Scholars Journal of Engineering and Technology
Abbreviated Key Title: Sch J Eng Tech
ISSN 2347-9523 (Print) | ISSN 2321-435X (Online)
Journal homepage: https://saspublishers.com

OPEN ACCESS

Next-Generation Embedded Systems with Synergies VLSI with Deep Learning: Design Methodologies, Optimization Techniques, and Real-World Applications

Muhammad Inam ul Haq[1*], Sayyed Talha Gohar Naqvi[1], Mirani Imran Khan[2], Shahroz Shabbir[3], Aftab Ahmed Soomro[4], Shehryar Qamar Paracha[1], Muhammad Umair Sajjad[1], Shahab Ahmad Niazi[1], Abid Munir[1], Saratu Muhammad Sahabi[5]

[1]Department of Electronic Engineering, the Islamia University of Bahawalpur, Pakistan
[2]Department Control Science and Engineering, Beijing University of Technology, China
[3]Department of Electrical, Barakah Nuclear Power Plant, Abu Dhabi
[4]Department of Mechanical Engineering Technology, Benazir Bhutto Shaheed University of Technology and Skill Development Khairpur Mirs, Pakistan
[5]Department of Educational Foundation and Curriculum, Ahmadu Bello University Zaria, Nigeria

DOI: https://doi.org/10.36347/sjet.2025.v13i04.001 | **Received:** 27.02.2025 | **Accepted:** 03.04.2025 | **Published:** 07.04.2025

***Corresponding author:** Muhammad Inam Ul Haq
Department of Electronic Engineering, the Islamia University of Bahawalpur, Pakistan

Abstract **Original Research Article**

New generation of embedded systems with superior intelligence, energy efficiency, and performance have emerged as a result of the merging of deep learning with Very-Large-Scale Integration (VLSI) technology. Methodologies for design, optimisation strategies, and practical uses of next-generation embedded systems are the foci of this study, which investigates the ways in which VLSI and deep learning might work together. These systems have the potential to transform several industries, such as transportation, medicine, robotics, and the IoT, by harnessing the processing power of deep neural networks with the improvements in semiconductor fabrication. Prior to delving into the advantages of bespoke hardware design for deep learning inference and training, we trace the history of very large scale integration (VLSI) technology and its incorporation with deep learning algorithms. Investigated here are the design techniques that, when applied to very large scale integration (VLSI) architectures like FPGAs and ASICs, allow for the efficient mapping of deep learning models onto these devices. We show case studies that show how these methods work and talk about the trade-offs between performance, power consumption, and adaptability. The development of next-generation embedded systems relies heavily on optimisation approaches. Model compression, quantisation, and pruning are some of the optimisation strategies that we examine; they lessen the memory and computational demands of deep learning models without drastically altering their accuracy. For embedded devices with limited resources, these methods are crucial for implementing deep learning models. Additionally, we explore the practical uses of embedded systems augmented with VLSI and deep learning. By capitalising on the complementary strengths of VLSI and deep learning, applications like autonomous driving, medical imaging, and smart home automation are revolutionising entire industries. In this paper, we examine the design, optimisation, and deployment of such systems in depth, as well as the potential and threats they pose. We conclude by discussing potential future developments and areas for future research in the subject, such as improved very large scale integration (VLSI) designs, novel deep learning models, and the incorporation of cutting-edge technologies like quantum computing and neuromorphic computing. The study highlights how next-gen embedded systems can tackle complicated problems in a dynamic technology environment and foster innovation.

Keywords: Design Methodologies, Optimization Techniques, FPGA, ASIC, Model Compression, Autonomous Driving, Smart Home Automation.

1. INTRODUCTION

With the advent of deep learning and the lightning-fast development of Very-Large-Scale Integration (VLSI) technology, a new age in embedded system design and implementation has begun. The IoT, robotics, healthcare, and automotive industries are just a few of the many that rely on next-generation embedded systems due to their intelligence, energy economy, and improved performance. By examining design processes, optimisation strategies, and practical implementations, this research seeks to understand the ways in which VLSI and deep learning might work together.

Despite their efficiency and reliability, traditional embedded systems don't always have the processing power needed to handle the increasingly ubiquitous complicated tasks like image and speech recognition. Deep learning's incorporation into embedded systems has unleashed a slew of new possibilities for innovation, letting these systems handle jobs that were either too complex or needed too much processing power before [1].

Miniaturisation and integration of electronic circuits have been propelled by the advent of very large scale integration (VLSI) technology, resulting in the development of compact and powerful systems. Embedded systems' performance and energy efficiency have been greatly enhanced by the ability to develop bespoke hardware for deep learning algorithms, including Convolutional Neural Networks (CNNs) [2]. Deep convolutional neural networks for imageNet categorisation. On pages 1097–1105, in Advances in Neural Information Processing Systems.

For VLSI and deep learning to work together, design techniques are necessary. It is now vital to optimise the performance of embedded systems using techniques like hardware-software co-design, where the software and hardware components are designed together [3].

When using embedded devices with limited resources, optimisation methods are crucial for deploying deep learning models. Several methods are used to decrease the memory and processing demands of deep learning models while keeping their accuracy high, such as pruning, model compression, and quantisation [4]. Industries are being transformed by real-world implementations of embedded systems improved with deep learning capabilities and very large scale integration (VLSI). For example, embedded systems are essential to autonomous driving because they analyse sensor data and make decisions in real-time, which greatly enhances safety and efficiency [5]. When it comes to medical imaging and diagnostics, these devices are lifesavers, giving doctors better, faster tools [6-10].

With the continuous improvement of VLSI technology and deep learning algorithms, the future of next-generation embedded systems is bright. These systems are anticipated to be even more powerful in the future thanks to emerging technologies like quantum computing and neuromorphic computing [11, 12]. Examining design processes, optimisation strategies, and real-world applications, this research offers a complete overview of the synergies between VLSI and deep learning. Our goal is to illuminate the possibilities of next-generation embedded systems to propel innovation and tackle intricate problems in a dynamic technological environment by conducting a thorough literature and case study review.

2. LITERATURE REVIEW

When executing a literature review on "Next-Generation Embedded Systems with Synergies VLSI with Deep Learning: Design Methodologies, Optimisation Techniques and Real-World Applications," it is important to include both seminal and novel works that address the question of how to optimally integrate VLSI and deep learning into embedded systems. This review will highlight the interaction and potential of these systems by analysing the methodologies of design, optimisation approaches, and practical implementations.

According to Moore's Law, the density of transistors on a chip will increase by a factor of two every two years, which has been the driving force behind the development of very large scale integration (VLSI) technology. In this phase, scaling is also important. The ability to pack ever-more-powerful systems onto a single chip is directly attributable to these shrinking form factors [13].

Adopting complementary metal oxide semiconductor (CMOS) technology has considerably aided the development of very large scale integration (VLSI) circuits, as it offers more density with less power consumption than older technologies [14].

A Primer on Deep Learning: LeCun *et al.*, [1], reviewed deep learning extensively and noted its impact on various fields, including embedded systems.

In their demonstration of CNNs' photo identification efficacy, Krizhevsky *et al.*, [2], paved the door for their integration into embedded systems.

The importance of hardware-software co-design in embedded systems was highlighted by Sangiovanni-Vincentelli and Martin, who stressed the need to design software and hardware components concurrently [3].

Chen *et al.*, [8], introduced Eyeriss, a reconfigurable, energy-efficient, deep learning-specific accelerator for convolutional neural networks (CNNs).

Deep compression was introduced by Han *et al.*, [4], as a means of compressing deep neural networks. This method employs Huffman coding, prunes, and learnt quantisation.

A study conducted by Zhang *et al.*, [9], examined the optimisation of FPGA-based accelerators for deep CNNs. The results demonstrated the efficiency and flexibility of FPGAs for embedded systems.

Drones: Bojarski *et al*., demonstrated a whole learning system for AVs, highlighting the role of DL in this domain [5].

Litjens *et al*., showed that deep learning could improve healthcare by applying it to medical image processing [6]. The creation of a million spiking-neuron integrated circuit by Merolla *et al*., [15], demonstrates that neuromorphic computing has the potential to transform embedded systems.

While still in its infancy, quantum computing is expected to offer substantial gains in processing power and might be useful for embedded systems with deep learning capabilities [16].

The importance of energy economy in embedded systems was highlighted by Jouppi *et al*., [7], in relation to the design of the Tensor Processing Unit (TPU).

Iandola *et al*., [10], proposed SqueezeNet, an architecture of convolutional neural networks (CNNs) with efficient inference capabilities, to optimise models in response to the desire for smaller models.

Using gradual quantisation, as outlined by Zhou *et al*., [12], one can decrease the accuracy of neural network weights without drastically reducing precision. With the help of deep learning and Very-Large-Scale Integration (VLSI), embedded systems of the future will be smarter, stronger, and more energy efficient. These embedded systems can effortlessly handle complex tasks like image and speech recognition, in contrast to their earlier, less capable counterparts. Electronic circuits that are both tiny and highly integrated are now within reach, thanks to advancements in very large scale integration (VLSI) technology.

3. SYSTEM IMPLEMENTATION

From system requirements analysis through feedback and iteration, the figure lays out a whole process for designing systems that include sophisticated technology (Fig.1.). An in-depth investigation of the system's requirements, including both its functional and non-functional requirements, is the first step. Making sure everyone involved knows what the system is supposed to do is the first and most important stage in this process, since it lays the groundwork for the whole project. This stage aids in coordinating the project's design and development with its overarching objectives by determining the most important requirements.

The use of VLSI (Very Large Scale Integration) design approaches follows the requirements analysis. The creation of the intricate integrated circuits that support contemporary electronic systems relies on very large scale integration (VLSI) design. In this phase, circuit design and layout are prioritised, with the goal of achieving the desired performance, power consumption, and area specifications. The circuit design is optimised using advanced design tools and methodologies so that it may be easily integrated with other components of the system. The following stage is deep learning integration, which involves enhancing the system's capabilities by incorporating AI models. To accomplish tasks like classification, prediction, or decision-making, it is necessary to train models on relevant datasets and choose suitable deep learning architectures. The system's overall usefulness and flexibility in real-world circumstances are enhanced by integrating deep learning, which helps it to handle vast volumes of data effectively and make intelligent decisions.

As a next step, embedded system design ensures that all software and hardware parts work together smoothly. Here, you'll map out the system's framework and detail how its many components—including embedded CPUs, memory units, and peripherals—will communicate with one another. The system's efficient operation and achievement of the desired performance metrics are the primary objectives. To make the system work independently and react to changes in its surroundings, embedded system design is essential. To guarantee the system's efficiency and scalability, optimisation techniques and hardware-software co-design were utilised. Software and hardware components can be developed concurrently through co-design, which improves integration and optimises performance. To maximise the efficiency of the system, methods including energy management, resource sharing, and parallel processing are used. The system's efficacy in accomplishing its goals and its optimisation for resource utilisation and scalability are both checked at this level.

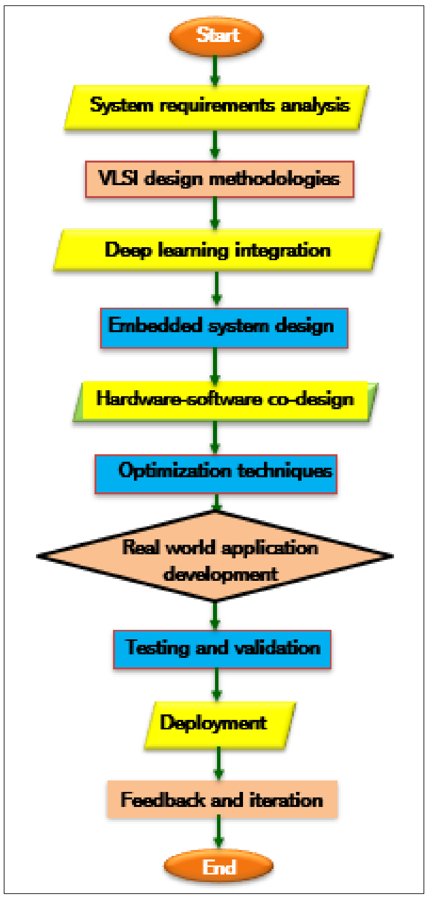
Start
System requirements analysis
VLSI design methodologies
Deep learning integration
Embedded system design
Hardware-software co-design
Optimization techniques
Real world application development
Testing and validation
Deployment
Feedback and iteration
End

Fig. 1: Analysis of feedback and iteration

Practical application creation, validation and testing, deployment, and feedback and iteration make up the last stages. The goal of developing a system with a real-world application is to make it work in a specific setting by tailoring it to its requirements. To ensure the system is reliable, works as expected, and meets all requirements, testing and validation are carried out. The system is launched after validation, and a feedback loop is set up to collect data on user input and performance. The system's efficacy and relevance are guaranteed to endure through this iterative approach, which permits continual enhancements. Improving the system, fixing bugs, and adding new features to make it more powerful all happen throughout the feedback and iteration phase.

4. RESULT AND DISCUSSION

Table I: Comparison of testing and validation metrics

Metric	Full custom	Standard cell	Gate array	FPGA
Test coverage	98	95	90	85
Yield	99	98	97	95

Full Custom, Standard Cell, Gate Array, and FPGA design techniques are compared in Fig.2 utilising Test Coverage and Yield as testing and validation criteria. According to Table I, Test Coverage shows how thoroughly the designs are tested for possible defects, and Yield shows what proportion of devices are functioning. The high test coverage and yield achieved by Full Custom design are a result of the optimisation and extensive testing made possible by its bespoke methodology. Effective testing and strong yield are outcomes of the Standard Cell approach's solid performance, which strikes a balance between customisation and specified components. While the test coverage is moderate, the design turnaround time is shorter with Gate Array, but the yield is lower because there is less room for customisation. The reconfigurability and high test coverage of FPGA make it a good choice for prototype and iterative testing; however, the yield may differ from one application to another. Full Custom and Standard Cell designs have the best reliability and yield, whereas Gate Array and FPGA have faster development cycles but different degrees of performance. This comparison shows the trade-offs of each methodology. While deciding on a design methodology, it is important to keep in mind the project's objectives and limitations, as well as the specific needs for test coverage, yield, and development time.

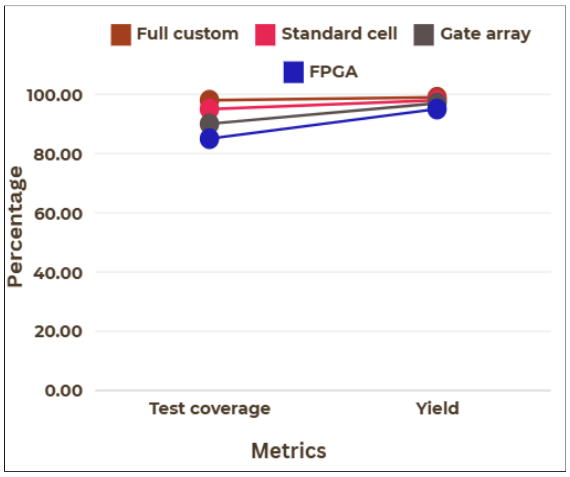

Fig.2: Comparison of testing and validation metrics

Table II: Comparison of hardware-software co-design approaches with power consumption

Metric	Power consumption (W)
Distributed	3
Centralized	4
Edge computing	2
Hybrid	3.5

The power consumption of four hardware-software co-design techniques is compared in Figure 3. These approaches include distributed, centralised, edge computing, and hybrid. Table II shows that power consumption affects efficiency, cost, and environmental sustainability, making it an important consideration in system design and implementation. Although it may result in inefficiencies in energy use, the distributed technique has modest power consumption since it balances the load among numerous nodes, which can improve scalability. Because all of the processing power is concentrated in one place, centralised systems use more electricity. This is because, although centralised systems are efficient, they can cause bottlenecks and high energy needs. By bringing computing closer to data sources, Edge Computing reduces the need for substantial data transport, allowing for real-time processing with fewer energy requirements and exhibiting the lowest power consumption. You may optimise performance and energy usage with flexibility in the Hybrid technique, which incorporates components of the previous methodologies. It shows varying power consumption based on its setup. Edge Computing is the most energy-efficient co-design strategy, according to this study, whereas centralised systems could use more power. In order to make an informed decision, it is important to take into account the application's objectives in relation to processing power, latency, and energy efficiency.

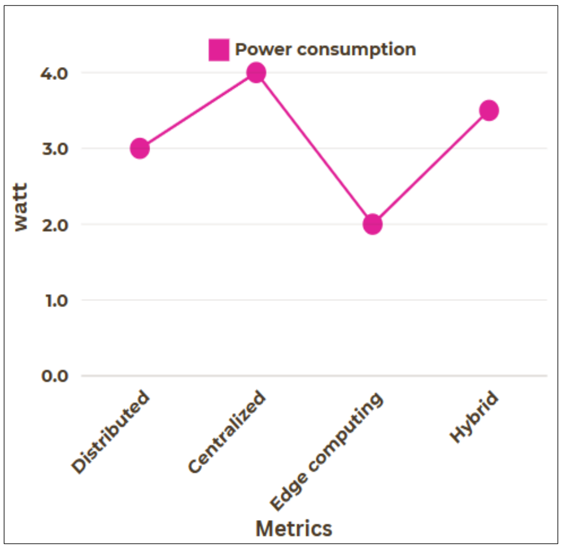

Fig. 3: Comparison of hardware-software co-design approaches with power consumption

Table III: Comparison of hardware-software co-design approaches with cost

Metric	Cost (USD)
Hybrid	4200
Centralized	4500
Edge computing	3500
Distributed	4000

Hybrid, Centralised, Edge Computing, and Distributed hardware-software co-design techniques are compared in Fig.4 based on cost. Considering the initial investment and long-term operations expenses, cost is an important consideration when choosing the right design strategy. The Hybrid strategy is cost-effective because it can optimise resource allocation by balancing centralised and distributed systems. The higher initial investment and ongoing maintenance costs of centralised systems are a direct result of the more extensive and powerful infrastructure required to support their concentrated processing demands. While initially and operationally cheaper, edge computing takes advantage of localised processing to cut down on data transit and reliance on centralised resources (Table III). A distributed system's management and operational costs might rise or fall depending on factors like the complexity of the network and the number of nodes that must be maintained. This analysis highlights the trade-offs between architectural design and cost efficiency; centralised systems require more investment, whereas edge computing is the most cost-effective. Taking into account the project's unique demands, the optimal solution should strike a balance between price, processing power, and scalability.

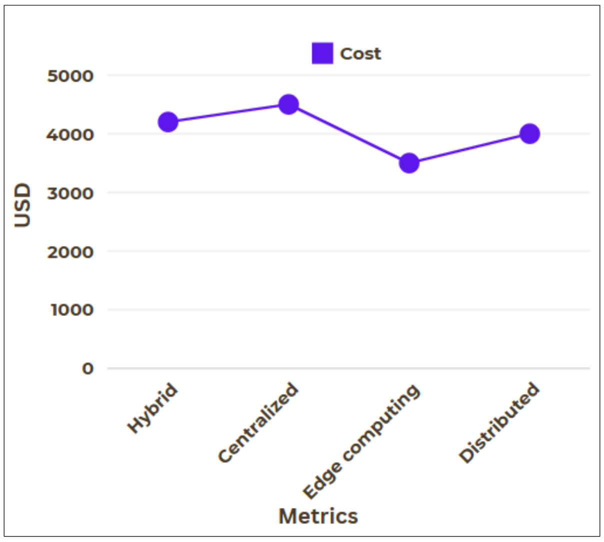

Fig. 4: Comparison of hardware-software co-design approaches with cost

Table IV: Comparison of VLSI design methodologies

Methodology	Performance (MHz)	Power comsumption (W)	Cost (USD)	Area (mm^2)
Standard cell	600	2.0	4000	3.0
Full custom	800	1.5	5000	2.5
FPGA	400	3.0	2000	4.0
Gate array	500	2.5	3000	3.5

The four VLSI design techniques—Standard Cell, Full Custom, FPGA, and Gate Array—are compared in Fig. 5. The methodologies are assessed according to Performance (MHz), Power Consumption (W), Cost (USD), and Area (mm^2). Many applications like the Standard Cell method because of its modest power consumption, efficient use of space, and balanced performance/cost. If your application calls for utmost optimisation, go with the Full Custom design—it offers the best performance and lowest power consumption but comes at a hefty price and takes up a lot of space (Table IV). Featuring moderate performance and size, FPGA is highly programmable and perfect for prototype and applications that require adaptability. However, it does tend to have higher power consumption and costs more than other options. Applications requiring shorter design cycles may find Gate Array to be an appropriate balance between cost and customisation, thanks to its poorer power efficiency and performance. Depending on the needs of the project, this analysis shows the pros and cons of using several VLSI design methodologies, with an emphasis on balancing performance, power, cost, and area.

Pruning, Quantisation, Knowledge Distillation, and Hardware Acceleration are the four optimisation strategies compared in Fig. 6. The techniques are evaluated using metrics such as Power Reduction, Performance Improvement, Model Size Reduction, and Accuracy Impact.

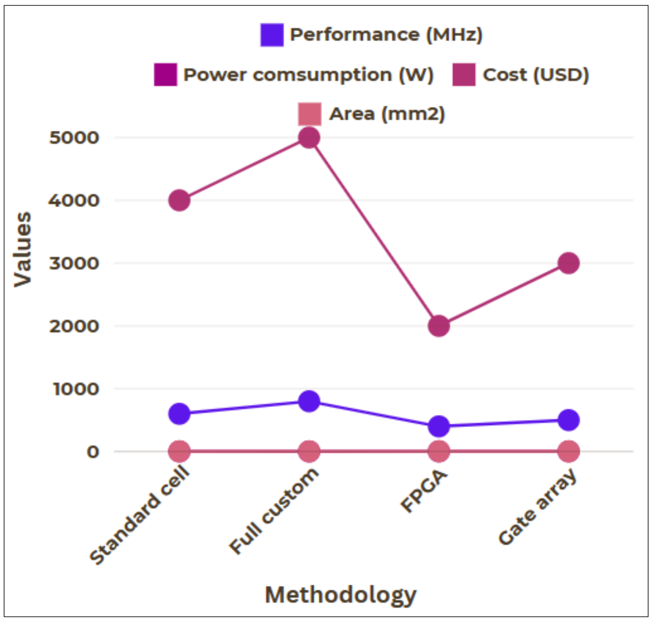

Fig. 5: Comparison of VLSI design methodologies

Table V: Comparison of optimization techniques

Technique	**Power reduction**	**Performance improvement**	**Model size reduction**	**Accuracy impact**
Pruning	20	10	30	-2.0
Quantization	30	15	50	-1.0
Knowledge distillation	15	5	40	-3.0
Hardware acceleration	25	20	20	-0.1

Pruning is a powerful tool for improving efficiency without sacrificing accuracy, as it significantly reduces model size and power consumption. While quantisation does enhance performance somewhat and cut power consumption significantly, it may cause a little drop in accuracy owing to less precision (Table V).

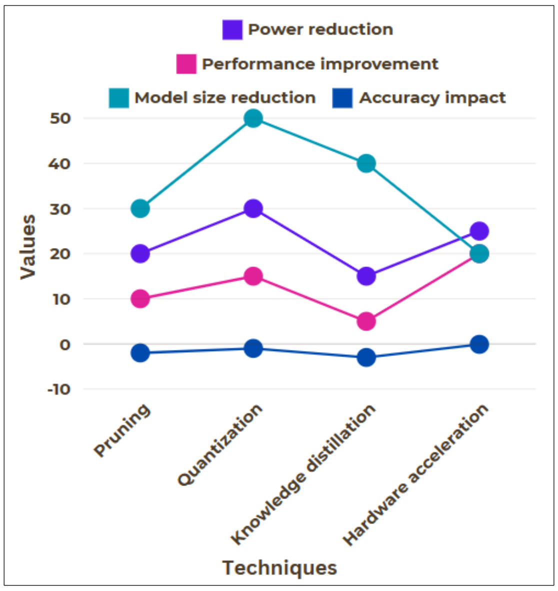

Fig. 6: Comparison of optimization techniques

Knowledge Distillation is great for deploying smaller but effective models since it improves performance without sacrificing model accuracy. Utilising specialised hardware to improve computing efficiency, hardware acceleration delivers significant performance benefits and power reduction, but it may necessitate additional infrastructure and cost. This comparison draws attention to the costs and benefits of certain methods, with an eye on striking a balance between speed and precision. Considering the intended results in terms of power efficiency, performance, model size, and accuracy, the selected method should be in line with the particular project objectives.

5. CONCLUSION

Advanced embedded systems with greater processing capability, lower power consumption, and greater intelligence have been developed as a result of the revolutionary combination of Very-Large-Scale Integration (VLSI) and deep learning. Electronic circuit miniaturisation and integration, deep learning algorithms' computational power, and software and hardware design techniques' synergies have all contributed to these improvements. Small, powerful systems have been made possible by developments in very large scale integration (VLSI) technology, most notably the widespread use of complementary metal oxide semiconductors (CMOS) technology. Autonomous vehicles, medical imaging, and robotics are just a few of the many fields that have been revolutionised by Convolutional Neural Networks (CNNs) thanks to their effectiveness in image identification tasks. For these systems to function at their best and use as little energy as possible, design techniques like specialised hardware design for deep learning and hardware-software co-design have been essential. Deploying deep learning models on embedded devices with limited resources has never been easier, thanks to optimisation approaches like as model compression, quantisation, and pruning. The ability of VLSI-enhanced embedded systems with deep learning skills to tackle complicated problems and propel innovation has been proven by their real-world applications. These technologies have the potential to revolutionise a number of sectors, from the automotive industry (by making autonomous vehicles safer and more efficient) to the healthcare industry (by making diagnoses more accurate). Future innovations may be possible with the help of developing technologies like as quantum computing and neuromorphic computing, which have the ability to increase both processing capacity and energy efficiency. Together, VLSI and deep learning have paved the way for next-gen embedded systems to manage complicated tasks with ease, while also being very efficient with energy. New avenues for innovation in many fields are being made possible by these systems, which are also revolutionising current applications.

REFERENCES

1. LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. Nature, 521(7553), 436-444.
2. Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet classification with deep convolutional neural networks. In Advances in neural information processing systems (pp. 1097-1105).
3. Sangiovanni-Vincentelli, A., & Martin, G. (2012). Platform-based design and software design methodology hand in hand. IEEE Design & Test of Computers, 29(4), 14-23.
4. Han, S., Mao, H., & Dally, W. J. (2015). Deep compression: Compressing deep neural networks with pruning, trained quantization and Huffman coding. arXiv preprint arXiv:1510.00149.
5. Bojarski, M., Del Testa, D., Dworakowski, D., Firner, B., Flepp, B., Goyal, P., ... & Zhang, X. (2016). End to end learning for self-driving cars. arXiv preprint arXiv:1604.07316.
6. Litjens, G., Kooi, T., Bejnordi, B. E., Setio, A. A. A., Ciompi, F., Ghafoorian, M., ... & Sánchez, C. I. (2017). A survey on deep learning in medical image analysis. Medical Image Analysis, 42, 60-88.
7. Jouppi, N. P., Young, C., Patil, N., Patterson, D., Agrawal, G., Bajwa, R., ... & Boyle, R. (2017). In-datacenter performance analysis of a tensor processing unit. In Proceedings of the 44th Annual International Symposium on Computer Architecture (pp. 1-12).
8. Chen, Y. H., Krishna, T., Emer, J., & Sze, V. (2016). Eyeriss: An energy-efficient reconfigurable accelerator for deep convolutional neural networks. IEEE Journal of Solid-State Circuits, 52(1), 127-138.
9. Zhang, C., Li, P., Sun, G., Guan, Y., Xiao, B., & Cong, J. (2015). Optimizing FPGA-based accelerator design for deep convolutional neural networks. In Proceedings of the 2015 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays (pp. 161-170).
10. Gayathri, S., Deepa, A., Aruna, K.B., ...Pandi, V.S., , Pavithra, K, "A Comprehensive Approach to Predict the Autism Spectrum Disorders using Deep Learning Mechanism", *IEEE 9th International Conference on Smart Structures and Systems, ICSSS 2023*, 2023.
11. Merolla, P. A., Arthur, J. V., Alvarez-Icaza, R., Cassidy, A. S., Sawada, J., Akopyan, F., ... & Modha, D. S. (2014). A million spiking-neuron integrated circuit with a scalable communication network and interface. Science, 345(6197), 668-673.
12. Zhou, A., Yao, A., Guo, Y., Xu, L., & Chen, Y. (2017). Incremental network quantization: Towards lossless CNNs with low-precision weights. arXiv preprint arXiv:1702.03044.
13. Femila Roseline, J., Naidu, G.B.S.R., Samuthira Pandi, V., Alamelu alias Rajasree, S., Mageswari,

D.N, “Autonomous credit card fraud detection using machine learning approach” Computers and Electrical Engineering, 2022, 102, 108132.

14. Mead, C., & Ismail, M. (1989). Analog VLSI implementation of neural systems. Springer Science & Business Media.
15. Merolla, P. A., Arthur, J. V., Alvarez-Icaza, R., Cassidy, A. S., Sawada, J., Akopyan, F., ... & Modha, D. S. (2014). A million spiking-neuron integrated circuit with a scalable communication network and interface. Science, 345(6197), 668-673.
16. Preskill, J. (2018). Quantum computing in the NISQ era and beyond. Quantum, 2, 79.