Baseline model for deep neural networks in resource-constrained environments: an empirical investigation

Raafi Careem[1], Md Gapar Md Johar[2], Prof. Dr Abdol 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

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

Received ## Mon. 20##, Revised ## Mon. 20##, Accepted ## Mon. 20##, Published ## Mon. 20##

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. Consequently, MobileNet emerges as a suitable baseline model for future DNNs developed specifically for RCE image classification. The study's conclusions endorse MobileNet as a baseline model for transfer learning techniques (used in DNN design), providing valuable insights for optimizing DNN models in resource-constrained scenarios. This approach enhances the creation of efficiency-focused and lightweight DNN models, improving their application and efficacy in resource-constrained environments. Future research will leverage the identified MobileNet model as a foundation to create a new DNN model tailored for efficiency-driven image classification applications in RCE devices.

Keywords: Basline 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 vehicle [1], [2] healthcare [3-5], agriculture [6-8], security [9], [10], and sports [11]. Particularly notable is their proficiency in image classification within the computer vision discipline [8], [12-14]. 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) [17], which are prevalent in real-time applications.

The significance of adapting DNNs for use in RCEs is underscored by the rapid growth of the Internet of Things (IoT) and the increasing demand for mobile devices [7], [18-20]. 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 sacrificing 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 [22-24]. Onboard implementation, the direct deployment of DNNs on RCE devices, has several benefits, including real-time image classification, 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 fields, 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 efficient and effective manner.

Considering the challenges of implementing DNN models in RCE, particularly due to their size and processing demands [17], [24], [25], several studies have explored techniques for creating lightweight models alongside their very deep counterparts. This simplification of very deep models is achieved through optimization and compression techniques [17], [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], to reduce the size of networks [31], [32] and lower inference costs [33] 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 amount of bits in floating-point values that indicate activations and weights. To improve image classification accuracy, Yang et al. [34] used activation quantization and weights. However, using fewer bits for weights could result in a loss of accuracy, which would affect the accuracy of neural networks. Knowledge distillation is another method, as used in [35], [36] , 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 [37], 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], [38], which utilizes model weights from previously trained models. By using feature representations that a pre-trained model has learnt, TL eliminate 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 effective 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 model to develop new models utilizing the TL approach, such as MobileNet [28], [39], [40] , MobileNetV2 [38], [41] , EfficientNetB0, EfficientNetB1, EfficientNetB2[29],[42], DenseNet121 [43], Xception [27], InceptionV3 [44], ResNet50 [45] and InceptionResNetV2 [46]. Each of these models offering its own set of advantages and limitation. This has been analyzed comprehensively by the authors in [17]. Identifying the most suitable pre-trained model from the aforementioned list to serve as a baseline model for the development of an efficiency-focused model in RCEs is crucial. However, a notable research gap exists, as there is a lack of experimental studies evaluating these models within the context of RCE scenarios.

The primary objective of this article is to address this gap by systematically evaluating the aforementioned pre-trained DNN models through a well-defined experimental methodology in an RCE scenario. Our aim is to use suitable evaluation metrics to identify the most suitable baseline model for designing a new DNN model in RCE settings. The results obtained from this research will serve as a roadmap for the development of productive and successful image classification applications, ensuring optimal performance in practical situations where resource limitations are a common constraint. Subsequent sections will delve into the experimental methodology, evaluation metrics, comparison of results, and findings, providing valuable insights to guide the development of efficient DNN models for RCE.

2. RESEARCH METHOD

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) [17].

A. Evaluation matrices

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,

offering a comprehensive view of the strengths and trade-offs 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 (citation). 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 efficacy 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 when 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. Experimental Procedure

The experimental method, depicted in Fig. 1, followed a systematic procedure. Initially, ten DNN models were sequentially deployed onto a predefined RCE scenario. Each of the ten images selected from Fig. 1(a) was individually presented to every deployed model, as illustrated in Fig. 1. Subsequent to the image input, predictions generated by each model were observed and recorded for both accuracy and inference time, as shown in Fig. 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 1 and 2. While Table 1 presented accuracy values, Table 2 outlined inference times for each model across all images. These recorded accuracy values and inference times were then used to compute mean accuracy (Mean Acc) and mean inference time (Mean Time), respectively.

Mean accuracy provided an average measure of prediction accuracy for each of the ten models, as depicted in Table 1. Simultaneously, mean time represented an average measure of prediction time for each model, as detailed in Table 2.

In Tables 1, Mean Acc for each DNN model is calculated as the simple average of recorded accuracyvalues across all images (N=10), using the formula (1) [49], [50]:

$$\text{Mean Acc} = \frac{1}{N}\sum_{i=1}^{N} Acc_i \quad (1)$$

where, N is total number of images and Acc_i represents the accuracy value for the ^{i}th image.

Similarly, in Tables 2, Mean Inference Time (Mean Time) is determined as the simple average of recorded time values for each model across all images (N=10) predictions, using the formula (2) [49]:

$$\text{Mean Time} = \frac{1}{N}\sum_{i=1}^{N} Time_i \qquad (2)$$

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

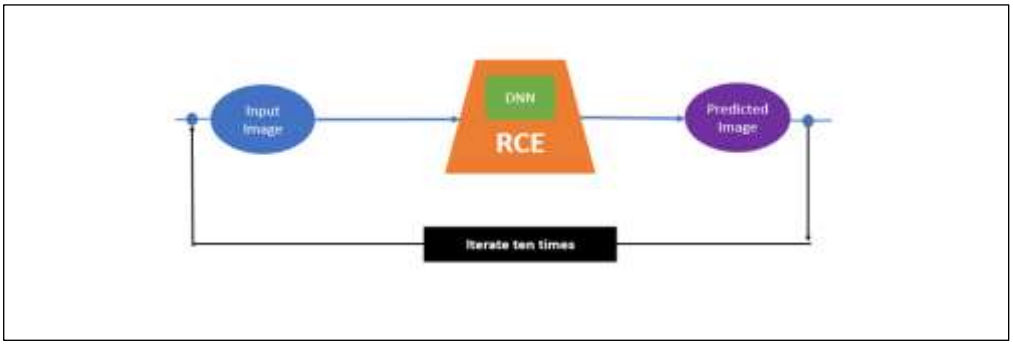

Figure 1. Overview of the experimental setup: green color box indicating the deployed DNN model on RCE, blue color oval shape is input image and purple is predicted image, orange color object refers RCE device

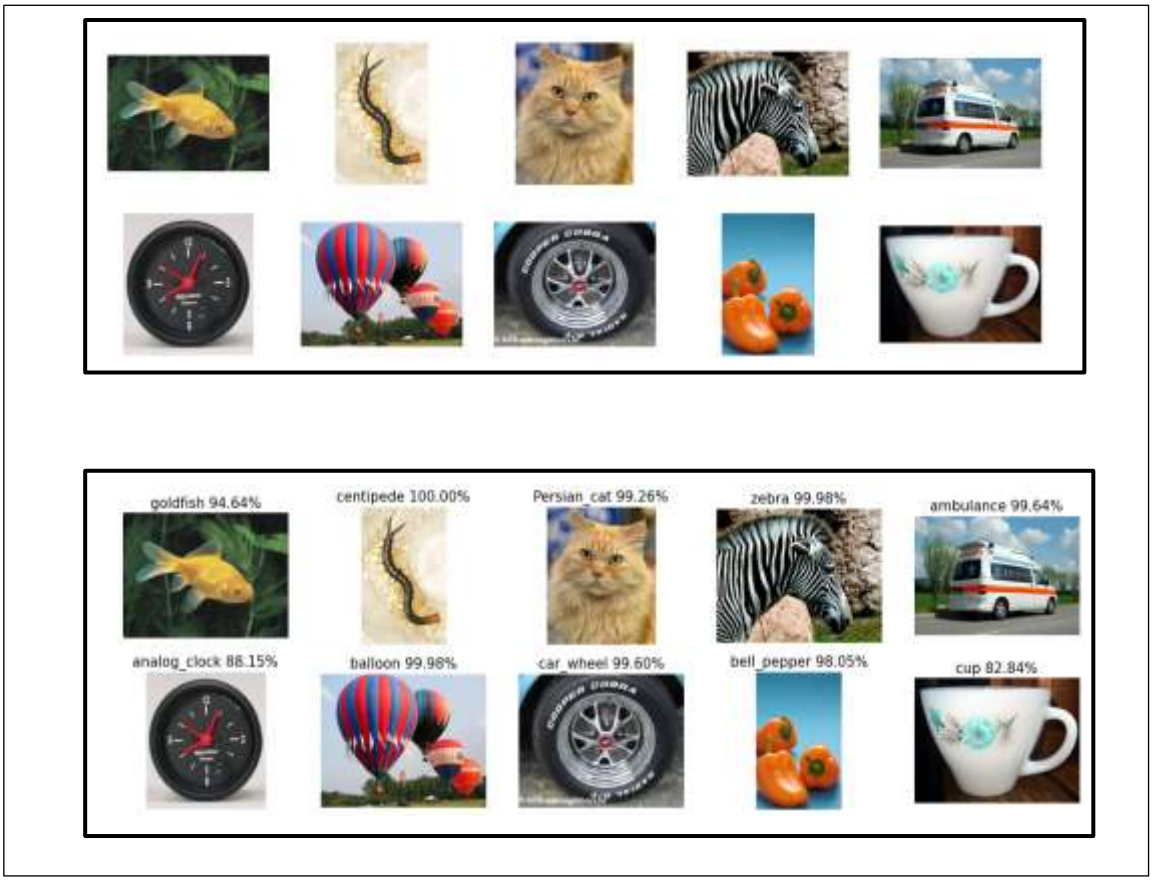

Figure 2. Set of images used for the experiment: (a) Images randomly selected from the ImageNet testing dataset; (b) Examples of predicted images from a model, with predicted labels displayed above each image along with their corresponding accuracy.

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 3 presents the systematic documentation on observations pertaining to these many criteria, which adds significant value to the assessment as a whole. The observations encompassing these diverse metrics are systematically documented and presented in Table 3. By incorporating these additional measures, the assessment goes beyond accuracy and inference time alone, providing valuable insights into computational efficiency and resource utilization for the predictions. This thorough examination is designed to facilitate decision-making regarding the potential role of these DNN models as baseline models, particularly in the development of new, efficiency-focused models tailored for deployment in resource-constrained scenario.

TABLE 1. RECORDED PREDICTION ACCURCY OF TEN DNN MODELS FOR TEN IMAGES

DNN Models	Goldfish	Centipede	Cat	Zebra	Ambulance	Balloon	Wheel	Clock	Bell Pepper	Cup	Mean Accuracy (%)
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
EfficientNetB0	92.28	67.15	82.50	88.14	97.43	90.15	88.17	44.16	96.20	59.72	80.59
EfficientNetB1	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
EfficientNetB2	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-ResNetV2	91.64	94.15	92.28	93.17	94.86	93.43	89.17	82.08	91.93	78.12	90.08

TABLE 2. RECORDED INFERENCE TIME FOR THE PREDICTION OF TEN IMAGES BY TEN DNN MODELS

DNN Models	Goldfish	Centipede	Cat	Zebra	Ambulance	Balloon	Wheel	Clock	Bell Pepper	Cup	Mean Time
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
EfficientNetB0	9.27	8.49	8.06	8.50	7.84	8.44	9.72	8.25	9.56	7.94	8.61
EfficientNetB1	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
EfficientNetB2	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-ResNetV2	26.43	26.72	27.17	27.00	26.88	27.11	25.80	26.08	26.32	26.41	26.59

3. RESULT AND DISCUSSION

The outcomes of the conducted experimentation involving various DNN models are documented in Table 3, offering a thorough overview of their performance based on essential evaluation metrics. Fig. 3 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 efficiency, boasting the lowest number of parameters (3.5m and 4.3m) and the smallest storage footprint (13.9MB and 16.4MB). 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 (55.9m) and requires the most storage (215MB). While offering high accuracy, it may be less practical for deployment in resource-constrained scenarios (see Fig. 3(a) and (b)).

MobileNet, with a mean inference time of 3.9s, 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 (26.6s), suggesting slower processing (see Fig 3.(c)). DenseNet121 and ResNet50 showcase the highest mean accuracy (88.9% and 96.2%, respectively) (see Fig. 3(d)), underscoring their excellence in image classification. However, it's important to note that these models come with a higher computational cost and storage demand (see Fig. 3(a) and (b)).

EfficientNetB0 and EfficientNetB1 strike a balance between accuracy and efficiency, demonstrating moderate values in both metrics. These models showcase a trade-off between resource utilization and prediction accuracy. On the other hand, EfficientNetB0 demonstrates a compromise, with the lowest mean accuracy (80.6%), highlighting the importance of considering trade-offs when selecting models for RCE.

EfficientNetB1, distinguished by its heightened model complexity, adeptly achieves a balanced synthesis of computational efficiency 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 efficiency without sacrificing predictive accuracy. Despite previous comprehensive study [17] suggesting EfficientNetB1 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 Fig. 3(e)).

Furthermore, MobileNet, despite its modest computational requirements, stands out for providing fast inference times and achieving high mean accuracy. This combination of efficiency 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 efficient 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.

TABLE 3. COMPARISION OF DNN MODELS WITH FOUR EVALUATION MATRIXES

`DNN Models	Parameters (m)	Storage (MB)	Mean Time (s)	Mean Acc (%)
MobileNetV2	3.5	13.9	5.6	81.9
MobileNet	4.3	16.4	3.9	97.6
EfficientNetB0	5.3	20.9	8.6	80.6
EfficientNetB1	7.9	30.8	14.2	83.0
DenseNet121	8.1	31.8	13.5	88.9

EfficientNetB2	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

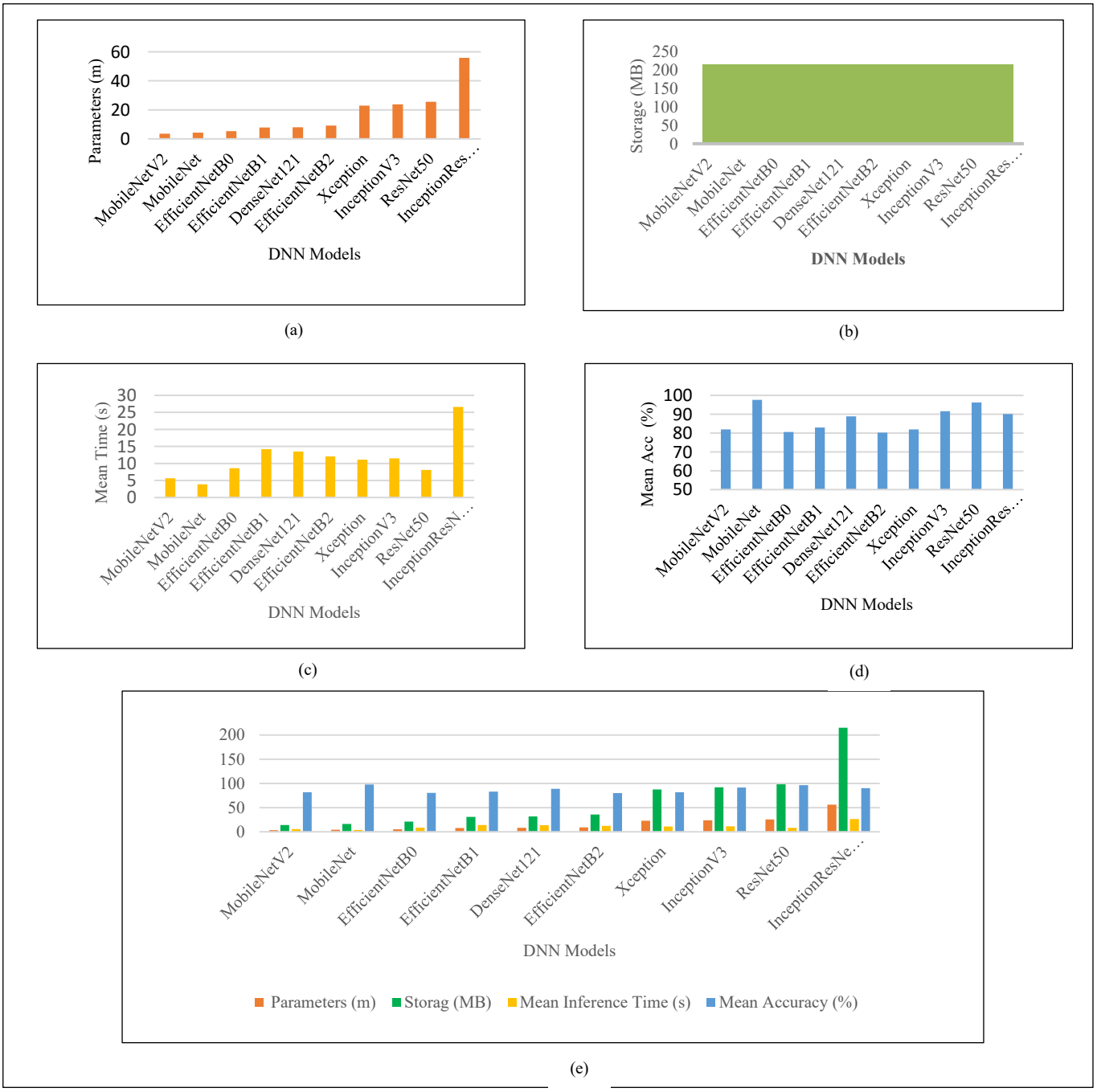

Figure 3. Bar Charts Comparing Evaluation Metrics Across Ten DNN Models: (a) compares the number of parameters in millions for each model; (b) compares the storage requirements of each model in megabytes; (c) illustrates the differences in inference time between models; (d) displays the accuracy differences in image prediction for each model; (e) provides an overall comparison of the four metrics across the ten models.

4. CONCLUSION

This paper presents an empirical study of various DNN(DNN) models in a Resource-Constrained Environment (RCE) has 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 efficient 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 solidifies its position as a promising choice for effective deployment. MobileNet, in summary, provides balanced performance in terms of parameters, storage, inference time, and accuracy, making it an ideal platform for creating effective DNN models that tackle the problems caused by resource constraints in real-world applications. This work brings substantial value to the field of deploying DNN models in resource-constrained conditions. When selecting baseline models for image classification applications specifically designed for RCE devices, it lays a foundation for decision-making.

The future research endeavors will leverage the identified MobileNet model as the baseline to develop a novel DNN model, termed GRMobiNet. This development aims to cater to efficiency-focused image classification applications specifically tailored for deployment in RCE devices.

[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, 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, p. 3992, 2023, doi:10.3390/s23083992.

[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 classification with histopathological image based on machine learning," *International Journal of Electrical & Computer Engineering (2088-8708),* vol. 13, 2023, doi:10.11591/ijece.v13i5.

[5] B. Cassidy, N. D. Reeves, J. M. Pappachan, N. Ahmad, S. Haycocks, D. Gillespie*, et al.*, "A cloud-based deep learning framework for remote detection of diabetic foot ulcers," *IEEE Pervasive Computing,* 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*, 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,* p. 100322, 2023, doi:10.1016/j.atech.2023.100322.

[8] A. Bhargava and A. Bansal, "Fruits and vegetables quality evaluation using computer vision: A review," *J. King Saud Univ. Comput. Inf. Sci.,* vol. 33, 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, pp. 296-312, 2023.

[10] 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,* vol. 12, pp. 338-354, 2023, doi:10.11591/eei.v12i1.4284.

[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)*, 2020, pp. 1-6, doi:10.1109/wf-iot48130.2020.9221198.

[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)*, 2022, pp. 994-999, doi:10.1109/icpeca53709.2022.9718847.

[14] X. Zhao, "Research and application of deep learning in image recognition," in *Journal of Physics: Conference Series*, 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, pp. 103-115, 2022, doi:10.1007/s11554-021-01164-1.

[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, doi:10.1016/j.sysarc.2023.102978.

[17] R. Careem, G. Johar, and A. Khatibi, "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, doi:10.11591/ijeecs.v33.i3.

[18] 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.

[19] H. Nguyen, "Real-time vehicle and pedestrian detection on embedded platforms," *J. Theor. Appl. Inf. Technol,* vol. 98, 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, doi:10.1016/j.pisc.2016.04.052.

[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 Artificial Intelligence,* vol. 126, p. 107086, 2023, doi:10.1016/j.engappai.2023.107086.

[22] 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.
[23] A. Ignatov, R. Timofte, A. Kulik, S. Yang, K. Wang, F. Baum*, 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.
[24] A. Ignatov, R. Timofte, W. Chou, K. Wang, M. Wu, T. Hartley*, et al.*, "Ai benchmark: Running deep neural networks on android smartphones," in *Computer Vision – ECCV 2018 Workshops*, 2019, pp. pp. 288–314, doi:10.1007/978-3-030-11021-5_19.
[25] 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.
[26] L. Zhao, L. Wang, Y. Jia, and Y. Cui, "A lightweight deep neural network with higher accuracy," *Plos one,* vol. 17, 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, doi:10.1109/cvpr.2017.195.
[28] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand*, et al.*, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861,* 2017.
[29] M. Tan and Q. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," in *International conference on machine learning*, 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, doi:10.1016/j.neucom.2021.07.045.
[31] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems,* vol. 25, pp. 1097-1105, 2012.
[32] 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,* vol. 11, p. 1719, 2021, doi:10.11591/ijece.v11i2.pp1719-1727.
[33] H. Li, A. Kadav, I. Durdanovic, H. Samet, and H. P. Graf, "Pruning filters for efficient convnets," *arXiv preprint arXiv:1608.08710,* 2016.
[34] J. Yang, X. Shen, J. Xing, X. Tian, H. Li, B. Deng*, 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.
[35] 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.
[36] 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.
[37] 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.
[38] 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.
[39] 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.
[40] A. Pujara, "Image Classification with MobileNet," *Published in Analytics Vidhya,* 2020.
[41] 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.
[42] 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.
[43] S. A. Albelwi, "Deep Architecture based on DenseNet-121 Model for Weather Image Recognition," *Int. J. Adv. Comput. Sci. Appl,* vol. 13, pp. 559-565, 2022, doi:10.14569/ijacsa.2022.0131065.
[44] 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.
[45] S. Mascarenhas and M. Agarwal, "A comparison between VGG16, VGG19 and ResNet50 architecture frameworks for Image Classification," in *2021 International conference on disruptive technologies for multi-disciplinary research and applications (CENTCON)*, 2021, pp. 96-99.
[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] C. Chen, P. Zhang, H. Zhang, J. Dai, Y. Yi, H. Zhang*, et al.*, "Deep learning on computational-resource-limited platforms: a survey," *Mobile Information Systems,* vol. 2020, pp. 1-19, 2020.
[48] F. MartEnez, 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, 2022, doi:10.11591/ijece.v12i6.pp6363-6372.
[49] 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.
[50] V. Labatut and H. Cherifi, "Accuracy measures for the comparison of classifiers," *arXiv preprint arXiv:1207.3790,* 2012.

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 in Uva Wellasa 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 ice 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.

Prof. Dr Abdol 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.