Original Research Article

Analyzing Baseline Models for Optimizing Deep Neural Networks in Resource-Constrained Environments

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

[1] *Department of Computer Science and Informatics, Uva Wellassa University, Badulla, Sri Lanka*
[2] *Management and Science University, Shah Alam, Malaysia*

A B S T R A C T

This study aims to identify a baseline model for optimizing deep neural network (DNN) models for deployment in resource-constrained environments (RCE). Although DNNs are excellent in many applications, their deployment on devices like wearables and mobile phones presents significant challenges. The study investigates six popular DNN models, including MobileNet (V1 and V2), ResNet50, InceptionV3, DenseNet121, and EfficientNetB1. To assess each model's advantages, disadvantages, usability, and effectiveness in RCE scenarios, a comprehensive review and empirical analysis were conducted. The analysis focuses on optimizing these models to function effectively given the limited computational power and memory of RCE devices. Key factors such as model size, computational complexity, and inference speed are examined to uncover performance trade-offs between accuracy and resource efficiency. The findings suggest that MobileNetV1 should serve as baseline models for building efficiency-focused DNN models for image classification on RCE devices. This recommendation is based on MobileNetV1's balance between performance and efficiency, making it an ideal starting point for further optimization.

Introduction

*T*he rapid advancement of DNNs has revolutionized numerous fields, delivering outstanding performance in tasks such as image classification [1-5], natural language processing [6-10], and autonomous driving [11,12]. These advanced models have proven to be very capable, surpassing the performance of conventional techniques and establishing new standards in a wide range of applications. Deploying DNNs in RCE [13] is still a difficult task, despite their outstanding results. Complex DNN models may not be able to run on devices such as mobile phones, wearables, and internet of things (IoT) devices due to their restricted computational power, memory, and battery life. Due to these limitations, considerable optimization work is required to guarantee that

DNNs may operate well and efficiently with the restricted resources at hand. Achieving successful deployment in these systems requires striking a balance between the trade-offs between computational complexity, resource consumption, and model accuracy. By finding viable baseline models and improving them for RCE scenarios, this study seeks to address these issues and open the way for a wider acceptance and deployment of DNN technology in common devices. The rising number of RCEs in recent years, driven by IoT and smart devices, has made the employment of DNNs in these situations necessary. Using DNNs on RCEs has several benefits, including improved privacy and security, low latency, decreased bandwidth consumption, and real-time image classification. [14]. Faster decision-making in a variety of industries, including smart homes, autonomous cars, healthcare, agriculture, and transportation, is made possible by DNNs on RCE devices. As a result, the requirement for the successful and efficient installation of DNNs on RCEs has grown. Many DNN models, including MobileNet [15-18], ResNet [2,4,19], InceptionNet [4,20-22], DenseNet [23], and EfficientNet [24,25] have been developed to solve these issues. Each model has advantages and disadvantages specific to RCE-based image classification. The features, algorithms, usability, strengths, and drawbacks of these cutting-edge DNN models for deploying DNNs in RCE devices are reviewed and analyzed in this work. It also suggests a foundational DNN model for RCE device classification of images, which would be helpful as a guide for further study.

Materials and Methods

Deploying s DNNs on RCE devices can be challenging due to their limited computational resources and energy constraints. These limitations necessitate careful consideration of model complexity, memory usage, and power consumption to ensure efficient operation. To address these challenges, researchers have developed a variety of DNN models specifically designed for RCE applications. These models aim to balance performance with resource efficiency, enabling practical deployment on devices such as mobile phones, wearables, and IoT sensors. In this study, we conduct two types of analyses to evaluate these DNN models: comprehensive review and empirical. The comprehensive review analysis involves a thorough review of existing literature and state-of-the-art DNN models, examining their theoretical foundations, architectural innovations, and potential for adaptation to RCE scenarios. This review provides a holistic understanding of the strengths and weaknesses of various models, such as ResNet, MobileNet, InceptionNet, DenseNet, and EfficientNet, in the context of RCE deployment. The empirical analysis, on the other hand, involves practical experimentation and performance evaluation of these models under RCE conditions. This includes assessing factors such as model size, computational complexity and inference speed [26]. By conducting these empirical tests, we aim to provide concrete data on how each model performs in real-world RCE scenarios, highlighting the trade-offs between accuracy and resource efficiency. These analyses are detailed in the following sections, offering insights into the optimization strategies that can enhance the deployment of DNNs in RCE.

Comprehensive review analysis

This compressive analysis has undertaken based on the review study of the models and Tables 1 and 2. The DNN frameworks ResNet50, DenseNet, MobileNet, Inception, and EfficientNet have all been created for image classification towards RCE devices. ResNet50 bypasses the vanishing gradient problem by using residual blocks and skip connections. While DenseNet uses concatenation for layers, mobile and embedded vision applications, MobileNet employs depthwise separable convolutions. EfficientNet, a small-scale variation, achieves high accuracy while maintaining a moderate model size and low computing cost. The advantages and disadvantages of the aforementioned DNN models used for image classification using RCE are contrasted in Table 1. In comparison to other DNN models, MobileNet is less accurate but more efficient and quicker due to its optimization for devices with limited processing capacity. The most advanced deep network

image classification models, ResNet50 and ResNet152V2, are risky of overfitting due to their complexity and resource consumption. Both InceptionV3 as well as InceptionResNetV2 are computationally efficient when working with different filter sizes, but they are challenging to train and optimize. DenseNet121 is memory-efficient and requires less parameters, but its computational cost and memory consumption are significant. Lastly, EfficientNetB0, B1, as well as B7 have a flexible architecture and are quick, accurate, and efficient. Table 2 and Figure 1 show that the EfficientNet architecture—which is designed to have excellent accuracy with fewer processes is the most accurate of the aforementioned DNN models. The network's depth, width, along with resolution are all equally scaled via the EfficientNet's compound scaling technique, which keeps processing budget constant. EfficientNetB1 is a minimal-scale variation that achieves great accuracy while maintaining a small model capacity and low computational cost, making it suitable for deployment on RCE devices. The Table 2 illustrates the effectiveness of EfficientNetB1 in terms of speed and storage utilization, as well as its versatility, by contrasting the advantages and disadvantages of various models. Given its high level of efficiency, it becomes sense to use EfficientNetB1 as a baseline model when proposing a new model to feed DNN deployment on RCE.

Table 1: Comparing the strengths and weaknesses of DNN Architectures

| DNN model architecture	Strength	Weakness	[Ref.]
MobileNet (v1,v2)	Designed for devices with limited computing power; efficient and fast; reduces computation requirements.	Reduced accuracy compared to alternative DNN models.	[15-18]
ResNet50	Multiple-layer DNN for complex feature extraction; skip connections to address vanishing gradient problems.	Complex plus resource-intensive; risk of over-fitting; and requires regularization.	[2,4,19]
InceptionV3	Accuracy-enhancing efficient computation using several filter sizes; combines Inception with residual connections for better accuracy; and versatile in computer vision applications.	Difficult to practice and improve; loss of detail in the image; and RCE requires a lot of storage.	[4,20-22]
DenseNet121	Dense connectivity improves gradient flow and the network; and fewer parameters result in memory efficiency.	High cost of computation	[23,27]
EfficientNetB1	Fast, efficient, and highly accurate	Complex architecture	[24,25]

Table 2: Evaluating the deployment of DNN models for image classification using the ImageNet dataset: a comparison of model size, inference time, and accuracy

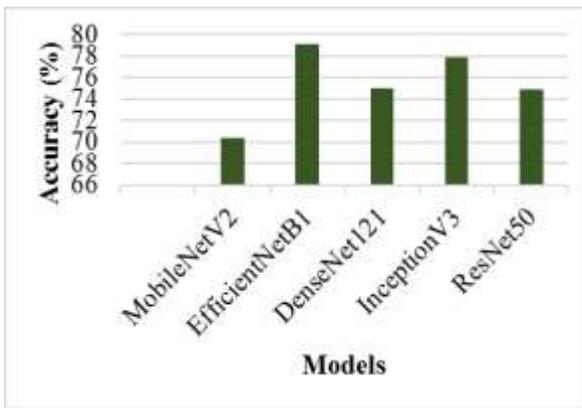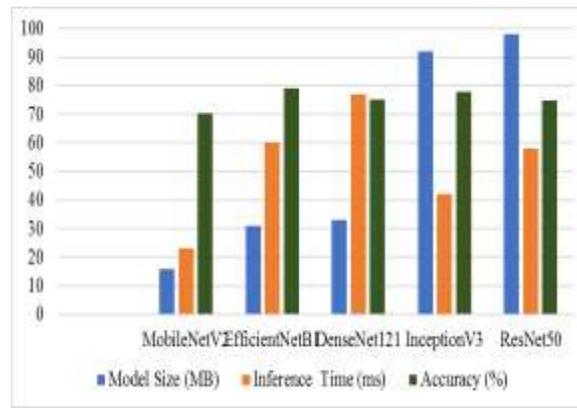| DNN models	Model size (in MB)	Inference time (in ms)	Accuracy (%)
MobileNetV1	16	23	70.4
MobileNetV2	14	26	71.3
EfficientNetB1	31	60	79.1
DenseNet121	33	77	75.0
InceptionV3	92	42	77.9
ResNet50	98	58	74.9

Sources: [13,28]
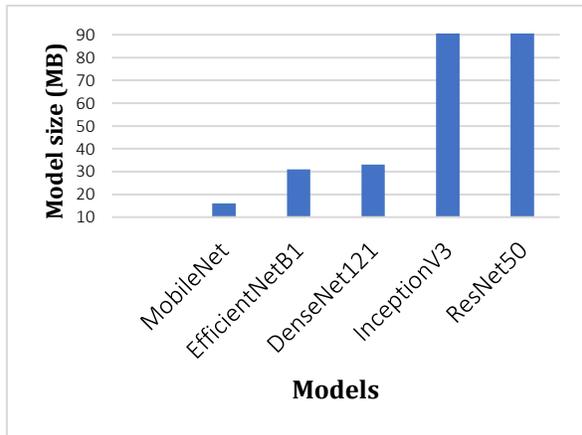
a
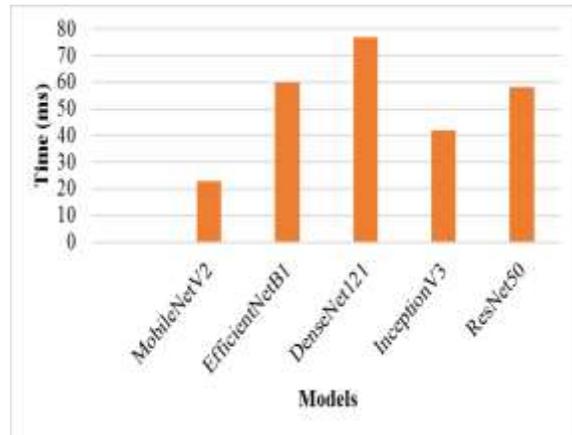
b

c

d

Figure 1: Bar charts to compare Five DNN models with respect to evaluation matrices: (a) displays the model size (MB); (b) compares the inference time (ms) of the models; (c) illustrates accuracy (%) of the models' prediction; and (d) gives a general overview of how the three matrices in the five models compare

Empirical analysis

Based on the aforementioned models' trials, an empirical study of models for optimization was carried out to evaluate their effectiveness.

Finding the best baseline model to use while installing DNNs in RCE was the goal. The experiments were carried out in a Python and TensorFlow environment using Keras models. It used an Intel 1.86GHz CPU and 4GB RAM to test our DNNs models in RCE. Although this hardware specification may appear outdated by current standards, it was deliberately chosen to simulate the limitations that many RCE devices still encounter today, such as older embedded systems, budget mobile devices, and IoT devices. These systems often have limited computational power and memory, making it critical to optimize DNN models for such environments. This ensures that the models can operate efficiently across a broader spectrum of devices, not just high-end systems with modern hardware for each model, a set of ten input images, as depicted in Figure 2, was used to generate predictions, and the resulting details were recorded in Table 2.

Table 3 indicates several different deep learning models, along with various performance and resource metrics, including the storage requirements of the model in MB, inference time in milli seconds, and the prediction accuracy of each model. Based on the data in the Table 3, we can make the following observations:

In view of memory limitations, RCE benefit more from lesser storage requirements. MobileNetV1 shows 97.64% accuracy with 16.4 MB storage requirement along with 3,88 ms inference time. At 13.9 MB, MobileNetV2 requires the least amount of storage, whereas EfficientNetB1 requires 30.8 MB. For real-time applications on RCE devices, a lower inference time is essential. With an inference time of 5.6 ms, MobileNetV2 has the highest speed, followed by ResNet50 at 8.1 ms. Better performance requires higher accuracy, and ResNet50 leads in this domain with 96.2% accuracy, followed by InceptionV3 at 91.6%. MobileNetV2 offers the lowest storage requirement and the fastest inference time, but suffers from the lowest accuracy. EfficientNetB1 strikes a balance with moderate storage and inference time while providing reasonable accuracy. DenseNet121 delivers good accuracy but comes with higher storage and inference time compared to MobileNetV2 and EfficientNetB1. InceptionV3 and ResNet50 achieve the highest accuracy. However, their significantly higher storage demands make them less ideal for RCE devices. In RCE scenario, EfficientNetB1 could be the best baseline model to use for developing a new DNN model because it strikes a balance between accuracy and storage and inference time; its reasonable storage requirement of 30.8 MB and its decent accuracy of 83.0% make it a suitable starting point for further optimization in RCE settings.

Figure 2: Ten randomly chosen images from the ImageNet testing dataset

It is especially recommended because it strikes a good balance without sacrificing too much in any one aspect, which is crucial for RCE devices that need to optimize across multiple constraints.

Table 3: Prediction information of bench mark DNNs models

Model	Storage (in MB)	Inference time (in ms)	Accuracy (%)
MobileNetV1	16.4	3.88	97.64
MobileNetV2	13.9	5.6	81.9
EfficientNetB1	30.8	14.2	83.0
DenseNet121	31.8	13.5	88.9
InceptionV3	91.8	11.5	91.6
ResNet50	98.2	8.1	96.2

Results and Discussion

Table 3 compares six different DNN models in terms of three crucial metrics: model storage size, inference time (speed), and accuracy. The focus is on identifying the most suitable baseline model for deployment in RCEs. In RCEs, devices have limited memory and computational power. Therefore, it is important to balance model storage requirements, prediction speed, and accuracy to achieve optimal performance. Model Storage refers to the memory space required by each model, measured in MB. A lower storage size is more suitable for devices in RCEs, which often have limited memory capacity. MobileNetV1, with a storage size of 16.4 MB, stands out for its highly compact nature, making it extremely efficient for RCEs with tight memory constraints. Even more compact is MobileNetV2 at 13.9 MB, further enhancing its suitability for environments with very limited storage space. On the other hand, EfficientNetB1 requires 30.8 MB of storage, making it less suitable for devices with restricted storage capacity. DenseNet121 comes in slightly larger at 31.8 MB, but remains manageable for RCEs with moderate memory availability. At the higher end, InceptionV3 requires 91.8 MB, making it less ideal for devices with very limited memory, while ResNet50 has the highest storage requirement at 98.2 MB, making it the least efficient option for devices with strict memory constraints. Inference time, measured in milliseconds (ms), refers to the time taken by a model to classify an image, with lower values indicating faster predictions, which is crucial for real-time applications in RCEs. Among the models compared, MobileNetV1 delivers the fastest inference time at 3.88 ms, making it ideal for real-time tasks on resource-limited devices. MobileNetV2, while slightly slower with an inference time of 5.6 ms, still offers sufficiently fast performance for real-time classification. In contrast, EfficientNetB1 requires a significantly longer inference time of 14.2 ms, which may not be suitable for real-time applications. DenseNet121, with a time of 13.5 ms, provides moderate speed, slower than MobileNetV1 but faster than EfficientNetB1. InceptionV3, despite its large size, offers a better inference time of 11.5 ms, quicker than DenseNet121 and EfficientNetB1 but still slower than MobileNetV1. Lastly, ResNet50, with a time of 8.1 ms, strikes a balance between inference speed and model size, slower than MobileNetV1 but faster than most other models. Accuracy is a crucial metric in evaluating how well a model performs in image classification tasks, especially in RCEs. MobileNetV1, with an accuracy of 97.64%, stands out as a top choice due to its high accuracy coupled with its small size and rapid inference, making it ideal for devices with limited resources. In contrast, MobileNetV2 suffers a notable drop in accuracy to 81.9%, which could limit its use in scenarios requiring high classification performance. EfficientNetB1 improves slightly over MobileNetV2 with an accuracy of 83.0%, but still lags behind MobileNetV1, despite being larger and slower. DenseNet121, with 88.9% accuracy, provides a significant improvement over MobileNetV2 and EfficientNetB1, making it a solid compromise between accuracy and efficiency for RCEs with moderate storage capacity. InceptionV3 boasts an impressive 91.6% accuracy but its large storage requirements may pose challenges for deployment on memory-constrained devices. ResNet50, with an accuracy of 96.2%, offers

excellent performance second only to MobileNetV1. However, its large size could limit its practicality in environments where memory is a major concern.

Conclusion

With an emphasis on image classification applications, this paper offers a thorough examination and empirical study of the difficulties and solutions associated with implementing DNNs in resource limited scenario. RCE devices have a lot of potential for DNN usage, but they have special difficulties because of their constrained resources. The article systematically examines these challenges and offers insights into neural network models tailored to RCEs. It focused on analyses of DNN models, including MobileNetV1, MobileNetV2, InceptionV3, DenseNet121, EfficientNetB1, and ResNet501. The models' applicability in resource constraint scenarios was examined by considering variables including accuracy, computational efficiency, and model size. Based on this analysis, MobileNetV1 emerges as the most suitable model for RCEs, as it provides an excellent balance of compact storage, fast inference time, and high accuracy. While models like MobileNetV2 and EfficientNetB1 may offer more compact solutions, they come with trade-offs in accuracy and speed. Models like DenseNet121, InceptionV3, and ResNet50 provide better accuracy but at the cost of larger storage requirements, which might not be feasible for deployment in many RCE. The proposed study emphasizes the significance of improving DNNs for resource limited settings, in line with the research topic, and provides a methodical examination of the models that can be used to accomplish this. It emphasizes how important it is to carefully choose and include DNN models and methodologies based on the particular requirements of the deployment environment. Reviewing and evaluating models, as well as examining the difficulties in utilizing DNNs in RCE devices, yielded significant insights. It provides academics and practitioners with a helpful road map for developing effective DNNs in image classification for resource-constrained scenarios and suggests using EfficientNetB1 being a baseline model for future

study. Although this study obtained encouraging results, it is important to acknowledge certain limitations. In particular, we utilized only six state-of-the-art benchmark models to compare and identify the baseline model for developing an efficiency-focused DNN. While these models are well-established in the literature, many other benchmark models, such as SqeezNet and Xception, were not included in this study. Future research should broaden the scope of model comparison by incorporating a wider range of benchmark models and evaluating them using comprehensive metrics. This approach will ensure a more robust selection of the baseline model. Building on these findings, the next step in this research involves developing an optimized DNN model that will utilize the identified baseline model, MobileNetV1. This new model will aim to balance accuracy and efficiency, catering specifically to RCEs. The performance of the proposed model will be evaluated against existing benchmark models to assess its effectiveness.

ORCID

Raafi Careem
https://orcid.org/0009-0001-8632-4019
Gapar Md Johar
https://orcid.org/0000-0001-7986-3834
Ali Khatibi
https://orcid.org/0000-0002-2531-7720

Reference

[1]. X. Zhao, Research and application of deep learning in image recognition, *Journal of Physics: Conference Series, IOP Publishing*, 2023, 012047. [Crossref], [Google Scholar], [Publisher]

[2]. J.R. Leow, W.H. Khoh, Y.H. Pang, H.Y. Yap, Breast cancer classification with histopathological image based on machine learning, *International Journal of Electrical & Computer Engineering (2088-8708)*, **2023**, *13*. [Crossref], [Google Scholar], [Publisher]

[3]. R.M. Jasim, T.S. Atia, Towards classification of images by using block-based CNN, *Bulletin of Electrical Engineering and Informatics*, **2023**, *12*, 373-379. [Crossref], [Google Scholar], [Publisher]

[4]. M. Shafiq, Z. Gu, Deep residual learning for image recognition: A survey, *Applied Sciences*, **2022**, *12*, 8972. [Crossref], [Google Scholar], [Publisher]

[5]. S.M. Mahmoud, H.A. Al-Jubouri, T.E. Abdoulabbas, Chest radiographs images retrieval using deep learning networks, *Bulletin of Electrical Engineering and Informatics*, **2022**, *11*, 1358-1369. [Crossref], [Google Scholar], [Publisher]

[6]. Y. Li, Research and application of deep learning in image recognition, 2022 IEEE 2nd international conference on power, electronics and computer applications (ICPECA), *IEEE*, **2022**, 994-999. [Crossref], [Google Scholar], [Publisher]

[7]. M. Kim, Y. Kwon, J. Kim, Y. Kim, Image classification of parcel boxes under the underground logistics system using CNN MobileNet, *Applied Sciences*, **2022**, *12*, 3337. [Crossref], [Google Scholar], [Publisher]

[8]. A.R. Choudhuri, B.G. Thakurata, B. Debnath, D. Ghosh, H. Maity, N. Chattopadhyay, R. Chakraborty, MNIST image classification using convolutional neural networks, *Modeling, Simulation and Optimization*, **2022**, 255-266. [Crossref], [Google Scholar], [Publisher]

[9]. Ò. Lorente, I. Riera, A. Rana, Image classification with classic and deep learning techniques, *Computer Vision and Pattern Recognition*, **2021**. [Crossref], [Google Scholar], [Publisher]

[10]. D.W. Otter, J.R. Medina, J.K. Kalita, A survey of the usages of deep learning for natural language processing, *IEEE Transactions on Neural Networks and Learning Systems*, **2020**, *32*, 604-624. [Crossref], [Google Scholar], [Publisher]

[11]. 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)*, **2022**, *12*. [Crossref], [Google Scholar], [Publisher]

[12]. S. Kuutti, R. Bowden, Y. Jin, P. Barber, S. Fallah, A survey of deep learning applications to autonomous vehicle control, *IEEE Transactions on Intelligent Transportation Systems*, **2020**, *22*, 712-733. [Crossref], [Google Scholar], [Publisher]

[13]. R. Careem, G. Johar, A. Khatibi, Deep neural networks optimization for resource-constrained environments: techniques and models, *Indonesian Journal of Electrical Engineering and Computer Science,* **2024**, *33*, 1843-1854. [Crossref], [Google Scholar], [Publisher]

[14]. J. Park, P. Aryal, S.R. Mandumula, R.P. Asolkar, An optimized dnn model for real-time inferencing on an embedded device, *Sensors*, **2023**, *23*, 3992. [Crossref], [Google Scholar], [Publisher]

[15]. Y. Gulzar, Fruit image classification model based on MobileNetV2 with deep transfer learning technique, *Sustainability*, **2023**, *15*, 1906. [Crossref], [Google Scholar], [Publisher]

[16]. L. Zhao, L. Wang, A new lightweight network based on MobileNetV3, *KSII Transactions on Internet and Information Systems (TIIS)*, **2022**, *16*, 1-15. [Crossref], [Google Scholar], [Publisher]

[17]. W. Wang, Y. Li, T. Zou, X. Wang, J. You, Y. Luo, A novel image classification approach via dense-MobileNet models, *Mobile Information Systems*, **2020**, *2020*, 7602384. [Crossref], [Google Scholar], [Publisher]

[18]. A. Pujara, Image classification with mobilenet, *Published in Analytics Vidhya,* **2020**. [Google Scholar], [Publisher]

[19]. A.R. Luaibi, T.M. Salman, A.H. Miry, Detection of citrus leaf diseases using a deep learning technique, *International Journal of Electrical and Computer Engineering*, **2021**, *11*, 1719. [Crossref], [Google Scholar], [Publisher]

[20]. X. Wan, F. Ren, D. Yong, Using inception-resnet v2 for face-based age recognition in scenic spots, *2019 IEEE 6th International Conference on Cloud Computing and Intelligence Systems (CCIS), IEEE*, **2019**, 159-163. [Crossref], [Google Scholar], [Publisher]

[21]. C. Szegedy, S. Ioffe, V. Vanhoucke, A. Alemi, Inception-v4, inception-resnet and the impact of residual connections on learning, *Proceedings of the AAAI Conference on Artificial Intelligence*, **2017**. [Crossref], [Google Scholar], [Publisher]

[22]. A.E. Minarno, L. Aripa, Y. Azhar, Y. Munarko, Classification of malaria cell image using inception-v3 architecture, *JOIV: International Journal on Informatics Visualization*, **2023**, *7*, 273-278. [Crossref], [Google Scholar], [Publisher]

[23]. S.A. Albelwi, Deep architecture based on DenseNet-121 model for weather image recognition, *International Journal of Advanced Computer Science and Applications*, **2022**, *13*. [Crossref], [Google Scholar], [Publisher]

[24]. S. Benkrama, N.E.H. Hemdani, Deep learning with efficientnetB1 for detecting brain tumors in mri images, ***2023 International Conference on Advances in Electronics, Control and Communication Systems (ICAECCS), IEEE***, **2023**, 1-6. [Google Scholar], [Publisher]

[25]. M. Tan, Q. Le, Efficientnet: Rethinking model scaling for convolutional neural networks, *International Conference on Machine Learning, PMLR*, **2019**, 6105-6114. [Google Scholar], [Publisher]

[26]. R. Careem, M. Gapar Md Johar, A. Khatibi, Baseline model for deep neural networks in resource-constrained environments: an empirical investigation, *International Journal of Computing and Digital Systems*, **2024**, *16*, 1147-1161. [Crossref], [Google Scholar], [Publisher]

[27]. M. Wei, Q. Wu, H. Ji, J. Wang, T. Lyu, J. Liu, L. Zhao, A skin disease classification model based on densenet and convnext fusion, *Electronics*, **2023**, *12*, 438. [Crossref], [Google Scholar], [Publisher]

[28]. R. Careem, G. Johar, A. Khatibi. Deep neural networks optimization for resource-constrained environments: Techniques and models, *Indonesian Journal of Electrical Engineering and Computer Science*, **2024**, *33*, 1843-1854. [Google Scholar]