ISSN: 2959-6386 (Online), Vol. 3, Issue 3

Journal of Knowledge Learning and Science Technology

journal homepage: https://jklst.org/index.php/home

Enhancing Machine Learning Performance: The Role of GPU-Based AI Compute Architectures

Bhuvi chopra

Product manager, Google

Abstract

This paper advances the field of GPU-based embedded intelligence (EI) by providing a comprehensive review of current and emerging architectures and applications. It covers key paradigms in GPU-based EI, focusing on architecture, technologies, and practical applications. The paper is structured as follows: (1) An overview and classification of GPU-based EI research, providing a broad perspective and concise summary of the paper's scope; (2) An in-depth discussion of various architectural technologies for GPU-based deep learning techniques and applications; and (3) A detailed examination of architectural technologies for GPU-based machine learning techniques and applications. This paper aims to offer valuable insights into the research area, encouraging further development of GPU-based EI for practical deployment and applications.

Keywords: embedded intelligence; GPU; multi-GPU; parallel architecture; machine learning

Article Information:
Article history: *Received:* 20-Jan-24 *Accepted:* 01-Mar-24 *Online:* 08- Mar -24 *Published*: 09- Mar -24

DOI: https://doi.org/10.60087/jklst.vol3.n3.p42

!Correspondences'author: Bhuvi chopra

Introduction

Embedded Intelligence (EI) in products or systems endows them with the capability to reflect on their operational performance. EI involves the integration of sensors, communication modules, and computational processing units into products or systems to achieve specific operational objectives. Recently, machine learning, deep learning, and artificial intelligence (AI) have been widely adopted across various platforms, imposing new requirements on existing computing systems and architectures. While these technologies can exist solely as software, they often

necessitate hardware components to build standalone intelligent machines. This interplay between "intelligence" and embedded systems is crucial.

There are several platforms for deploying machine learning, deep learning, and AI, including: (1) Graphics Processing Units (GPUs); (2) Field Programmable Gate Arrays (FPGAs); (3) Central Processing Units (CPUs); (4) Application-Specific Integrated Circuits (ASICs); and (5) Field Programmable System-on-Chip (FPSoC). Recent advancements in GPU architecture have significantly increased computational power. Originally designed for fast graphics rendering, GPUs feature hundreds of smaller cores optimized for parallel processing. Initially created to enhance video game responsiveness, GPUs have since revolutionized the IT industry, extending their utility to a broad range of applications, including high-performance computing systems.

GPUs serve as hardware accelerators, significantly speeding up the training and inference processes in machine learning, deep learning, and AI. Their core density and power efficiency make them suitable for meeting real-time requirements and the intensive computational demands of these technologies. Machine learning algorithms have been widely adopted across various hardware platforms, including GPUs, for their energy efficiency, compact form factor, and affordability. Modern smartphones incorporate hardware to accelerate machine learning algorithms, and software frameworks have been optimized for embedded platforms. Additionally, hardware accelerators like the edge Tensor Processing Unit (TPU) are becoming commoditized.

Similarly, GPUs are extensively used to accelerate deep learning, proving to be highly effective for many applications. Emerging deep learning cloud services provided by AI service providers further boost the use of deep learning in numerous business-critical processes. Large companies' deep learning platforms offer customized hardware, including servers, storage, and networking communications, to support high computational workloads. However, centralized cloud environments for deep learning entail longer latency, higher energy consumption, and financial overheads. Consequently, research and development platforms often focus on cost-effective GPUs for developing limited-scale computational clusters to handle diverse deep learning workloads. Recent trends, such as competitions like the "Low-Power Image Recognition

Challenge" (LPIRC), emphasize a balance between performance accuracy, computational throughput, and power consumption, reflecting the growing focus on efficiency and effectiveness in AI development.

Current development trends in GPU-based embedded intelligence (EI) are moving towards: (1) Utilizing lower precision arithmetic, such as shifting from 32-bit to 16-bit representations; (2) Exploiting operations on sparse matrices, where mechanisms like ZeroSkip can take advantage of the many zero weights in convolutional neural networks (CNNs), as discussed further in Section 3.6; and (3) Implementing binary neural networks (BNNs), which are deep neural networks (DNNs) using binary representations for weights and activation values, elaborated in Section 3.1. These advancements make deep learning approaches for small, power-efficient devices highly attractive across various domains.

Examples of GPU-based EI in real-world applications with significant social impacts include predictive systems for disaster early warning and management (e.g., large-scale water supply systems management, flood and fire simulation and forecasting), and in the electronics industry (e.g., circuit solvers for electronic systems with numerous components). Further examples will be discussed in Sections 3 and 4.

Currently, there is a lack of comprehensive surveys or reviews on GPU-based embedded intelligence research and development. Most reviews on machine learning, deep learning, or AI do not focus on hardware or embedded intelligence. This paper aims to fill this gap by providing a comprehensive review and several representative studies on the emerging and current paradigms in GPU-based EI research and development, with a focus on enabling technologies, applications, and challenges.

The overview and classifications of GPU-based EI research and development are summarized in Table 1. The research works are categorized as follows: (1) An overview and classification of GPU-based EI research, providing a full spectrum and concise summary of the paper’s scope; (2) A detailed discussion of various architectural technologies for deep learning techniques and applications; and (3) A detailed discussion of various architectural technologies for machine learning techniques and applications. This paper aims to offer valuable insights into the research

area and encourage further development of GPU-based EI for practical deployment and applications.

The remainder of the paper is structured as follows: Section 2 presents an overview and classification of EI research on GPUs. Sections 3 and 4 discuss GPU-based deep learning and machine learning techniques and applications, respectively. Section 5 concludes the paper.

Table1. Classification Descriptors for Embedded Intelligence ResearchonGPU.

Classification Descriptor	References
GPU-based Deep Learning Technologies for EI	
Architecture frame work and strategy	[11–25]
Scheduling and communication	[26–29]
Image processing and computer vision	[30–40]
Medical or health	[41–44]
Modeling or prediction	[45–51]
Convolution or performance analysis	[6,52–54]
VLSI placement	[55]
GPU-based Machine Learning Technologies for EI	
Architecture platform	[56–65]
Applications	[66–77]

GPU:Graphic Process Unit; EI: Embedded Intelligence.

Overview and Classifications of EI Research on GPU Architecture

Table 1 provides an overview and classification of research on GPU-based architecture technologies and applications for embedded intelligence (EI). This table offers a comprehensive view of the research landscape and serves as a concise summary of the paper's scope. The research is classified into two main descriptors: (1) GPU-based deep learning technologies for EI and (2) GPU-based machine learning technologies for EI.

The first classification descriptor, GPU-based deep learning technologies for EI, is further divided into seven sub-descriptors:

- Architecture framework and strategy
- Scheduling and communication
- Image processing and computer vision
- Medical or health applications
- Modeling or prediction
- Convolution or performance analysis
- VLSI placement

The second classification descriptor, GPU-based machine learning technologies for EI, is divided into two sub-descriptors:
- Architecture platform
- Applications

The right column in the table lists the relevant works and references corresponding to each sub-descriptor, facilitating quick access for readers to the reviewed works.

Deep Learning on GPU Architecture

Deep learning approaches and techniques have been proposed and deployed to address many real-world problems such as bioinformatics, manufacturing, robotics, computer vision, and natural language processing. Some well-known deep learning models include convolutional neural networks (CNNs) like AlexNet and GoogleNet. Commercial cloud services offered by large technology companies have expanded the adoption of deep learning in various business-critical processes. Additionally, several deep learning frameworks such as TensorFlow (from Google), CNTK (from Microsoft), and Caffe2 (from Facebook) have been developed to facilitate training on GPU-enabled computational clusters.

Architecture Framework and Strategy

Ultra-deep neural networks (UDNNs) have been proposed to produce high-quality models. However, training UDNNs is resource-intensive and time-consuming, limiting training

efficiency on modern GPUs due to limited DRAM capacity. To address this, the authors in [11] proposed a new architecture called AccUDNN, an accelerator designed to optimize limited GPU memory resources and speed up UDNN training. The architecture of AccUDNN consists of several interconnected modules:

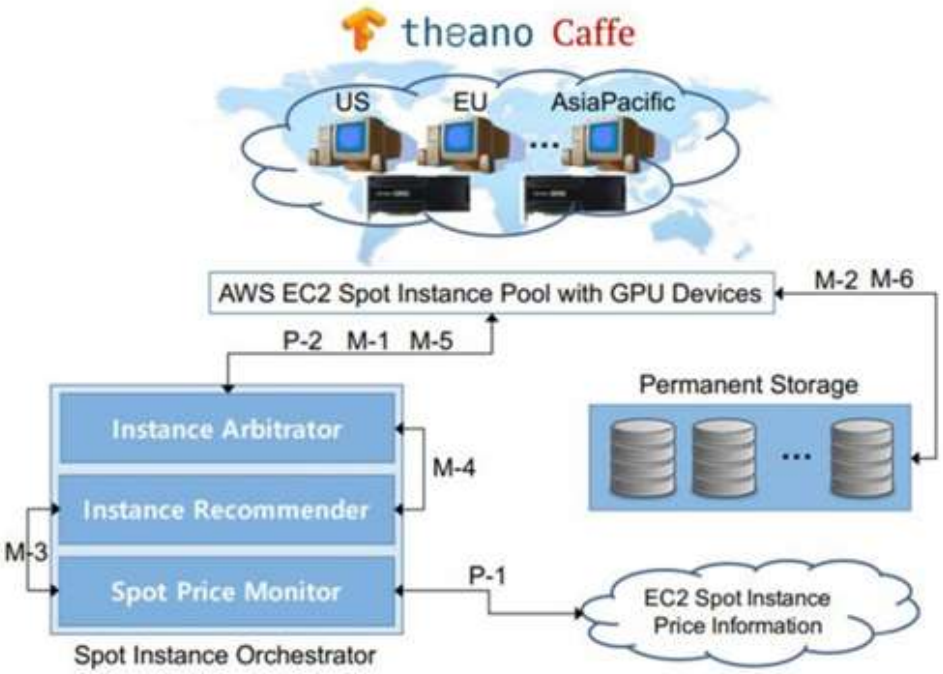
1. The Information Collector gathers features and attributes to build the performance model.
2. The Performance Model Builder analyzes runtime characteristics and behavior in terms of computational performance, memory utilization, and communication requirements.
3. The Constraint Unit develops conditions to prevent performance degradation.
4. The Hyperparameter Tuner computes the optimal minibatch size to meet efficiency constraints.

Their experimental results showed that the proposed architecture reduced the memory requirements for training ResNet-152 from 24 GB to 8 GB.

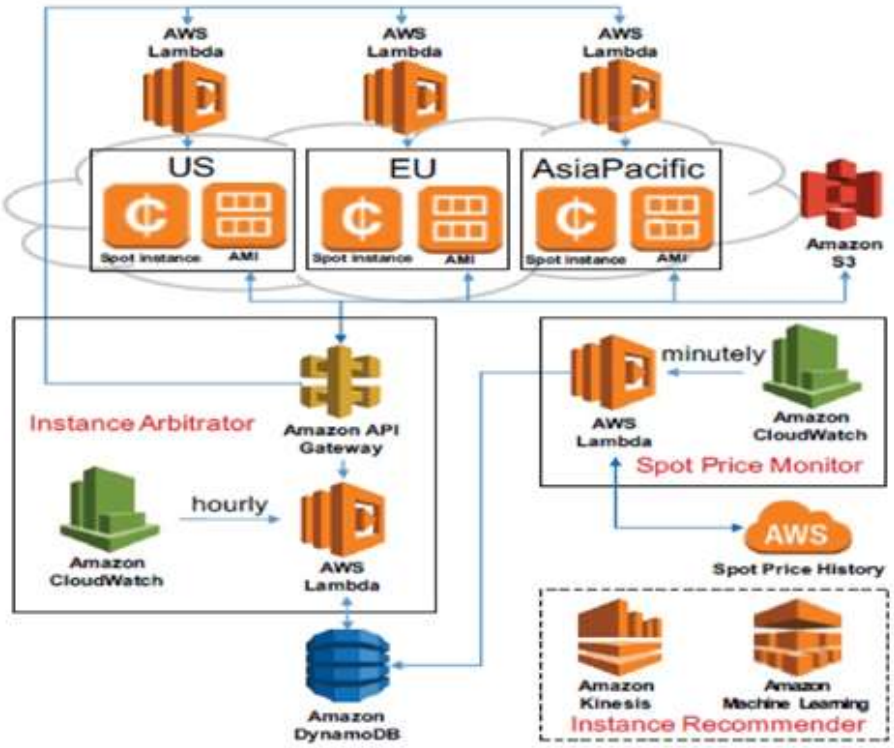
The authors in [12] proposed the DeepSpotCloud architecture to execute deep learning tasks with cost efficiency and fault tolerance. Figure 1 illustrates the system architecture of DeepSpotCloud, which includes several important modules:

1. The Spot Instance Orchestrator executes tasks for spot price monitoring, recommendation, and arbitration.
2. The Spot Price Monitor accesses the current GPU spot price.
3. The Instance Arbitrator monitors running instances to identify interrupted tasks.

Their experimental results demonstrated significant cost savings with a marginal increase in task running time.


theano Caffe
US
EU
AsiaPacific
AWS EC2 Spot Instance Pool with GPU Devices
M-2 M-6
P-2 M-1 M-5
Permanent Storage
Instance Arbitrator
Instance Recommender
Spot Price Monitor
M-4
M-3
P-1
EC2 Spot Instance Price Information
Spot Instance Orchestrator

(a)


AWS Lambda
AWS Lambda
AWS Lambda
US
EU
AsiaPacific
Spot instance
AMI
Amazon S3
Instance Arbitrator
Amazon API Gateway
hourly
Amazon CloudWatch
AWS Lambda
Amazon DynamoDB
minutely
AWS Lambda
Amazon CloudWatch
Spot Price Monitor
AWS
Spot Price History
Amazon Kinesis
Amazon Machine Learning
Instance Recommender

The authors in [13] proposed a scalable architecture for large-scale DNN training in a distributed environment. The framework comprises a four-tier technology stack:

1. Hardware Infrastructure: Nvidia CUDA GPU cluster and nodes
2. Nvidia CUDA Drivers
3. Middleware: HDFS storage and cluster resource management using Apache Flink
4. Application Framework: Enables the management of large data storage (GB scale)

The framework also includes a learning pipeline for scaling across distributed environments, a neural network package, a deep architecture builder, a GPU executor, a linear algebra library, and data format parsers.

A significant challenge in training DNNs, such as CNNs, is the high demand for computational resources and memory bandwidth. Accurate modeling of GPU performance relative to available computational and memory resources is crucial for design optimization. The authors in [14] proposed DeLTA, an analytical GPU model for CNNs that considers various parameters (arithmetic performance, memory hierarchy traffic, data reuse) to optimize computation throughput and memory bandwidth. Their work utilized two NVIDIA Pascal GPUs (P100 and TITAN Xp) and a Volta GPU (V100) and was validated on four CNN architectures (AlexNet, VGG, GoogLeNet, ResNet). Their experimental results demonstrated that the DeLTA architecture could be used for resource space exploration and identifying trade-offs using various scaling parameters to meet different design requirements.

The authors in [15] proposed GRAMARCH, a heterogeneous 3D Network-on-Chip (NoC)-enabled GPU and ReRAM (Resistive Random-Access Memory) architecture that leverages the advantages of ReRAM and GPUs for 3D NoC. Figure 2 illustrates the GRAMARCH architecture, which consists of two layers:

1. Bottom Layer: Contains the GPU and Last Level Cache (LLC) tiles
2. Top Layer: Contains the ReRAM for storage and computation, including eDRAM buffers, in-situ multiple-accumulate units (IMA), output registers, shift-and-add, sigmoid, and max-pool

units

Their experimental results showed a performance improvement of up to 53 times compared to conventional GPUs for image segmentation. Furthermore, the authors in [16] proposed AccuReD, an M3D-enabled architecture combining ReRAM arrays with GPU cores for training CNNs with high performance and accuracy. Their experimental results indicated that the proposed architecture could accelerate CNN training processes by up to twelve times compared to conventional GPU platforms.

Machine Learning in GPU Architecture

Machine learning (ML) algorithms have seen widespread adoption across various domains and hardware platforms in recent years. This section describes different types of machine learning techniques for embedded intelligence on GPUs.

Architecture/Platform/Framework and Strategy

The authors in [56] proposed a parallel approach called the H-ELM (hierarchical extreme learning machine) algorithm, which is based on GPU and Flink, an in-memory cluster computing platform. Figure 7 illustrates the architecture and workflow of H-ELM. Flink uses Java interfaces to communicate with the GPU. The GFlink architecture is depicted in Figure 8.

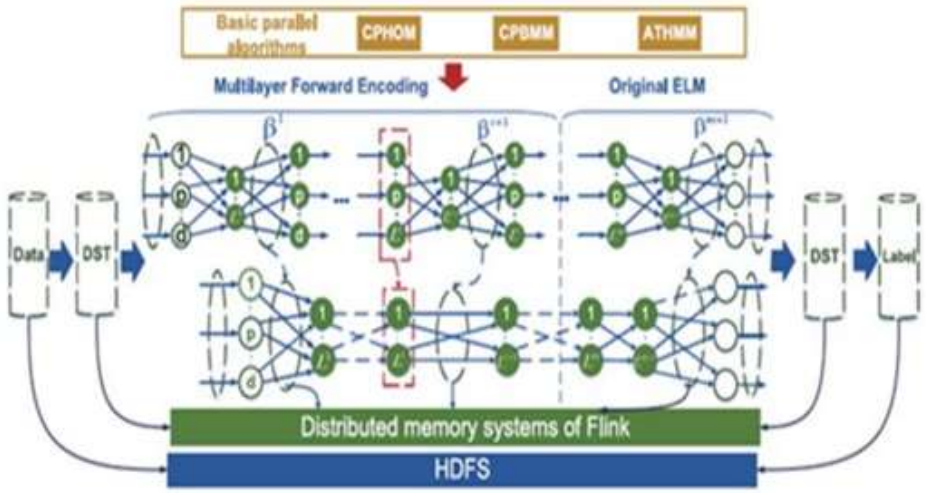


Figure7. H-ELMGPUarchitectureandworkflow[56].(Reprintedwithpermissionfromref.[56].Copyright 2021 IEEE).

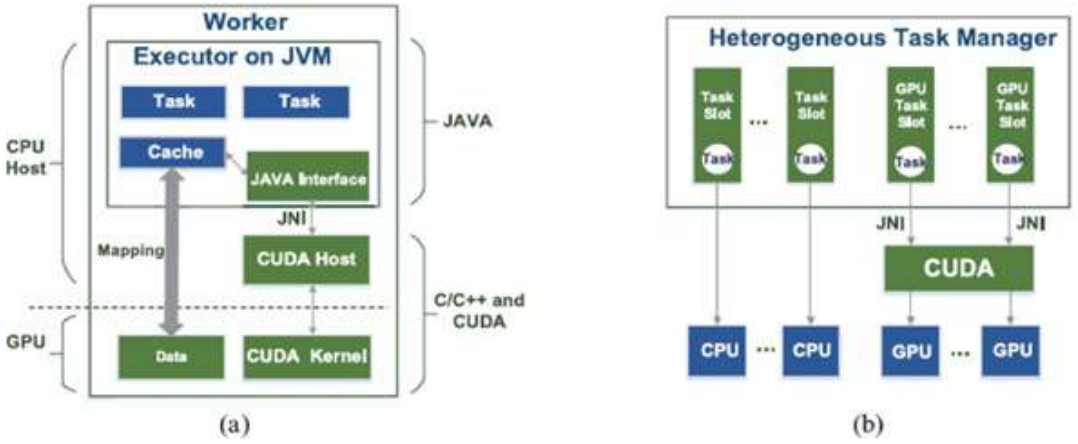


Figure8. GFlink:(a)Architectureofaworknode,(b)Heterogeneoustaskmanagement[56]. (Reprinted with permission from ref. [56]. Copyright 2021 IEEE).

The authors in [57] introduced a GPU architecture integrated with the Spark Big Data framework. HeteroSpark clusters can be conceptualized as Spark clusters with GPUs connected to Spark worker nodes. The HeteroSpark architecture integrated GPU accelerators into the Spark framework to enhance data parallelism and algorithm acceleration. Experimental validation of the HeteroSpark architecture using popular machine learning applications demonstrated an 18-fold performance improvement.

In [58], the authors proposed an efficient GPU-based MapReduce framework to accelerate Support Vector Machine (SVM) learning. GPUs were utilized for parallel numerical calculations, while MapReduce facilitated parallel task scheduling and processing. Their approach employed the MapReduce computational model to parallelize SVM search tasks, resulting in significant performance enhancements for SVM learning.

[59] introduced fast and low-precision learning for a GPU-accelerated spiking neural network (SNN) simulator architecture called ParallelSpikeSim. This simulator employed unsupervised learning and stochasticity to achieve fast and accurate learning with low-precision operations. Experimental results showcased a two to three times speedup in learning performance compared to deterministic SNN architectures across simple and complex datasets.

In [60], the authors proposed an event-based and time-driven SNN simulator for a hybrid CPU-GPU platform. They conducted a comparative study of different simulation methods (event-driven and time-driven) across various computational platforms. Their experimental work implemented the event-driven neural simulator based on lookup tables (EDLUT) in CPU/GPU clusters, resulting in improved spike propagation and queue management times.

[61] presented a neural accelerated architecture for GPUs termed NGPU, enabling scalable integration of neural accelerators for large numbers of GPU cores. NGPU featured improvements such as elimination of fetch/decode during neural execution, reduction of memory/register file accesses, and implementation of the sigmoid using a lookup table. Experimental results demonstrated an average 2.4 times performance improvement speedup and a 2.8 times average energy reduction across a diverse set of benchmarks.

[62] proposed a novel machine learning approach to determine optimal GPU memory requirements for CUDA applications. Their workflow involved two phases: Offline learning and Online inference. The Offline learning phase utilized the NSight CUDA Profiler to collect profiling metrics. Their experimental results accurately predicted optimal memory requirements for discrete memory or unified memory space using various classifiers.

Finally, in [63], the authors introduced a generic sparsity pattern termed Regularized Multi Block (RMB) sparsity pattern, along with an efficient storage format (CRMB) and a fast GPU algorithm for processing the RMB Matrix Multiplication (MM). Their work demonstrated that the RMB sparsity pattern enabled efficient implementations for parallel algorithms and reduced storage for sparse matrices.

Applications

The authors in [66] introduced an approach to optimize GPU energy consumption using dynamic voltage and frequency scaling (DVFS). Their method implemented the DVFS energy management model within a GPU. Experimental results conducted on three GPU platforms (Tesla, Fermi, and Kepler) showcased improvements in both performance and power. Additionally, in [67], another approach called EDVFS utilized GPU and memory coordination to save energy. The EDVFS method adjusted voltage and frequency based on extracted runtime characteristics, achieving maximum energy savings of 10.63% and average energy savings of 2.68% compared to traditional DVFS.

In [68], a GPU-based approach for PLV (phase locking value) biomarkers was proposed, resulting in a 21.3 times improvement in search space efficiency and reduced complexity for on-device processing. Furthermore, [69] presented an efficient GPU-based implementation of multivariate empirical mode decomposition (MEMD) for neural data processing, achieving a performance boost of 6 to 16 times compared to traditional PC-based implementations.

[70] explored GPU usage for simulating large-scale neuronal networks based on the AdEx neuron-model, achieving a fifty-fold performance improvement compared to reference multicore implementations. Similarly, [71] introduced a GPU Simulator of MLMVN, demonstrating a thirty-fold performance enhancement for the MLMVN learning process.

In [72], a novel approach for ECG recognition over GPU platforms using the probabilistic neural network (PNN) was proposed. Experimental results showed improved computational time and algorithm performance compared to other learning models such as SVM and ELM.

[73] proposed a fast soma cell detection approach in knife-edge scanning microscopy (KESM) for high-throughput imagery using GPU-accelerated machine learning, achieving real-time cell detection exceeding traditional KESM data rates. Meanwhile, [74] presented a parallel implementation of chaos neural networks for an embedded GPU using OpenCL, resulting in a pseudo-random number generator that was 49% faster than AES in counter mode.

[75] proposed an approach for anomaly-based intrusion detection system (IDS) using GPU-accelerated neural architecture, showcasing a thirty-fold performance improvement compared to CPU implementations. Additionally, [76] introduced an approach for robot trajectory generation and collision-free trajectory computation for robot swarms using GPU, achieving feasible and collision-free trajectories within seconds. Lastly, [77] proposed the GPU WiSARD Vessel Tracker GWVT, which utilized the WiSARD weightless neural network implemented on a GPU for maritime vessel tracking. Experimental results indicated improved performance compared to CPU trackers.

Conclusions

This paper has provided an extensive survey of the research domain concerning embedded intelligence (EI) within GPU-based architectures and hardware implementations. The review encompassed both contemporary deep learning methodologies and traditional machine learning approaches. Additionally, the discussion extended to GPU memory scheduling and communication, underscoring their significance in advancing EI technologies.

The exploration of GPU-based EI applications spanned various domains, including image processing, computer vision, medical applications, modeling or prediction, convolution, performance analysis, and VLSI placement. These discussions aimed to underscore the broad potential of EI technologies for real-world deployments.

By offering insights into this burgeoning field, this paper endeavors to serve as a valuable resource, inspiring researchers to delve deeper into this pivotal and evolving technological

domain.

References List:

[1]. Prakash, S., Malaiyappan, J. N. A., Thirunavukkarasu, K., & Devan, M. (2024). Achieving Regulatory Compliance in Cloud Computing through ML. AIJMR-Advanced International Journal of Multidisciplinary Research, 2(2).

[2]. Malaiyappan, J. N. A., Prakash, S., Bayani, S. V., & Devan, M. (2024). Enhancing Cloud Compliance: A Machine Learning Approach. AIJMR-Advanced International Journal of Multidisciplinary Research, 2(2).

[3]. Devan, M., Prakash, S., & Jangoan, S. (2023). Predictive Maintenance in Banking: Leveraging AI for Real-Time Data Analytics. Journal of Knowledge Learning and Science Technology ISSN: 2959-6386 (online), 2(2), 483-490.

[4]. Eswaran, P. K., Prakash, S., Ferguson, D. D., & Naasz, K. (2003). Leveraging Ip For Business Success. International Journal of Information Technology & Decision Making, 2(04), 641-650.

[5]. Prakash, S., Malaiyappan, J. N. A., Thirunavukkarasu, K., & Devan, M. (2024). Achieving Regulatory Compliance in Cloud Computing through ML. AIJMR-Advanced International Journal of Multidisciplinary Research, 2(2).

[6]. Malaiyappan, J. N. A., Prakash, S., Bayani, S. V., & Devan, M. (2024). Enhancing Cloud Compliance: A Machine Learning Approach. AIJMR-Advanced International Journal of Multidisciplinary Research, 2(2).

[7]. Biswas, A. (2019). Media Insights Engine for Advanced Media Analysis: A Case Study of a Computer Vision Innovation for Pet Health Diagnosis. International Journal of Applied Health Care Analytics, 4(8), 1-10.

[8] Chopra, B., & Raja, V. (2024). Toward Enhanced Privacy in Digital Marketing: An Integrated Approach to User Modeling Utilizing Deep Learning on a Data Monetization Platform. Journal of Artificial Intelligence General science (JAIGS) ISSN: 3006-4023, 1(1), 91-105.

[9]. Raja, V. (2024). Fostering Privacy in Collaborative Data Sharing via Auto-encoder Latent Space Embedding. Journal of Artificial Intelligence General science (JAIGS) ISSN: 3006-4023, 4(1), 152-162.

[10]. Raja, V. ., & chopra, B. . (2024). Exploring Challenges and Solutions in Cloud Computing: A Review of Data Security and Privacy Concerns. Journal of Artificial Intelligence General Science (JAIGS) ISSN:3006-4023, 4(1), 121–144. https://doi.org/10.60087/jaigs.v4i1.86