REVIEW ARTICLE

AI Enabled Resource-Constrained Computing Architectures for IoT Devices

Ishfaq Ahmad Malik[1,*] **and Gousia Habib**[1]

[1] Yogananda School of AI, Computer and Data Sciences, Shoolini University, Solan 173229, Himachal Pradesh, India

Abstract

Deep learning is a great success primarily because it encodes large amounts of data and manipulates billions of model parameters. Despite this, it is challenging to deploy these cumbersome deep models on devices with limited resources, such as mobile phones and embedded devices, due to the high computational complexity and the amount of storage required. Various techniques are available to compress and accelerate models for this purpose. Knowledge distillation is a novel technique for model compression and acceleration, which involves learning a small student model from a large teacher model. Then, that student network is fine-tuned on any downstream task to be applicable for resource-constrained applications. This paper explores various state-of-the-art model compression techniques, including knowledge distillation, for compressing large deep neural networks to make them deployable on resource-constrained devices.

Keywords: knowledge distillation, high-performance computing, transformers, DistilBERT, self-attention mechanism.

Submitted: 14 August 2025
Accepted: 18 October 2025
Published: 13 November 2025

Vol. 1, **No.** 3, 2025.
10.62762/TMI.2025.225921

***Corresponding author:**
✉ Ishfaq Ahmad Malik
ishfaqmalik@shooliniuniversity.com

1 Introduction

In the 2012 ImageNet competition, the AlexNet model outperformed all other models. It was inevitable that neural networks would grow in popularity. Many state-of-the-art were broken by 2015. Generally, neural networks can be applied to almost any problem. The success of VGG Net further demonstrated the benefit of using deeper models or ensembles of models. Deep learning is a great success primarily because it encodes large amounts of data and manipulates billions of model parameters. Despite this, it is challenging to deploy these cumbersome deep models on devices with limited resources, such as mobile phones and embedded devices, due to the high computational complexity and the amount of storage required. Despite the overwhelming success of large-scale deep models, their massive computational complexity and storage requirements make it difficult for them to be deployed in real-time applications, especially on devices with limited resources, such as video surveillance [1] and autonomous driving vehicles [2].

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean proposed a strategy in which shallow models are trained under the guidance of pre-trained ensembles. This process is called knowledge distillation [3] because you are distilling knowledge from a pre-trained model into a new model. Because this seems like a teacher guiding a student, it can also be referred to as teacher-student learning. Their main goal was to see how well distillation works by

Citation
Malik, I. A., & Habib, G. (2025). AI Enabled Resource-Constrained Computing Architectures for IoT Devices. *ICCK Transactions on Machine Intelligence*, 1(3), 138–147.

training a neural network with two layers of 1200 linear rectified hidden units on all 60,000 training cases. According to [4], the network was strongly regularized through dropout and weight constraints. Dropout is a method of training an exponentially large ensemble of models with similar weights. Aside from that, the input images had jitters as large as two pixels in any direction. There were 67 test errors with this net, while 146 errors were achieved with a smaller net with two hidden layers of 800 rectified linear hidden units without regularization. However, if the smaller net was regularized by adding the additional task of matching the soft targets produced by the large net at a temperature of 20, it achieved 74 test errors. While the transfer set does not contain translations, soft targets can still transfer much knowledge to the distilled model, including information about how to generalize from translated training data.

The other objective of their research was to examine the effects of ensembling Deep Neural Networks (DNN) acoustic models used in Automatic Speech Recognition (ASR). They show that distillation achieves the desired result of distilling many models into one that works better than a model learned directly from the same training data of the same size. Since that paper, knowledge distillation has recently gained more research interest [26]. Knowledge distillation systems have three key components: knowledge, distillation algorithms, and teacher-student relationship architecture. The basic architecture of the knowledge distillation is given in Figure 1 For students to learn in a knowledge distillation process, types of knowledge, distillation strategies, and teacher-student architecture are crucial. This section will distill the different categories of knowledge.

For a vanilla knowledge distillation, a large deep model's logits are used as the teacher knowledge. The different types of knowledge that can be distilled from teacher to students are given by the following subsections as [5].

Response-based Knowledge Distillation: A response-based knowledge is a neural response from the last output layer of a teacher model. A teacher model is directly mimicked to produce a final prediction. Response-based knowledge distillation is simple yet effective for model compression and is widely used for various tasks and purposes.

Representation-based Knowledge distillation: With increasing abstraction, deep neural networks can learn multiple levels of feature representation. The process is called representation learning. In this manner, a student model can be trained based on the output from the last layer and the output from intermediate layers, i.e., feature maps.

Relation-Based Knowledge Distillation: Response-based and feature-based knowledge are based on the outputs of specific layers in the teacher model. A relation-based knowledge approach further explores how different layers or data samples are related.

After discussing various types of knowledge that can be distilled from teacher to student, it is very important to know the various types of knowledge distillation modes, which are described as: **Offline Distillation**: The whole training process thus has two stages: 1) first, the large teacher model is trained on a set of training samples before distillation; 2) the teacher model extracts knowledge in the form of logits or intermediate features, which are then used to guide the distillation of the student model.

Online Distillation The research community has increasingly focused attention on some issues in offline distillation despite its simplicity and effectiveness. We propose online distillation to overcome the limitations of offline distillation, especially when obtaining a large-capacity, high-performance teacher model is difficult. Using online distillation, the student and teacher models are updated simultaneously, and the entire framework can be trained from end to end.

Self Distillation: The teacher and student models use the same networks during self-distillation. As a special case of online distillation, this is possible. Zhang et al. [4] proposed a new self-distillation method in which knowledge from the deeper sections is conveyed to the shallow sections.

2 Resource Constrained Convolutional Neural Networks

CNNs have remarkable success in computer vision, satellite imagery, biomedical imaging, and cosmogenic images [6]. However due to the over-parameterization of CNNS, they are computationally intensive, limiting their practical application to the resource-constrained environment. Much research is on making these CNN models light and sparse, such as quantization, weight multiplexing, and mixed precision. All the techniques perform well but face the serious limitation of reducing accuracy while achieving the compression ratio. This issue has gained more research interest,

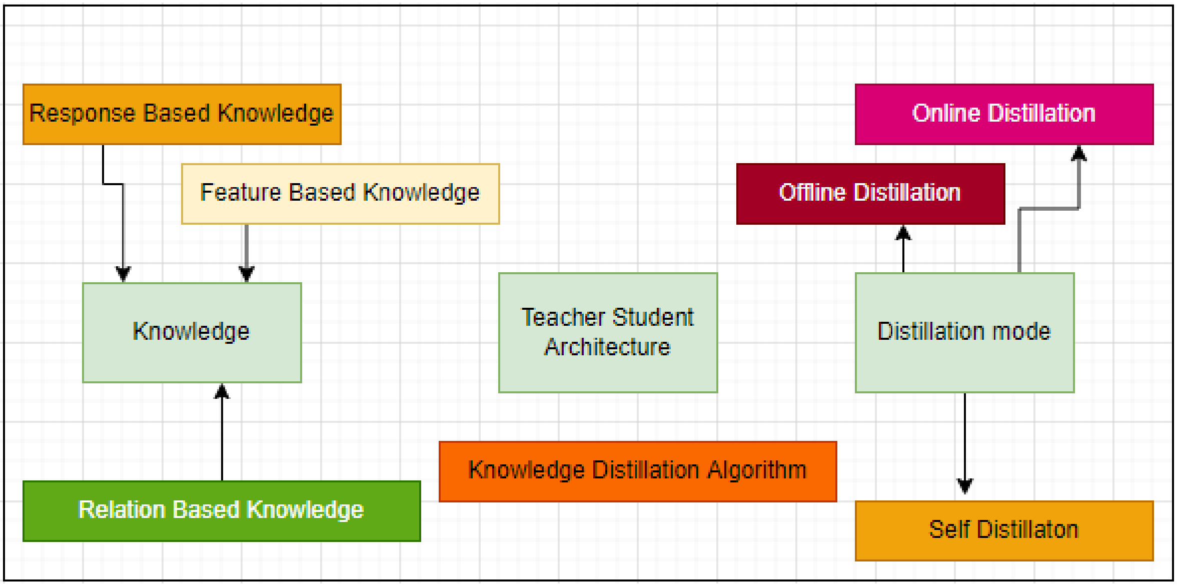

Figure 1. Generic framework of knowledge distillation.

and novel compression techniques for knowledge distillation exist for the compression of CNN models. This technique makes these models lightweight and sparse with a 1% or 1.5% accuracy drop. For instance, recent work on object detection demonstrates up to 85% model size reduction and 6× inference speedup on IoT hardware like Raspberry Pi using contrastive KD [28].

A knowledge distillation process can transfer the teacher's knowledge from the teacher CNN model to the smaller student CNN model by taking the original huge model as the teacher model. This allows the model to be compressed and sparse, making it feasible in resource-constrained environments.

3 Major Limitation of Neural Networks(NNs)

It is important to note that even though neural networks are highly powerful, they have several limitations. A large amount of labelled data is often required for efficient training, and high-performance hardware is often required. Interpretability of neural networks continues to be challenging, as they often appear as black boxes without transparency in their decision-making.

There is a risk that neural networks will not be robust, and deployment may require significant computer resources. Additionally, training data biases can result in biased models, and hyper-parameter tuning is a highly sensitive process. Whenever it comes to image-related problems, CNN is the go-to model. When it comes to accuracy, they are miles ahead of the competition. Various other applications are also possible using it, including recommendation systems, natural language processing, and more.

With CNN, unlike its predecessors, essential features are detected automatically without human intervention. In one example, it learns to distinguish between these characteristics by looking at many images of cats and dogs. Additionally, CNN is computationally efficient. It performs parameter sharing and uses special convolution operations and pooling operations. It allows CNN models to run on any device, which makes them universally appealing.

Despite its remarkable success, CNN has become a universally accepted model for computer vision tasks. CNNs face critical limitations, including their inability to encode positional relationships between interrelated features. It is necessary to use extensive filters to encode the combination of these features. An example might be that large filters are necessary to encode "eyes above the nose and mouth." Therefore, CNN does not consider the inter-relationship between the features, which degrades its performance compared to more powerful deep learning models such as transformers. Large receptive fields are required for long-range dependency tracking within an image. Expanding the convolution kernels increases the network's representational capacity and reduces the efficiency of the computations and statistics obtained using local convolutions.

Self-Attention modules It is a type of attention mechanism in which CNNs can help model long-range dependence without compromising computational and statistical efficiency. This module is complementary to convolutions and helps model long-range, multi-level dependencies across regions of an image. Self-attention involves every sequence element interacting with one another to determine who they should pay more attention to. This way, "long-term" information and dependencies between sequence elements can be captured.

Our goal was to demonstrate how self-attention can effectively resolve some of the limitations of convolutional networks. The Transformers model, which is based on attention rather than CNN, may be able to replace CNNs, mainly known as vision transformers, completely.

4 Brief Introduction to Revolutionary Deep Learning models "Transformers"

Transformers are dominant sequence transduction models, including encoder and decoder neural networks based on recurrent or convolutional neural networks. In addition, the best models incorporate an attention mechanism to connect the encoder and decoder. With attention mechanisms, dependencies can be modelled without considering their distance from the inputs and outputs of a task. Self-attention, or intra-attention, is a mechanism by which different positions of a single sequence are related to computing a representation. As an efficient way to comprehend reading, abstract summarization, entail text, and learn task-independent sentence representations, self-attention has been successfully applied to various tasks [7].

The Transformer is the first transduction model to rely exclusively on self-attention to represent input and output without using sequence-aligned RNNs or convolution. Building on the self-attention mechanism, the most competitive neural sequence transduction models typically employ an encoder-decoder structure.

An encoder maps a sequence of symbol representations $(x_1, .., x_n)$ to a sequence of continuous representations $(z_1, .., z_n)$. Based on z, the decoder generates an output sequence $(y_1, ..., y_m)$ of symbols over time. Model steps are auto-regressive, consuming previously generated symbols as input for the next step. As shown in Figure 2, the Transformer architecture employs stacked self-attention layers and fully connected layers in both the encoder (left) and decoder (right).

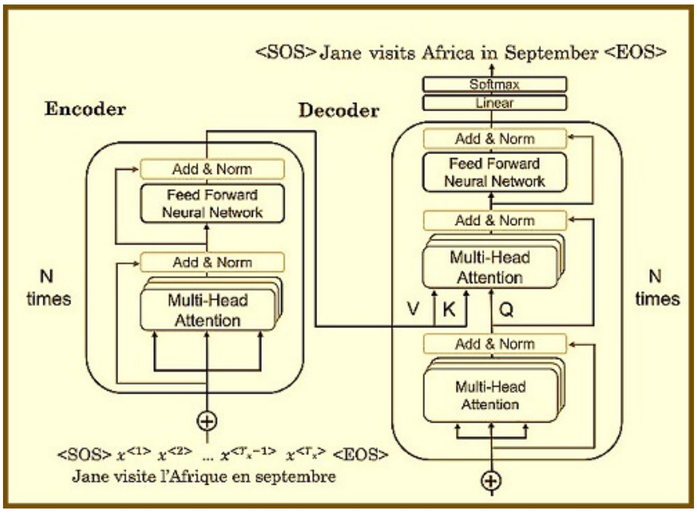
<SOS> Jane visits Africa in September <EOS>
Softmax
Linear
Encoder
Decoder
Add & Norm
Feed Forward Neural Network
Add & Norm
Multi-Head Attention
N times
Add & Norm
Feed Forward Neural Network
Add & Norm
Multi-Head Attention
V
K
Q
Add & Norm
Multi-Head Attention
N times
<SOS> x<1> x<2> ... x<Tx-1> x<Tx> <EOS>
Jane visite l'Afrique en septembre

Figure 2. Generic framework of Transformer.

However, Transformer architecture remains limited in its applications to computer vision despite its de facto status as the standard for natural language processing. It is used either to replace components of convolutional networks while keeping their overall structure intact or to use attention in conjunction with convolutional networks. In this [8] study, we show that a pure transformer applied directly to sequences of image patches can perform very well on image classification tasks without relying on CNNs.The Vision Transformer (ViT) achieves excellent results compared to state-of-the-art convolutional networks when trained on large quantities of data and transferred to multiple mid-sized or small image recognition benchmarks (ImageNet, CIFAR-100, VTAB, etc.). It requires significantly fewer computational resources to train than state-of-the-art convolutional networks.

5 Open Challenges in Transformers

Despite enabling sample-efficient learning, Transformers may have lower performance ceilings due to their inherent inductive biases. Recent studies have shown that Vision Transformers (ViTs) outperform CNNs for classifying images using self-attention layers. Their cost is increased by the time and effort required for pre-training on large external datasets and distillation from pre-trained convolutional networks. Even sequential tasks require very large datasets to train and are computationally very heavy. The large memory, energy power, and computational requirements of transformer models demand high-performance computing systems (HPCs).

Another major limitation of deep learning models, such as CNNs and transformers, is that they are heavily over-parameterized and need a lot of computation and

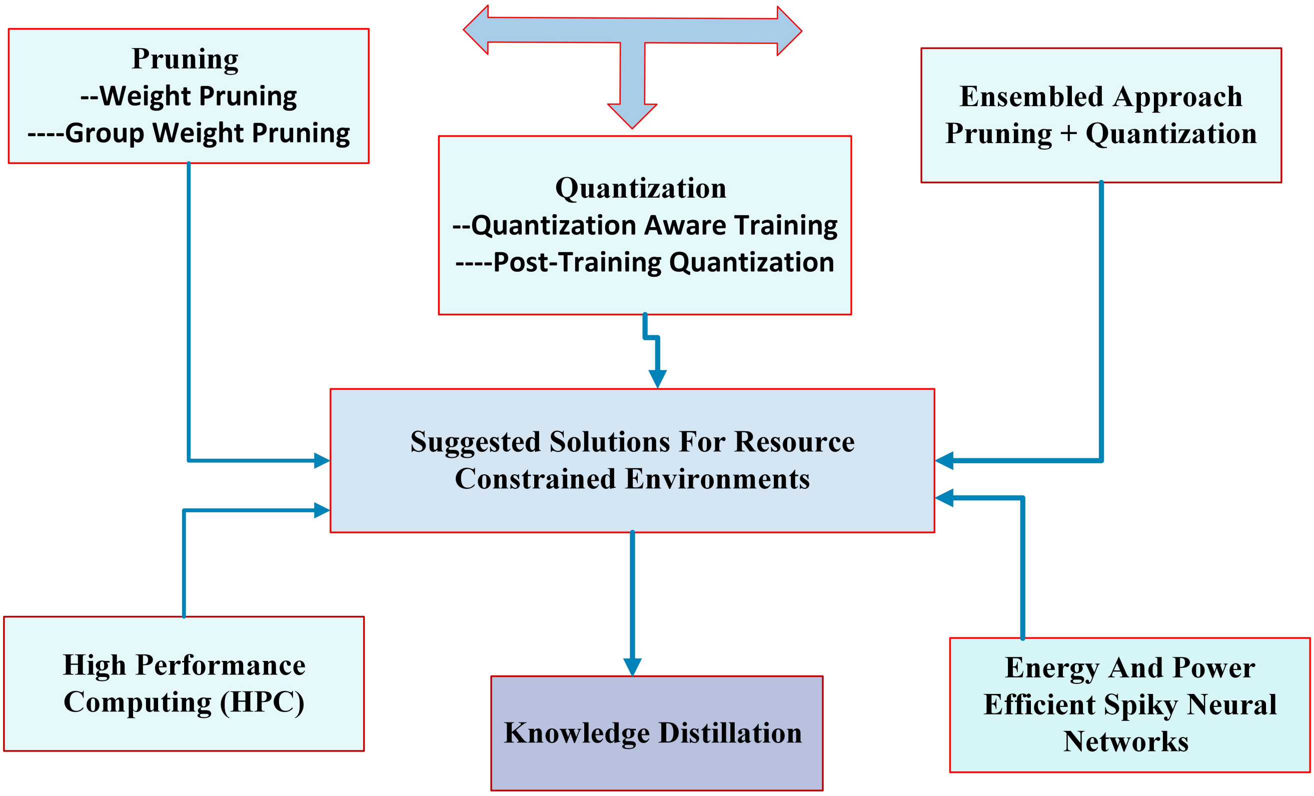
Pruning
--Weight Pruning
----Group Weight Pruning
Quantization
--Quantization Aware Training
----Post-Training Quantization
Ensembled Approach
Pruning + Quantization
Suggested Solutions For Resource Constrained Environments
High Performance Computing (HPC)
Knowledge Distillation
Energy And Power Efficient Spiky Neural Networks

Figure 3. Solutions for deployment of ML models on resource constrained environment.

memory resources. These requirements limit their applications in resource-constrained environments such as deploying IoT and embedded devices. It has gained much attention, and exhaustive research is going on to reduce the model size to make it feasible for such environments. This is an open challenge to be researched at the current time.

6 Suggested Solutions To Mitigiate the Challenges

The Taxonomy of Suggested Solutions is illustrated by Figure 3

6.1 High-performance computing HPC

Engineering is always looking for ways to optimize existing hardware features for maximum power and performance at the lowest cost. The process is accomplished through system engineering and software optimization. A high-performance computer (HPC) processes data and rapidly performs complex calculations. In terms of performance, a laptop or desktop with a 3 GHz processor can perform around 3 billion calculations per second. Compared with HPC solutions capable of performing quadrillions of calculations every second, that's much faster than any human can achieve. A supercomputer is one of the most famous types of HPC solutions. A supercomputer's task is accomplished by thousands of computing nodes working in conjunction with one another. In this case, parallel processing is used. Distributed computing systems are like thousands of computers networked to speed up work.

With advances in technology such as the Internet of Things (IoT), artificial intelligence (AI), and 3-D imaging, the size and amount of data organizations have to deal with are growing exponentially. Real-time processing is essential for many purposes, including streaming a live sporting event, tracking the development of a storm, testing new products, or studying stock market trends. For organizations to stay competitive, IT infrastructure that processes, stores, and analyzes vast amounts of data must be lightning-fast and reliable. The three main components of HPC solutions are computing, network, and storage. Using clustering technology, high-performance computing can be built by connecting compute servers. Multiple software programs and algorithms are running simultaneously on clustered servers. The cluster is networked to the data storage system to capture the output. Each component works seamlessly with the others to accomplish various tasks.

Every component must keep pace with the others to function at maximum efficiency. Storage components must be able to feed and ingest data to and from compute servers as quickly as processing takes place. In addition, the networking components must be able to move data between compute servers and storage at high speeds. The HPC infrastructure will perform poorly if one component cannot keep pace with the rest.

Using Intel, Xeon Platinum processors, this paper [9] will describe the design, development, and implementation of the High-Performance Computing (HPC) and Artificial Intelligence (AI) Intel® Server System family.

6.2 Pruning

Deep learning has attained remarkable success across a range of applications, including computer vision, machine translation, voice recognition, and language understanding, often surpassing human-level performance. However, deploying deep neural networks (DNNs) in real-world applications necessitates not only high accuracy but also efficiency in terms of latency and energy consumption. This need has led to the development of model compression techniques, which aim to reduce the size and computational requirements of DNNs without significantly compromising their performance. Two primary techniques used for model compression are pruning and quantization.

Pruning is a widely used technique in neural networks that entails the removal of unnecessary or less critical weights, thereby decreasing the model's size and the number of operations needed during inference. This results in a reduction in latency and energy consumption. There are two primary approaches to pruning:

Individual weights pruning: Individual weights pruning involves zeroing out the least significant weights in the network [17].

Group weights pruning: Group weights pruning, on the other hand, entails removing entire groups of weights, such as neurons, channels, or even layers, which can lead to more efficient utilization of hardware accelerators and increased computational efficiency. Techniques for group weights pruning can be found in works like. By reducing the number of active weights, pruning can significantly decrease the memory footprint of the network and the number of computations required during inference [18].

6.3 Quantization

Quantization is a technique that reduces the precision of the weights and activations in a neural network by using lower bit-width representations, such as 8-bit integers. This approach results in two main benefits: memory savings, as lower bit-width representations require less storage, and reduced computation, as operations on lower bit-width data types are faster and consume less power. Quantization techniques

Table 1. Summarised distilled methods.

Distilled Model	Size (millions)	Training Time	Performance	Data	KD Method Employed
BERT	Base: 110 Large: 340	Base: 8 V100 12 days* Large: 64 TPU Chips 4 days	Outperforms state-of-the-art	16 GB BERT data	BERT (MLM and NSP)
RoBERTa	Base: 110 Large: 340	Large: 1024 V100 1 day	2-20% improvement	160 GB	BERT without NSP
DistilBERT	Base: 66	Base: 8 V100 3.5 days	3% degradation	16 GB BERT data	BERT Distillation
TinyBERT [9]	14.5	Speedup = 9.4X	77.0	GLUE [16]	Attention-based
BERT(6-PKD) [10]	67.0	Speedup = 2.0X	81.5	GLUE	Multi-layer distillation
MKD-LSTM(L=2) [13]	10.2	–	73.0	GLUE	Multi-task distillation
MobileBERT Tiny [11]	15.1	–	77.0	GLUE	–
BERT(4-PKD) [12]	52.2	Speedup = 3.0X	72.6	GLUE	Multi-layer distillation
DistilBERT-4 [14]	52.2	Speedup = 3.0X	71.9	GLUE	Response-based
DistilBERT-6 [15]	67.0	Speedup = 3.0X	76.8	GLUE	Response-based

can be categorized into two types: quantization-aware training (QAT) and post-training quantization (PTQ) [19]. QAT involves simulating quantization effects during training, allowing the model to learn to compensate for the reduced precision. This often results in higher accuracy compared to PTQ. PTQ applies quantization to a pre-trained model without retraining and is faster and more convenient, making it particularly beneficial for large language models or scenarios where fine-tuning is impractical [20].

6.4 Ensemble of Pruning and Quantization

Enhancing the efficiency of models by reducing their size is achieved by combining pruning and quantization. This can be accomplished through fine-tuning with pruning in the loop, which integrates pruning into the training process, allowing the model to adapt to the pruned structure. Alternatively, post-training pruning can be used, which prunes the model after it has been fully trained, making it more suitable for deployment on resource-constrained environments. To make full use of the benefits of these optimizations, specialized hardware support is required [21].

6.5 Knowledge Distillation

The first section of the paper gives an introduction to knowledge distillation. We have different ways of distilling knowledge from teacher to student. We can either use homologous knowledge distillation techniques or cross-architectural knowledge distillation.

In homologous knowledge distillation, we can have a deeper CNN model as a teaching model and a shallow CNN network as a student model. Suppose we can use ResNEt-101 as a teacher model responsible for disseminating knowledge. The Resnet 50 model can be taken as a student model for mimicking the knowledge from the teacher model. It could be from CNN-CNN or Transformer to Transformer.

Through the cross-architectural knowledge distillation technique, we can have a heterogeneous architecture. It could be from CNN-Transformers or Transformer-CNNs. There is a large architectural gap between the teacher and the student model. So, distilling and mimicking knowledge in such a heterogeneous environment is an open research problem that needs to be researched.

Much research is currently on knowledge distillation techniques for compressing deep learning models, particularly deep CNNs and transformers, to make them deployable in resource-constrained environments such as IoT devices [27]. The summary of various distilled or compressed versions of the transformer models using the knowledge distillation technique is summarized in Table 1.

6.6 Transition to Spiky Neural Networks(SNNs) for Energy and Power Efficiency

The spiking neural network (SNN) is distinguished by its extremely low power consumption, making it appealing for implementing edge intelligence in resource-limited environments. The idea of merging artificial intelligence (AI) with edge computing, often referred to as edge intelligence, has garnered considerable interest from researchers. Edge intelligence aims to shift the workload of AI algorithms from centralized servers to network edges, resulting in reduced processing delays and required bandwidth.

This approach could benefit a multitude of applications, ranging from autonomous driving to video surveillance. However, implementing edge intelligence presents significant challenges, primarily due to the limited computational power and restricted battery life of edge devices. Moreover, as network architectures continue to scale, deploying large models onto edge devices becomes impractical. Therefore, it is crucial to explore energy-efficient neural network (NN) models that support the use of edge intelligence.

Spiking neural networks (SNNs) have emerged as a promising solution for achieving low-power edge intelligence [22]. Unlike conventional neural networks that use real numbers, SNNs process information through binary spike trains, which consist of sequences of action potentials or "spikes" generated by neurons. This spike-based computing approach results in high energy efficiency and always-on operation, making SNNs particularly suitable for resource-constrained edge devices. Due to these desirable properties, SNNs have gained extensive attention from both academia and industry. Recently, distributed SNNs have been introduced [23] to address the need for integrating information from multiple sources, with sensors and processors located in different places.

This distributed computing paradigm can utilize idle computation resources and enhance communication efficiency through collaboration, enabling the development of large-scale neural networks.

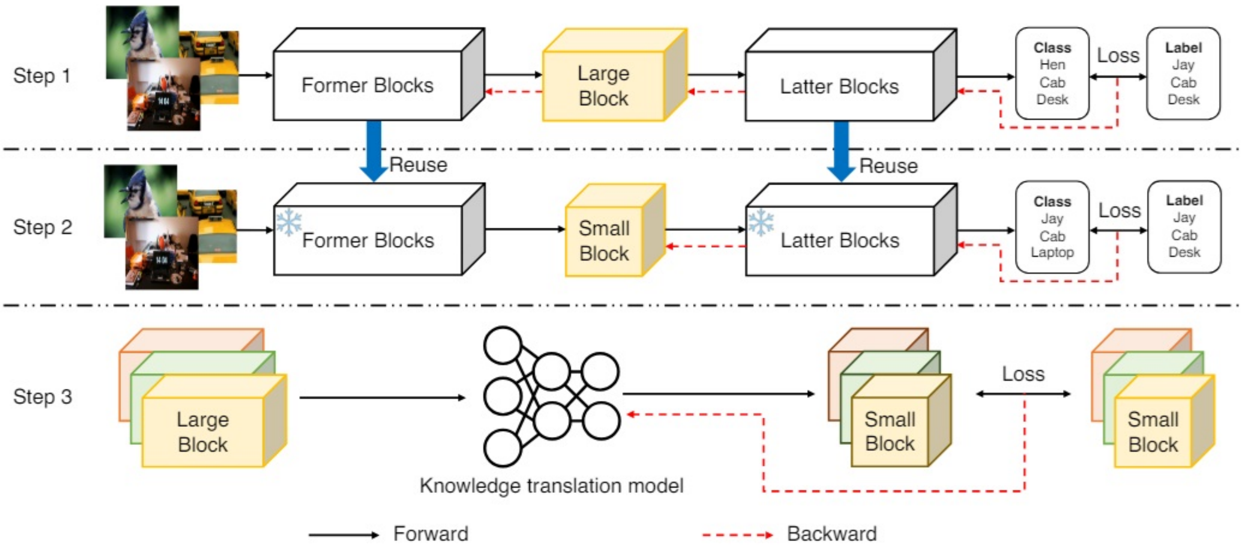

Figure 4. Overview of knowledge translation.

Additionally, SNNs are well-suited for distributed deployment because spiking neurons are event-driven and generate minimal amounts of data, which can significantly reduce communication costs, especially in wireless environments. Potential applications of distributed SNNs include security alarms, environmental monitoring, intelligent robots [24], and healthcare. It is believed that such a distributed neuromorphic computing paradigm will contribute to realizing the full potential of edge AI systems.

Among all the techniques or solutions discussed in above sections, Knowledge distillation, or KD, is widely viewed as an effective technique for compressing models. In this process, a smaller, student model is trained with the guidance of a larger, pre-trained teacher model or an ensemble of models. Numerous strategies have been proposed since the original concept, which mimic various aspects of the teacher, such as its representation space, decision boundary, or intra-data relationships. Some methods replace the traditional one-way knowledge distillation from a static teacher with collaborative learning involving a group of students. Despite recent progress, there is still a lack of clarity regarding where knowledge is stored within deep neural networks and the optimal method for capturing and transferring knowledge from the teacher to the student remains an open question.

Knowledge distillation involves transferring knowledge from a large pre-trained teacher model to a smaller, untrained student model. The goal is to align the outputs of the teacher and student models to facilitate the transfer of knowledge, which leads to improved performance for the

student model. Initially, this knowledge transfer occurs in the form of logits, but it can also take other forms, such as layer-wise features and cross-layer features. However, it is important to note that knowledge distillation is not limited to any specific architectural requirements between the teacher and student models. Despite this, it does require a higher training overhead since the student model must be trained a new.

6.7 Knowledge Translation(KT)

Rather than attempting to translate the entire model, the another alternative is to focus on translating specific components, such as a large intermediate block within a network, into a smaller intermediate block. The process consists of three main stages: generating the input data (Step 1), creating the target data (Step 2), and training the knowledge translation model (Step 3), as depicted in Figure 4 [25].

7 Conclusion and Future Scope

This article investigates the process of knowledge distillation, which involves transferring knowledge from larger, complex teacher models to smaller, efficient student models using different distillation algorithms. We discuss the architecture of these models for both sequence-to-sequence and computer vision applications, focusing on techniques such as pruning and quantization that help reduce model size and computational requirements. The paper explores the challenges of maintaining performance while optimizing models for resource-limited environments like IoT and embedded devices.

Furthermore, we provide an extensive overview of major distilled models suitable for mobile and embedded platforms, summarizing their performance, size, and speed improvements in a detailed table. The paper also identifies key challenges and proposes open research opportunities to further advance this field. By doing so, we offer insights into how to effectively deploy advanced models in practical applications, ensuring they meet the constraints of limited-resource settings without compromising on efficiency and accuracy.

Furthermore, this paper examines significant challenges in knowledge distillation, with particular focus on maintaining performance while reducing model size and computational demands. It proposes open research opportunities to address these challenges. Furthermore, the paper delves into optimizing models for IoT and embedded devices through techniques such as pruning and quantization. Lastly, the paper presents a comprehensive table summarizing key distilled models suitable for mobile and embedded platforms, highlighting their performance, size, and speed improvements, providing a clear overview of their applicability in resource-constrained environments.

Future Scope: Augmentation techniques can be employed to introduce inductive bias into transformers to enhance their performance. By utilizing hybrid parallelism that combines data and model parallelism, both CNNs and transformers can meet their computational requirements and improve inferences. Attention mechanisms can be integrated into CNNs to make them more robust models. Additionally, regularization methods can be developed to refine the transformers' inductive bias, making them both more sparse and lightweight.

Future research can also concentrate on creating adaptive learning rates and dynamic neural architectures that adjust based on the task's complexity, as well as examining federated learning to enhance model performance while safeguarding data privacy in distributed environments.

From the comparison Table 1, it is clear that attention-based knowledge-based methods perform well when implemented on the same data set as GLUE. Attention-distilled models dramatically reduce the number of parameters with only 1–1.5%.

Data Availability Statement

Not applicable.

Funding

This work was supported without any funding.

Conflicts of Interest

The authors declare no conflicts of interest.

Ethical Approval and Consent to Participate

Not applicable.

References

[1] Bouguettaya, A., Zarzour, H., Taberkit, A. M., & Kechida, A. (2022). A review on early wildfire detection from unmanned aerial vehicles using deep learning-based computer vision algorithms. *Signal Processing, 190*, 108309. [CrossRef]

[2] Le Mero, L., Yi, D., Dianati, M., & Mouzakitis, A. (2022). A survey on imitation learning techniques for end-to-end autonomous vehicles. *IEEE Transactions on Intelligent Transportation Systems, 23*(9), 14128-14147. [CrossRef]

[3] Lin, S., Xie, H., Wang, B., Yu, K., Chang, X., Liang, X., & Wang, G. (2022, June). Knowledge Distillation via the Target-aware Transformer. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 10905-10914). IEEE. [CrossRef]

[4] Zhang, L., Chen, X., Tu, X., Wan, P., Xu, N., & Ma, K. (2022, June). Wavelet Knowledge Distillation: Towards Efficient Image-to-Image Translation. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 12454-12464). IEEE. [CrossRef]

[5] Yang, C., Zhou, H., An, Z., Jiang, X., Xu, Y., & Zhang, Q. (2022). Cross-image relational knowledge distillation for semantic segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 12319-12328). [CrossRef]

[6] Hassan, S., Andrianomena, S., & Doughty, C. (2020). Constraining the astrophysics and cosmology from 21 cm tomography using deep learning with the SKA. *Monthly Notices of the Royal Astronomical Society, 494*(4), 5761–5774. [CrossRef]

[7] Subakan, C., Ravanelli, M., Cornell, S., Bronzi, M., & Zhong, J. (2021). Attention is all you need in speech separation. In *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 21–25). [CrossRef]

[8] Dosovitskiy, A. (2020). An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929.*

[9] Jiao, X., Yin, Y., Shang, L., Jiang, X., Chen, X., Li, L., ... & Liu, Q. (2020, November). Tinybert: Distilling bert for natural language understanding. In *Findings of the association for computational linguistics: EMNLP 2020* (pp. 4163-4174). [CrossRef]

[10] Nityasya, M. N., Wibowo, H. A., Chevi, R., Prasojo, R. E., & Aji, A. F. (2022). Which student is best? a comprehensive knowledge distillation exam for task-specific bert models. *arXiv preprint arXiv:2201.00558*.

[11] Cui, B., Li, Y., & Zhang, Z. (2021). Joint structured pruning and dense knowledge distillation for efficient transformer model compression. *Neurocomputing, 458*, 56–69. [CrossRef]

[12] Haider, M. H., Valarezo-Plaza, S., Muhsin, S., Zhang, H., & Ko, S. B. (2024, May). Optimized Transformer Models: ℓ' BERT with CNN-like Pruning and Quantization. In *2024 IEEE International Symposium on Circuits and Systems (ISCAS)* (pp. 1-5). IEEE. [CrossRef]

[13] Tahaei, M., Charlaix, E., Nia, V., Ghodsi, A., & Rezagholizadeh, M. (2022). KroneckerBERT: Significant Compression of Pre-trained Language Models Through Kronecker Decomposition and Knowledge Distillation. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* (pp. 2116–2127). [CrossRef]

[14] Sanh, V., Debut, L., Chaumond, J., & Wolf, T. (2019). DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

[15] Adoma, A. F., Henry, N. M., & Chen, W. (2020, December). Comparative analyses of bert, roberta, distilbert, and xlnet for text-based emotion recognition. In *2020 17th international computer conference on wavelet active media technology and information processing (ICCWAMTIP)* (pp. 117-121). IEEE. [CrossRef]

[16] Feng, L., Yu, J., Cai, D., Liu, S., Zheng, H., & Wang, Y. (2021). ASR-GLUE: A new multi-task benchmark for asr-robust natural language understanding. *arXiv preprint arXiv:2108.13048*.

[17] Tian, Y., Chen, H., Guo, T., Xu, C., & Wang, Y. (2023). Towards higher ranks via adversarial weight pruning. *Advances in Neural Information Processing Systems, 36*, 1189-1207.

[18] de Resende Oliveira, F. D., Batista, E. L. O., & Seara, R. (2024). On the compression of neural networks using ℓ0-norm regularization and weight pruning. *Neural Networks, 171*, 343-352. [CrossRef]

[19] Xu, K., Li, Z., Wang, S., & Zhang, X. (2024). PTMQ: Post-training Multi-Bit Quantization of Neural Networks. In *Proceedings of the AAAI Conference on Artificial Intelligence* (Vol. 38, No. 14, pp. 16193–16201). [CrossRef]

[20] Zhang, A., Yang, Z., Wang, N., Qi, Y., Xin, J., Li, X., & Yin, P. (2025). Comq: A backpropagation-free algorithm for post-training quantization. *IEEE Access*. [CrossRef]

[21] Kuzmin, A., Nagel, M., Van Baalen, M., Behboodi, A., & Blankevoort, T. (2023). Pruning vs quantization: Which is better?. *Advances in neural information processing systems, 36*, 62414-62427.

[22] Hemmati, A., Raoufi, P., & Rahmani, A. M. (2024). Edge artificial intelligence for big data: a systematic review. *Neural Computing and Applications, 36*(19), 11461-11494. [CrossRef]

[23] Eshraghian, J. K., Ward, M., Neftci, E. O., Wang, X., Lenz, G., Dwivedi, G., ... & Lu, W. D. (2023). Training spiking neural networks using lessons from deep learning. *Proceedings of the IEEE, 111*(9), 1016-1054. [CrossRef]

[24] Che, C., Zheng, H., Huang, Z., Jiang, W., & Liu, B. (2024). Intelligent robotic control system based on computer vision technology. *arXiv preprint arXiv:2404.01116*.

[25] Sun, W., Chen, D., Chen, J., Feng, Y., Chen, C., & Wang, C. (2024). Knowledge Translation: A New Pathway for Model Compression. *arXiv preprint arXiv:2401.05772*.

[26] Mansourian, A. M., Ahmadi, R., Ghafouri, M., Babaei, A. M., Golezani, E. B., Ghamchi, Z. Y., ... & Kasaei, S. (2025). A Comprehensive Survey on Knowledge Distillation. *arXiv preprint arXiv:2503.12067*.

[27] Cantini, R., Orsino, A., & Talia, D. (2024). Xai-driven knowledge distillation of large language models for efficient deployment on low-resource devices. *Journal of Big Data, 11*(1), 63. [CrossRef]

[28] Setyanto, A., Sasongko, T. B., Fikri, M. A., Ariatmanto, D., Agastya, I. M. A., Rachmanto, R. D., ... & Kim, I. K. (2025). Knowledge Distillation in Object Detection for Resource-Constrained Edge Computing. *IEEE Access*. [CrossRef]

Ishfaq Ahmad Malik received his PhD in Mathematics from NIT Srinagar in 2019, and Masters degree in Mathematics from University of Kashmir in 2010. Qualified NET with AIR-013 and GATE with a score 360. His interests are Functional Analysis, Machine Learning and Differential equation. (Email: ishfaqmalik@shooliniuniversity.com)

Gousia Habib received her PhD degree in Computer Science from NIT Srinagar in 2023, Pursued Post doc from IIT Delhi, NUS Singapore and presently doing post doc from University of Helsinki. Her interests are Machine learning, AI and Reinforcement learning. (Email: gousiya.cstaff@iitd.ac.in)