 × 48***	***74***	***20.15***	***17.30***	***84.15***	***Baseline 2***
6	***8-bit***	***48 × 48***	***68***	***19.15***	***18.10***	***83.20***	***Minimum***
7	4-bit	48 × 48	57	17.15	13.61	64.15	
8	2-bit	48 × 48	47	14.20	12.25	54.19	

4. Experimental Setup and Results

In recent years, edge devices such as FPGAs have become prominent alternatives, with power-efficient and fast real-time accelerators in the area of deep learning. In this section, we discuss the deployment of the proposed QSSN architecture on the FPGA. The proposed QSSN architecture was deployed on the Xilinx ZCU104 FPGA, as illustrated in Figure 7. The implementation began with the configuration of the hardware architecture, which was performed by a pre-placed bit file and .tcl file in the PL block. Afterward, the PS block initialized the video camera's USB, basic HDMI-related settings, and QSSN weight and threshold load. Once the camera captured the targeted image, an AXI4-Stream sent the image to the QSSN system in the PL. After the completion, the predicted image was sent to the PS terminal and displayed on the screen.

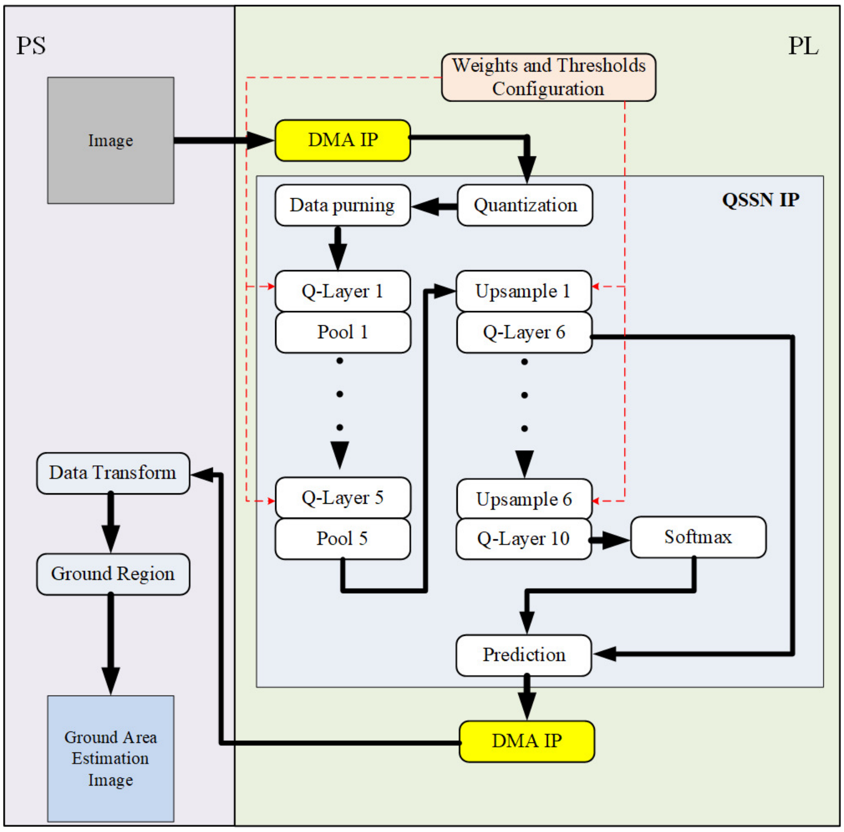

Figure 7. The proposed QSSN system architecture on the FPGA.

Table 4 shows a performance comparison of the quantized neural networks. The first four listed architectures were implemented on the FPGA platform and performed only classifications. The binary CNN (BCNN) simple 2-bit quantized CNN showed an accuracy of about 88%, whereas the third achieved up to 64%. However, the 8-bit CNN showed 87% accuracy with lower GOPS than the BCNN. The bottom two listed architectures, i.e., SFCN and U-Net, performed semantic segmentation. U-Net had better GOPS than SFCN; however, SFCN and U-Net achieved accuracies of less than 70%, which makes both methods inefficient in real-time robotics applications. In this study, the QSSN architecture achieved an accuracy of up to 85.15%, which is suitable for real-time robotics applications.

Table 4. Performance Comparison of Quantized Neural Networks.

Architecture	Precision	Platform	GOPS	GOPS/W	Acc. (%)
BCNN	1-bit	ZC702	722	218.78	88.61
BCNN	1-bit	ZC7020	207.8	44.21	88.68
BCNN (YOLOv2)	1-bit	VC707	1877	102.62	64.16
QCNN	8-bit	ZC7Z045	84.3	8.65	87
SFCN	8-bit	ZCU104	165.4	137.9	65.9
U-Net	3-bit	Alveo 200	11,059	243.9	67.8
Proposed QSSN	8-bit	ZCU104	1942	430.79	85.15

The proposed QSSN achieved exceptional performance in terms of computational speed, power efficiency, and required memory storage. In addition, the parameter size of the QSSN was compressed by up to six times; therefore, the storage demand was reduced, as well as the complexity. Because of the quantized bits, all arithmetic operations were performed as bitwise operations so the computational speed was higher than in full-precision neural network modules. The QSSN achieved accuracy that, although less than full-precision architectures, is still negligible because for robotics applications, an accuracy higher than 80.00% is more acceptable. According to the accuracy requirement, the QSSN arbitrated the bit lengths in different layers; hence, it provided an arbitrary bit size. Therefore, according to the edge device's memory and required performance, it was possible to alter the bit length. The foremost accomplishment of this proposed algorithm was the power efficiency, which was ten times better than full-precision architectures.

5. Conclusions

This study proposes a QSSN, which provides an architecture with improved required storage, enhanced power efficiency, and faster computational speed than conventional segmentation architectures. The QSSN accuracy is not as high as full-precision architectures but it is negligibly tolerated. The QSSN achieved an accuracy of 85.15% with quantized 8-bit precision, which is sufficiently acceptable for real-time robotics applications. The most important edge device requirement is low-operational power. The QSSN architecture is six times more compact than full-precision architectures, which enables QSSN deployment on edge devices; hence, the QSSN is ten times more power-efficient than full-precision architectures. This study also reveals that quantization below 8-bit precision causes a drastic reduction in accuracy with a minimal reduction in storage size. Therefore, 4-bit and 2-bit precision are not suitable, and the maximum low precision is 8-bit. In the future, the output of the Taguchi block may include other performance factors such as parameter size or operation power or a combination of two or more performance factors. In addition, the quantization of each convolutional layer can be performed separately; however, it will make computations more complex but we can still achieve higher accuracy with reduced storage memory.

Author Contributions: Conceptualization, C.-C.S.; Formal analysis, C.-C.S. and W.-K.K.; Investigation, C.-C.S. and W.-K.K.; Methodology, A.A.; Validation, A.A.; Visualization, A.A.; Writing—original draft, A.A.; Writing—review and editing, A.A. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported in part by the Ministry of Science and Technology of Taiwan under the contract of MOST-108-2221-E-150-038.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Dhanachandra, N.; Manglem, K.; Chanu, Y.J. Image segmentation using K-means clustering algorithm and subtractive clustering algorithm. *Procedia Comput. Sci.* **2015**, *54*, 764–771. [CrossRef]
2. Plath, N.; Toussaint, M.; Nakajima, S. Multi-class image segmentation using conditional random fields and global classification. In Proceedings of the 26th Annual International Conference on Machine Learning, Montreal, QC, Canada, 14–18 June 2009; pp. 817–824.
3. Wang, P.; Chen, P.; Yuan, Y.; Liu, D.; Huang, Z.; Hou, X.; Cottrell, G. Understanding convolution for semantic segmentation. In Proceedings of the 2018 IEEE Winter Conference on Applications of Computer Vision (WACV), Lake Tahoe, NV, USA, 12–15 March 2018; pp. 1451–1460.
4. Shotton, J.; Johnson, M.; Cipolla, R. Semantic texton forests for image categorization and segmentation. In Proceedings of the 2008 IEEE Conference on Computer Vision and Pattern Recognition, Anchorage, AK, USA, 23–28 June 2008; pp. 1–8.
5. Schroff, F.; Criminisi, A.; Zisserman, A. Object Class Segmentation using Random Forests. In Proceedings of the BMVC, Leeds, UK, 1–4 September 2008; pp. 1–10.
6. Chen, B.k.; Gong, C.; Yang, J. Importance-Aware Semantic Segmentation for Autonomous Driving System. In Proceedings of the IJCAI, Melbourne, Australia, 19–25 August 2017; pp. 1504–1510.
7. Teichmann, M.; Weber, M.; Zoellner, M.; Cipolla, R.; Urtasun, R. Multinet: Real-time joint semantic reasoning for autonomous driving. In Proceedings of the 2018 IEEE Intelligent Vehicles Symposium (IV), Changshu, China, 26–30 June 2018; pp. 1013–1020.
8. Jiang, F.; Grigorev, A.; Rho, S.; Tian, Z.; Fu, Y.; Jifara, W.; Adil, K.; Liu, S. Medical image semantic segmentation based on deep learning. *Neural Comput. Appl.* **2018**, *29*, 1257–1265. [CrossRef]

9. Zeng, G.; Zheng, G. Holistic decomposition convolution for effective semantic segmentation of medical volume images. *Med. Image Anal.* **2019**, *57*, 149–164. [CrossRef]
10. Li, B.; Liu, S.; Xu, W.; Qiu, W. Real-time object detection and semantic segmentation for autonomous driving. In Proceedings of the MIPPR 2017: Automatic Target Recognition and Navigation, Xiangyang, China, 28–29 October 2017; Volume 10608, pp. 167–174.
11. Tseng, Y.H.; Jan, S.S. Combination of computer vision detection and segmentation for autonomous driving. In Proceedings of the 2018 IEEE/ION Position, Location and Navigation Symposium (PLANS), Monterey, CA, USA, 23–26 April 2018; pp. 1047–1052.
12. Zhang, Y.; Chen, H.; He, Y.; Ye, M.; Cai, X.; Zhang, D. Road segmentation for all-day outdoor robot navigation. *Neurocomputing* **2018**, *314*, 316–325. [CrossRef]
13. Pauwels, K.; Tomasi, M.; Alonso, J.D.; Ros, E.; Van Hulle, M.M. A comparison of FPGA and GPU for real-time phase-based optical flow, stereo, and local image features. *IEEE Trans. Comput.* **2011**, *61*, 999–1012. [CrossRef]
14. Coates, A.; Baumstarck, P.; Le, Q.; Ng, A.Y. Scalable learning for object detection with GPU hardware. In Proceedings of the 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems, St. Louis, MO, USA, 10–15 October 2009; pp. 4287–4293.
15. Wang, X.; Han, Y.; Leung, V.C.; Niyato, D.; Yan, X.; Chen, X. Convergence of edge computing and deep learning: A comprehensive survey. *IEEE Commun. Surv. Tutor.* **2020**, *22*, 869–904. [CrossRef]
16. Heidari, A.; Jabraeil Jamali, M.A.; Jafari Navimipour, N.; Akbarpour, S. Deep Q-learning technique for offloading offline/online computation in blockchain-enabled green IoT-edge scenarios. *Appl. Sci.* **2022**, *12*, 8232. [CrossRef]
17. Heidari, A.; Toumaj, S.; Navimipour, N.J.; Unal, M. A privacy-aware method for COVID-19 detection in chest CT images using lightweight deep conventional neural network and blockchain. *Comput. Biol. Med.* **2022**, *145*, 105461. [CrossRef]
18. Filho, C.P.; Marques, E., Jr.; Chang, V.; Dos Santos, L.; Bernardini, F.; Pires, P.F.; Ochi, L.; Delicato, F.C. A Systematic Literature Review on Distributed Machine Learning in Edge Computing. *Sensors* **2022**, *22*, 2665. [CrossRef]
19. Long, J.; Shelhamer, E.; Darrell, T. Fully convolutional networks for semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 3431–3440.
20. Visin, F.; Ciccone, M.; Romero, A.; Kastner, K.; Cho, K.; Bengio, Y.; Matteucci, M.; Courville, A. Reseg: A recurrent neural network-based model for semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, Las Vegas, NV, USA, 26 June–1 July 2016; pp. 41–48.
21. Medsker, L.R.; Jain, L. Recurrent neural networks. *Des. Appl.* **2001**, *5*, 64–67.
22. Ronneberger, O.; Fischer, P.; Brox, T. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*; Springer: Berlin/Heidelberg, Germany, 2015; pp. 234–241.
23. Çiçek, Ö.; Abdulkadir, A.; Lienkamp, S.S.; Brox, T.; Ronneberger, O. 3D U-Net: Learning dense volumetric segmentation from sparse annotation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*; Springer: Berlin/Heidelberg, Germany, 2016; pp. 424–432.
24. Badrinarayanan, V.; Kendall, A.; Cipolla, R. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 2481–2495. [CrossRef]
25. Chen, L.C.; Papandreou, G.; Kokkinos, I.; Murphy, K.; Yuille, A.L. Semantic image segmentation with deep convolutional nets and fully connected crfs. *arXiv* **2014**, arXiv:1412.7062.
26. Noh, H.; Hong, S.; Han, B. Learning deconvolution network for semantic segmentation. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 1520–1528.
27. Eigen, D.; Fergus, R. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 2650–2658.
28. Papandreou, G.; Chen, L.; Murphy, K.; Yuille, A.L. Weakly-and semi-supervised learning of a DCNN for semantic image segmentation. *arXiv* **2015**, arXiv:1502.02734.
29. Hft, N.; Schulz, H.; Behnke, S. Fast semantic segmentation of RGB-D scenes with GPU-accelerated deep neural networks. In *Joint German/Austrian Conference on Artificial Intelligence (Künstliche Intelligenz)*; Springer: Berlin/Heidelberg, Germany, 2014; pp. 80–85.
30. Wang, J.; Sun, K.; Cheng, T.; Jiang, B.; Deng, C.; Zhao, Y.; Liu, D.; Mu, Y.; Tan, M.; Wang, X.; et al. Deep high-resolution representation learning for visual recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, *43*, 3349–3364. [CrossRef]
31. Wang, W.; Zhou, T.; Yu, F.; Dai, J.; Konukoglu, E.; Van Gool, L. Exploring cross-image pixel contrast for semantic segmentation. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, QC, Canada, 10–17 October 2021; pp. 7303–7313.
32. Zhou, T.; Wang, W.; Konukoglu, E.; Van Gool, L. Rethinking Semantic Segmentation: A Prototype View. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 19–20 June 2022; pp. 2582–2593.
33. Zhou, T.; Li, L.; Bredell, G.; Li, J.; Konukoglu, E. Volumetric memory network for interactive medical image segmentation. *Med. Image Anal.* **2022**, *83*, 102599. [CrossRef]
34. Strudel, R.; Garcia, R.; Laptev, I.; Schmid, C. Segmenter: Transformer for semantic segmentation. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, QC, Canada, 10–17 October 2021; pp. 7262–7272.
35. Hatamizadeh, A.; Tang, Y.; Nath, V.; Yang, D.; Myronenko, A.; Landman, B.; Roth, H.R.; Xu, D. Unetr: Transformers for 3d medical image segmentation. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, Waikoloa, HI, USA, 4–8 January 2022; pp. 574–584.

36. Khoram, S.; Li, J. Adaptive quantization of neural networks. In Proceedings of the International Conference on Learning Representations, Vancouver, BC, Canada, 30 April–3 May 2018.
37. Gupta, S.; Agrawal, A.; Gopalakrishnan, K.; Narayanan, P. Deep learning with limited numerical precision. In Proceedings of the International Conference on Machine Learning, Lille, France, 7–9 July 2015; pp. 1737–1746.
38. Hubara, I.; Courbariaux, M.; Soudry, D.; El-Yaniv, R.; Bengio, Y. Binarized neural networks. *Adv. Neural Inf. Process. Syst.* **2016**, *29*, 4107–4115.
39. Rastegari, M.; Ordonez, V.; Redmon, J.; Farhadi, A. Xnor-net: Imagenet classification using binary convolutional neural networks. In *European Conference on Computer Vision*; Springer: Berlin/Heidelberg, Germany, 2016; pp. 525–542.
40. Zhou, S.; Wu, Y.; Ni, Z.; Zhou, X.; Wen, H.; Zou, Y. Dorefa-net: Training low bitwidth convolutional neural networks with low bitwidth gradients. *arXiv* **2016**, arXiv:1606.06160.
41. Tang, W.; Hua, G.; Wang, L. How to train a compact binary neural network with high accuracy? In Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, San Francisco, CA, USA, 4–9 February 2017.
42. Lin, X.; Zhao, C.; Pan, W. Towards accurate binary convolutional neural network. In Proceedings of the Advances in Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; pp. 345–353.
43. Darabi, S.; Belbahri, M.; Courbariaux, M.; Nia, V.P. BNN+: Improved binary network training. In Proceedings of the Sixth International Conference on Learning Representations, Vancouver, BC, Canada, 30 April–3 May 2018.
44. Vandersteegen, M.; Van Beeck, K.; Goedemé, T. Integer-Only CNNs with 4 Bit Weights and Bit-Shift Quantization Scales at Full-Precision Accuracy. *Electronics* **2021**, *10*, 2823. [CrossRef]
45. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.
46. Vogel, S.; Springer, J.; Guntoro, A.; Ascheid, G. Efficient acceleration of cnns for semantic segmentation on fpgas. In Proceedings of the 2019 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays, Seaside, CA, USA, 24–26 February 2019; p. 309.
47. Shimoda, M.; Sada, Y.; Nakahara, H. Filter-wise pruning approach to FPGA implementation of fully convolutional network for semantic segmentation. In *International Symposium on Applied Reconfigurable Computing*; Springer: Berlin/Heidelberg, Germany, 2019; pp. 371–386.
48. Miyama, M. FPGA implementation of 3-bit quantized CNN for semantic segmentation. *J. Phys. Conf. Ser.* **2021**, *1729*, 012004. [CrossRef]
49. Jacob, B.; Kligys, S.; Chen, B.; Zhu, M.; Tang, M.; Howard, A.; Adam, H.; Kalenichenko, D. Quantization and training of neural networks for efficient integer-arithmetic-only inference. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 2704–2713.
50. Rosa, J.L.; Robin, A.; Silva, M.; Baldan, C.A.; Peres, M.P. Electrodeposition of copper on titanium wires: Taguchi experimental design approach. *J. Mater. Process. Technol.* **2009**, *209*, 1181–1188. [CrossRef]
51. Athreya, S.; Venkatesh, Y. Application of Taguchi method for optimization of process parameters in improving the surface roughness of lathe facing operation. *Int. Ref. J. Eng. Sci.* **2012**, *1*, 13–19.
52. Ju, Y.; Guo, J.; Liu, S. A deep learning method combined sparse autoencoder with SVM. In Proceedings of the 2015 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery, Xi'an, China, 17–19 September 2015; pp. 257–260.
53. Hsu, Q.C.; Do, A.T. Minimum porosity formation in pressure die casting by Taguchi method. *Math. Probl. Eng.* **2013**, *2013*, 920865. [CrossRef]
54. Cui, F.; Su, Y.; Xu, S.; Liu, F.; Yao, G. Optimization of the physical and mechanical properties of a spline surface fabricated by high-speed cold roll beating based on taguchi theory. *Math. Probl. Eng.* **2018**, *2018*, 8068362. [CrossRef]
55. Andrzejak, R.G.; Lehnertz, K.; Mormann, F.; Rieke, C.; David, P.; Elger, C.E. Indications of nonlinear deterministic and finite-dimensional structures in time series of brain electrical activity: Dependence on recording region and brain state. *Phys. Rev. E* **2001**, *64*, 061907. [CrossRef] [PubMed]