Indonesian Journal of Electrical Engineering and Computer Science
Vol. 33, No. 3, March 2024, pp. 1843~1854
ISSN: 2502-4752, DOI: 10.11591/ijeecs.v33.i3.pp1843-1854

Deep neural networks optimization for resource-constrained environments: techniques and models

Raafi Careem[1], Md Gapar Md Johar[2], Ali Khatibi[3]
[1]Department of Computer Science and Informatics, Uva Wellassa University, Badulla, Sri Lanka
[2]Software Engineering and Digital Innovation Centre, Management and Science University, Shah Alam, Malaysia
[3]School of Graduate Studies, Management and Science University, Shah Alam, Malaysia

Article Info

Article history:

Received Oct 16, 2023
Revised Dec 25, 2023
Accepted Jan 3, 2024

Keywords:

Deep neural network
Image classification
Optimization model
Optimization technique
RCE

ABSTRACT

This paper aims to present a comprehensive review of advanced techniques and models with a specific focus on deep neural network (DNN) for resource-constrained environments (RCE). The paper contributes by highlighting the RCE devices, analyzing challenges, reviewing a broad range of optimization techniques and DNN models, and offering a comparative assessment. The findings provide potential optimization techniques and recommend a baseline model for future development. It encompasses a broad range of DNN optimization techniques, including network pruning, weight quantization, knowledge distillation, depthwise separable convolution, residual connections, factorization, dense connections, and compound scaling. Moreover, the review analyzes the established optimization models which utilizes the above optimization techniques. A comprehensive analysis is conducted for each technique and model, considering its specific attributes, usability, strengths, and limitations in the context of effective deployment in RCEs. The review also presents a comparative assessment of advanced DNN models' deployment for image classification, employing key evaluation metrics such as accuracy and efficiency factors like memory and inference time. The article concludes with the finding that combining depthwise separable convolution, weight quantization, and pruning represents potential optimization techniques, while also recommending EfficientNetB1 as a baseline model for the future development of optimization models in RCE image classification.

Corresponding Author:

Raafi Careem
Department of Computer Science & Informatics, Uva Wellassa University
Badulla, Sri Lanka
Email: mraafi@gmail.com

1. INTRODUCTION

Deep neural networks (DNNs) have been widely used for image classification tasks due to their exceptional accuracy and performance [1]–[4]. Nevertheless, deploying these valuable models on devices with limited resources, including mobile phones and embedded systems, remains challenging task due to the low computing and energy capacities of these resources [5], [6]. These devices are collectively referred to as resource-constrained environment (RCE), characterized by constrained computing capacities, as illustrated in Table 1, where central processing unit (CPU) frequency ranging from 1.2 GHz to 2.84 GHz, and random-access memory (RAM) capacity varies from 1 GB to 8 GB.

The rapid growth of the internet of things (IoT) and the growing popularity of mobile devices have highlighted the importance of adapting DNNs for use in these RCEs [4], [7]–[9]. Given the limitation of RCE devices, deploying DNNs on such devices requires careful consideration of several challenges including increased model size and computational complexity [10], [11]. Furthermore, RCE devices, which operate under limited resources necessitate optimization approaches that reduce model size and resource consumption while preserving accuracy.

Table 1. Typical resource-constraint environment with their specifications

Device name	CPU model	CPU frequency (GHz)	RAM (GB)
Raspberry Pi 4B	Broadcom BCM2711	1.5	2-8
NVIDIA Jetson Nano	NVIDIA Carmel ARMv8.2	1.43	4
Google Coral Dev Board	NXP i.MX 8M	1.5	1-4
Google Pixel	Qualcomm Snapdragon 821	2.15	4
iPhone XR	Apple A12 Bionic	2.49	3
Google Pixel 3	Qualcomm Snapdragon 845	2.5	4
Google Pixel 2	Qualcomm Snapdragon 835	2.35	4
Google Pixel 4	Qualcomm Snapdragon 855	2.84	6
Xiaomi Mi 10	Qualcomm Snapdragon 865	2.84	8
Samsung S10	Exynos 9820	2.73	8
Huawei P30 Pro	Kirin 980	2.6	8
Sony Xperia ZL	Quad Core	1.5	2
940MX	Core Speed	1.2	4

Source: depicted from [9], [12]–[14]

The rapid popularity of RCEs in recent years, driven by smart devices and IoT, has intensified the demand for implementing DNNs in these environments [10], [11]. The advantages of deploying DNNs on RCEs are numerous, including real-time image classification, low latency, reduced bandwidth usage, and enhanced privacy and security [15]. DNNs on RCE devices enable rapid data processing, facilitating faster decision-making in various domains, including agriculture, transportation, healthcare, smart homes, and autonomous vehicles.

However, deploying very deep neural networks on RCE devices poses challenges due to their substantial size and high computational requirements [10]. In response to these challenges, numerous studies have emerged, exploring various techniques and DNN models. Han *et al.* [16] employed a pruning technique to reduce the size of networks in AlexNet [17]–[20] by 9x without compromising accuracy in image classification. Similarly, Li *et al.* [21] utilized filter pruning techniques to lower inference costs for DNN models, such as VGG-16 [22] and residual network (ResNet)-110 [23], on the CIFAR10 [24] image classification dataset. Despite its benefits, pruning the DNN architecture can result in accuracy loss and introduce complexity during model training. Another approach is the quantization of the network, where floating-point numbers representing weights and activations are quantized to a smaller number of bits. For example, Yang *et al.* [25] applied weights and activation quantization to enhance image classification accuracy. However, the use of fewer bits for weights may lead to a loss of precision, impacting neural network accuracy. Knowledge distillation, as used in [26], [27] involves transferring knowledge from a large and complex DNN (teacher network) to a smaller and simpler DNN (student network). While distillation has improved accuracy [28], it may result in information loss during the transfer and require additional computational cost for training the large model. Dense connection, an optimization technique used in [29], [30] connects all layers in the DNN architecture directly to each other, facilitating efficient information flow to prevent loss. On the other hand, Chollet [31] applied depthwise separable convolution techniques, which use depthwise convolution and pointwise convolution to train and run more complex models [32], [33] with limited computational resources. However, the use of depthwise convolution techniques in DNN models reduces the prediction accuracy of the trained model during inference time [33]. Tan and Le in [34] introduced the compound scaling technique, involving scaling the depth, width, and resolution of a neural network simultaneously to achieve better accuracy. While compound scaling can improve accuracy, it demands additional computational resources, memory, and CPU [35]. Accordingly, several DNN models such as MobileNet [32], [33], [36]–[39], ResNet [18], [20], [23], [40]–[44], InceptionNet [18], [40], [45], [46], DenseNet [29], [44], [47], and EfficientNet [34], [48] have incorporated these techniques for RCE-based image classification, each offering its own set of advantages and limitations.

The objective of this paper is to review and analyse the aforementioned techniques and DNN models in order to identify the suitable techniques and base line model to propose a new DNN model for DNN based image classification on RCEs. The paper offers a comprehensive review of these state-of-the-art optimization techniques and models for deploying DNNs in RCE devices by analyzing their attributes, usability, strengths and weaknesses. Furthermore, it introduces a novel DNN model considering the advantages of combining multiple optimization techniques with an appropriate baseline DNN model for image classification in RCE devices, serving as a valuable reference for future research. The insight presented in this paper can be a useful reference for researchers and professionals engages in the development of DNN for RCE devices.

2. RESEARCH METHOD

The research method commenced with a comprehensive review of significant research articles, including peer-reviewed journals and conference proceedings, in the state-of-the-art field up to the year 2023. The review encompassed papers sourced from reputable online databases for computer science research such as Google Scholar, Scopus, Elsevier, IEEE Xplore, ACM, Science Direct, Web of Science, Emerald, and Springer. Initially, a focused review aimed to acquire existing knowledge on DNN-based image classification for resource limited devices RCE from journals and articles. Keywords such as “deep neural network,” “image classification,” “resource-constrained environment,” and “optimization techniques” were employed to establish a foundation of knowledge on the implementation of DNN-based image classification applications on RCE devices. This study identified potential DNN optimization techniques, including network pruning, weight quantization, knowledge distillation, depthwise separable convolution, residual connections, factorization, dense connections, and compound scaling. The extracted optimization techniques underwent a comprehensive analysis using a Table 2, examining their principles, usages, strengths, and weaknesses concerning the implementation of these techniques for deploying DNN models on RCE devices.

Another phase of the review used the keywords “DNN models utilizing the identified optimization techniques” to identify potential DNN models incorporating the above optimization techniques for image classification applications on RCE devices. The extracted DNN models then underwent a comparative analysis of their architectural structure, usage, strengths, and weaknesses using a Table 3. Furthermore, the study performed an additional analysis by tabulating empirical data, including “DNN models”, “model size,” “inference time,” and “accuracy,” to identify the trade-offs among these factors influencing DNN deployment on resource-limited devices RCE (refer to Table 4). This analysis aims to determine the most promising DNN base model for the development of new models for image classification in RCE devices in the future.

2.1. Optimization techniques

Deploying DNNs on RCE devices can be difficult because these devices have limited computational resources and energy constraints. To address these challenges, researchers have created various optimization techniques. These techniques can help reduce the size and complexity of DNNs while maintaining their performance. The following section review each of these techniques including its principles along with their strengths and weaknesses.

2.1.1. Network pruning

Network pruning [21], [49], [50] is a technique used to reduce the computational cost and memory requirements of DNNs by eliminating unimportant connections and weights. It can be implemented during or after training, based on the importance of weights. Studies have shown that pruning can significantly reduce the size of DNNs while maintaining their accuracy. For example, Han *et al.* in [16] used network pruning to reduce the size of AlexNet by 9x without affecting its accuracy. Similar to this, Li *et al.* in [21] employed pruning to lower inference costs for VGG-16 and ResNet-110 on the CIFAR10 image classification datasets by up to 34% and 38%, respectively. Pruning has several advantages [16], [21], such as making DNNs more efficient for memory-limited devices, especially for RCE devices with limited memory and computational resources. Pruned networks require fewer weights and computations, allowing them to be trained and used for inference faster than unpruned ones. However, pruning the network in some cases can have a negative impact on prediction accuracy [51].

2.1.2. Quantization

Weight quantization is a method that reduces the precision of weights in DNNs by using fewer bits instead of floating-point numbers. It can significantly reduce memory and energy requirements while maintaining accuracy [25]. Weight quantization is beneficial for DNNs training and deployment, as it allows for memory savings and faster inference times. However, it can also result in a loss of precision, especially for smaller models, and introduce errors that accumulate across layers. Therefore, a careful balance between accuracy and memory usage is essential to achieve the desired level of accuracy without sacrificing memory efficiency.

2.1.3. Knowledge distillation

Knowledge distillation [26], [27] is a technique that involves transferring knowledge from a larger, more complex neural network, known as the teacher network, to a smaller, simpler one, known as the student network. The teacher network generates soft targets or probability distributions for each input, which the student network uses to learn from. Knowledge distillation has been shown to significantly improve the accuracy and performance of small and simple neural networks. For example, Fu *et al.* [28] proposes a method of knowledge distillation called interactive knowledge distillation for training a light-weight student network under the guidance of a well-trained, large teacher network that outperformed a larger and more complex DNN on several image classification tasks.

2.1.4. Depthwise separable convolution

Depthwise separable convolution [7], [31] is a technique used in DNNs to maintain high accuracy while reducing the number of parameters and computations required. It combines two types of convolutions: depthwise convolution and pointwise convolution. Depthwise convolution applies different weights to each input channel independently, while pointwise convolution operates on a single pixel at a time and combines information from different channels of the input. To create a depthwise separable convolution, a depthwise convolution is applied to the input data, producing a set of feature maps with the same spatial dimensions as the input.

Then, a pointwise convolution is applied to the output of the depthwise convolution, using a 1x1 filter to combine information from different channels of the output feature maps. This resulting convolutional layer requires fewer parameters and computational cost than a traditional convolutional layer, making it particularly useful in applications with limited computational resources. Depthwise separable convolution has several advantages, including reducing model size by significantly reducing the number of model parameters, faster training and inference, and improved regularization and generalization of the model. However, it may lead to a reduction in accuracy compared to traditional convolutional layers, particularly when the input data contains complex spatial patterns. Optimizing depthwise separable convolutional layers can be challenging due to the significant impact of hyperparameters on model accuracy.

2.1.5. Residual connections

Residual connections, also known as skip connections, are a technique used in DNNs to overcome the vanishing gradient problem [52], [53] and improve information flow [32], [36], [54], [55]. In a feedforward neural network, each layer nonlinearly transforms the input, and the output of one layer is fed as input to the next. The final layer produces the network's prediction. The vanishing gradient problem can hinder convergence or prevent it in deep neural networks with many layers. Residual connections solve this problem by enabling the network to skip one or more layers instead of passing through each layer in sequence. A shortcut connection is added that bypasses one or more layers and feeds directly to a later layer. The key idea behind residual connections is that the network can learn the identity function as a special case, allowing it to pass the input through the residual connection without applying any nonlinear transformations.

2.1.6. Factorization of convolution

Factorization of convolution [56], [57] is a technique used in DNN structures to reduce the computational load of convolutional layers while retaining their expressive abilities. This involves breaking down a traditional convolution operation into smaller operations, each with fewer parameters than the original filter. This can be particularly useful on resource-limited devices like mobile phones or embedded systems. Advantages of factorization of convolution include increased computational efficiency and reduced number of parameters. Factorization can make convolutional layers more efficient to train and run, resulting in faster inference times and lower memory requirements. It can also reduce the number of parameters in a model, preventing overfitting and making models more robust to noisy or sparse input data.

2.1.7. Dense connection

Dense connection [29], [30], [47] is a technique in DNNs that connects all layers directly to each other, unlike traditional feedforward neural networks. It involves feeding the output of each layer as an input to all subsequent layers, creating a dense graph of connections. This technique improves gradient flow during backpropagation, addressing the vanishing gradient problem and enabling faster and more stable training of deep neural networks. It promotes feature reuse and prevents overfitting, leading to better generalization and performance on new data. Dense connection reduces the number of parameters required in the network. It reduces the complexity of the model resulting in faster training and inference times and reducing the risk of overfitting. It has been shown to improve performance of deep neural networks, particularly on image classification and recognition tasks. However, it requires a large number of connections between layers, increasing computational cost, memory usage, and complexity. The implementation of dense connection requires careful tuning and hyperparameter selection to achieve optimal performance.

2.1.8. Compound scaling

Compound scaling [34] and Swish activation [58] are two techniques used in DNN to enhance their performance. Compound scaling involves scaling the depth, width, and resolution of a neural network simultaneously to improve accuracy and efficiency. This approach is based on the idea that networks with more parameters are more accurate but also more computationally expensive. By scaling the network architecture appropriately, it can lead to faster training and inference times. Compound scaling can significantly improve model accuracy while reducing training time and allowing for better resource management. However, it can add complexity to the design process and risk overfitting if the network is scaled too aggressively.

2.2. Deep neural network models

DNN models use a range of optimization techniques to operate at optimal efficiency in deep learning applications. These models, referred to as optimization models, play a crucial role in enhancing the efficiency and effectiveness of DNNs. The popular optimization models for image classification in RCE devices have been discussed in the following section.

2.2.1. ResNet

ResNet is a kind of DNN based on ResNet that is made up of residual blocks [18], [23], [59]–[61]. These blocks have skip or shortcut connections that let identity mappings get through weight layers. There are several versions of ResNet [62] for image classification tasks. For example, the ResNet-50 model a version of ResNet, which has 50 layers with 48 convolutional layers, one MaxPool layer, and one average pool layer, has been a popular choice. ResNet's skip connections enhance the output of weight layers and identity mappings to prevent the vanishing gradient problem in deep layers. ResNet uses bottleneck layers with 1x1 filters to decrease the amount of parameters, and the weight layers are commonly made up of two 3x3 convolutional layers. With these bottleneck layers, two layers are replaced by three layers of 1x1, 3x3, and 1x1 filters, with the 1x1 filters being used to first lower and then increase the number of weights. ResNet152 is another version ResNet which is a 152-layer ResNet that is the deepest version of ResNet. It has the most parameters and is the most complex, but it can achieve even higher accuracy on some tasks. ResNet50 is a deep neural network that has shown remarkable accuracy in image classification tasks, particularly on benchmark datasets like ImageNet. Its 50-layer depth allows it to capture intricate patterns and features within images, making it an excellent choice for recognizing complex visual elements.

2.2.2. DenseNet

Huang *et al.* in [29] developed DenseNet, a kind of DNN that expands on the ResNet-introduced residual learning paradigm and DenseNet block. In a DenseNet block, as opposed to ResNet, each layer is joined to all the succeeding layers with equivalent feature map dimensions through concatenation rather than addition. The ResNet bottleneck architecture, which uses a 1x1 filter to limit the number of input channels before being routed into the 3x3 convolution layer, is comparable to the architecture of the convolution layers in DenseNet. A compression or transition layer is used between DenseNet blocks to reduce the amount of feature mappings. DenseNet uses a lot less training data than ResNet while still being about as accurate. The network is made up of a convolution layer, four DenseNet blocks, three transition layers, and a fully connected (FC) layer at the bottom. DenseNet includes multiple versions, among which DenseNet-121 [60] is the smallest version with 121 layers. It has 4 dense blocks and 3 transition layers, and it is typically used for small-scale image recognition tasks. DenseNet is a densely connected convolutional neural network architecture that enhances information flow, memory efficiency, and speed of training.

2.2.3. MobileNet

Embedded and mobile vision applications made use of DNNs of the MobileNet design [32], [60]. MobileNet is specifically designed to run on low-processing-power hardware, such as smartphones and IoT devices. The fundamental idea of MobileNet is that it uses of depthwise separable convolutions, which separate the standard convolution process into a depthwise convolution and a pointwise convolution. While the depthwise convolution applies a single filter to each input channel, the pointwise convolution utilizes a 1x1 filter to aggregate the outputs. By doing this, precision is maintained while fewer computations and parameters are needed. MobileNet also uses other techniques, such as linear bottlenecks with shortcut links, to improve speed and reduce model size. Additionally, the MobileNets may be modified using the width and resolution multipliers, also known as model shrinkage hyperparameters. The width multiplier is used to proportionately reduce the number of channels in the network at each layer, which lowers the overall number of parameters for that layer. By changing the input resolution for the model, the implicit setting of the resolution multiplier lowers the computational cost of the model. MobileNet is an efficient DNN designed for resource-constrained devices like smartphones and IoT devices. It balances model size, accuracy, and performance by using depthwise separable convolutions and techniques like linear bottlenecks and shortcut connections.

2.2.4. Inception network

A DNN architecture called inception [45], [60] was developed by Google researchers to address scalability issues in DNNs, notably for image recognition applications. As part of the inception idea, many filters of different sizes are included at every level of the network to capture features at various scales. Each layer of an inception network uses convolutional layers of different sizes, such as 1x1, 3x3, and 5x5. The output from these layers is then aggregated and fed to the layer below. This lowers the number of parameters while preserving the network’s ability to learn and gather data at different sizes. The inception architecture has undergone a number of revisions, with each one bringing new changes and advancements. Two of them have enhanced the original Inception architecture: inception-v3 [18] and inception ResNet-v2 [63], [64]. With 48 layers, inception-v3 use “inception modules” to quickly calculate filters of various sizes inside a layer. For improved gradient flow and feature representation learning, inception ResNet-v2, with 164 layers, combines the inception architecture with residual connections.

2.2.5. EfficientNet

EfficientNet, introduced by Tan and Le [34], is a DNN model designed to achieve high accuracy with fewer parameters and processing than other models like ResNet and inception. It consists of convolutional layers, global average pools, and a fully connected layer for classification. The convolution layers are organized in repeating blocks, with skip connections enhancing the gradient flow and information flow. EfficientNet has achieved state-of-the-art performance on various image classification benchmarks, including ImageNet, and is well-suited for deployment on limited resources, such as mobile devices and edge computing systems. EfficientNetB1 [48], a variation of EfficientNet, offers state-of-the-art performance while maintaining a small model size and low computational cost. It is suitable for applications on devices with RCEs due to its modest model size, manageable parameters, and relatively high accuracy. EfficientNetB1 achieves efficiency by employing swish activation functions and squeeze excitation (SE) blocks (a special kind of block to which gives priority to the weights according to its features), increasing the network's expressiveness, and compound scaling, scaling the network's depth, width, and resolution in a principled manner. EfficientNetB1 is a viable alternative for developing new DNN models for low-resource devices, as it achieves a reasonable balance between accuracy and efficiency.

3. RESULTS AND DISCUSSION

The article explore the challenges associated with deploying DNNs on devices with limited resources and introduces various optimization techniques and models suitable for deploying DNNs on RCEs. Section 2 introduces various optimization techniques and offering a comprehensive overview by applying these techniques to diverse DNN models. The ensuing Table 2 encapsulates the discussed optimization techniques for image classification on RCE devices, detailing their respective advantages and drawbacks.

Table 2 illustrates that there are many different strategies that appeal inside the DNNs, each with their own advantages and disadvantages. For instance, network pruning provides the attractive prospects of a smaller model and faster inference. However, there are certain limitations, such as the trade-off between accuracy and increased training complexity. Contrarily, weight quantization reduces memory requirements and speeds up inference while potentially degrading model performance and accuracy.

Knowledge distillation works as a way to transfer knowledge from a complicated DNN to a straightforward one, encouraging effectiveness and quick inference. However, it might result in information loss and a rise in computing expenses. Meanwhile, depthwise separable convolution reduces model size and speeds up training, but there is a chance that accuracy will suffer. Residual connections and factorization provide problems like overfitting and the requirement for demanding tuning along with enticing benefits like accuracy gains and increased operational efficiency. Although careful trade-off is required and may increase model complexity and memory usage, factorization and residual connections may promise improvements in speed and performance. Dense connections reduce the number of parameters and improve performance, but they also increase the complexity of the architecture and add computational and memory expenses. Finally, compound scaling improves training effectiveness, precision, and resource management, while they frequently come with increased computing cost.

Table 2. Analysis of DNN optimization techniques for image classification

DNN deployment optimization techniques	Strengths	Weaknesses	Example DNN models using the technique	Connected references
Network pruning	Reduced model size; increased inference speed	Loss of accuracy; increased training complexity	ResNet, VGG16, VGG19	[21], [49], [50]
Weight quantization	Reduced memory; faster inference; lower power	Reduced model accuracy; reduce performance	MobileNetV2	[25], [65], [66]
Knowledge distillation	Efficient models; faster inference	loss of information; computational cost	MobileNetV3	[26]–[28], [65], [67], [68]
Depthwise separable convolution	Reduction in model size; faster training and inference	complex spatial patterns	MobileNet, Xception, EfficientNet,	[7], [31], [41]
Residual connections	Improved accuracy; easier training; improved gradient flow	Increased complexity	ResNet, DenseNet, and Inception-v4.	[32], [41], [44], [54], [55], [69], [70]
Factorization of convolution	Computational efficiency; reduced number of parameters; improved accuracy	Increased model complexity	InceptionV3, MobileNet, Xception, EfficientNet, and ShuffleNet, InceptionResNetV2	[56], [57]
Dense connections	Reduced network parameters; better performance	Increased computational cost increased memory usage more complex architecture	DenseNet, DenseNet121	[29], [30], [44], [47]
Compound scaling and swish activation	Improved accuracy; efficient training and resource management	Increased computational cost	EfficientNet MobileNetV3	[34], [39], [48], [58]

Table 3. Comparing the strengths and weaknesses of DNN architectures

DNN model architecture	Strengths	Weaknesses	Comparative analysis	Connected references
MobileNet (MobileNetV2)	Optimized for limited computing power devices; efficient and speedy; computation reduction	Lower accuracy compared to other CNN models; performance limited by hardware capabilities; width and resolution multipliers may require trial and error to optimize	Excels in efficiency and speed but sacrifices some accuracy when compared to other models	[32], [33], [36]–[39], [60]
ResNet (ResNet50, ResNet152V2)	Deep network with larger number of layers for complex feature learning; uses skip connections to prevent vanishing gradient problem.	Complex network; resource-intensive. Overfitting risk; regularization essential	Stands out for its deep network and skip connections but faces challenges with complexity and resource usage when compared with MobileNet	[18], [23], [40]–[42], [59]–[61], [69], [71]–[74]
Inception (InceptionV3, InceptionResNetV2)	Efficient computation with varied filter sizes for accuracy	Difficult to train and fine-tuning; Loss of image details; demands high memory for RCE	Offers efficient computation and versatility but at the cost of potential difficulties in training and high memory demands than MobileNet	[18], [40], [45], [46], [60], [63], [75]
DenseNet (DenseNet121)	Dense connectivity improves network flow and gradient flow; fewer parameters - memory efficiency	High computational cost; potential to accuracy loss	Effective but higher memory consumption and computational cost Compared to MobileNet and EfficienetNet (B0 and B1)	[29], [47], [60], [72], [75],
EfficientNet (EfficientNetB0-EfficientNetB7	Efficient and Fast; highly Accurate; versatile Architecture	complex architecture	EfficientNet demonstrates efficiency, speed, and accuracy, but it has complex architecture	[34], [48]

Each optimization technique has its own advantages and disadvantages. The appropriate optimization technique depends on the specific task requirements and trade-offs between different factors. Therefore, choosing the right optimization technique is essential to achieve improved performance, reduced model size and faster inference for DNN deployment on RCE devices. The combination of depthwise separable convolution, weight quantization, and pruning techniques can be considered as a promising approach to address the challenges of model size, inference speed, and memory footprint on RCE devices. Depthwise separable convolution can simplify the model structure and increase inference speed by factorizing the standard convolution. Weight quantization can further reduce memory usage by representing the model weights using fewer bits. However, this may come at the cost of some reduction in model accuracy. Pruning can also help reduce model size by removing redundant connections and weights. By taking these factors into account, future research can develop efficient and effective DNN models for image classification on RCE devices.

Concurrently, sections 3 of this paper presents DNN models which utilizing the said optimization techniques for image classification towards RCE devices. The DNN model ResNet employs skip connections and residual blocks to get around the vanishing gradient issue where as DenseNet uses concatenation for layers. MobileNet models employs depthwise separable convolutions to reduce the model size where as a small-scale variant called EfficientNet manages to preserve a modest model size and minimal computing cost while achieving great accuracy. The Table 3 compares the strengths and weaknesses of the different DNN models for image classification on RCE. MobileNet is optimized for limited computing power devices and is efficient and speedy but has lower accuracy compared to other DNN models. ResNet50 and ResNet152V2 are state-of-the-art image classification models with deep networks but are complex and resource-intensive. InceptionV3 and InceptionResNetV2 are efficient in computation with varied filter sizes but are difficult to train and fine-tune. DenseNet121 has fewer parameters and is memory efficient but has high memory consumption and computational cost. Finally, EfficientNetB0, B1, and B7 are efficient, fast, and highly accurate with a versatile architecture.

The subsequent Table 4 demonstrates the deployment information of the state-of-the-art DNN models for image classification with ImageNet [76] dataset in Keras deep learning platform [35], including evaluation matrix, accuracy and efficiency (memory and inference time). Each model has designed to maintain a compromise between accuracy, model size, and the time required for inference. The most accurate models, such as EfficientNetB7 and InceptionResNetV2, achieve high accuracy (84.30% and 80.30%), meaning the percentage of images for which the model correctly identifies the main object, but have a large model size (256 MB and 215 MB) and a long time per inference step (1,579 ms and 130 ms). These models are computationally intensive and require a significant amount of memory and processing power (heavy-weight models). On the other hand, models with lower accuracy (light-weight models) such as MobileNet and EfficientNetB0 (70.4% and 77.1%), have smaller model sizes (16 MB and 29 MB) and shorter inference times (22.6 ms and 46 ms). These models are optimized for RCEs where computational resources are limited. However, they sacrifice some accuracy in order to achieve faster inference times and a smaller memory utilization.

Table 4 highlights that DNN models, the EfficientNet architecture, which is made to have great accuracy with less processing than other models. The EfficientNet’s compound scaling method scales the network's depth, width, and resolution uniformly while maintaining the computing budget. In order to be deployed on RCE devices, EfficientNetB1 is a small-scale variant that achieves excellent accuracy while preserving a small model size and affordable computational cost. Table 4 highlights EfficientNetB1's efficiency in terms of speed and memory usage and high accuracy. Therefore, using EfficientNetB1 as a reference model to propose a new model for DNN deployment on RCE is a suitable choice given its high accuracy and efficient design.

Table 4. Comparison of DNN models’ deployment for image classification for ImageNet dataset in terms of evaluation matrix model size, inference time and accuracy

DNN models	Model size (MB)	Inference time (ms)	Accuracy (%)
MobileNet	16	23	70.4
EfficientNetB0	29	46	77.1
EfficientNetB1	31	60	79.1
DenseNet121	33	77	75.0
InceptionV3	92	42	77.9
ResNet50	98	58	74.9
InceptionResNetV2	215	130	80.3
ResNet152V2	232	108	78.0
EfficientNetB7	256	1,579	84.3

Source: depicted from [35]

4. CONCLUSION

This article provides a comprehensive exploration of the optimization techniques and connected DNN models related to deploying DNNs in RCEs with a specific focus on image classification applications. RCE devices offer tremendous potential for DNN applications, but their limited resources present unique challenges. The paper analyzed a wide range of optimization techniques, from network pruning to weight quantization, knowledge distillation, depthwise separable convolution, and more. Each technique was discussed in terms of its attributes, usability, strengths and weaknesses, highlighting the need for a strategic integration of these methods to address the requirements of RCE deployment. The article further discussed the optimization models, including ResNet, MobileNet, InceptionNet, DenseNet, and EfficientNet with their variations. These models were analyzed for their suitability in RCE scenarios, considering factors such as computational efficiency, model size, memory usage, and accuracy. The analysis from Table 4 shows that among these models, EfficientNetB1 emerged as the strong candidate due to its balance of efficiency and accuracy. The article aligns with its research title by emphasizing the importance of optimizing DNNs for RCEs and presents a systematic view of the available techniques and models to achieve this goal. It highlights the critical role of careful selection and integration of optimization techniques and models, depending on the specific demands of the deployment scenario. The process of analyzing the challenges of using DNNs on RCE devices, analyzing optimization techniques, and reviewing DNN models presented important insights. For effective DNN deployment on RCEs, the suggested approach of combining depthwise separable convolution, weight quantization, and pruning approaches shows promise. Furthermore, the recommendation to use EfficientNetB1 as a benchmark model for future research offers academics and practitioners a useful road map for creating efficient DNNs for image classification in resource-constrained settings. Furthermore, this study has identified a critical research gap concerning the absence of optimization models precisely tailored for DNNs in RCEs for image classification. Despite the existence of optimization models, they fall short in adequately addressing the unique constraints and limitations inherent in RCEs, including restricted memory and processing power. Consequently, the development of optimization models becomes imperative, ones that can effectively balance accuracy, computational complexity, and memory utilization to facilitate high-quality image classification on RCEs.

Moving forward, future optimization models should specifically target these identified performance objectives. The aim would be to achieve a minimum accuracy threshold of 80%, all while keeping memory utilization below 16 megabytes. These objectives align precisely with the specific constraints prevalent in RCEs. Addressing this research gap not only contributes to the advancement of optimization techniques but also ensures the practical applicability of DNNs in real-world scenarios characterized by resource constraints. The next step in this research involves the development of a DNN model that incorporates the four identified techniques(depthwise separable convolution, weight quantization, and pruning) into the EfficientNetB1 architecture. This novel model is strategically designed to strike a balance between accuracy and efficiency, with a specific focus on addressing the challenges posed by RCEs. The effectiveness of the proposed model will be assessed through a comprehensive evaluation, benchmarking its performance against existing models such shown in Table 4. This step aims not only to advance the field of DNN optimization but also to provide a practical and efficient solution tailored for image classification in RCEs.

REFERENCES

[1] A. Goel, C. Tung, Y.-H. Lu, and G. K. Thiruvathukal, "A survey of methods for low-power deep learning and computer vision," in *2020 IEEE 6th World Forum on Internet of Things (WF-IoT)*, 2020, pp. 1-6, doi: 10.1109/wf-iot48130.2020.9221198.

[2] Y. Li, "Research and application of deep learning in image recognition," in *2022 IEEE 2nd International Conference on Power, Electronics and Computer Applications (ICPECA)*, 2022, pp. 994-999, doi: 10.1109/icpeca53709.2022.9718847.

[3] H. Liu, D. Wang, K. Xu, P. Zhou, and D. Zhou, "Lightweight convolutional neural network for counting densely piled steel bars," *Automation in Construction*, vol. 146, p. 104692, 2023, doi: 10.1016/j.autcon.2022.104692.

[4] Y. Chen, J. Bin, and C. Kang, "Application of machine vision and convolutional neural networks in discriminating tobacco leaf maturity on mobile devices," *Smart Agricultural Technology*, p. 100322, 2023, doi: 10.1016/j.atech.2023.100322.

[5] I. Martinez-Alpiste, G. Golcarenarenji, Q. Wang, and J. M. Alcaraz-Calero, "Smartphone-based real-time object recognition architecture for portable and constrained systems," *Journal of Real-Time Image Processing*, vol. 19, pp. 103-115, 2022, doi: 10.1007/s11554-021-01164-1.

[6] G. Li, X. Ma, Q. Yu, L. Liu, H. Liu, and X. Wang, "CoAxNN: optimizing on-device deep learning with conditional approximate neural networks," *Journal of Systems Architecture*, vol. 143, p. 102978, 2023, doi: 10.1016/j.sysarc.2023.102978.

[7] T. Lawrence and L. Zhang, "IoTNet: an efficient and accurate convolutional neural network for IoT devices," *Sensors*, vol. 19, p. 5541, 2019, doi: 10.3390/s19245541.

[8] A. Turner, "August 2023 mobile user statistics: discover the number of phones in the world and smartphone penetration by country or region," *Bankmycell*, *https://www.bankmycell.com/blog/how-many-phones-are-in-the-world*, 2023.

[9] V. Kamath and A. Renuka, "Deep learning based object detection for resource constrained devices-systematic review, future trends and challenges ahead," *Neurocomputing*, 2023, doi: 10.1016/j.neucom.2023.02.006.

[10] A. Ignatov *et al.*, "Ai benchmark: running deep neural networks on android smartphones," in *Computer Vision – ECCV 2018 Workshops*, 2019, pp. 288–314, doi: 10.1007/978-3-030-11021-5_19.

[11] A. Ignatov *et al.*, "Ai benchmark: all about deep learning on smartphones in 2019," in *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, 2019, pp. 3617-3635, doi: 10.1007/978-3-030-11021-5_19.

[12] H. Nguyen, "Real-time vehicle and pedestrian detection on embedded platforms," *Journal of Theoretical and Applied Information Technology*, vol. 98, pp. 3405-3415, 2020.

[13] R. K. Bedi, J. Singh, and S. K. Gupta, "Analysis of multi cloud storage applications for resource constrained mobile devices," *Perspectives in Science*, vol. 8, pp. 279-282, 2016, doi: 10.1016/j.pisc.2016.04.052.

[14] S. Mazhar, N. Atif, M. Bhuyan, and S. R. Ahamed, "Block attention network: a lightweight deep network for real-time semantic segmentation of road scenes in resource-constrained devices," *Engineering Applications of Artificial Intelligence*, vol. 126, p. 107086, 2023, doi: 10.1016/j.engappai.2023.107086.

[15] J. Park, P. Aryal, S. R. Mandumula, and R. P. Asolkar, “An optimized DNN model for real-time inferencing on an embedded device,” *Sensors*, vol. 23, p. 3992, 2023, doi: 10.3390/s23083992.

[16] S. Han, J. Pool, J. Tran, and W. Dally, “Learning both weights and connections for efficient neural network,” *Advances in neural information processing systems,* vol. 28, 2015.

[17] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” *Communications of the ACM,* vol. 60, pp. 84-90, 2017, doi: 10.1145/3065386.

[18] J. R. Leow, W. H. Khoh, Y. H. Pang, and H. Y. Yap, “Breast cancer classification with histopathological image based on machine learning,” *International Journal of Electrical & Computer Engineering (2088-8708),* vol. 13, 2023, doi: 10.11591/ijece.v13i5.

[19] H. A. Al-Jubouri and S. M. Mahmmod, “A comparative analysis of automatic deep neural networks for image retrieval,” *TELKOMNIKA (Telecommunication Computing Electronics and Control),* vol. 19, pp. 858-871, 2021, doi: 10.12928/telkomnika.v19i3.18157.

[20] A. R. Luaibi, T. M. Salman, and A. H. Miry, “Detection of citrus leaf diseases using a deep learning technique,” *International Journal of Electrical and Computer Engineering (IJECE),* vol. 11, p. 1719, 2021, doi: 10.11591/ijece.v11i2.pp1719-1727.

[21] H. Li, A. Kadav, I. Durdanovic, H. Samet, and H. P. Graf, “Pruning filters for efficient convnets,” *arXiv preprint arXiv: 1608.08710,* 2016.

[22] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *arXiv preprint arXiv: 1409.1556,* 2014.

[23] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770-778, doi: 10.1109/cvpr.2016.90.

[24] S. Divya, B. Adepu, and P. Kamakshi, “Image enhancement and classification of CIFAR-10 using convolutional neural networks,” in *2022 4th International Conference on Smart Systems and Inventive Technology (ICSSIT)*, 2022, pp. 1-7, doi: 10.1109/icssit53264.2022.9716555.

[25] J. Yang *et al.*, “Quantization networks,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 7308-7316, doi: 10.1109/cvpr.2019.00748.

[26] J. H. Cho and B. Hariharan, “On the efficacy of knowledge distillation,” in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 4794-4802, doi: 10.1109/iccv.2019.00489.

[27] J. Gou, B. Yu, S. J. Maybank, and D. Tao, “Knowledge distillation: a survey,” *International Journal of Computer Vision,* vol. 129, pp. 1789-1819, 2021, doi: 10.1007/s11263-021-01453-z.

[28] S. Fu, Z. Li, Z. Liu, and X. Yang, “Interactive knowledge distillation for image classification,” *Neurocomputing,* vol. 449, pp. 411-421, 2021, doi: 10.1016/j.neucom.2021.04.026.

[29] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, “Densely connected convolutional networks,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4700-4708, doi: 10.1109/cvpr.2017.243.

[30] W. Wang, Y. Li, T. Zou, X. Wang, J. You, and Y. Luo, “A novel image classification approach via dense-MobileNet models,” *Mobile Information Systems,* vol. 2020, 2020, doi: 10.1155/2020/7602384.

[31] F. Chollet, “Xception: Deep learning with depthwise separable convolutions,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1251-1258, doi: 10.1109/cvpr.2017.195.

[32] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, “Mobilenetv2: Inverted residuals and linear bottlenecks,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 4510-4520, doi: 10.1109/cvpr.2018.00474.

[33] A. G. Howard *et al.*, “Mobilenets: efficient convolutional neural networks for mobile vision applications,” *arXiv preprint arXiv:1704.04861*, 2017.

[34] M. Tan and Q. Le, “Efficientnet: rethinking model scaling for convolutional neural networks,” in *International conference on machine learning*, 2019, pp. 6105-6114.

[35] Keras, “Keras applications,” *https://keras.io/api/applications/,* 2023.

[36] A. Howard *et al.*, “Searching for mobilenetv3,” in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 1314-1324, doi: 10.1109/iccv.2019.00140.

[37] Q. Xiang, X. Wang, R. Li, G. Zhang, J. Lai, and Q. Hu, “Fruit image classification based on Mobilenetv2 with transfer learning technique,” in *Proceedings of the 3rd international conference on computer science and application engineering*, 2019, pp. 1-7, doi: 10.1145/3331453.3361658.

[38] M. Kim, Y. Kwon, J. Kim, and Y. Kim, “Image classification of parcel boxes under the underground logistics system using CNN MobileNet,” *Applied Sciences,* vol. 12, p. 3337, 2022, doi: 10.3390/app12073337.

[39] L. Zhao and L. Wang, “A new lightweight network based on MobileNetV3,” *KSII Transactions on Internet & Information Systems,* vol. 16, 2022, doi: 10.3837/tiis.2022.01.001.

[40] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, “Inception-v4, inception-resnet and the impact of residual connections on learning,” in *Thirty-first AAAI conference on artificial intelligence*, 2017, doi: 10.1609/aaai.v31i1.11231.

[41] L. Dang, P. Pang, and J. Lee, “Depth-wise separable convolution neural network with residual connection for hyperspectral image classification,” *Remote Sensing,* vol. 12, p. 3408, 2020, doi: 10.3390/rs12203408.

[42] M. Shafiq and Z. Gu, “Deep residual learning for image recognition: a survey,” *Applied Sciences,* vol. 12, p. 8972, 2022, doi: 10.3390/app12188972.

[43] A. R. Ajel, A. Q. Al-Dujaili, Z. G. Hadi, and A. J. Humaidi, “Skin cancer classifier based on convolution residual neural network,” *International Journal of Electrical & Computer Engineering (IJECE),* vol. 13, 2023, doi: 10.11591/ijece.v13i6.pp6240-6248.

[44] F. Martínez, C. Hernández, and F. Martínez, “Evaluation of deep neural network architectures in the identification of bone fissures,” *TELKOMNIKA (Telecommunication Computing Electronics and Control),* vol. 18, pp. 807-814, 2020, doi: 10.12928/telkomnika.v18i2.14754.

[45] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, “Rethinking the inception architecture for computer vision,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2818-2826, doi: 10.1109/cvpr.2016.308.

[46] X. Wan, F. Ren, and D. Yong, “Using inception-Resnet v2 for face-based age recognition in scenic spots,” in *2019 IEEE 6th International Conference on Cloud Computing and Intelligence Systems (CCIS)*, 2019, pp. 159-163, doi: 10.1109/ccis48116.2019.9073696.

[47] S. A. Albelwi, “Deep architecture based on DenseNet-121 model for weather image recognition,” *International Journal of Advanced Computer Science and Applications,* vol. 13, pp. 559-565, 2022, doi: 10.14569/ijacsa.2022.0131065.

[48] S. Benkrama and N. E. H. Hemdani, "Deep learning with EfficientNetB1 for detecting brain tumors in MRI images," in *2023 International Conference on Advances in Electronics, Control and Communication Systems (ICAECCS)*, 2023, pp. 1-6, doi: 10.1109/icaeccs56710.2023.10104761.

[49] P. Molchanov, A. Mallya, S. Tyree, I. Frosio, and J. Kautz, "Importance estimation for neural network pruning," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 11264-11272, doi: 10.1109/cvpr.2019.01152.

[50] L. Chen, Y. Chen, J. Xi, and X. Le, "Knowledge from the original network: restore a better pruned network with knowledge distillation," *Complex and Intelligent Systems,* pp. 1-10, 2021, doi: 10.1007/s40747-020-00248-y.

[51] R. Yazdani, M. Riera, J.-M. Arnau, and A. González, "The dark side of DNN pruning," in *2018 ACM/IEEE 45th Annual International Symposium on Computer Architecture (ISCA)*, 2018, pp. 790-801, doi: 10.1109/isca.2018.00071.

[52] X. Wang, Y. Qin, Y. Wang, S. Xiang, and H. Chen, "ReLTanh: an activation function with vanishing gradient resistance for SAE-based DNNs and its application to rotating machinery fault diagnosis," *Neurocomputing,* vol. 363, pp. 88-98, 2019, doi:10.1016/j.neucom.2019.07.017.

[53] L. Alzubaidi *et al.*, "Review of deep learning: concepts, CNN architectures, challenges, applications, future directions," *Journal of big Data,* vol. 8, pp. 1-74, 2021, doi: 10.1186/s40537-021-00444-8.

[54] H. Qassim, A. Verma, and D. Feinzimer, "Compressed residual-VGG16 CNN model for big data places image recognition," in *2018 IEEE 8th annual computing and communication workshop and conference (CCWC)*, 2018, pp. 169-175, doi: 10.1109/ccwc.2018.8301729.

[55] R. H. Bedeir, R. O. Mahmoud, and H. H. Zayed, "Automated multi-class skin cancer classification through concatenated deep learning models," *IAES International Journal of Artificial Intelligence (IJ-AI),* vol. 11, p. 764, 2022, doi: 10.11591/ijai.v11.i2.pp764-772.

[56] K. Wu, Y. Guo, and C. Zhang, "Compressing deep neural networks with sparse matrix factorization," *IEEE transactions on neural networks and learning systems,* vol. 31, pp. 3828-3838, 2019, doi: 10.1109/tnnls.2019.2946636.

[57] S. Swaminathan, D. Garg, R. Kannan, and F. Andres, "Sparse low rank factorization for deep neural network compression," *Neurocomputing,* vol. 398, pp. 185-196, 2020, doi: 10.1016/j.neucom.2020.02.035.

[58] P. Ramachandran, B. Zoph, and Q. V. Le, "Searching for activation functions," *arXiv preprint arXiv:1710.05941,* 2017.

[59] F. MartEnez, H. Montiel, and F. Martínez, "Comparative study of optimization algorithms on convolutional network for autonomous driving," *International Journal of Electrical & Computer Engineering (IJECE),* vol. 12, 2022, doi: 10.11591/ijece.v12i6.pp6363-6372.

[60] H. Imaduddin, F. Y. Ala, A. Fatmawati, and B. A. Hermansyah, "Comparison of transfer learning method for COVID-19 detection using convolution neural network," *Bulletin of Electrical Engineering and Informatics (BEEI),* vol. 11, pp. 1091-1099, 2022, doi: 10.11591/eei.v11i2.3525.

[61] A. Taslim, S. Saon, M. Muladi, and W. N. Hidayat, "Plant leaf identification system using convolutional neural network," *Bulletin of Electrical Engineering and Informatics (BEEI),* vol. 10, pp. 3341-3352, 2021, doi: 10.11591/eei.v10i6.2332.

[62] M. A. I. Aquil and W. H. W. Ishak, "Evaluation of scratch and pre-trained convolutional neural networks for the classification of Tomato plant diseases," *IAES International Journal of Artificial Intelligence (IJ-AI),* vol. 10, p. 467, 2021, doi: 10.11591/ijai.v10.i2.

[63] S. Suprayitno, W. A. Fauzi, K. Ain, and M. Yasin, "Real-time military person detection and classification system using deep metric learning with electrostatic loss," *Bulletin of Electrical Engineering and Informatics (BEEI),* vol. 12, pp. 338-354, 2023, doi: 10.11591/eei.v12i1.4284.

[64] K. Okokpujie, E. Noma-Osaghae, S. N. John, C. Ndujiuba, and I. P. Okokpujie, "Comparative analysis of augmented datasets performances of age invariant face recognition models," *Bulletin of Electrical Engineering and Informatics (BEEI),* vol. 10, pp. 1356-1367, 2021, doi: 10.11591/eei.v10i3.3020.

[65] A. Polino, R. Pascanu, and D. Alistarh, "Model compression via distillation and quantization," *arXiv preprint: 1802.05668,* 2018.

[66] T. Liang, J. Glossner, L. Wang, S. Shi, and X. Zhang, "Pruning and quantization for deep neural network acceleration: a survey," *Neurocomputing,* vol. 461, pp. 370-403, 2021, doi: 10.1016/j.neucom.2021.07.045.

[67] J. Yim, D. Joo, J. Bae, and J. Kim, "A gift from knowledge distillation: Fast optimization, network minimization and transfer learning," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4133-4141, doi: 10.1109/cvpr.2017.754.

[68] A. Alkhulaifi, F. Alsahli, and I. Ahmad, "Knowledge distillation in deep learning and its applications," *PeerJ Computer Science,* vol. 7, p. e474, 2021, doi: 10.7717/peerj-cs.474.

[69] S. M. Mahmoud, H. A. Al-Jubouri, and T. E. Abdoulabbas, "Chest radiographs images retrieval using deep learning networks," *Bulletin of Electrical Engineering and Informatics (BEEI),* vol. 11, pp. 1358-1369, 2022, doi: 10.11591/eei.v11i3.3478.

[70] I. Nasri, M. Karrouchi, H. Snoussi, K. Kassmi, and A. Messaoudi, "DistractNet: a deep convolutional neural network architecture for distracted driver classification," *IAES International Journal of Artificial Intelligence (IJ-AI),* vol. 11, p. 494, 2022, doi: 10.11591/ijai.v11.i2.

[71] P. Varghese and A. S. Saroja, "Biologically inspired deep residual networks," *IAES International Journal of Artificial Intelligence (IJ-AI),* vol. 12, pp. 1873-1882, 2023, doi: 10.11591/ijai.v12.i4.

[72] R. M. Jasim and T. S. Atia, "Towards classification of images by using block-based CNN," *Bulletin of Electrical Engineering and Informatics (BEEI)*, vol. 12, pp. 373-379, 2023, doi: 10.11591/eei.v12i1.4806.

[73] N. M. Al-Moosawi and R. S. Khudeyer, "ResNet-n/DR: automated diagnosis of diabetic retinopathy using a residual neural network," *TELKOMNIKA (Telecommunication Computing Electronics and Control),* vol. 21, pp. 1051-1059, 2023, doi: 10.12928/telkomnika.v21i5.24515.

[74] R. Y. Patil, S. Gulvani, V. B. Waghmare, and I. K. Mujawar, "Image based anthracnose and red-rust leaf disease detection using deep learning," *TELKOMNIKA (Telecommunication Computing Electronics and Control),* vol. 20, pp. 1256-1263, 2022, doi: 10.12928/telkomnika.v20i6.24262.

[75] R. A. Pratiwi, S. Nurmaini, D. P. Rini, M. N. Rachmatullah, and A. Darmawahyuni, "Deep ensemble learning for skin lesions classification with convolutional neural network," *IAES International Journal of Artificial Intelligence (IJ-AI),* vol. 10, p. 563, 2021, doi: 10.11591/ijai.v10.i3.

[76] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: a large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*, 2009, pp. 248-255, doi: 10.1109/cvpr.2009.5206848.

BIOGRAPHIES OF AUTHORS

Raafi Careem is a Ph.D. scholar in Computer Science at School of Graduate Studies, Management and Science University, Malaysia. He earned his M.Sc. in Computer Science from the University of Peradeniya, Sri Lanka and holds B.Sc. (Hons.) degree in Computer Science from the South Eastern University of Sri Lanka. He is an Associate Fellow of the Higher Education Academy (AFHEA) received from Auckland University of Technology, New Zealand. He is currently affiliated with the Department of Computer Science & Informatics, Uva Wellassa University, Sri Lanka. His research interests are deep neural network, machine learning, artificial intelligence, intelligent system, image classification, and android application development. He can be contacted at email: mraafi@gmail.com.

Md Gapar Md Johar is Senior Vice President System, Technology and Innovation of Management and Science University, Malaysia. He is a professor in Software Engineering. He holds Ph.D. in Computer Science, M.Sc. in Data Engineering and B.Sc. (Hons) in Computer Science and Certified E-Commerce Consultant. He has more than 40 years of working and teaching experience in various organizations include Ministry of Finance, Ministry of Public Enterprise, Public Service Department, Glaxo Malaysia Sdn Bhd and Cosmopoint Institute of Technology. His research interests include learning content management system, knowledge management system, blended assessment system, data mining, RFID, e-commerce, image processing, character recognition, data analytics, artificial intelligent, and healthcare management system. He can be contacted at email: mdgapar@msu.edu.my.

Ali Khatibi is Senior Vice President and a professor at the School of Graduate Studies, Management and Science University (MSU), Malaysia, with a career spanning 41 years in academia and industry. Throughout his tenure, he has held numerous senior academic and administrative positions at MSU, contributing significantly to research, teaching, and administration. As a Professor of Marketing, he has been honored as a Senior Research Fellow, receiving both Gold and Silver Medals for his contributions to invention and innovation research. With over 400 publications, more than 5,000 citations, and supervision of over 150 Master's and Ph.D. candidates, he has made a substantial impact in academia. Additionally, he has served as Editor-in-Chief and authored several books. He can be contacted at email: alik@msu.edu.my.