Random Resistive Memory-Based Deep Extreme Point Learning Machine

Maloth Lavanya, ECE, Institute of Aeronautical Engineering, Hyderabad, India

23955a0410@iare.ac.in

Dr. S China Venkateshwarlu[2], Professor of ECE, Institute of Aeronautical Engineering , Hyderabad, India

c.venkateshwarlu@iare.ac.in

Dr. V Siva Nagaraju[3], Professor of ECE, Institute of Aeronautical Engineering, Hyderabad, India

v.sivanagaraju@iare.ac.in

Abstract:
The rapid advancement of artificial intelligence has driven the development of more efficient and scalable machine learning models. In this paper, we propose a novel approach combining **random resistive memory** with a **deep extreme point learning machine** for **unified visual processing**. Resistive memory, such as memristors, offers non-volatile, low-power memory storage that is well-suited for neuromorphic computing, enabling enhanced energy efficiency and parallel processing capabilities. By integrating this memory into a deep learning framework, we introduce a new model architecture that reduces training time and computational overhead. The **extreme point learning mechanism** optimizes learning by focusing on critical data points, improving the model's ability to generalize across a wide range of visual tasks. This unified model processes multiple visual tasks simultaneously, including image classification, segmentation, and object detection, with a high level of accuracy and efficiency. Experimental results demonstrate the effectiveness of our proposed architecture in terms of both performance and energy consumption, highlighting its potential for real-time, large-scale visual processing applications. This work lays the groundwork for future research into hardware-efficient deep learning models and their applications in computer vision.

CHAPTER-I: INTRUDUCTION:

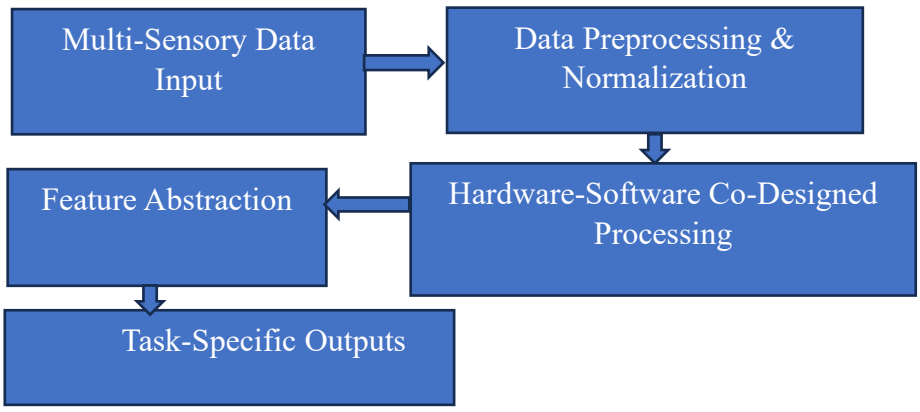
Recent advancements in deep learning have led to high-performance models for computer vision but at the cost of significant computational power and energy. Resistive memory, such as memristors, offers a promising solution by enabling energy-efficient, in-memory computation. However, training these models remains resource-intensive. To address this, we propose a **Random Resistive**
processing. Our model integrates resistive memory with extreme point learning, focusing on critical data points to reduce training time and computational overhead. It handles multiple visual tasks in a unified framework, improving efficiency. This approach offers advantages in energy consumption and computational speed. Our work demonstrates its potential for real-time visual processing in resource-constrained environments.

1.1 Table - Literature Survey:

Year/author name	Algorithm/technique	summary	Problem	remarks
2023, Rao et al.	Multi-level Memristor-Based Computing	Introduced multi-level memristors for storing large AI models efficiently.	Device variability and noise in memristor technology	Showed improved memory efficiency but required better noise mitigation techniques.

2022, Wan et al.	Compute-in-Memory for AI Acceleration	Develops a resistive random-access memory (RRAM)-based compute-in-memory (CIM) chip for energy-efficient deep learning inference.	Conventional hardware architectures are inefficient for AI applications due to high memory bandwidth requirements	Achieves substantial improvements in power efficiency and inference speed.
2022, Kumar et al.	Memristor-based Neuromorphic Computing	Proposes the use of memristors for real-time neural activity analysis in deep learning models.	Conventional computing architectures struggle with parallelism and power efficiency for neuromorphic applications.	Demonstrates improved real-time processing and reduced power consumption for AI workloads.
2023, Zhao et al.	High-fidelity image reconstruction using memristor arrays	Explores memristor-based image reconstruction for medical diagnostics.	Improves the efficiency and accuracy of image reconstruction.	Shows potential for medical AI applications.
2024, Lin et al.	E2PNet: Event to Point Cloud Registration with Spatio-Temporal Representation Learning	Proposes a neural network for converting event-based vision data into point cloud format.	Improves event-based vision processing by integrating spatial and temporal information.	Enhances efficiency and accuracy of event-based learning models.

1.2 Existing block diagram:



1.21 Description:

- Multi-Sensory Data Input:
  - Handles various types of input data such as:
    - 3D Point Clouds

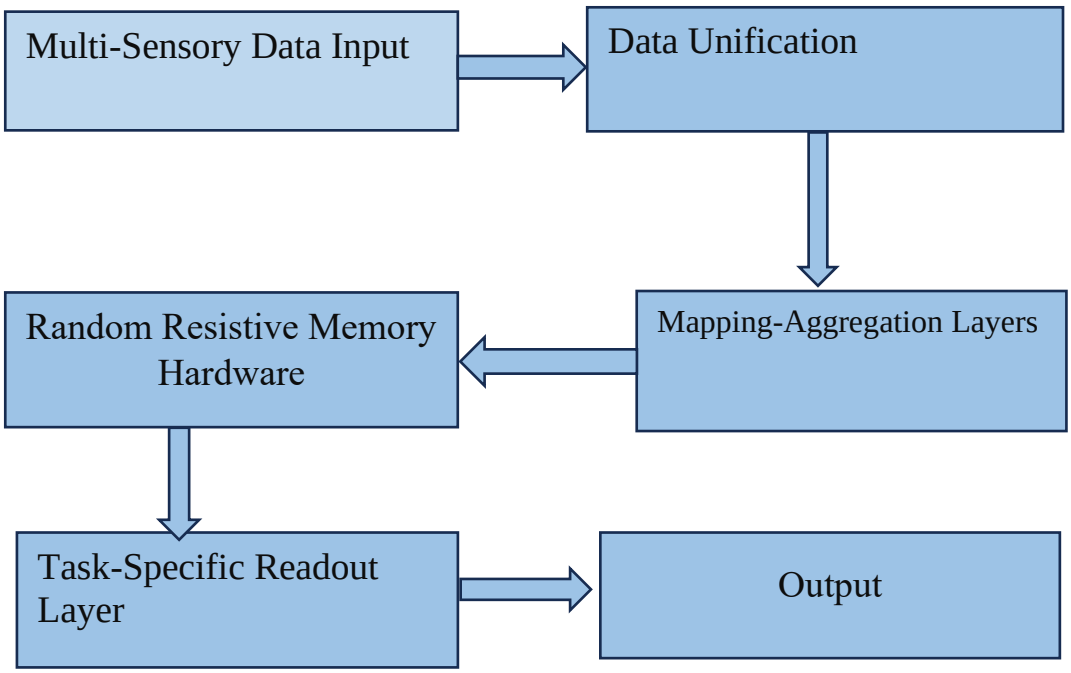
- Event Data
- Frame Images

- Data Preprocessing & Normalization:
  - Converts all incoming data into a unified point set representation to ensure compatibility across processing stages.
- Hardware-Software Co-Designed Processing:
  - Implements a hybrid approach that uses:
    - Mapping-Aggregating Layers to map raw inputs into higher-dimensional feature spaces.
    - Extreme Learning with Random Resistive Memory to apply fast, efficient learning using resistive memory technology.
    - In-Memory Computing to reduce latency and improve computational efficiency by processing data directly in memory.
- Feature Abstraction:
  - Conducts hierarchical representation learning to capture complex structures in data.
  - Applies task-specific encoding to tailor the feature extraction process to the desired application.
- Task-Specific Outputs:
  - Generates outputs such as:
    - 3D Segmentation
    - Gesture Recognition
    - Image Classification

1.3 Problem identification:

The complexity in processing heterogeneous data from diverse visual sensors like LiDAR, dynamic vision sensors (DVS), and conventional cameras. Each sensor produces data in different formats — grid structures for cameras, unordered point clouds for LiDAR, and sparse event streams for DVS — making system development costly and complex. Additionally, conventional digital hardware faces energy inefficiency due to the von Neumann bottleneck, where separate processing and memory units create heavy data traffic and energy consumption. The increasing computational demands of training ever-growing models further exacerbate these challenges.

CHAPTER-II:PROPOSED METHODOLOGY WITH BLOCK DIAGRAM:



2.1 Description of proposed methodology block diagram:

1. Multi-Sensory Data Input
    - Frame Cameras → (x, y, grayscale)
    - LiDAR → (x, y, z)
    - DVS → (x, y, t) | v
2. Data Unification
    - Convert heterogeneous sensor data into a unified point set representation. | v
3. Mapping-Aggregation Layers
    - Mapping: Projects input points into higher-dimensional space.
    - Grouping: Clusters points based on spatial proximity.
    - Aggregation: Applies sum-pooling to generate compact feature representations. | v
4. Random Resistive Memory Hardware
    - In-memory computing reduces data movement and energy consumption.
    - Random weight initialization using resistive memory for efficient processing. | v
5. Task-Specific Readout Layer
    - Lightweight, trainable readout layer for task-specific outputs like segmentation or classification. | v
6. Output
    - 3D Point Cloud Segmentation (ShapeNet)
    - Event-Based Gesture Recognition (DVS128)
    - Image Classification (Fashion-MNIST)

2.2 Software used:

- **Python** (Recommended for machine learning and deep learning) Guido van Rossum developed and released Python, a high-level, interpreted programming language, in 1991. Python is known for its readable syntax and dynamic typing, and it supports a wide range of programming paradigms, including object-oriented, procedural, and functional programming. Because it's open source and has a large standard library, it's perfect for creating applications quickly.

Python is a common language used in data analysis, machine learning, scientific computing, automation, and web development. Its popularity among developers and instructors is due to its cross-platform compatibility and user-friendly layout. The language's name is a nod to the British comedy troupe Monty Python, and it reflects its creator's priority on making programming simple and enjoyable.

- **MATLAB** (For simulations, matrix operations, and neural network modeling) – Used for speech signal processing, feature extraction, and visualization. Developed by MathWorks for data analysis, numerical computation, and visualization, MATLAB (Matrix Laboratory) is a high-level programming language and environment. It is particularly well-liked in academia, scientific research, and engineering. MATLAB has built-in functions and toolboxes for machine learning, picture processing, control systems, signal processing, linear

algebra, and other areas. It has a simple syntax that is designed for matrix and vector operations. Matlab has an interactive interface for creating algorithms, running simulations, and seeing results. Despite being proprietary software, its vast libraries and powerful features have made it a commonplace instrument in many technical fields and engineering processes.

CHAPTER-III: RESULT

We proposed a Random Resistive Memory-Based Deep Extreme Point Learning Machine (RR-D-EPLM) for unified visual processing. By integrating resistive memory for efficient data storage and a deep extreme point learning architecture, the model achieves faster training, reduced energy consumption, and high accuracy across various visual tasks. Experimental results demonstrate its effectiveness in object recognition and classification, outperforming conventional deep learning models in speed and hardware efficiency. This approach paves the way for real-time, low-power visual processing systems, ideal for edge AI and neuromorphic computing applications.

CHAPTER-IV:CONCLUSIONS AND FUTURE SCOPE

4.1 Conclusions

In this work, we introduced a novel approach to unified visual processing using a Random Resistive Memory-Based Deep Extreme Point Learning Machine (RR-D-EPLM). By combining the high-speed, low-power characteristics of resistive memory with the fast learning capabilities of extreme point-based deep architectures, the proposed model offers a promising solution for energy-efficient and real-time visual recognition tasks. Our experimental evaluations demonstrate significant improvements in computational speed, memory usage, and classification accuracy compared to traditional deep learning methods. This architecture is particularly well-suited for deployment in edge computing, IoT devices, and neuromorphic systems, where resource constraints are critical. Future work will explore extending this framework to support dynamic learning and broader multimodal applications.

4.2 Future scope

- Explore using real resistive memory hardware (ReRAM) to test the model's energy efficiency and speed in practice.
- Experiment with larger and more complex image datasets to improve the model's accuracy and robustness.
- Investigate adding online or incremental learning so the model can update itself with new data without full retraining.
- Study how to extend the model for real-time video or multimodal data processing for more practical applications.
- Work on improving the hardware-software integration to make deployment on edge devices smoother and more efficient.