AGH UNIVERSITY OF KRAKOW

FIELD OF SCIENCE ENGINEERING AND TECHNOLOGY

SCIENTIFIC DISCIPLINE AUTOMATION, ELECTRONICS, ELECTRICAL ENGINEERING AND SPACE TECHNOLOGIES

DOCTORAL THESIS

DEEP NEURAL NETWORKS QUANTIZATION AND ACCELERATION METHODS FOR REAL TIME ENERGY EFFICIENT VISION SYSTEMS

AUTHOR: DOMINIKA PRZEWŁOCKA-RUS

FIRST SUPERVISOR: MAREK GORGOŃ, PROF.
ASSISTING SUPERVISOR: TOMASZ KRYJAK, PHD

COMPLETED IN: AGH OF KRAKOW, FACULTY OF ELECTRICAL ENGINEERING, AUTOMATICS, COMPUTER SCIENCE AND BIOMEDICAL ENGINEERING, DEPARTMENT OF AUTOMATIC CONTROL AND ROBOTICS

KRAKÓW 2024

AKADEMIA GÓRNICZO-HUTNICZA IM. STANISŁAWA STASZICA W KRAKOWIE

DZIEDZINA NAUK INŻYNIERYJNO-TECHNICZNYCH

DYSCYPLINA AUTOMATYKA, ELEKTRONIKA, ELEKTROTECHNIKA I TECHNOLOGIE KOSMICZNE

ROZPRAWA DOKTORSKA

METODY KWANTYZACJI I AKCELERACJI GŁĘBOKICH SIECI NEURONOWYCH DLA ENERGOOSZCZĘDNYCH SYSTEMÓW WIZYJNYCH CZASU RZECZYWISTEGO

AUTORKA: DOMINIKA PRZEWŁOCKA-RUS

PROMOTOR ROZPRAWY: PROF. DR HAB. INŻ. MAREK GORGOŃ
PROMOTOR POMOCNICZY: DR INŻ. TOMASZ KRYJAK

PRACA WYKONANA: AKADEMIA GÓRNICZO-HUTNICZA IM. STANISŁAWA STASZICA W KRAKOWIE, WYDZIAŁ ELEKTROTECHNIKI, AUTOMATYKI, INFORMATYKI I INŻYNIERII BIOMEDYCZNEJ, KATEDRA AUTOMATYKI I ROBOTYKI

KRAKÓW 2024

To supervisors, Professor Marek Gorgon and Doctor Tomasz Kryjak, for many years of cooperation, guidance and supervision in the preparation of this dissertation,
To my mother Monika, for making it possible for me to follow this path,
To my best friend and husband Sebastian, for constant support and patience -

- thank you.

Promotorom, prof. dr. hab. inż. Markowi Gorgoniowi oraz dr. Tomaszowi Kryjakowi, za wieloletnią współpracę, opiekę i nadzór nad przygotowaniem niniejszej rozprawy,
Mamie Monice za umożliwienie podjęcia tej drogi,
Najlepszemu Przyjacielowi i Mężowi Sebastianowi za wsparcie i cierpliwość -

- dziękuję.

Abstract

Designing low-latency, high-accuracy and energy-efficient vision systems requires an approach characterised by consistency between the algorithmic solution and the target platform on which it will run. The main challenge is to implement highly memory and computationally complex neural networks in small low-power devices, such as SoCs (System on Chips), FPGAs (Field-Programmable Gate Arrays) or ultimately ASICs (Application-Specific Integrated Circuits). A number of methods are therefore being used to reduce this complexity, either by reducing the size of the model or by simplifying the computations, particularly multiply and accumulate operations: one of these is the integer quantisation of network parameters. Linear 8-bit quantisation has become a certain standard, while it is other special schemes that will allow to develop advanced systems with much higher performance. This research proposes methods for training and efficient hardware implementation of neural networks quantised to weights of powers of two, allowing the development of 4-bit models with efficiency comparable to full-precision networks and, at the same time, allowing a significant reduction in computational complexity by changing the multiplication to a bit-shift operation. Furthermore, the possibility of using different quantisation schemes (linear, logarithmic, binary and mixed) for neural networks used in advanced vision systems was also analysed, proposing models dedicated to low-power devices, for classification, tracking and object detection tasks. As a result of appropriate algorithm design and implementation in embedded platforms (so-called hardware aware algorithm co-design), it is shown that proper model quantisation methods enable the implementation of complex vision systems with high accuracy, low latency and low power consumption.

Streszczenie

Projektowanie energooszczędnych systemów wizyjnych o małej latencji i wysokiej skuteczności działania wymaga podejścia charakteryzującego się spójnością między rozwiązaniem algorytmicznym i docelową platformą, na której zostanie ono uruchomione. Głównym wyzwaniem jest implementacja złożonych pamięciowo-obliczeniowo sieci neuronowych w niewielkich urządzeniach małej mocy, jak SoC (System on Chip), FPGA (Field-Programmable Gate Array) czy docelowo ASIC (Application-Specific Integrated Circuit). Stosuje się zatem szereg metod pozwalających na redukcję tej złożoności, poprzez zmniejszenie rozmiarów modelu lub uproszczenie obliczeń, w szczególności operacji mnożąco-akumulujących: jedną z nich jest kwantyzacja parametrów sieci do liczb całkowitych. Pewnym standardem stała się kwantyzacja liniowa 8-bitowa, natomiast to inne, specjalne schematy pozwolą na realizację zaawansowanych systemów o znacznie wyższej wydajności. W ramach przeprowadzonych badań zaproponowano metody uczenia i wydajnej implementacji sprzętowej sieci kwantyzowanych do wag o wartościach potęg dwójki, pozwalając na realizację modeli 4-bitowych o skuteczności porównywalnej do sieci pełnej precyzji i jednocześnie umożliwiając znaczną redukcję złożoności obliczeniowej poprzez zmianę mnożenia na operację przesunięcia bitowego. Ponadto przeanalizowano też możliwości użycia różnych schematów kwantyzacji (liniowej, logarytmicznej, binarnej i mieszanej) dla sieci neuronowych używanych w zaawansowanych systemach wizyjnych, proponując modele dedykowane urządzeniom niewielkiej mocy, dla zadań klasyfikacji, śledzenia oraz detekcji obiektów. W wyniku odpowiedniego projektowania algorytmów i ich implementacji w platformach wbudowanych pokazano, że odpowiednie metody kwantyzacji modeli umożliwiają realizację systemów o wysokiej skuteczności działania, małej latencji i niskim poborze energii.

Contents

Acronyms and Abbreviations 2

1. Introduction 4

2. Synthetic overview of the dissertation 11

2.1. Powers-of-Two quantisation 11

2.2. Other quantisation schemes 15

2.3. Conclusions 18

3. Publication Series - Full Texts 21

3.1. XNOR CNNs in FPGA: real-time detection and classification of traffic signs in 4K – a demo 27

3.2. Optimisation of a Siamese neural network for real-time energy efficient object tracking 29

3.3. Quantised Siamese tracker for 4K/UltraHD video stream – a demo 31

3.4. Exploration of hardware acceleration methods for an XNOR traffic signs classifier 33

3.5. Power-of-two quantization for low bitwidth and hardware compliant neural networks 35

3.6. Towards real-time and energy efficient Siamese tracking - a hardware-software approach 37

3.7. Energy efficient hardware acceleration of neural networks with power-of-two quantisation 39

3.8. Power-of-Two Quantized YOLO Network for Pedestrian Detection with Dynamic Vision Sensor 41

3.9. PowerYOLO: Mixed Precision Model for Hardware Efficient Automotive Detection with Event Data 43

Acronyms and Abbreviations

ADAS Advanced Driver Assistance Systems

AI Artificial Intelligence

APoT Additive Powers-of-Two

ASIC Application-Specific Integrated Circuit

BAC Bitshift and Accumulate

CPU Central Processing Unit

DVS Dynamic Vision Sensor

eGPU Embedded Graphics Processing Unit

FPGA Field-Programmable Gate Array

FPS Frames Per Second

GAN Generative Adversarial Network

GFLOPS Giga Floating Point Operations per Second

GPU Graphics Processing Unit

ILSVRC ImageNet Large Scale Visual Recognition Challenge

LiDAR Light Detection and Ranging

LLM Large Language Model

MAC Multiply and Accumulate

ML Machine Learning

POC Proof of Concept

PoT Powers-of-Two

QAT Quantization-Aware Training

RCNN Region-based Convolutional Neural Network

RL Reinforcement Learning

ROI Region of Interest

SoC System on Chip

SOTA State-of-the-Art

STE Straight Through Estimator

TinyML Tiny Machine Learning

YOLO You Only Look Once

1. Introduction

The recent years of advances in the development of applications using so-called Artificial Intelligence (AI) place humanity at the verge of a technological revolution, although it is actually not yet entirely clear *of what kind*. Leaving beyond discussion the voices of the overly optimistic enthusiasts and the fatalists on the other extreme, it is very likely that in the not too distant future we shall have the opportunity to establish a kind of symbiosis between natural (exhibited by the human species) and artificial intelligence. The current revolution in many ways resembles the advent of the Internet, and many of the voices accompanying that groundbreaking invention seem strangely relevant today, in the context of the AI and, especially, the huge (!) amount of data we must deal with. Noted in 1997 [Reirer, 1997], "The good news is that everything is widely available. The bad news is that everything is widely available, (...) if you search for an item even with a very good search engine, most of the results you get will be irrelevant", originally referring to the Internet, can be easily used to comment on so popular nowadays applications based on generative models like ChatGPT or DALL-E. At the same time it is evident, drawing from more than 25 years of experience, that although indeed Internet search engines generate a lot of information that is useless from the user's point of view, the technology today is much more mature, and our knowledge (and perhaps already intuition, resulting from exposure to technology from an early age – especially for generations who do not remember the days without the Internet) allows us to use these tools efficiently. The same is likely to happen to the aforementioned AI based applications, or rather their next generations. Yet another challenge is the question of the credibility of the data – a second quote from the 1997 source can be used here: "You used to spend hours getting the information you needed. Now you spend hours verifying the information you have gotten." Does this not sound like something so up-to-date that it could have been posted just yesterday on some social network in the context of, say, another DeepFake (albeit the infamous video staging the surrender of the Ukrainian President circulating on the Internet in 2022)? It is up to the user of the language model to verify the obtained information, as it may be a hallucination, or refer to non-existent sources, or simply be wrong (if only because of an inaccurately formulated question).

Just as in the past it was difficult to predict how the world would change due to widespread Internet access, it is now impossible to determine exactly what transformation the ongoing AI revolution will bring. It is known, however, that in addition to the above-mentioned challenges, mankind has to face others – in particular when applying AI to autonomous systems, which now hold more responsibility than ever before. Autonomous vehicles (military drones or cars) are an obvious example of such, and so are Advanced Driver Assistance Systems (ADAS). The machines are (or will be) often equipped with cameras and other sensors – various types of radars, thermal imaging or ultrasonic devices. They form the vehicle's perception system, providing necessary information about the environment. Data collected from some non-camera sensors can also be analysed using digital image processing (or similar) algorithms: in particular, from Light Detection and Ranging (LiDAR) sensors generating a 3D point cloud, or event cameras – Dynamic Vision Sensor (DVS) – recording an event cloud. Proper analysis of the collected data enables making appropriate decisions for controlling the vehicle. In particular, for such analysis, one should consider using neural networks (mainly convolutional, but also graph or so-called transformers), specialised for finding and reproducing patterns in detection, tracking or segmentation tasks. The control system itself can be

designed in a *standard* way, i.e., algorithm is trained *offline* on huge amounts of data representing real-world scenarios. The other option uses Reinforcement Learning (RL), where an agent (autonomous vehicle) explores the environment in some pseudo-random way, and learns which actions are profitable with the help of a reward signal.

At the same time, in the considered applications it is necessary to operate with low latency, i.e. the response of the system must be received with minimal delay: data must be processed at a speed that guarantees timely delivery of the necessary information relevant to the control, supervision or monitoring of the external process. Such systems are referred to as real time vision systems [Gorgoń, 2013]. Finally, the whole system should operate on devices with limited energy budget, often battery-powered. Simultaneous fulfilment of real-time and high-accuracy conditions requires using proper devices, with adequate computing performance – in particular, with multi-threading, e.g., multiple Central Processing Units (CPUs) or Graphics Processing Units (GPUs). However, adding a third requirement – energy efficiency – will define a problem that forces a shift in approach of algorithm development from a paradigm of gigabyte-sized solutions (the *large* neural networks that form the basis of the most exciting applications, treated as the starting point of this introduction) to compact algorithms running in low-power embedded devices.

Low-power embedded devices include microcontrollers (and microprocessors), System on Chips (SoCs) and Field-Programmable Gate Array (FPGA) platforms, dedicated Application-Specific Integrated Circuit (ASIC) chips, but also Embedded Graphics Processing Units (eGPUs) or neuromorphic platforms. The characteristics of these devices are different: microcontrollers and eGPUs are general-purpose processors, i.e. execution of the set of instructions (an algorithm) is bounded to the underlying electronic circuit architecture. This implies that the possibilities for optimising the computations are limited, but due to the versatility of such platforms (and relative ease of programming), and access to libraries that can optimise the software for a particular hardware architecture, such choice is often justified and sufficient – particularly when using appropriate methods to optimise the neural network model itself. Obviously, eGPUs are also characterised by the possibility of significant parallelisation of computations. The FPGA platforms (including SoCs) and dedicated ASICs allow the hardware architecture to be adapted to the designed solution. Once the ASIC has been fabricated, the further possibilities of configuring the device architecture are limited, while the architecture of FPGAs and SoCs can be configured (reconfigured) after the electronic system is built, and even during the execution of computing tasks. The ASIC chip, which architecture is best suited to the computational task and optimisation, will therefore have an even lower power consumption than the FPGA. Whereas the ASIC is the so-called end product, in the early stages it is the FPGA chip that is often initially used as the Proof of Concept (POC) for a given architecture. FPGA is also preferential if the algorithm architecture may change during product life-cycle. In both of these cases, one is dealing with a completely different approach to *programming* – one does not define the input processing instructions, but the entire electronic circuit architecture, ultimately a dataflow-based architecture. This opens up new possibilities in the efficient implementation of neural networks. The last group of mentioned devices are neuromorphic platforms, which, due to their specificity, are difficult to categorise into any of the distinguished groups. Firstly, these devices are not currently commercially available, but rather are the subject of research at several universities and corporations. While they are not reconfigurable in the same way as FPGAs, their architecture is closely related to the only type of model they support – spiking neural networks. Processing data event-driven (coherently with the 3rd generation spiking neuron model), it is expected to guarantee very low power consumption and fast data processing.

Regardless of the choice of computational platform (summarised in Table 1.1), there are number of methods for reducing the memory-computational complexity of neural networks that can be used independently, such as quantisation, pruning and other simplifications of operations. Furthermore, it is increasingly common for the standard AI/Machine Learning (ML) tools (e.g. PyTorch) to have extensions that allow appropriate compression and fast inference of neural networks on general-purpose processors, while optimising how the computations are performed for the appropriate hardware (i.e. drawing on the already established architecture in an efficient manner).

Table 1.1: The differences between hardware platforms used for deep neural networks inference: multithreading (ability to parallelise computations in neural networks), reconfigurability (**R** – flexibility of defining connections between electronic elements) and programmability (**P** – uploading virtually *any* program understood as a sequence of instructions described using a standard programming language, e.g. C, Python, etc.).

Platform	**Power*** **[W]**	**Multithreading**	**R**	**P**
CPU	65-150	Limited possibilities	No	Yes
GPU	250-400	Yes (massive)	No	Yes
eGPU	5 - 10	Yes (>60x lower than for GPU)	No	Yes
FPGA/SoC-FPGA	< 10	Yes, limited only by the number of electronic elements	Yes	Partial (for SoC-FPGA)
ASIC	< FPGA†	Yes, by design	No	No
Neuromorphic	1‡	Yes	No	Yes

* indicative values – may vary for specific platforms

† expected <1 [W]: in this context it's best to compare ASICs to similar architectures designed in FPGAs

‡ according to Loihi2 documentation [Intel Labs, 2021]

However, the most can be achieved by combining these two areas – software (model-based) optimisations with the way computations are performed. As a whole, the aforementioned methods are referred to as Tiny Machine Learning (TinyML) methods.

It is worth at once to mention the reason for the necessity of model-based optimisations, and at the same time to show how the development of hardware platforms (in this particular case, the GPUs) has influenced the development of neural networks. While many breakthroughs have been made in the area of machine learning algorithms over the years, perhaps the most relevant for the field of image processing are those related to the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) [Russakovsky et al., 2015]. Firstly, until the launch of the ImageNet database, the efforts were oriented towards improving models and algorithms, without a proper emphasis on the manifold of data nowadays known as *training data*. ImageNet was the first large-scale database and quickly became a certain benchmark standard for digital image processing tasks, while also underpinning the now-classic assumption that AI algorithms go hand in hand with *big data*. Secondly, in the 2012 edition of ILSVRC, a score below an error of 25% was reached for the first time (with a difference to the next place of 9.8 percentage points). This was achieved using the AlexNet model [Krizhevsky et al., 2012], a deep convolutional neural network, the architecture that had hitherto been dismissed by many as impractical – deep learning (i.e., learning networks with multiple layers) was not feasible with CPUs, and the high-end GPUs at the time had little memory (around 3GB). As network depth was crucial for efficiency, AlexNet training was distributed across two GPUs, splitting the network in half and ensuring adequate communication between nodes. This, and a number of other details in the architecture (drawing on the first convolutional neural networks from the LeNet family [LeCun et al., 1989]) demonstrated not only the capabilities of deep models, but was a key to further incredible advances in the field of artificial intelligence and its use in machine vision, along with the parallel development of now publicly available libraries for the efficient implementation of these solutions in GPUs. Next generations of general-purpose GPUs with ever-increasing computing power and internal memory resources were also not without influence. Thus, starting with the classification task, deep convolutional neural networks began to dominate virtually all digital image processing tasks: detection (You Only Look Once (YOLO) family of networks), tracking (Region-based Convolutional Neural Networks (RCNNs), Siamese networks), segmentation (YOLO networks, Siamese networks, endcoder-decoder architectures), image generation (encoder-decoder architectures, Generative Adversarial Networks (GANs), dif-

fusion models). Today, models using transformer architectures ([Kolesnikov et al., 2021]), which use tokenisation mechanisms, self-attention and classical fully connected feedforward neural networks, are gaining popularity.

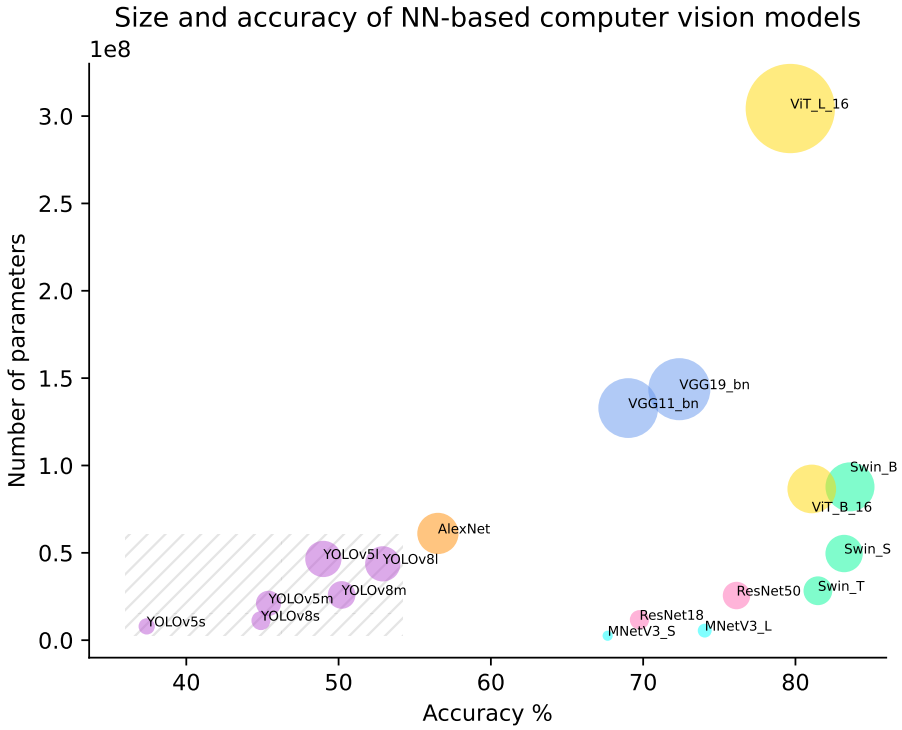

Figure 1.1: Relationship between the number of parameters and model accuracy: for the classification task (AlexNet, VGG networks, ResNet, small MobileNet models, ViT transformers and Swin) for the ImageNet dataset; for the detection task (YOLO networks, marked with grey background) mAP50-95 on the COCO dataset. The classification models highlighted in the diagram are often backbones to more advanced systems.

Figure 1.1 shows the number of parameters and accuracy of selected neural network models proposed from 2012 (AlexNet), up to 2021 (transformer-based architectures – ViT and Swin), or in the case of the highlighted YOLO detectors, 2023. Architectures designed for the classification task are often the backbone for other models used in more complex applications (in Siamese networks, encoders, decoders, etc.), so it is worth looking at the practices for this *simplest* task. After the publication of the ground-breaking AlexNet architecture, much work was devoted to designing even deeper (and therefore larger) architectures, such as the VGG family of networks. Although increasing the depth of the network initially provided a stable accuracy raise, another obstacle was the problem of vanishing gradients in the hidden layers (during back-propagation, the propagated gradients become smaller and smaller, up to tiny values, causing no change to the weights of neural network), and the problem of accuracy saturation-degradation when adding even more layers. In response, the ResNet family with residual connections [He et al., 2016] was developed. At the same time, more and more attention began to be focused on network size, designing solutions with small architectures, such as MobileNet. Similarly, different backbone sizes were used in the YOLO family, as well as a number of other procedures involving, among other things, the preprocessing of training data, so that subsequent generations gained in accuracy. As a general rule of thumb, however, a larger model is more likely to perform better in terms of accuracy - taking into account any architecture tricks that improve the training process (such as residual connections). Although the size of the discussed models is definitely small compared to the Large Language Models (LLMs), in the context of embedded solutions representing a significant part of vision system solutions, it is large enough to cause real problems in their implementation in real

applications: autonomous vehicles, driver assistance systems, wearable tech and others. The number of parameters translates into the number of performed operations, mainly of the Multiply and Accumulate (MAC) type, counted in Giga Floating Point Operations per Second (GFLOPS). For example, ResNet18 performs 1.81 GFLOPS during inference, while SwinB performs 15.43 GFLOPS. The implementation of neural networks in low-power devices is therefore limited primarily by their size – methods must therefore be used that (1) reduce the number or size of parameters, so that the model occupies memory measured in kilobytes rather than megabytes or even gigabytes; and (2) allow efficient implementation of these algorithms, using appropriate parallelisation of operations, simplifying even the basic processing elements as much as possible.

One should also reflect on the more global issue of the environmental impact of AI solutions – according to a report published by Stanford University in 2023 [Maslej et al., 2023], 25 times more carbon dioxide was emitted during the training of the BLOOM model (alternative to ChatGPT) than is attributed to a single passenger travelling by air from New York to San Francisco. This is the influence of both the immensity of the training data (*big data*), the complexity of the model and thus the performed computations, as well as the energy characteristics of GPU, which is currently the most common platform of choice for accelerating these and similar models – and it is not an isolated case.

Similar problems can be spotted with generative models inference – IEEE Spectrum [Wells, 2023] cites that a single interaction with LLM can lead to power consumption comparable to lighting an LED bulb for an hour; the [Samsi et al., 2023] shows that in order to complete a meaningful interaction (as a sequence of consecutive prompts) with the 65B LLaMA model, a minimum of 8 V100 GPUs, each with 32 GB RAM, or 4 A100 GPUs, each with 80 GB RAM, is required (in the mentioned experiments, the maximum power consumption per GPU is limited to 250W).

It is therefore also essential to examine what accompanies this ongoing revolution, and what will become increasingly important even if one day the market and society saturates with yet more applications of AI in general, or generative AI in particular, and which is already relevant, observing the constant attempts to automate virtually every area of life, or the usual drive to improve its quality with small devices collecting and analysing vast amounts of data. It is thus necessary to develop methods to use complex and accurate artificial intelligence models in applications with a limited computational budget, resulting from the need to use low-power devices (in particular battery-powered devices), or directly from the impossibility of using large GPU farms and a need to carry out computations on personal computers (devices). Or the already mentioned ecological factors. The development of such methods may not be sufficient on its own, and it is only when we move away from general-purpose platforms to specialised hardware, and with hardware aware algorithm co-design, that we will finally be able to master efficient implementation of highly memory-computational complex deep learning algorithms.

Research problems, contributions and scientific novelty

The goal of the conducted research was to analyse and develop methods enabling the implementation of functional real-time vision systems using deep neural network algorithms in low-power devices. In particular, it involved proposing methods of reducing the memory and computational complexity of algorithms, to a *size* suitable for relatively small FPGA platforms (or ultimately dedicated circuits), while maintaining the accuracy of the core solution, and using appropriate organisation of computations (performed in parallel) to meet the low latency requirements. This dissertation takes the form of a series of publications that describe the results of the conducted research in relation to the leading research hypothesis: ***appropriate methods for quantising the parameters of neural networks allow a significant reduction in the memory and computational complexity of the models, while guaranteeing the preservation of a high accuracy and enabling implementation of real-time vision systems in low latency and low power hardware platforms***.

The author's main contribution to the discipline of *automatics, electronic, electrical engineering and space technology* can be summarised in the following points:

1. The proposal of methods for training neural networks quantised to weights of powers-of-two values, the design of a hardware architecture of special MAC operator (and the convolution layer), taking into account the particular form of such a network, and a method of fusion of the convolution and batch normalisation layers, taking into account the particular form of such a network.

2. A series of experiments and analysis of the impact of different quantisation schemes (linear, logarithmic), with different target bit-widths, for neural networks used in advanced vision systems in embedded devices, to assess the impact on the accuracy, memory-computational complexity and energy efficiency of such systems.

Chapter 2 presents a synthetic discussion of the dissertation with the main conclusions. The full publications are included in Chapter 3, together with a table summarising the author's contributions to each article.

2. Synthetic overview of the dissertation

One method of reducing the memory-computational complexity of neural network models is to quantise the parameters and/or activations to integers. An obvious consequence of such quantisation is a reduction in the size of the model in terms of the number of bytes needed to store its parameters – the commonly used 8-bit quantisation allows a reduction in memory complexity with a factor of almost 4x (with respect to the 32-bit floating-point models normally used in high-end GPUs). Furthermore, the use of integers also has the effect of reducing computational complexity (simplifying the computational architecture), mainly for solutions on FPGAs or ASICs, but also on CPUs (e.g. even 2-3 times more Frames Per Second (FPS) for ResNet-50 in 8-bit integer precision, with relation to an optimised 32-bit floating point model, depending on the Intel Core i7 processor version, using the OpenVINO tool [Intel, 2024]) and GPUs (16x increase in computational throughput using quantisation to 8-bit integers [Wu et al., 2020]). However, especially interesting in the context of low-power and low-latency solutions are other special quantisation schemes, up to very low bit-widths, which allow the reduction of MAC operations to much simpler equivalents: for logarithmic quantisation to powers of two to bit-shifting, for binary weights to XNOR operations. Reducing the number of used electronic elements affects both the size of the system (in terms of the number of necessary/used components) and ultimately the power consumption. At the same time, it is also necessary to choose a solution that guarantees a satisfactory level of accuracy (particularly in safety-related applications), often measured against base models, i.e. the full precision, not subjected to quantisation. Linear 8-bit quantisation often results in only a slight decrease in accuracy, while the use of lower bit-widths for this scheme is generally associated with significant decreases. This may be different for logarithmic quantisation, which by design is intended to model the distribution of quantisation levels in a way closer to the distribution of weights in the convolution or fully connected layer. For this reason, it is often possible to maintain high accuracy even at lower bit-widths. For the most radical form of quantisation, i.e. binary networks, high accuracy is achievable for relatively *simple* problems such as well-defined classification, especially when using larger architectures (and thus guaranteeing quite a bit of parameter redundancy).

Overview of the scope of the research with key results is presented below.

2.1. Powers-of-Two quantisation

This research proposes two methods for training the Powers-of-Two (PoT) neural networks (based on Straight Through Estimator (STE) and by introducing Adaptive Learning Rate to compensate for unequal distances between quantisation levels), with the best results achieved with the method drawing on the STE approach. First, the full precision model is trained and then the Quantization-Aware Training (QAT) is performed in a standard loop, running a forward pass with weights quantised according to the Equation 2.1 (the weights w are scaled to the interval $[-1, 1]$, $bitwidth$ is the target bit-width without the sign bit, FSR is the Full Scale Range, specifying the quantisation extremes). Backpropagation is then run on floating-point numbers and, after updating the weights, the

Table 2.1: Training results for different ResNet models and ImageNet dataset, for different quantisation methods (e.g. 4L/8U means quantising convolution layers logarithmically to 4-bitwidth and quantising fully connected layers using 8-bit linear quantisation), compared to other PoT-related SOTA methods. The difference in classification performance relative to the floating-point model is given in brackets

Network	**Precision (C/FC)**	**Proposed** [Przewłocka-Rus et al., 2022]	**DeepShift** [Elhoushi et al., 2021]	**APoT** [Li et al., 2020]
ResNet18	4L/32F	69.982% (+0.224)	-	70.7% (+0.5)
ResNet18	4L/8U	69.868% (+0.11)	-	-
ResNet18	4L/4L	69.526% (-0.232)	69.56% (-0.198)	-
ResNet50	5L/5L	76.468% (+0.338)	76.33% (+0.216)	
ResNet50	4L/32F	76.404% (+0.274)	-	76.6% (+0.2)
ResNet50	4L/8U	76.384% (+0.254)	-	-
ResNet50	4L/4L	76.314% (+0.184)	-	-

model is re-quantised (and thus quantisation is performed between successive iterations).

$$
\begin{aligned}
LQ(\text{w, bitwidth, FSR}) &= \begin{cases} 0, & \text{if } w = 0 \\ 2^{\tilde{w}}, & \text{otherwise} \end{cases} \\
\tilde{w} &= \text{clip}(\text{round}(log_2(|w|)), \text{FSR} - 2^{\text{bitwidth}}, \text{FSR}) \\
\text{clip(w, min, max)} &= \begin{cases} 0, & w \leq min \\ max - 1, & w \geq max \\ w, & \text{otherwise} \end{cases}
\end{aligned}
\tag{2.1}
$$

The method was tested for a number of architectures and compared with State-of-the-Art (SOTA) solutions, showing that the proposed approach achieves results comparable (often better) to other methods using the PoT weights [Elhoushi et al., 2021] and variations of the PoT weights – Additive Powers-of-Two (APoT) [Li et al., 2020]. The APoT method is similar to the PoT scheme except that the weights are quantised to the values of the sums of powers of two, allowing a denser distribution of quantisation levels at the expense of increased computational complexity. Representative results of mentioned experiments are shown in Table 2.1, a comprehensive comparison for multiple architectures is available in article [Przewłocka-Rus et al., 2022].

In addition, a number of experiments were carried out comparing logarithmically and linearly quantised 4-bit models, showing the advantage of PoT quantisation - a summary of the results is presented in Table 2.2 - and proving the potential for significant simplifications in terms of memory-computational complexity, also for embedded devices. Firstly, it is shown that, for the classification task, the introduction of 4-bit PoT weights allows

Table 2.2: Comparison of 4-bit models quantised logarithmically (to PoT values) and linearly

Network	Baseline	Log STE	Uniform
ResNet20 CIFAR 10	91.77%	91.6% (-0.17)	91.22% (-0.55)
ResNet20 CIFAR 100	68.68%	68.51% (-0.17)	65.47% (-3.21)
ResNet18 ImageNet	69.76%	69.868% (+0.11)	57.83% (-10.93)

the compression of weights almost twice as much as in case of classical 8-bit quantisation, while maintaining the accuracy of the 8-bit (or even floating-point 32-bit) network.

A dedicated hardware module for MAC operation was also designed, in which the multiplication operation is replaced by a bit-shift operation (schematic shown in the left-hand side of Figure 2.1). A comparison was made between hardware-designed computational elements for 4-bit weights and 8-bit activations for linear, PoT, APoT quantisation, and with the standard computational element for a model with weights and activations both linearly quantised to 8-bit width. Thus, it is shown that the module for 4-bit PoT quantisation uses the smallest number of electronic components, which also translates into several times lower energy requirements: 6x for PoT versus 8x8 linear MAC and 2x versus 4x8 linear MAC. Extensive description of the proposed training method, hardware design, experiments and more quantitative results are included in the paper [Przewłocka-Rus et al., 2022] (TinyML Research Symposium 2022, USA), prepared during a research internship at Meta Reality Labs (formerly Facebook Reality Labs) in 2022.

Next, a hardware implementation of the PoT convolution layer, with appropriate weight encoding, in the Zynq UltraScale+ MPSoC ZCU104 platform was proposed (a simplified schematic is shown in the right-hand side of the Figure 2.1). It was shown that a layer using a MAC module dedicated for PoT quantisation - named a Bitshift and

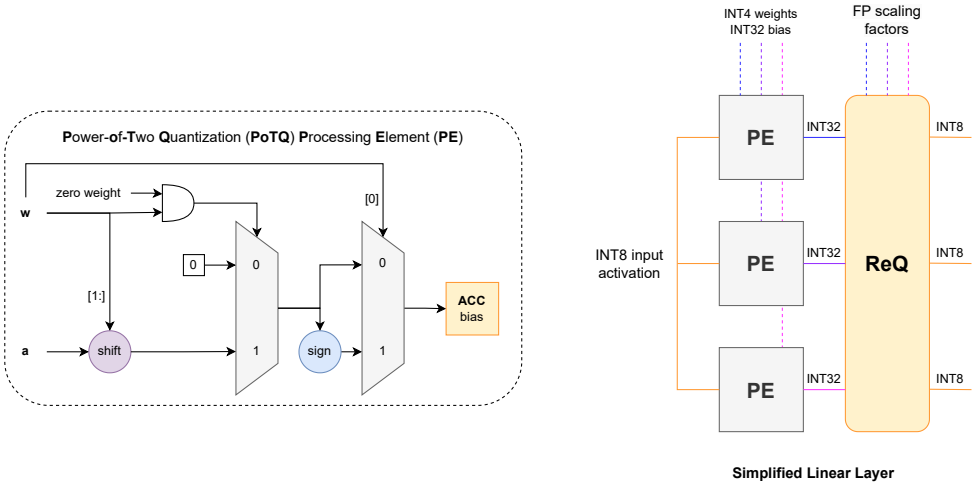

Figure 2.1: Simplified quantised neural network layer with a MAC element based on logarithmic PoT quantisation. After loading the appropriate weights and biases from memory, the elements process the activations from the previous layer using an efficient bit-shift operation. The output activations are then re-quantised using appropriate scaling values.

Accumulate (BAC) module - uses about 0.6 of the energy required for the standard MAC layer, for 8-bit activations

and 4-bit weights, while increasing possible operation frequency of the chip.

To extend the generality of PoT quantisation towards the possibilities guaranteed with linear quantisation, methods of pruning and layer fusion were proposed to enable further reductions in memory-computation complexity.

For low bit-widths, linear quantisation introduces automatic pruning of values smaller than the lowest quantisation level, without affecting the final number of quantisation levels. This is different for logarithmic quantisation, where automatic pruning has no place, and direct zeroing of weights with the lowest values leads to a reduction in the number of quantisation levels (which can have a significant impact on the accuracy of the solution). To enable correct pruning for logarithmic quantisation, a double normalisation method was therefore proposed, introducing a *death zone* of a fixed width, moving the smallest quantisation level further away from zero. With this trick, it is possible to prune weights similarly to linear quantisation: without reducing the number of quantisation levels, and, as shown in the experiments, for redundant networks, such as ResNet20 for the CIFAR10 dataset, it is possible to prune more than 40% of connections without any loss in network accuracy (with 70% of pruned weights, the loss was less than a percentage point). A detailed description of the pruning method, as well as experiments with convolution layer acceleration using the BAC module, are available in the paper [Przewłocka-Rus and Kryjak, 2022a], published at the International Conference on Computer Vision and Graphics (ICCVG) 2022.

Furthermore, to reduce the number of computations during inference, fusion of convolution layers with batch normalisation layers is usually performed, according to the Equation (2.2):

$$y_{bn} = \frac{x_{conv} - \mu}{\sqrt{\sigma^2 - \epsilon}}\gamma + \beta = \frac{\gamma}{\sqrt{\sigma^2 - \epsilon}}x_{conv} - \mu\frac{\gamma}{\sqrt{\sigma^2 - \epsilon}} + \beta \tag{2.2}$$

where x_{conv} is the output of convolution layer. Therefore weights w and bias b are properly modified for inference: $w_{fused} = \frac{\gamma}{\sqrt{\sigma^2-\epsilon}} * w$ oraz $b_{fused} = b - \mu\frac{\gamma}{\sqrt{\sigma^2-\epsilon}} + \beta$. Obviously, such a modification does not affect the accuracy of the neural network, but it reduces the number of multiplication and addition operations, and the number of parameters. Introducing exactly such a fusion into a PoT network would cause the weights to no longer be in the form of powers of two, and therefore the important property enabling the use of BAC computational elements would be lost. To reduce the number of operations it is therefore proposed to introduce two different manipulations leading to similar simplifications. As the bias is not quantised logarithmically, it can be successfully modified according to the scheme known from standard fusion method. However, the weight-related multiplier is instead merged with the scaling factor of the quantisation operation. In this way, all additional calculations associated with the batch normalisation layer are reduced to a minimum, with only a slight increase in memory complexity compared to the model after full/standard fusion – instead of a single scaling factor for the entire layer, each output map has a separate one.

All of the proposed methods allow to design efficient computer vision systems with a SOTA ratio of model complexity and computational architecture to solution accuracy, as shown in the example of the mixed precision model (4-bit PoT weights, 8-bit linearly quantised activations) of PowerYOLO for pedestrian and vehicle detection. The proposed solution achieved an accuracy of mAP50-95 0.301 (8.3% decrease with relation to the baseline model), while reducing the size by 8x, and introducing significant simplifications in the computational architecture by changing the multiplication operation to a bit-shift operation. Detailed descriptions of the methods and quantitative results are available in [Przewłocka-Rus and Kryjak, 2023] published at the 26th Euromicro Conference on Digital System Design (DSD) in 2023, and [Przewłocka-Rus et al., 2024] published at the 27th Euromicro Conference on Digital System Design (DSD) in 2024.

In summary, new methods for training PoT neural networks and other methods to further reduce their memory-computational complexity were proposed, analogous to the practices standard for linear quantisation (pruning, layers fusion), but adapted to this special logarithmic quantisation scheme. The first such comprehensive study on the implementation of the PoT neural networks in embedded devices was conducted, demonstrating the energy and

Table 2.3: Post-implementation resources usage for custom XNOR convolutional neural network accelerator implemented in Zynq UltraScale+ MPSoC ZCU104

Resource	**Utilisation**	**Available**	**Utilisation %**
LUT	79302	230400	34.42
LUTRAM	350	101760	0.34
FF	44170	460800	9.59
BRAM	152	312	48.72
DSP	902	1728	52.20

frequency gains compared to a standard linear quantisation scheme, resulting from the use of the BAC element. It is also shown that PoT quantisation for low bit-widths maintains accuracy on par with the full precision models, in opposition to linear quantisation. The results as a whole demonstrate the viability of using the proposed methods in the design of real-time vision systems for low-power embedded devices.

2.2. Other quantisation schemes

A number of experiments using other quantisation schemes were also proposed: radical binary quantisation (XNOR neural networks) and linear quantisation. Special attention was given to the possibilities of accelerating such models in FPGA/SoC devices for advanced vision systems, showing how such applications can be implemented, and the energy gains resulting from the choice of an appropriate platform.

For the traffic sign detection system, a hardware accelerator for XNOR neural network was designed (details in [Przewłocka and Kryjak, 2019] from 2019 Conference on Design and Architectures for Signal and Image Processing (DASIP) and [Przewłocka-Rus et al., 2021] from 2021 International Conference on Computer Recognition Systems (CORES)), with activations and weights in binary form, and thus allowing the convolution operation to be reduced to one given by Equation 2.3, where W, X and N are respectively: a flattened vector of weights, input activations and the size of the convolution window.

$$Y = \text{BitCount}(X \bar{\oplus} W) << 2 - N \tag{2.3}$$

The accelerator was implemented in a semi-pipelined manner: the computations in each convolution layer are parallelised filter-wise, and each output map is written to dedicated BRAM, as shown in Fig. 2.2. For the proposed binary network architecture (similar to the LeNet5 network, and achieving an accuracy of 96.28% on a GTSRB dataset [Houben et al., 2013]), for a 32x32 input, the accelerator running on a Zynq UltraScale+ MPSoC ZCU104 platform allowed the processing of almost 450 FPS with a power consumption of 4.396W, and clock frequency 100MHz. The summary of post-implementation resources usage for the designed accelerator is presented in Table 2.3.

As the field of neural network acceleration in embedded devices is in dynamic development, *in the meantime* open-source libraries Brevitas and FINN were released, allowing the training and implementation of different precision networks in Xilinx AMD MPSoC platforms. These tools allow the use of linear quantisation and different bit-widths for weights and activations, including enabling XNOR networks. A comparison experiment was therefore carried out for the traffic sign detection system, designing a second solution with the aforementioned libraries.

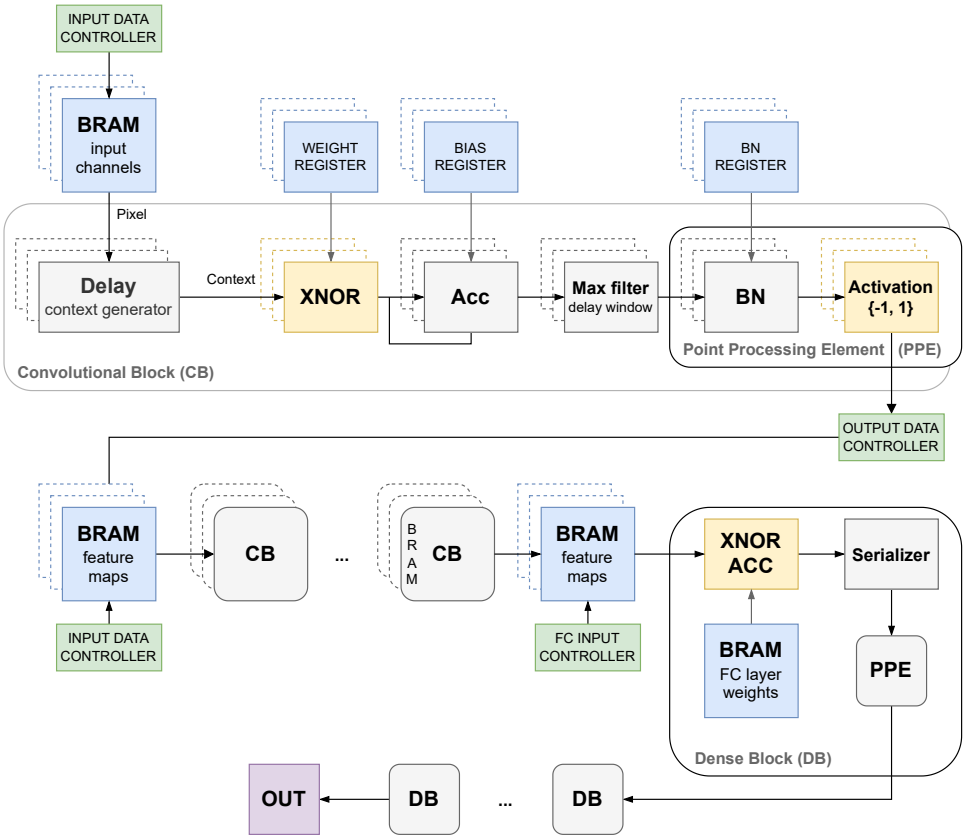

Figure 2.2: A simplified scheme of the designed semi-pipelined XNOR convolutional neural network accelerator. The proposed architecture consists of three main components: Convolutional Blocks for convolutional and batch normalisation layers computations, Dense Blocks for fully connected layers and BRAM sets for inputs and outputs from layers. The operations in layers are streamlined, and the data flow is organised using properly designed data controllers.

The network trained with the Brevitas tool achieved a lower accuracy (95%), while with the appropriate configuration of the accelerator generated by the FINN tool, a detector processing over 580 FPS with an energy consumption of 3.547W was proposed (increasing the energy efficiency by more than 1.2 with relation to the previous solution).

Using the same tools, a number of experiments on acceleration of object tracking systems were conducted, in particular for the solution marking SOTA at the start of the research, based on Siamese Neural Networks [Bertinetto et al., 2016]. Detailed descriptions can be found in ([Przewłocka et al., 2020] published at the ICCVG 2020 conference, [Przewłocka-Rus and Kryjak, 2021] published at the 31st International Conference on Field-Programmable Logic and Applications (FPL) in 2021, and [Przewłocka-Rus and Kryjak, 2022b] published at the DASIP 2022 conference. It should be emphasised that acceleration of Siamese-based tracking algorithms has not previously been the subject of research reported in the international literature. Also worth noting is the existing general disparity in the availability of software solutions (accelerated with power-hungry high-end GPUs) using complex machine learning algorithms, compared to their efficient implementation in low-power devices. The conducted research was designed to help closing this gap. The experiments showed that for redundant architectures (so-called *big* neural networks), linear quantisation of the hidden layers can positively affect the accuracy of the model. In addition, special attention was given to the challenges of direct implementation of SOTA solutions in FPGA/SoC devices and the necessity of the *hardware aware algorithm co-design* approach. An alternative Siamese network

Table 2.4: Post-implementation resources usage for Siamese neural network implemented in Zynq UltraScale+ MPSoC ZCU104 using FINN tool

Resource	Utilisation	Available	Utilisation %
LUT	154068	230400	66.87
LUTRAM	12760	101760	12.54
FF	105984	460800	23
BRAM	173	312	55.45

architecture was proposed that reduces the number of parameters 6.7 times relative to the [Bertinetto et al., 2016] benchmark solution, while still guaranteeing high tracking accuracy, with a decrease of only 6% relative to the benchmark solution. As mentioned earlier, the need to design compact architectures is driven by the limited num-

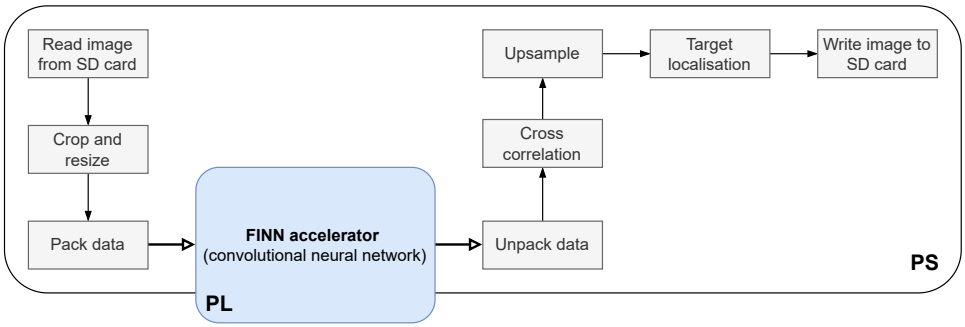

Figure 2.3: Overview of the proposed hardware-software system. A single branch of the Siamese network is accelerated using the FINN framework in PL (Programmable Logic – FPGA chip). The Python script is run on the ARM processor (Programming System – PS), handling the input and output, communicating with the accelerator and post-processing the network output.

ber of computational elements in embedded platforms. A further reduction was made by quantising the weights of the hidden layers to 4-bit integers, and the first and last layers to 8-bit integers, using linear quantisation. Thus, a small architecture in terms of memory complexity was proposed, which allowed the tracking system to operate with an accuracy close to the full-precision model (with a drop of less than 2%). The network was then accelerated on the Zynq UltraScale+ MPSoC ZCU104 platform achieving almost 50 FPS (for input size 256x256) with a power consumption of 5.5W, and clock frequency 100MHz. The resources usage is summarized in Table 2.4.

The full tracking algorithm, simplified to single-scale Region of Interest (ROI) processing, partially in the processor part of the Zynq chip, achieves 17 FPS and consumes 5.5W, compared to the original system running on the NVIDIA GeForce GTX Titan X with a power consumption of 250W (about 45 times more)) and achieving 83 FPS. The designed computational architecture for real time Siamese tracking with a neural network hardware acceleration is presented in Fig. 2.3: additional pre- and postprocessing operations, such as cropping ROI, cross correlating known object features with network's output and upsampling are done using the processor part of SoC platform.

2.3. Conclusions

Extensive descriptions of the research and results are available in the relevant publications gathered in the Chapter 3, but nonetheless, as a summary of the conducted research and the proposed methods, two important facts are worth emphasising. Firstly, as was shown with the work on logarithmic PoT quantisation, special quantisation schemes tailored to the distribution of weights in the layers of neural networks can not only guarantee a much lower decrease in accuracy than for linearly quantised models, with respect to the full precision model, but also allow for significant simplifications in the computational architecture of neural network accelerators, leading to energy gains. Secondly, methods for acceleration of advanced vision systems in FPGA/SoC devices were also developed, proposing appropriate modifications of the SOTA solutions in such a way that their implementation in embedded devices is feasible, also with the use of existing frameworks.

In summary, the author's major original contributions include:

1. Methods for training neural networks with weights quantised to powers of two values, enabling accuracy on par with full precision models even for low bit-width (4 bits), which is difficult to achieve using linear quantisation.

2. Design of a hardware architecture that implements a MAC operation for PoT networks using bit-shifting instead of multiplication. The proposed single computational element for 4x8 quantisation (width of weights x width of activations) allows a reduction in energy requirements of 2x relative to the module for a linearly quantised model with the same bit widths.

3. Design of a hardware architecture for the PoT convolution layer that uses about 0.6 of the energy required for the standard layer and increases the operating frequency range.

4. Method of convolution and batch normalisation layers fusion, taking into account a special form of PoT neural network.

5. A series of experiments and analysis of the impact of different quantisation schemes (linear, logarithmic), with different target bit-widths, for neural networks used in advanced vision systems, to assess the impact on the accuracy, memory-computational complexity and energy efficiency of such systems.

6. Design of software-hardware advanced and energy-efficient neural networks-based vision systems: traffic sign detection, object tracking and pedestrian and vehicle detection.

The proposed quantisation methods and their appropriate use to reduce the memory and computational complexity of the selected neural network models, together with appropriate computational architecture, enable the implementation of real-time vision systems in low-power devices, proving the validity of the thesis. The results of the above described research were published in a series of 9 publications summarised in Table 2.5, together with the number of citations (total: 41 (37), as of 30th of July 2024, the number without self-citations is given in brackets).

Table 2.5: Simplified list of publications of the presented series, with number of citations (without autocitations in brackets)

Reference	**Title**	**Citations**
[Przewłocka and Kryjak, 2019]	*XNOR CNNs in FPGA: real-time detection and classification of traffic signs in 4K – a demo*	1 (1)
[Przewłocka et al., 2020]	*Optimisation of a Siamese neural network for real-time energy efficient object tracking*	6 (5)
[Przewłocka-Rus and Kryjak, 2021]	*Quantised Siamese tracker for 4K/UltraHD video stream – a demo*	0 (0)
[Przewłocka-Rus et al., 2021]	*Exploration of hardware acceleration methods for an XNOR traffic signs classifier*	0 (0)
[Przewłocka-Rus et al., 2022]	*Power-of-two quantization for low bitwidth and hardware compliant neural networks*	25 (23)
[Przewłocka-Rus and Kryjak, 2022b]	*Towards real-time and energy efficient Siamese tracking – a hardware-software approach*	4 (4)
[Przewłocka-Rus and Kryjak, 2022a]	*Energy efficient hardware acceleration of neural networks with power-of-two quantisation*	3 (2)
[Przewłocka-Rus and Kryjak, 2023]	*Power-of-Two Quantized YOLO Network for Pedestrian Detection with Dynamic Vision Sensor*	2 (2)
[Przewłocka-Rus et al., 2024]	*PowerYOLO: Mixed Precision Model for Hardware Efficient Automotive Detection with Event Data*	n/a

3. Publication Series - Full Texts

This chapter contains the full texts of the publications that form this dissertation. The publications are arranged chronologically and are preceded by a table listing the author's contributions to each article.

Table 3.1: Listing of the author's contribution to each publication in the presented series

Publication	**Contr. %**	**Detailed contribution**
Dominika Przewłocka, Marcin Kowalczyk and Tomasz Kryjak, **'XNOR CNNs in FPGA: real-time detection and classification of traffic signs in 4K – a demo'**, in *DASIP 2019: Conference on Design and Architectures for Signal and Image Processing*, 16–18 October 2019, Montréal, Canada	75%	• development and training of binary (XNOR) traffic sign classifier suitable for implementation in embedded devices • hardware implementation of traffic sign detection algorithm in FPGA device • hardware implementation of convolution and fully connected blocks specific for XNOR networks • analysis of the results and preparation of the publication
Dominika Przewłocka, Mateusz Wąsala, Hubert Szolc, Krzysztof Błachut and Tomasz Kryjak, **'Optimisation of a Siamese neural network for real-time energy efficient object tracking'**, in *Chmielewski, L.J., Kozera, R., Orłowski, A. (eds) Computer Vision and Graphics. ICCVG 2020. Lecture Notes in Computer Science, vol 12334. Springer, Cham.*, DOI10.1007/978-3-030-59006-2_14	50%	• proposition of appropriate architecture of neural network • scheduling and analyzing results of all experiments • preparation of the publication

Continued on next page

Table 3.1: Listing of the author's contribution to each publication in the presented series (Continued)

Publication	Contr. %	Detailed contribution
Dominika Przewlocka-Rus and Tomasz Kryjak, **'Quantised Siamese tracker for 4K/UltraHD video stream – a demo'**, in *2021 31st International Conference on Field-Programmable Logic and Applications (FPL)*, 30 August - 3 September 2021, Dresden, Germany. E-ISBN 978-1-6654-3759-2, DOI10.1109/FPL53798.2021.00089	90%	• development and training of quantised Siamese neural network • designing hardware architecture of Siamese neural network based tracker for 4K/UHD data stream • comparative implementation with FINN tool • analysis of the results and preparation of the publication
Dominika Przewłocka-Rus, Marcin Kowalczyk and Tomasz Kryjak, **'Exploration of hardware acceleration methods for an XNOR traffic signs classifier'**, in *Choraś, M., Choraś, R.S., Kurzyński, M., Trajdos, P., Pejaś, J., Hyla, T. (eds) Progress in Image Processing, Pattern Recognition and Communication Systems. CORES IP&C ACS 2021. Lecture Notes in Networks and Systems, vol 255. Springer*, DOI:10.1007/978-3-030-81523-3_4	75%	• design and implementation of hardware accelerator of XNOR neural networks • development and training of XNOR traffic sign classifier • hardware acceleration of traffic sign classifier with the designed architecture and FINN tool; comparison of both approaches • analysis of the results and preparation of the publication

Continued on next page

Table 3.1: Listing of the author's contribution to each publication in the presented series (Continued)

Publication	Contr. %	Detailed contribution
Dominika Przewłocka-Rus, Syed Shakib Sarwar, H. Ekin Sumbul, Yuecheng Li and Barbara De Salvo, **'Power-of-two quantization for low bitwidth and hardware compliant neural networks'**, in *TinyML Research Symposium 2022*, 28 march 2022, San Jose, USA (available online: `https://cms.tinyml.org/wp-content/uploads/talks2022/2203.05025.pdf`)	80%	• proposition of 2 different methods for training PoT quantised neural networks • designing the hardware architecture of 3 processing elements for MAC operation with weights quantised using linear, PoT and APoT schemes • demonstration of reduction in computational and memory complexity • analysis of the results and preparation of the publication
Dominika Przewłocka-Rus and Tomasz Kryjak, **'Towards real-time and energy efficient Siamese tracking – a hardware-software approach'** in *Desnos, K., Pertuz, S. (eds) Design and Architecture for Signal and Image Processing. DASIP 2022. Lecture Notes in Computer Science, vol 13425. Springer, Cham.*, `DOI:10.1007/978-3-031-12748-9_13`	90%	• hardware-software implementation of Siamese neural network based tracker on MPSoC ZCU104 platform • designing the Siamese neural network with reduced computational and memory complexity • analysis of the results and preparation of the publication

Continued on next page

Table 3.1: Listing of the author's contribution to each publication in the presented series (Continued)

Publication	Contr. %	Detailed contribution
Dominika Przewlocka-Rus and Tomasz Kryjak, **'Energy efficient hardware acceleration of neural networks with power-of-two quantisation'**, in *Chmielewski, L.J., Orłowski, A. (eds) Computer Vision and Graphics. ICCVG 2022. Lecture Notes in Networks and Systems, vol 598. Springer, Cham.*, DOI10.1007/978-3-031-22025-8_16	90%	• designing the hardware architecture of BAC processing element as essential processing element in convolution layers of PoT networks, with proper weight encoding • hardware implementation of convolution layer with BAC element and comparison with linearly quantised layer • proposition and development of proper for PoT neural networks pruning algorithm • analysis of the results and preparation of the publication
Dominika Przewlocka-Rus and Tomasz Kryjak, **'Power-of-Two Quantized YOLO Network for Pedestrian Detection with Dynamic Vision Sensor'**, in *26th Euromicro Conference on Digital System Design (DSD)*, 6-8 September 2023, Durres, Albania	90%	• development of pedestrian detection system with YOLO network quantised to 4-bit PoT weights, using event data from DVS sensor • training of the quantised neural network • analysis of the results and preparation of the publication

Continued on next page

Table 3.1: Listing of the author's contribution to each publication in the presented series (Continued)

Publication	**Contr. %**	**Detailed contribution**
Dominika Przewlocka-Rus, Tomasz Kryjak, and Marek Gorgon, **'PowerYOLO: Mixed Precision Model for Hardware Efficient Automotive Detection with Event Data'**, in *27th Euromicro Conference on Digital System Design (DSD)*, 28-30 August 2024, Sorbonne University, Paris, France	80%	• design of mixed precision YOLO network, with 4-bit PoT weights and remaining parameters quantised to 8 bits for pedestrian and vehicles detection using event data • proposition of convolution and batch normalisation layers fusion appropriate for PoT weights • training of the quantised neural network • analysis of the results and preparation of the publication

3.1. XNOR CNNs in FPGA: real-time detection and classification of traffic signs in 4K – a demo

Dominika Przewłocka, Marcin Kowalczyk and Tomasz Kryjak, **'XNOR CNNs in FPGA: real-time detection and classification of traffic signs in 4K – a demo'**, in *DASIP 2019: Conference on Design and Architectures for Signal and Image Processing*, 16–18 October 2019, Montréal, Canada

The full text of the publication is removed from the online version of the dissertation due to copyright concerns.

3.2. Optimisation of a Siamese neural network for real-time energy efficient object tracking

Dominika Przewłocka, Mateusz Wąsala, Hubert Szolc, Krzysztof Błachut and Tomasz Kryjak, **'Optimisation of a Siamese neural network for real-time energy efficient object tracking'**, in *Chmielewski, L.J., Kozera, R., Orłowski, A. (eds) Computer Vision and Graphics. ICCVG 2020. Lecture Notes in Computer Science, vol 12334. Springer, Cham.*, DOI10.1007/978-3-030-59006-2_14

The full text of the publication is removed from the online version of the dissertation due to copyright concerns.

3.3. Quantised Siamese tracker for 4K/UltraHD video stream – a demo

Dominika Przewlocka-Rus and Tomasz Kryjak, **'Quantised Siamese tracker for 4K/UltraHD video stream – a demo'**, in *2021 31st International Conference on Field-Programmable Logic and Applications (FPL)*, 30 August - 3 September 2021, Dresden, Germany. E-ISBN 978-1-6654-3759-2, DOI10.1109/FPL53798.2021.00089

The full text of the publication is removed from the online version of the dissertation due to copyright concerns.

3.4. Exploration of hardware acceleration methods for an XNOR traffic signs classifier

Dominika Przewłocka-Rus, Marcin Kowalczyk and Tomasz Kryjak, **'Exploration of hardware acceleration methods for an XNOR traffic signs classifier'**, in *Choraś, M., Choraś, R.S., Kurzyński, M., Trajdos, P., Pejaś, J., Hyla, T. (eds) Progress in Image Processing, Pattern Recognition and Communication Systems. CORES IP&C ACS 2021. Lecture Notes in Networks and Systems, vol 255. Springer*, DOI:10.1007/978-3-030-81523-3_4

The full text of the publication is removed from the online version of the dissertation due to copyright concerns.

3.5. Power-of-two quantization for low bitwidth and hardware compliant neural networks

Dominika Przewłocka-Rus, Syed Shakib Sarwar, H. Ekin Sumbul, Yuecheng Li and Barbara De Salvo, **'Power-of-two quantization for low bitwidth and hardware compliant neural networks'**, in *TinyML Research Symposium 2022*, 28 march 2022, San Jose, USA (available online: `https://cms.tinyml.org/wp-content/uploads/talks2022/2203.05025.pdf`)

The full text of the publication is removed from the online version of the dissertation due to copyright concerns.

3.6. Towards real-time and energy efficient Siamese tracking - a hardware-software approach

Dominika Przewłocka-Rus and Tomasz Kryjak, **'Towards real-time and energy efficient Siamese tracking – a hardware-software approach'** in *Desnos, K., Pertuz, S. (eds) Design and Architecture for Signal and Image Processing. DASIP 2022. Lecture Notes in Computer Science, vol 13425. Springer, Cham.*, DOI:10.1007/978-3-031-12748-9_13

The full text of the publication is removed from the online version of the dissertation due to copyright concerns.

3.7. Energy efficient hardware acceleration of neural networks with power-of-two quantisation

Dominika Przewlocka-Rus and Tomasz Kryjak, **'Energy efficient hardware acceleration of neural networks with power-of-two quantisation'**, in *Chmielewski, L.J., Orłowski, A. (eds) Computer Vision and Graphics. ICCVG 2022. Lecture Notes in Networks and Systems, vol 598. Springer, Cham.*, DOI10.1007/978-3-031-22025-8_16

The full text of the publication is removed from the online version of the dissertation due to copyright concerns.

3.8. Power-of-Two Quantized YOLO Network for Pedestrian Detection with Dynamic Vision Sensor

Dominika Przewlocka-Rus and Tomasz Kryjak, **'Power-of-Two Quantized YOLO Network for Pedestrian Detection with Dynamic Vision Sensor'**, in *26th Euromicro Conference on Digital System Design (DSD)*, 6-8 September 2023, Durres, Albania

The full text of the publication is removed from the online version of the dissertation due to copyright concerns.

3.9. PowerYOLO: Mixed Precision Model for Hardware Efficient Automotive Detection with Event Data

Dominika Przewlocka-Rus, Tomasz Kryjak and Marek Gorgon, **'PowerYOLO: Mixed Precision Model for Hardware Efficient Automotive Detection with Event Data'**, in *27th Euromicro Conference on Digital System Design (DSD)*, 28-30 August 2024, Sorbonne University, Paris, France

The full text of the publication is removed from the online version of the dissertation due to copyright concerns.

Bibliography

[Bertinetto et al., 2016] Bertinetto, L., Valmadre, J., Henriques, J. F., Vedaldi, A., and Torr, P. H. S. (2016). Fully-convolutional siamese networks for object tracking. In Hua, G. and Jégou, H., editors, *Computer Vision – ECCV 2016 Workshops*, pages 850–865, Cham. Springer International Publishing.

[Elhoushi et al., 2021] Elhoushi, M., Chen, Z., Shafiq, F., Tian, Y. H., and Li, J. Y. (2021). Deepshift: Towards multiplication-less neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 2359–2368.

[Gorgoń, 2013] Gorgoń, M. (2013). *Układy FPGA w rekonfigurowalnych systemach wizyjnych czasu rzeczywistego — FPGA-based real-time reconfigurable vision systems*. Warszawa : Akademicka Oficyna Wydawnicza EXIT. ISBN: 978-83-7837-035-2.

[He et al., 2016] He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778.

[Houben et al., 2013] Houben, S., Stallkamp, J., Salmen, J., Schlipsing, M., and Igel, C. (2013). Detection of traffic signs in real-world images: The German Traffic Sign Detection Benchmark. In *International Joint Conference on Neural Networks*, number 1288.

[Intel, 2024] Intel (2024). *OpenVINO toolkit*. Intel. `https://www.intel.com/content/www/us/en/developer/tools/openvino-toolkit/overview.html`.

[Intel Labs, 2021] Intel Labs (2021). *Taking Neuromorphic Computing to the Next Level with Loihi 2 Technology Brief*. Intel. `https://download.intel.com/newsroom/2021/new-technologies/neuromorphic-computing-loihi-2-brief.pdf`.

[Kolesnikov et al., 2021] Kolesnikov, A., Dosovitskiy, A., Weissenborn, D., Heigold, G., Uszkoreit, J., Beyer, L., Minderer, M., Dehghani, M., Houlsby, N., Gelly, S., Unterthiner, T., and Zhai, X. (2021). An image is worth 16x16 words: Transformers for image recognition at scale.

[Krizhevsky et al., 2012] Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In Pereira, F., Burges, C., Bottou, L., and Weinberger, K., editors, *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc.

[LeCun et al., 1989] LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., and Jackel, L. D. (1989). Backpropagation applied to handwritten zip code recognition. *Neural Computation*, 1(4):541–551.

[Li et al., 2020] Li, Y., Dong, X., and Wang, W. (2020). Additive powers-of-two quantization: An efficient non-uniform discretization for neural networks. In *International Conference on Learning Representations*.

[Maslej et al., 2023] Maslej, N., Fattorini, L., Brynjolfsson, E., Etchemendy, J., Ligett, K., Lyons, T., Manyika, J., Ngo, H., Niebles, J. C., Parli, V., Shoham, Y., Wald, R., Clark, J., and Perrault, R. (April 2023). *The AI Index 2023 Annual Report*. AI Index Steering Committee, Institute for Human-Centered AI, Stanford University. https://aiindex.stanford.edu/report/.

[Przewłocka and Kryjak, 2019] Przewłocka, D. and Kryjak, T. (2019). XNOR CNNs in FPGA: real-time detection and classification of traffic signs in 4K – a demo. *DASIP 2019: Conference on Design and Architectures for Signal and Image Processing*. 16–18 October 2019, Montréal, Canada.

[Przewłocka et al., 2020] Przewłocka, D., Wąsala, M., Szolc, H., Błachut, K., and Kryjak, T. (2020). Optimisation of a Siamese neural network for real-time energy efficient object tracking. *Chmielewski, L.J., Kozera, R., Orłowski, A. (eds) Computer Vision and Graphics. ICCVG 2020. Lecture Notes in Computer Science, vol 12334. Springer, Cham.*

[Przewłocka-Rus et al., 2021] Przewłocka-Rus, D., Kowalczyk, M., and Kryjak, T. (2021). Exploration of hardware acceleration methods for an XNOR traffic signs classifier. *Choraś, M., Choraś, R.S., Kurzyński, M., Trajdos, P., Pejaś, J., Hyla, T. (eds) Progress in Image Processing, Pattern Recognition and Communication Systems. CORES IP&C ACS 2021. Lecture Notes in Networks and Systems, vol 255. Springer*.

[Przewłocka-Rus and Kryjak, 2021] Przewłocka-Rus, D. and Kryjak, T. (2021). Quantised Siamese tracker for 4K/UltraHD video stream – a demo. *2021 31st International Conference on Field-Programmable Logic and Applications (FPL)*. 30 August - 3 September 2021, Dresden, Germany.

[Przewłocka-Rus and Kryjak, 2022a] Przewłocka-Rus, D. and Kryjak, T. (2022a). Energy efficient hardware acceleration of neural networks with power-of-two quantisation. *Chmielewski, L.J., Orłowski, A. (eds) Computer Vision and Graphics. ICCVG 2022. Lecture Notes in Networks and Systems, vol 598. Springer, Cham.*

[Przewłocka-Rus and Kryjak, 2022b] Przewłocka-Rus, D. and Kryjak, T. (2022b). Towards real-time and energy efficient Siamese tracking – a hardware-software approach. *Desnos, K., Pertuz, S. (eds) Design and Architecture for Signal and Image Processing. DASIP 2022. Lecture Notes in Computer Science, vol 13425. Springer, Cham.*

[Przewłocka-Rus and Kryjak, 2023] Przewłocka-Rus, D. and Kryjak, T. (2023). Power-of-Two Quantized YOLO Network for Pedestrian Detection with Dynamic Vision Sensor. *26th Euromicro Conference on Digital System Design (DSD)*. 6-8 September 2023, Durres, Albania.

[Przewłocka-Rus et al., 2024] Przewłocka-Rus, D., Kryjak, T., and Gorgon, M. (2024). PowerYOLO: Mixed Precision Model for Hardware Efficient Automotive Detection with Event Data. *27th Euromicro Conference on Digital System Design (DSD)*. 28-30 August 2024, Sorbonne University, Paris France.

[Przewłocka-Rus et al., 2022] Przewłocka-Rus, D., Sarwar, S. S., Sumbul, H. E., Li, Y., and Salvo, B. D. (2022). Power-of-two quantization for low bitwidth and hardware compliant neural networks. *TinyML Research Symposium 2022*. 28 march 2022, San Jose, USA. Available at:https://cms.tinyml.org/wp-content/uploads/talks2022/2203.05025.pdf.

[Reirer, 1997] Reirer, S. (1997). *Skeptics Cite Overload Of Useless Information : Internet Arrives At a Crossroads*. International Herald Tribune. https://www.nytimes.com/1997/03/14/news/skeptics-cite-overload-of-useless-information-internet-arrives-at-a.html.

[Russakovsky et al., 2015] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., and Fei-Fei, L. (2015). ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252.

[Samsi et al., 2023] Samsi, S., Zhao, D., McDonald, J., Li, B., Michaleas, A., Jones, M., Bergeron, W., Kepner, J., Tiwari, D., and Gadepally, V. (2023). From words to watts: Benchmarking the energy costs of large language model inference. pages 1–9.

[Wells, 2023] Wells, S. (2023). *Generative AI's Energy Problem Today Is Foundational Before AI can take over, it will need to find a new approach to energy*. IEEE Spectrum. `https://spectrum.ieee.org/ai-energy-consumption`.

[Wu et al., 2020] Wu, H., Judd, P., Zhang, X., Isaev, M., and Micikevicius, P. (2020). Integer quantization for deep learning inference: Principles and empirical evaluation. *CoRR*, abs/2004.09602.