International Journal of Computing and Digital Systems
ISSN (2210-142X)
Int. J. Com. Dig. Sys. 16 , No.1 (Sep-2024)

http://dx.doi.org/10.12785/ijcds/160184

Baseline Model for Deep Neural Networks in Resource-Constrained Environments: An Empirical Investigation

Raafi Careem [1], Md Gapar Md Johar [2] and Ali Khatibi [3]

[1]*Department of Computer Science & Informatics, Uva Wellassa University, 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*

Received 6 Mar. 2024, Revised 20 May 2024, Accepted 21 May 2024, Published 1 Sep. 2024

Abstract: This paper presents an empirical study on advanced Deep Neural Network (DNN) models, with a focus on identifying potential baseline models for efficient deployment in resource-constrained environments (RCE). The systematic evaluation encompasses ten state-of-the-art pre-trained DNN models: ResNet50, InceptionResNetV2, InceptionV3, MobileNet, MobileNetV2, EfficientNetB0, EfficientNetB1, EfficientNetB2, DenseNet121, and Xception, within the context of an RCE setting. Evaluation criteria, such as parameters (indicating model complexity), storage space (reflecting storage requirements), CPU usage time (for real-time applications), and accuracy (reflecting prediction truth), are considered through systematic experimental procedures. The results highlight MobileNet's excellent trade-off between accuracy and resource requirements, especially in terms of CPU and storage consumption, in experimental scenarios where image predictions are performed on an RCE device. Utilizing the identified baseline model, a new model, GRM-MobileNet, was developed by implementing compound scaling and global average pooling techniques. GRM-MobileNet exhibits a substantial reduction of 23.81% in parameters compared to MobileNet, leading to a model size that is 23.88% smaller. Moreover, GRM-MobileNet demonstrates a significant improvement in accuracy, achieving a remarkable gain of 28.12% over MobileNet. Although the enhancement in inference time for GRM-MobileNet compared to MobileNet is modest at 1.66%, the overall improvements underscore the effectiveness of the employed strategies in enhancing the model's performance. A future study will examine other model optimization strategies, including factorization and pruning, which ultimately lead to faster inference without compromising accuracy, in an effort to improve the efficiency of the GRM-MobileNet model and its inference time.

Keywords: Baseline model, Deep neural network, Image classification, Optimization model, RCE

1. INTRODUCTION

Deep neural networks (DNNs) have gained widespread adoption across diverse domains, showcasing superior performance in applications such as autonomous vehicles [1], [2], healthcare [3], [4], [5], agriculture [6], [7], [8], security [9], [10], and sports [11]. Particularly notable is their proficiency in image classification within the computer vision discipline [8], [12], [13], [14]. However, deploying DNN models on devices, while promising higher accuracy, introduces challenges related to resource requirements, specifically in terms of memory and CPU utilization [15], [16]. These challenges become particularly pronounced when implementing DNN models in devices with limited resources, often denoted as resource-constrained environments (RCE).

RCEs are computing environments or systems with limited resources, such as memory, processor power, storage capacity, and energy sources. These environments are characterized by their inability to perform complex computations or handle large amounts of data, as opposed to more durable computing configurations such as desktop computers or servers. RCEs are frequently seen in real-time applications, especially in devices like mobile and Internet of Things (IoT) products, which usually have constrained hardware. A list of common RCE devices operating in this resource-constrained setting is given in the article [17], which is given below in Table I, where CPU frequencies range from 1.2 GHz to 2.84 GHz and RAM capacities range from 1 GB to 8 GB.

The significance of adapting DNNs for use in RCEs is underscored by the rapid growth of the IoT and the

E-mail address: mraafi@gmail.com, mdgapar@msu.edu.my, alik@msu.edu.my

increasing demand for mobile devices [7], [18], [19], [20]. Deploying DNNs on RCE devices, given their limitations, necessitates carefully considering a number of issues, such as increased model size and computational complexity [21], [22]. Furthermore, optimization strategies are essential to minimize model size and resource consumption without sacri cing accuracy, addressing the inherent constraints of RCE devices.

The rising popularity of RCEs can be attributed to the IoT and the widespread use of smart devices, leading to an increased demand for the integration of DNNs in these settings [22], [23], [24]. Onboard implementation, the direct deployment of DNNs on RCE devices, has several bene ts, including real-time image classi cation, reduced latency, lower bandwidth consumption, and strengthened privacy, as well as security measures [2]. Rapid data processing made possible by onboard DNNs in RCEs speeds up decision-making in a variety of elds, such as autonomous cars, smart homes, transportation, healthcare, and agriculture. As such, it is now more important than ever to deploy DNNs on RCEs in an e cient and e ective manner.

This article's main goal is to employ a well-de ned experimental technique to systematically evaluate benchmark pre-trained DNN models in an RCE scenario. Based on the discovered DNN model, the article will then design a new e ciency-focused DNN model that is speci cally tted to these settings. The outcomes of this study will function as a guide for the creation of productive image classi cation applications, promising top performance in real-world scenarios where resource constraints are frequently present.

The reminder of this article is structured as follows: In Section 2, related literature in this eld is reviewed. In Section 3, the employed methodology—which includes evaluation metrics and datasets as well as the experimental process for determining the baseline model and creating the new DNN model (GRM-MobileNet) are described. In Section 4, the created GRM-MobileNet is assessed for e cacy and performance by comparing its results with those of benchmark DNN models. In the last section, Section 5 summarizes the article's ndings, acknowledges its limitations, and outlines directions for future works.

2. related works

A number of studies have looked into methods for developing lightweight models alongside their very deep counterparts in light of the diculties in implementing DNN models in RCE, especially given their size and processing requirements [25], [26]. This simpli cation of very deep models is achieved through optimization and compression techniques [26]. For example, depthwise separable convolution methods in the optimization paradigm utilized by Chollet [27], namely depthwise convolution as well as pointwise convolution to use less computer power to train and run larger complex models. However, it is important to recognize that using depthwise convolution techniques in DNN models results in lower prediction accuracy when the model is being inferred [28]. In order to improve accuracy, Tan and Le [29] proposed the compound scaling technique, which simultaneously increases a neural network's depth, width, and resolution. This is another important tactic. Although using compound scaling has the potential to increase accuracy, there are additional needs in terms of memory utilization, CPU usage, and computational resources. In compression techniques, removing unimportant weights and links from a DNN is called pruning [30], [31] to reduce the size of networks and lower inference costs for DNN models. Pruning the DNN model has advantages, but it can also increase complexity and cause accuracy loss when training a model. An alternative method involves quantizing the network, which involves reducing the number of bits in oating-point values that indicate activations and weights. To improve image classi cation accuracy, Yang et al. [32] used activation quantization and weights. However, using fewer bits for weights could result in a loss of accuracy, which would a ect the accuracy of neural networks. Knowledge distillation is another method, as used in [33], [34], which is moving knowledge from a large, complicated DNN (teacher network) model to a smaller, more straightforward DNN (student network). Although distillation has increased accuracy [35], there is a chance that information will be lost in the transfer, and training the huge model will cost in terms of computation. An alternative approach is applying transfer learning (TL) [2], [36], which utilizes model weights from previously trained models. By using feature representations that a pre-trained model has learned, TL eliminates the requirement to train an entirely new model from scratch. This results in decreased training time and a reduction in generalization error when pre-trained models are incorporated into a new model [2]. In the TL method, the weights of previously trained models can be used to initial the weights for the new model, facilitating a more e ective training process. Consequently, the TL approach deliberately employs a relevant pre-trained DNN model as a fundamental starting point to build a unique model that is suited to the particular needs of a particular application.

Currently, several pre-trained DNN models exist that have the potential as baseline models to develop new models utilizing the TL approach, such as MobileNet [28], [37], MobileNetV2 [38], [39], E cientNetB0, E cientNetB1, E cientNetB2 [29], [40] DenseNet121 [41], Xception [42], InceptionV3 [43], ResNet50 [44], and InceptionResNetV2 [45]. Each of these models o ers its own set of advantages and limitations.

Identifying the most suitable pre-trained model to serve as a baseline for developing an e ciency-focused model in RCEs is crucial. Recent studies, such as Mana et al. [37], conducted an empirical study using chest X-ray images to identify the best model among eight benchmark pre-trained models: VGG16, ResNet50, DenseNet121, Xception, ResNet152V2, E cientNet, DenseNet201, and MobileNet. Similarly, another study [46] compared lightweight DNNs, including DenseNet121, E cientNet, MobileNetV2,

TABLE I. Typical resource-constraint environment with their speci cations depicted from [17]

Device Name	CPU Model	CPU Power (GHz)	Memory (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

MobileNetV3, Shu eNetV2, etc., for tomato leaf disease identi cation tasks. Both studies evaluated the models using accuracy, precision, recall, and F1-score. However, they primarily focused on accuracy for comparing models' performances, overlooking important elements like the number of parameters, model size, and inference time, which are critical for deploying models in RCE scenarios.

A notable research gap exists, as there is a lack of experimental studies evaluating these models within the context of RCE scenarios. This article attempts to ll this gap by conducting a systematic evaluation of the potential pre-trained DNN models in an RCE context, utilizing a well-de ned experimental methodology. It also aims to create a novel DNN model that is e cient and speci cally intended for use in these situations.

3. research method

The research methodology of this paper comprises two phases. The rst phase focuses on identifying the most suitable baseline model for developing an e ciency-focused DNN model tailored for RCE settings. This phase involves evaluating various existing models to determine their suitability based on factors such as performance and computational e ciency (evaluation metrics) to resource limitations.

In the second phase, once the baseline model is identi ed, the research progresses to design and develop the new DNN model. This phase entails leveraging the chosen baseline model as a foundation and implementing a DNN model to enhance its e ciency for RCE scenarios. The development process includes incorporating techniques such as compound scaling and global average pooling to improve the model's performance while ensuring its compatibility with RCE settings. Additionally, the newly developed model undergoes validation to assess its e ectiveness and performance against the given evaluation metrics, thereby ensuring its suitability for real-world deployment in RCE scenarios.

A. Evaluation metrics

The assessment matrices covered in this paper are essential resources for understanding the complex aspects of DNN model performance in resource-constrained settings. These matrices include important elements such as parameters that indicate the complexity of the models, storage space that indicates the amount of storage needed, CPU utilization time for real-time applications, and accuracy that measures how accurate the models are [17], [20], [22], [26], [47], [48]. Every criterion is carefully investigated using methodical experimental techniques, giving a comprehensive view of the strengths and trades displayed by several pre-trained DNN models.

1) Parameters

In a DNN, parameters are the weights and biases that are learned by the model during training. They establish how the model is put together and how it converts input data into predictions. In general, models with higher parameter counts are more complex. Gaining knowledge of parameter numbers helps one understand how sophisticated the model architecture is and how much computing it requires. Models with fewer parameters may be favored in contexts with limited resources since they need less computing power.

2) Storage space

The memory needed to hold the complete DNN model—including its architecture, parameters, and any extra data—is referred to as storage space. Models with lower storage footprints are required in resource-constrained contexts due to limited storage capacity. Determining the e cacy of implementing a model in settings with limited memory resources requires analyzing storage requirements.

3) CPU Usage Time

CPU use time is important for applications that need real-time responsiveness since it shows how long a DNN model needs to analyze and predict an input. Models that require real-time decision-making and have shorter CPU usage durations are favored in cases where resources are

limited. Analyzing this criterion provides light on how effective the model is in real-world, time-sensitive situations.

4) Accuracy

A DNN model's accuracy is a performance metric that assesses how accurate its predictions are. It shows the proportion of accurately anticipated cases to all instances. One key measure of a model's ability to correctly classify input data is its accuracy. Higher accuracy in image classification duties indicates that the model can generate accurate predictions. While accuracy is important, it must be weighed against other factors in order to balance resource efficiency with predictive performance.

B. Datasets

This study employed two sets of images from the larger and widely recognized ImageNet1K dataset [49]. First, the research used a selection of 10 randomly selected images from the ImageNet1K testing dataset to conduct the empirical investigation to determine the best-performing baseline model. To provide a thorough assessment of the models' performance, these pictures depict a variety of items. The selected images encompass a variety of categories, specifically: goldfish, centipede, Persian cat, zebra, ambulance, analog clock, balloon, car wheel, bell pepper, and cup. Each of these images is shown in Figure 1a.

This diverse selection of images is designed to test the models' ability to accurately classify a wide range of objects, providing insights into their generalization capabilities. By including images from different categories, the study aims to simulate real-world conditions where models must identify and classify various objects accurately. The choice of these particular categories ensures that the empirical study covers a broad spectrum of object types, from animals and vehicles to everyday items, thus making the evaluation robust and comprehensive.

In this work, a new dataset called ImageNet10 was constructed in order to shorten the training time and enable the quick construction and validation of new models. The ImageNet10 dataset is a carefully created as a subset of the ImageNet1K dataset. Images from ten different categories, each depicting a range of items, are included in this subset. As seen in Figure 2, the categories covered include balloon, goldfish, panda, parrot, shark, cock, dog, lion, horse, and airplane.

The ImageNet10 dataset consists of 11,000 images in total. The dataset is split into three sections to enable effective model training and evaluation: 7,700 photos, or 70% of the total, are set aside for training, 1,650 (15%) images for validation, and 1,650 (15%) images for testing. This distribution ensures a comprehensive approach to model training and performance evaluation, allowing for fine-tuning and validation during the development process, as well as an unbiased assessment of the model's generalization capabilities on unseen data. By employing this dataset, the study aims to streamline the development cycle and enhance the efficiency of the model training process.

C. Experimental Procedure

1) Baseline Model

An empirical analysis of the DNN models was conducted through an experiment involving the identified models to assess their performance through suitable evaluation metrics. The objective was to determine the most suitable baseline model for deploying DNNs in an RCE scenario. The experimental environment was implemented using Python 3.8.18, Tensor Flow 2.3.0, NumPy 1.18.5, Matplotlib 3.4.3, and Pandas 1.2.4 within the Keras 2.4.0 framework. The RCE environment used in the experiment contained an Intel 1.86 GHz X4 central processing unit (CPU) and 4 GB of random-access memory (RAM).

The experimental method, depicted in Figure 3, followed a systematic procedure. Initially, ten DNN models were sequentially deployed into a predefined RCE scenario. Each of the ten images selected from 1(a) was individually presented to every deployed model, as illustrated in Figure 3. Subsequent to the image input, predictions generated by each model were observed and the recorded for both accuracy and inference time (Figure 1(b)). This process, from deployment to observation, was repeated for each of the ten DNN models, ensuring a consistent evaluation across the identical set of images. Following the experimental phase, recorded accuracy and prediction time data were methodically organized into Tables II and III respectively. While Table II presented accuracy values, Table III outlined inference times for each model across all images. These recorded accuracy values and inference times were then used to compute mean accuracy (MAcc) and mean inference time (MTime), respectively.

MAcc provided an average measure of prediction accuracy for each of the ten models, as depicted in Table II. Simultaneously, mean time represented an average measure of prediction time for each model, as detailed in Table III. In Tables II, MAcc for each DNN model is calculated as the simple average of recorded accuracy values across all images ($N=10$), using the formula (1) [50], [51]:

$$MAcc = \frac{1}{N}\sum_{i=1}^{N} Acc_i \quad (1)$$

where, N is total number of images and Acc_i represents the accuracy value for the ith image. Similarly, in Table III, MTime is determined as the simple average of recorded time values for each model across all images ($N=10$) predictions, using the formula (2) [49]:

$$MTime = \frac{1}{N}\sum_{i=1}^{N} Time_i \quad (2)$$

where, N is the total number of images and $Time_i$ represents the time value for the ith image.

Figure 1. Set of images used for the experiment: (a) Images randomly selected from the ImageNet1K testing dataset; (b) Examples of predicted images from a model, with predicted labels displayed above each image along with their corresponding accuracy.

Figure 2. Images of ImageNet10 dataset

Figure 3. Overview of the experimental setup

The empirical analysis of the selected DNN models included two other crucial measures in addition to accuracy and inference time to provide a more thorough knowledge of their performance attributes. These metrics include the number of parameters in million (M), indicating the model's complexity and depth; and the size of the model in megabytes (MB), reflecting its storage requirements. Table VII presents the systematic documentation on observations pertaining to these many criteria, which adds significant value to the assessment as a whole. The interpretation of the data in this table is presented in detail in Section 4 of this article.

2) New Model (GRM- MobileNet)

The identified baseline model, MobileNet, which was chosen for its promising performance in resource-constrained contexts, served as the foundation for the development of the new model, called as GRM-MobileNet. Leveraging MobileNet as the base model, the design process incorporated RCE optimization techniques recommended by authors in [17], [26], [29], [40] namely compound scaling and global average pooling (GAP). These techniques aim to enhance the model's efficiency and adaptability to resource-constrained settings, as used in EfficientNetB1 [29]. By integrating these optimization strategies into the MobileNet architecture, GRM-MobileNet is tailored to meet the specific requirements of efficiency-focused image classification applications in RCE settings.

Compound scaling is an optimization technique that adjusts the network depth, width, and resolution of a DNN architecture to modify its size and complexity. Balancing the values of these factors can lead to enhanced classification efficiency [26], [29]. This technique involves varying the depth and resolution values within specific ranges, typically from 0.0 to 1.6, to find the optimal configuration for evaluation metrics such as parameters, model size, accuracy, and prediction time. Through empirical experimentation, researchers seek to identify the most suitable combination of depth and resolution values that maximize performance across these metrics. The findings from the experiments indicate that there exists an optimal model configuration with a width value of 1.0 and a resolution value of 1.2, demonstrating the effectiveness of compound scaling in optimizing DNN architectures for various performance criteria. The identified values were utilized to design the new model, GRM-MobileNet. A part of the architecture of the model is depicted in graphical format in Figure 4 and in tabulated format in Table IV.

The designed GRM-MobileNet underwent training and experimentation using the ImageNet10 dataset. The training process utilized an Intel i7-3770 CPU @ 3.5GHz (8 CPUs) with 12 GB of RAM, providing the computational resources necessary for model training and evaluation. Table V shows the classification report for the testing dataset, indicating a 73% prediction accuracy for the ImageNet10 dataset. The detailed interpretation of the table is presented in Section 4.

The trained GRM-MobileNet model underwent a comparative analysis with five benchmark models, namely MobileNet, MobileNetV2, EfficientNetB0, EfficientNetB1, and ResNet50, to evaluate its performance. Accordingly, these models, along with GRM-MobileNet, were deployed on an RCE device equipped with a CPU operating at 1.86 GHz and 4 GB of RAM. The experimentation involved testing the models using the ImageNet10 testing dataset. The given evaluation metrics, including parameter numbers, model size, accuracy, and inference time, were recorded for each model. The results of this comparative analysis are presented in Table VI. This table illustrates that the GRM-MobileNet offers the highest accuracy, lowest model size, and parameter numbers with moderated inference time.

4. RESULT AND DISCUSSION

This section presents the experimental results of this study. In Section A, the comparison results of benchmark DNN models are discussed to identify the most suitable benchmark model for developing a new efficiency-focused DNN model tailored for RCE settings. Section B focuses on evaluating the newly developed DNN model, GRM-MobileNet. This evaluation assesses the performance and effectiveness of GRM-MobileNet against predefined criteria, including metrics such as accuracy, model size, and inference time. The goal is to ensure that GRM-MobileNet

TABLE II. Recorded prediction accuracy of ten DNN models for ten images

DNN Models	Gold sh	Centipede	Cat	Zebra	Ambulance	Balloon	Wheel	Clock	BP	Cup	MAcc %
MobileNetV2	89.68	93.49	75.32	96.42	85.42	67.51	88.54	68.83	92.37	61.82	81.94
MobileNet	100.00	100.00	97.07	99.99	98.85	100.00	99.95	94.08	99.93	86.57	97.64
E cientNetB0	92.28	67.15	82.50	88.14	97.43	90.15	88.17	44.16	96.20	59.72	80.59
E cientNetB1	89.07	88.10	93.35	91.57	96.19	91.03	90.85	40.62	94.23	54.61	82.96
DenseNet121	98.04	100.00	93.43	99.94	99.48	99.52	94.32	48.42	99.95	55.41	88.85
E cientNetB2	84.21	82.49	89.88	88.73	89.77	91.31	84.40	54.62	89.19	47.48	80.21
Xception	86.94	90.00	90.68	85.64	98.02	80.80	91.82	49.78	88.06	57.08	81.88
InceptionV3	99.08	97.37	88.36	90.87	94.18	95.18	93.88	87.33	94.71	75.25	91.62
ResNet50	94.64	100.00	99.26	99.98	99.64	99.98	99.60	88.15	98.05	82.84	96.21
Inception-Res-NetV2	91.64	94.15	92.28	93.17	94.86	93.43	89.17	82.08	91.93	78.12	90.08

TABLE III. Recorded inference time (s) for the prediction of ten images by ten DNN models

DNN Models	Gold sh	Centipede	Cat	Zebra	Ambulance	Balloon	Wheel	Clock	BP	Cup	MTime (s)
MobileNetV2	5.75	5.16	5.17	5.36	5.96	6.00	5.79	5.92	5.59	5.51	5.62
MobileNet	3.70	4.07	3.71	3.91	3.55	3.64	4.01	4.38	3.95	3.84	3.88
E cientNetB0	9.27	8.49	8.06	8.50	7.84	8.44	9.72	8.25	9.56	7.94	8.61
E cientNetB1	13.18	14.26	14.06	14.29	16.00	14.59	13.71	14.69	13.20	14.06	14.20
DenseNet121	15.90	14.25	14.15	14.23	14.80	12.50	11.97	12.16	11.88	13.39	13.52
E cientNetB2	14.35	11.89	11.50	12.20	11.50	11.72	12.00	11.53	12.11	12.15	12.10
Xception	10.49	12.71	11.24	11.33	10.88	10.78	10.99	10.63	10.52	11.39	11.10
InceptionV3	11.75	10.92	10.81	11.10	11.10	12.51	11.88	11.39	12.10	11.33	11.49
ResNet50	8.73	7.82	8.18	7.75	8.82	7.98	8.00	7.98	7.84	7.75	8.09
Inception-Res-NetV2	26.43	26.72	27.17	27.00	26.88	27.11	25.80	26.08	26.32	26.41	26.59

TABLE IV. A part of GRM-MobileNet architecture

No.	Layer_name	Input_shape	Output_shape	Parammeter No.
1	input_1	[(None , 268, 268, 3)]	[(None , 268, 268, 3)]	0
2	conv1_pad	(None, 268, 268, 3)	(None, 269, 269, 3)	0
3	conv1	(None, 269, 269, 3)	(None, 134, 134, 32)	864
4	conv1_bn	(None, 134, 134, 32)	(None, 134, 134, 32)	128
5	conv1_relu	(None, 134, 134, 32)	(None, 134, 134, 32)	0
***	***	***	***	***
***	***	***	***	***
83	conv_dw_13_bn	(None, 8, 8, 1024)	(None, 8, 8, 1024)	4096
84	conv_dw_13_relu	(None, 8, 8, 1024)	(None, 8, 8, 1024)	0
85	conv_pw_13	(None, 8, 8, 1024)	(None, 8, 8, 1024)	1048576
86	conv_pw_13_bn	(None, 8, 8, 1024)	(None, 8, 8, 1024)	4096
87	conv_pw_13_relu	(None, 8, 8, 1024)	(None, 8, 8, 1024)	0
88	global_average_pooling2d	(None, 8, 8, 1024)	(None, 1024)	0
89	dense	(None, 1024)	(None, 10)	10250

Figure 4. A part of GRM-MobileNet architecture

TABLE V. Classi cation report of the GRM- MobileNet

Images	precision	recall	f1-score	support
aeroplane	0.92	0.5	0.65	165
balloon	0.65	0.53	0.58	165
cock	0.87	0.67	0.76	165
dog	0.74	0.76	0.75	165
gold sh	0.59	0.96	0.73	165
horse	0.61	0.88	0.72	165
lion	0.75	0.76	0.75	165
panda	0.69	0.92	0.79	165
parrot	0.91	0.59	0.71	165
shark	0.97	0.76	0.85	165
accuracy			0.73	1650
macro avg	0.77	0.73	0.73	1650
weighted avg	0.77	0.73	0.73	1650

TABLE VI. Comparison of GRM-MobileNet with benchmark DNN models with four evaluation matrices

DNN Models	Parameters (M)	Model Size (MB)	Accuracy (%)	Time for inference step (ms)
MobileNet	4.2	49.0	45.09	437.6
MobileNetV2	3.5	41.1	30.18	276.4
E cientNetB0	5.3	61.8	10.00	407.3
E cientNetB1	7.9	91.0	11.33	585.5
ResNet50	25.6	294.0	35.27	3097.6
GRM-MobileNet	3.2	37.3	73.21	430.3

meets the requirements for deployment in real-world RCE scenarios.

A. Baseline Model

The outcomes of the experimentation involving various DNN models are documented in Table VII, offering a thorough overview of their performance based on essential evaluation metrics. Figure 5 visually represents a comparative analysis of these DNN models. The metrics used for comparison encompass the number of parameters in millions (m), storage requirements in megabytes (MB), memory utilization during prediction, prediction time in seconds (s), and prediction accuracy. This comparative approach allows for the extraction of valuable insights into the distinctive characteristics of each model.

MobileNetV2 and MobileNet stand out for their simplicity and e ciency, boasting the lowest number of parameters (3.5 m and 4.3 m) and the smallest storage footprint (13.9 MB and 16.4 MB). These models are particularly suitable for applications where computational resources are limited (RCE). On the other end of the spectrum, InceptionResNetV2 exhibits a complex architecture with the highest number of parameters (55.9 m) and requires the most storage (215 MB). While offering high accuracy, it may be less practical for deployment in resource-constrained scenarios (see Figure 5(a) and 5(b)). MobileNet, with a mean inference time of 3.9 s, emerges as the fastest model in our evaluation, making it well-suited for real-time applications. In contrast, InceptionResNetV2 demonstrates the longest mean inference time (26.6 s), suggesting slower processing (see Figure 5(c)). DenseNet121 and ResNet50 showcase the highest mean accuracy (88.9% and 96.2%, respectively) (see Figure 5(d)), underscoring their excellence in image classi cation. However, it's important to note that these models come with a higher computational cost and storage demand (see Figure 5(a) and 5(b)).

E cientNetB0 and E cientNetB1 strike a balance between accuracy and e ciency, demonstrating moderate values in both metrics. These models showcase a trade-off between resource utilization and prediction accuracy. On the other hand, E cientNetB0 demonstrates a compromise, with the lowest mean accuracy (80.6%), highlighting the importance of considering trade-offs when selecting models for RCE. E cientNetB1, distinguished by its heightened model complexity, adeptly achieves a balanced synthesis of computational e ciency and model accuracy. Its parameters, storage requirements, and inference time, considered collectively, position it as a versatile and well-rounded option for applications in settings where resource constraints necessitate e ciency without sacri cing predictive accuracy. Despite previous comprehensive study [17] suggesting E cientNetB1 as a preferable base model for DNN development in RCE scenarios, the results of the current empirical study advocate MobileNet as a more suitable candidate (see Figure 5(e)).

Furthermore, MobileNet, despite its modest computational requirements, stands out for providing fast inference times and achieving high mean accuracy. This combination of e ciency and commendable predictive performance makes MobileNet a reliable and adaptable choice for developing new DNN models within resource-constrained contexts. The model's ability to deliver e cient results without compromising accuracy makes it particularly valuable for scenarios where computational resources are limited, showcasing its versatility and suitability for a variety of applications.

B. New Model (GRM-MobileNet)

The classi cation report in Table V provides valuable insights into the performance of the GRM-MobileNet model across di erent classes. In this table, the precision metric measures the accuracy of positive predictions for each class. For instance, the precision for ”aeroplane” indicates that out of all the images predicted as “aeroplane”, 92% were correctly classi ed. The recall matrix, also known as sensitivity, measures the ability of the model to correctly identify instances of a class. For example, the recall for ”gold sh” suggests that the model correctly identi ed 96% of all gold sh images in the dataset. The F1-score is the harmonic mean of precision and recall, providing a balanced measure of a model's performance. It takes into account both false positives and false negatives. For instance, the F1-score for ”parrot” is 0.71, indicating a relatively good balance between precision and recall for this class. The overall accuracy of the model is 73%, meaning that 73%

TABLE VII. Comparison of DNN models with four evaluation matrices

DNN Models	Parameters (M)	Model Size (MB)	Mean Time (ms)	Accuracy (%)
MobileNetV2	3.5	13.9	5.6	81.9
MobileNet	4.3	16.4	3.9	97.6
E cientNetB0	5.3	20.9	8.6	80.6
E cientNetB1	7.9	30.8	14.2	83.0
DenseNet121	8.1	31.8	13.5	88.9
E cientNetB2	9.2	35.8	12.1	80.2
Xception	22.9	87.7	11.1	81.9
InceptionV3	23.9	91.8	11.5	91.6
ResNet50	25.6	98.2	8.1	96.2
InceptionResNetV2	55.9	215	26.6	90.1

Figure 5. Bar charts comparing evaluation metrics across ten DNN models: (a) compares the number of parameters in millions for each model; (b) compares the storage requirements of each model in megabytes; (c) illustrates the variations in inference time between models; (d) displays the accuracy differences in image prediction for each model; (e) provides an overall comparison of the four metrics across the ten models.

of all predictions made by the model were correct. This classi cation report demonstrates that the GRM-MobileNet model achieves reasonably good performance across various classes.

Table VI and Figure 6(a, b and d) illustrate that while MobileNetV2 exhibits a lower number of parameters (3.5 M) and model size (41.1 MB), its accuracy (30.18%) is relatively lower compared to other models. On the other hand, both E cientNetB0 and E cientNetB1 demonstrate higher parameter counts and model sizes, resulting in lower accuracies compared to other models. Despite having the highest number of parameters (25.6 M) and model size (294.0 MB), ResNet50 achieves only moderate accuracy (35.27%). Conversely, GRM-MobileNet, the proposed model, presents a relatively lower number of parameters (3.2 M) and model size (37.3 MB) in comparison to other models, while achieving the highest accuracy (73.21%). Furthermore, its inference time (430.3 ms) is comparable to other models, indicating e cient performance (Figure 6(c)). Thus, based on the information provided in Table VI and Figure 6, the GRM-MobileNet model stands out as it outperforms the benchmark models in terms of accuracy while maintaining a relatively smaller model size and inference time, rendering it a promising choice for resource-constrained scenarios.

5. Conclusions and Future Work

This paper began with an empirical study of various DNN models in an RCE that revealed valuable insights into their performance attributes. Among the models evaluated, MobileNet emerges as the most suitable candidate for the development of a new DNN model in RCE scenarios. This determination is based on a holistic assessment of MobileNet's characteristics, showcasing a favorable combination of key metrics. MobileNet exhibits a relatively low number of parameters (3.5 million), indicating a manageable level of model complexity and depth. Furthermore, the model demands a compact storage requirement of 13.9 megabytes, making it e cient in terms of resource utilization. Notably, MobileNet achieves a fast mean inference time of 3.9 seconds, enhancing its suitability for real-time applications in RCE. The model's commendable mean accuracy of 97.6% further solidi es its position as a promising choice for e ective deployment. MobileNet, in summary, provides balanced performance in terms of parameters, storage, inference time, and accuracy, making it an ideal platform for creating e ective DNN models that tackle the problems caused by resource constraints in real-world applications. This work brings substantial value to the eld of deploying DNN models in resource-constrained conditions. When selecting baseline models for image classi cation applications speci cally designed for RCE devices, it lays the foundation for decision-making. Accordingly, in the second phase of this paper, utilizing the identi ed baseline model, a new model, GRM-MobileNet was designed and created by applying compound scaling and global average pooling techniques. The GRM-MobileNet was trained and experimented with using ImageNet10 dataset, which is a subset of the ImageNet1K dataset with 10 image categories. The constructed model, GRM-MobileNet, has the highest accuracy (73.21%) despite having comparatively fewer parameters (3.2 M) and a smaller model size (37.3 MB) than the other benchmark models above. Table VIII shows the percentage improvement of the GRM-MobileNet model, which was developed from the MobileNet baseline model.

TABLE VIII. Improvement of GRM-MobileNet compare with baseline model, MobileNet

DNN Models	Para -meters (M)	Model Size (MB)	Acc (%)	Inference Time (ms)
MobileNet	4.2	49.0	45.09	437.6
GRM-MobileNet	3.2	37.3	73.21	430.3
Improvement (%)	23.81	23.88	28.12	1.66

This table (VIII) presents a comparison between MobileNet and GRM-MobileNet models across various evaluation metrics. GRM-MobileNet demonstrates improvements across all parameters compared to MobileNet. Speci cally, there is a 23.81% reduction in parameters, leading to a smaller model size by 23.88%. Additionally, GRM-MobileNet achieves a signi cant improvement in accuracy, with a 28.12% increase compared to MobileNet. Despite these enhancements, the improvement in inference time is marginal, with only a 1.66% decrease observed for GRM-MobileNet compared to MobileNet. In general, GRM-MobileNet showcases notable advancements over MobileNet, particularly in terms of model e ciency and predictive performance for the RCE settings.

Despite the promising results obtained in this study, several limitations should be acknowledged. Firstly, we utilized only 10 benchmark models to compare and identify the baseline model for developing the GRM-MobileNet model. While these models are well-established in the literature, numerous other benchmark models were not included in this study. Future research will expand the scope of model comparison by incorporating a broader range of benchmark models and evaluating them using comprehensive metrics to ensure a more robust selection of the baseline model.

Secondly, the development of the GRM-MobileNet model was based on a dataset comprising only 10 categories of images. This limited scope may not fully capture the model's potential performance across a wider variety of image classi cations. In future work, we aim to utilize the ImageNet1K dataset, which contains 1000 categories of images, to train and evaluate the model. This will provide a more extensive evaluation of the model's capabilities and generalizability. Expanding the dataset will allow us to better assess the model's performance in diverse and complex image classi cation tasks, thereby enhancing the reliability and applicability of the GRM-MobileNet model in real-world scenarios.

Figure 6. Bar charts comparing evaluation metrics between GRM-MobileNet and ve benchmark DNN models: (a) compares the number of parameters in millions for each model; (b) compares the storage requirements of each model in megabytes; (c) illustrates the variances in inference time between models; (d) displays the accuracy differences in image prediction for each model.

Thirdly, the comparative study of the current GRM-MobileNet model shows that the inference time is only marginally reduced (1.66%). To address this, future work will employ model optimization strategies such as factorization and pruning to increase the GRM-MobileNet model's e ciency by reducing its inference time. Factorization approaches will decompose the model's parameters into more e cient representations, thereby reducing the computational load during inference. Pruning will systematically remove unnecessary or insigni cant weights and connections from the neural network, thereby reducing the model's size without compromising its functionality. These modi cations aim to enhance the speed and responsiveness of GRM-MobileNet, making it more suitable for real-time applications.

The goal of these developments is to satisfy the requirements of e ciency-driven image classi cation applications intended for deployment on RCE devices, which often have constrained memory and processing capacity. With these methods in place, the GRM-MobileNet model will be more capable of providing high-performance image classi cation while meeting the demanding requirements of RCE environments, ensuring practical usefulness and e ectiveness in real-world situations. By addressing these limitations in future research, we aim to develop a more

comprehensive and e cient DNN model that is capable of meeting the diverse demands of RCE based image classi cation applications.

References

[1] S. Kuutti, R. Bowden, Y. Jin, P. Barber, and S. Fallah, "A survey of deep learning applications to autonomous vehicle control," IEEE Transactions on Intelligent Transportation Systems, vol. 22, no. 2, pp. 712–733, 2020.

[2] 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, no. 8, p. 3992, 2023.

[3] S. Yang, F. Zhu, X. Ling, Q. Liu, and P. Zhao, "Intelligent health care: Applications of deep learning in computational medicine," Frontiers in Genetics, vol. 12, p. 607471, 2021.

[4] J. R. Leow, W. H. Khoh, Y. H. Pang, and H. Y. Yap, "Breast cancer classi cation with histopathological image based on machine learning." International Journal of Electrical & Computer Engineering (2088-8708), vol. 13, no. 5, 2023.

[5] B. Cassidy, N. D. Reeves, J. M. Pappachan, N. Ahmad, S. Haycocks, D. Gillespie, and M. H. Yap, "A cloud-based deep learning framework for remote detection of diabetic foot ulcers," IEEE Pervasive Computing, vol. 21, no. 2, pp. 78–86, 2022.

[6] L. Santos, F. N. Santos, P. M. Oliveira, and P. Shinde, "Deep learning applications in agriculture: A short review," in Robot 2019: Fourth Iberian Robotics Conference: Advances in Robotics, Volume 1. Springer, 2020, pp. 139–151.

[7] 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, vol. 5, p. 100322, 2023.

[8] A. Bhargava and A. Bansal, "Fruits and vegetables quality evaluation using computer vision: A review," Journal of King Saud University-Computer and Information Sciences, vol. 33, no. 3, pp. 243–257, 2021.

[9] I. H. Sarker, A. I. Khan, Y. B. Abushark, and F. Alsolami, "Internet of things (iot) security intelligence: a comprehensive overview, machine learning solutions and research directions," Mobile Networks and Applications, vol. 28, no. 1, pp. 296–312, 2023.

[10] S. Suprayitno, W. A. Fauzi, K. Ain, and M. Yasin, "Real-time military person detection and classi cation system using deep metric learning with electrostatic loss," Bulletin of Electrical Engineering and Informatics, vol. 12, no. 1, pp. 338–354, 2023.

[11] P. Yao, "Real-time analysis of basketball sports data based on deep learning," Complexity vol. 2021, pp. 1–11, 2021.

[12] 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). IEEE, 2020, pp. 1–6.

[13] Y. Li, "Research and application of deep learning in image recognition," in 2022 IEEE 2nd international conference on power, electronics and computer applications (ICPECA) IEEE, 2022, pp. 994–999.

[14] X. Zhao, "Research and application of deep learning in image recognition," in Journal of Physics: Conference Series, vol. 2425, no. 1. IOP Publishing, 2023, p. 012047.

[15] 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, no. 1, pp. 103–115, 2022.

[16] 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.

[17] C. Raa , J. Gapar, and K. Ali, "Deep neural networks optimization for resource-constrained environments: techniques and models," Indonesian Journal of Electrical Engineering and Computer Science, vol. 33(3), pp. 1843–1854, 2024.

[18] V. Kamath and A. Renuka, "Deep learning based object detection for resource constrained devices-systematic review, future trends and challenges ahead," Neurocomputing, 2023.

[19] H. Nguyen, "Real-time vehicle and pedestrian detection on embedded platforms," J. Theor. Appl. Inf. Technol, vol. 98, no. 21, pp. 3405–3415, 2020.

[20] 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.

[21] 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 Arti cial Intelligence, vol. 126, p. 107086, 2023.

[22] T. Lawrence and L. Zhang, "Iotnet: An e cient and accurate convolutional neural network for iot devices," Sensors, vol. 19, no. 24, p. 5541, 2019.

[23] A. Ignatov, R. Timofte, A. Kulik, S. Yang, K. Wang, F. Baum, M. Wu, L. Xu, and L. Van Gool, "Ai benchmark: All about deep learning on smartphones in 2019," in 2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW). IEEE, 2019, pp. 3617–3635.

[24] A. Ignatov, R. Timofte, W. Chou, K. Wang, M. Wu, T. Hartley, and L. Van Gool, "Ai benchmark: Running deep neural networks on android smartphones," in Proceedings of the European Conference on Computer Vision (ECCV) Workshops, 2018, pp. 0–0.

[25] A. Mohi ud din and S. Qureshi, "A review of challenges and solutions in the design and implementation of deep graph neural networks," International Journal of Computers and Applications, vol. 45, no. 3, pp. 221–230, 2023.

[26] L. Zhao, L. Wang, Y. Jia, and Y. Cui, "A lightweight deep neural network with higher accuracy," Plos one, vol. 17, no. 8, p. e0271225, 2022.

[27] 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.

[28] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: E cient

convolutional neural networks for mobile vision applications," arXiv preprint arXiv:1704.04861, 2017.

[29] M. Tan and Q. Le, "E cientnet: Rethinking model scaling for convolutional neural networks," in International conference on machine learning. PMLR, 2019, pp. 6105–6114.

[30] 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.

[31] S. Vadera and S. Ameen, "Methods for pruning deep neural networks," IEEE Access, vol. 10, pp. 63 280–63 300, 2022.

[32] J. Yang, X. Shen, J. Xing, X. Tian, H. Li, B. Deng, J. Huang, and X.-s. Hua, "Quantization networks," in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2019, pp. 7308–7316.

[33] J. H. Cho and B. Hariharan, "On the e cacy of knowledge distillation," in Proceedings of the IEEE/CVF international conference on computer vision, 2019, pp. 4794–4802.

[34] J. Gou, B. Yu, S. J. Maybank, and D. Tao, "Knowledge distillation: A survey," International Journal of Computer Vision, vol. 129, no. 6, pp. 1789–1819, 2021.

[35] S. Fu, Z. Li, Z. Liu, and X. Yang, "Interactive knowledge distillation for image classi cation," Neurocomputing, vol. 449, pp. 411–421, 2021.

[36] L. Balderas, M. Lastra, and J. M. Benítez, "Optimizing dense feed-forward neural networks," Neural Networks, vol. 171, pp. 229–241, 2024.

[37] M. S. A. Reshan, K. S. Gill, V. Anand, S. Gupta, H. Alshahrani, A. Sulaiman, and A. Shaikh, "Detection of pneumonia from chest x-ray images utilizing mobilenet model," in Healthcare, vol. 11, no. 11. MDPI, 2023, p. 1561.

[38] Y. Gulzar, "Fruit image classi cation model based on mobilenetv2 with deep transfer learning technique," Sustainability, vol. 15, no. 3, p. 1906, 2023.

[39] L. Yong, L. Ma, D. Sun, and L. Du, "Application of mobilenetv2 to waste classi cation," Plos one, vol. 18, no. 3, p. e0282336, 2023.

[40] S. Benkrama and N. E. H. Hemdani, "Deep learning with e cientnetb1 for detecting brain tumors in mri images," in 2023 International Conference on Advances in Electronics, Control and Communication Systems (ICAECCS). IEEE, 2023, pp. 1–6.

[41] S. A. Albelwi, "Deep architecture based on densenet-121 model for weather image recognition," International Journal of Advanced Computer Science and Applications, vol. 13, no. 10, 2022.

[42] S. A. Joshi, A. M. Bongale, P. O. Olsson, S. Urolagin, D. Dharrao, and A. Bongale, "Enhanced pre-trained xception model transfer learned for breast cancer detection," Computation, vol. 11, no. 3, p. 59, 2023.

[43] A. E. Minarno, L. Aripa, Y. Azhar, and Y. Munarko, "Classi cation of malaria cell image using inception-v3 architecture," JOIV: International Journal on Informatics Visualization, vol. 7, no. 2, pp. 273–278, 2023.

[44] S. Raveena and R. Surendran, "Resnet50-based classi cation of co ee cherry maturity using deep-cnn," in 2023 5th International Conference on Smart Systems and Inventive Technology (ICSSIT). IEEE, 2023, pp. 1275–1281.

[45] 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). IEEE, 2019, pp. 159–163.

[46] Y. Zhong, Z. Teng, and M. Tong, "Lightmixer: A novel lightweight convolutional neural network for tomato disease detection," Frontiers in Plant Science, vol. 14, p. 1166296, 2023.

[47] C. Chen, P. Zhang, H. Zhang, J. Dai, Y. Yi, H. Zhang, and Y. Zhang, "Deep learning on computational-resource-limited platforms: a survey," Mobile Information Systems, vol. 2020, pp. 1–19, 2020.

[48] F. Martnez, H. Montiel, and F. Martínez, "Comparative study of optimization algorithms on convolutional network for autonomous driving." International Journal of Electrical & Computer Engineering (2088-8708), vol. 12, no. 6, 2022.

[49] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein et al., "Imagenet large scale visual recognition challenge," International journal of computer vision, vol. 115, pp. 211–252, 2015.

[50] 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, no. 3, pp. 858–871, 2021.

[51] V. Labatut and H. Cherif, "Accuracy measures for the comparison of classi ers," arXiv preprint arXiv:1207.3798, 2012.

Raa 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 located with the Department of Computer Science and Informatics in Uva Wellasa University, Sri Lanka. His research interests are deep neural network, machine learning, arti cial intelligence, intelligent system, image classi cation, and android application development. He can be contacted at email: mraa@gmail.com.