EAI Endorsed Transactions
on Industrial Networks and Intelligent Systems
Research Article EAI.EU

Vehicle Type Classification with Small Dataset and Transfer Learning Techniques

Quang-Tu Pham[1], Dinh-Dat Pham[1], Khanh-Ly Can[1,2], To-Hieu Dao[1,2,3], Hoang-Dieu Vu[1,2,3,+]

[1]Faculty of Electrical and Electronic Engineering, Phenikaa University, Yen Nghia, Hanoi, 12116, Vietnam
[2]PHENIKAA Research and Technology Institute (PRATI), A&A Green Phoenix Group JSC, No.167 Hoang Ngan,Trung Hoa, Cau Giay, Hanoi, 11313, Vietnam
[3]Graduate University of Sciences and Technology, Vietnam Academy of Science and Technology, Hanoi, Vietnam

Abstract

This study delves into the application of deep learning training techniques using a restricted dataset, encompassing around 400 vehicle images sourced from Kaggle. Faced with the challenges of limited data, the impracticality of training models from scratch becomes apparent, advocating instead for the utilization of pre-trained models with pre-trained weights. The investigation considers three prominent models—EfficientNetB0, ResNetB0, and MobileNetV2—with EfficientNetB0 emerging as the most proficient choice. Employing the gradually unfreeze layer technique over a specified number of epochs, EfficientNetB0 exhibits remarkable accuracy, reaching 99.5% on the training dataset and 97% on the validation dataset. In contrast, training models from scratch results in notably lower accuracy. In this context, knowledge distillation proves pivotal, overcoming this limitation and significantly improving accuracy from 29.5% in training and 20.5% in validation to 54% and 45%, respectively. This study uniquely contributes by exploring transfer learning with gradually unfreeze layers and elucidates the potential of knowledge distillation. It highlights their effectiveness in robustly enhancing model performance under data scarcity, thus addressing challenges associated with training deep learning models on limited datasets. The findings underscore the practical significance of these techniques in achieving superior results when confronted with data constraints in real-world scenarios.

Received on 27 December 2023; accepted on 02 March 2024; published on 07 March 2024

Keywords: Small Dataset, Deep Learning, Transfer Learning, Knowledge Distillation

doi:10.4108/eetinis.v11i2.4678

1 Introduction

[1] Deep learning and Convolutional Neural Networks (CNN) have demonstrated remarkable efficacy in vehicle classification, as substantiated by numerous studies [1], [2], [3], [4]. Nevertheless, their inherent effectiveness often relies on a crucial factor—an extensive dataset for training robust classification models. This requisite becomes particularly challenging in scenarios where data availability is limited, a circumstance addressed in this study.

The present investigation focuses on vehicle classification, utilizing a meticulously curated dataset comprising 400 images sourced from Kaggle. Figure 1 illustrates representative samples extracted from the dataset. The dataset is carefully structured, featuring four distinct classes—truck, bus, car, and motorcycle—each containing precisely 100 images. This deliberate curation introduces a unique challenge due to the restricted data available for each class. To ensure a well-balanced representation within the limited dataset, a thoughtful partitioning is employed, creating distinct training and validation subsets, both maintaining an even 50:50 ratio.

This emphasis on a balanced representation is pivotal, especially when dealing with a limited dataset, as it mitigates biases that might emerge if one class is overrepresented. The challenge of achieving accurate and robust vehicle classification in the face of such constraints underscores the importance of innovative methodologies, such as transfer learning and knowledge distillation, which are explored in this study to enhance model performance under data scarcity.

[1] + Corresponding author: Hoang-Dieu Vu (email: dieu.vuhoang@phenikaa-uni.edu.vn

Many studies [5] [6] have advocated for transfer learning as a common and effective choice to bolster accuracy in such constrained scenarios. The study [6] further highlights that the similarity of features between the source and target domains significantly influences the performance of transfer learning. This is particularly pertinent in the current context, given that ImageNet [7] includes vehicle images, aligning with the nature of the problem under investigation. Furthermore, employing advanced training techniques, such as fine-tuning all layers with the same learning rate and gradual unfreezing with three layer groups and discriminative learning rates, can provide additional enhancements to model performance.

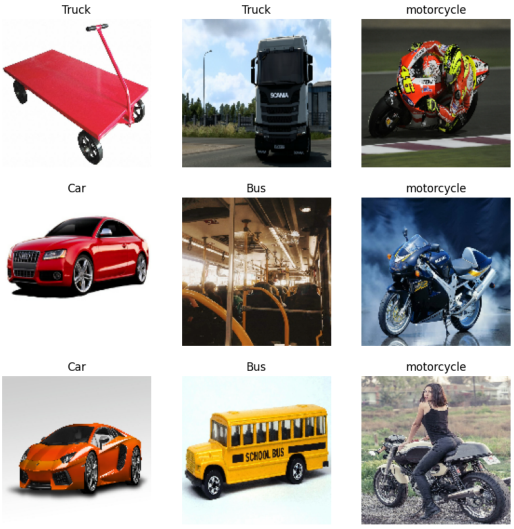


Figure 1. Sample Images from the Dataset

In light of the aforementioned considerations, the present study adopts a strategic approach, incorporating transfer learning by employing pre-trained models with pre-existing weights derived from the expansive ImageNet dataset—an extensive repository comprising millions of images across thousands of categories. This approach enables the network to capitalize on the general features acquired during its initial training on this vast dataset. And among the models under consideration—EfficientNetB0, ResNetB0, and MobileNetV2—EfficientNetB0 emerges as the most adept choice, especially when addressing the complexities associated with a dataset featuring both clear and unclear views of objects.

Furthermore, within the framework of a highly limited dataset as in this study, adhering to the recommendation proposed by [6] to fine-tune all layers with the same learning rate may not yield optimal results. Conversely, gradual unfreezing proves to be a promising methodology, showcasing superior performance in enhancing model accuracy.

Moving beyond conventional training methodologies, the investigation explores the transformative potential of knowledge distillation as a pivotal remedial measure, overcoming inherent limitations in training models from scratch. This approach notably boosts accuracy in both training and validation sets, demonstrating a substantial improvement from initial levels. Furthermore, this principle extends to scenarios involving fine-tuning all layers. While not deemed an optimized solution for our constrained dataset, knowledge distillation emerges as a viable alternative, yielding commendable enhancements. This is evident in a significant increase in accuracy, further validating the efficacy of knowledge distillation across different training scenarios.

Our contributions to this research encompass the following key aspects:

- The benefits of this work for vehicle classification are evident in its ability to address the challenges posed by limited datasets. By introducing strategic methodologies tailored to data scarcity, such as transfer learning, gradual unfreezing, and knowledge distillation, this study aims to advance the field of vehicle classification, offering practical solutions for real-world applications where data constraints are prevalent.

- The exploration of training techniques, specifically the application of transfer learning with gradually unfreeze layers and knowledge distillation. Through this exploration, we aim not only to discern the potential of these techniques in optimizing model performance within the constraints of a small dataset but also to conduct a comprehensive performance comparison among models. This investigation contributes valuable insights into effective strategies for enhancing model efficacy under conditions of data scarcity, providing a nuanced understanding of training methodologies in deep learning and facilitating informed model selection.

- The study introduces a novel approach by employing the gradually unfreeze layer technique over a specified number of epochs, resulting in EfficientNetB0 achieving remarkable accuracy, reaching 99.5% on the training dataset and 97% on the validation dataset. In contrast, training models from scratch yields notably lower accuracy. Furthermore, the study highlights the pivotal role of knowledge distillation, overcoming this limitation and significantly improving accuracy from 29.5% in training and 20.5% in validation to 54% and 45%, respectively.

2. Methodology

2.1. Description of proposed solution/design

In the pursuit of addressing the challenge of vehicle type classification within the domain of computer vision, three neural network architectures have been selected for examination: MobileNetV2, EfficientNetB0 and ResNet50. These architectures hold significance in the field of computer vision due to their distinct features and design principles. These models are imported from the TensorFlow framework and are subjected to specific design considerations. Furthermore, the research encompasses an exploration of the potential of knowledge distillation, with specific emphasis on the distillation of knowledge from ResNet50 to EfficientNetB0. This additional approach is undertaken to compare its performance with the standalone models and to provide a comprehensive evaluation of the impact of knowledge distillation within the context of limited data availability.

MobileNetV2 [8]:. MobileNetV2 is a lightweight and efficient convolutional neural network (CNN) architecture designed for mobile and edge devices. Introduced as an improvement over its predecessor, MobileNetV1 [9], MobileNetV2 incorporates several key features to enhance its performance while maintaining a low computational footprint.

The original paper on MobileNetV2 [8] does not provide a formal architectural graph of the model; however, in a related study [10], a comprehensive architectural graph of the model has been made available.

The architecture of MobileNetV2 is characterized by a streamlined design that leverages inverted residuals and linear bottlenecks. Inverted residuals involve the use of lightweight depthwise separable convolutions followed by linear bottlenecks, which help reduce the computational cost of the network. This design choice contributes to the model's efficiency by minimizing the number of parameters and computations.

The core building block of MobileNetV2 is the inverted residual with linear bottleneck. It consists of a sequence of operations, including a lightweight depthwise separable convolution, a 1x1 convolution for feature integration, and linear bottlenecks to maintain representational capacity. The linear bottlenecks utilize shortcut connections to enable information flow across the network and mitigate the risk of information loss during the depthwise separable convolution operations.

MobileNetV2 also introduces a novel feature called "linear bottleneck residual connection," which facilitates the direct flow of information between layers. This connection aids in preserving crucial features and gradients throughout the network, contributing to better training and performance.

The architecture incorporates a global depthwise separable convolutional layer at the end of the network to capture contextual information globally, enhancing the model's ability to understand relationships between different features. Additionally, MobileNetV2 employs a width multiplier and a resolution multiplier as hyperparameters, allowing users to balance the trade-off between computational efficiency and model accuracy.

Overall, MobileNetV2 stands out for its efficiency and effectiveness in resource-constrained environments. Its architectural innovations, including inverted residuals, linear bottlenecks, and global depthwise separable convolutions, make it a compelling choice for applications on mobile devices and edge computing platforms where computational resources are limited.. In this study, the base model of MobileNetV2 is imported, excluding the top layer, to leverage its inherent capabilities for vehicle type classification.

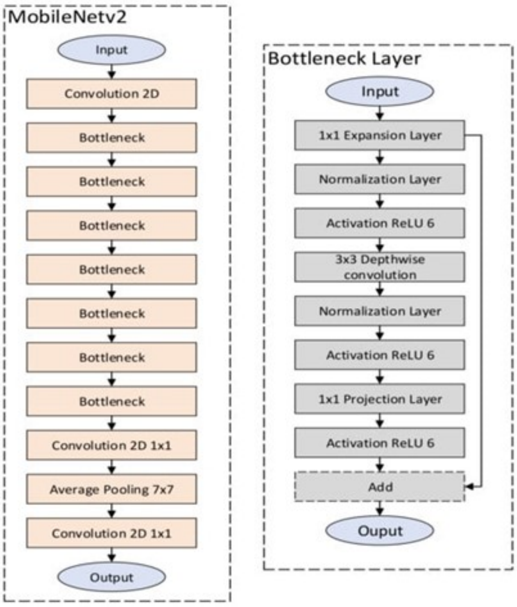


Figure 2. MobileNetV2 Architecture [10]

EfficientNetB0 [11]:. EfficientNetB0 is part of an efficient convolutional neural network (CNN) family designed with a focus on achieving high performance while optimizing computational efficiency. The architecture features systematic scaling of depth, width, and resolution. Compound scaling is employed, ensuring a balanced expansion of model capacity across dimensions.

Similar to MobileNetV2, the original publication for EfficientNetB0 [11] did not include a specific graphical representation of its architecture. Nonetheless, a related study [12] has released a readily accessible architectural diagram.

The baseline architecture starts with a 3x3 convolutional layer for initial feature extraction. Repeating blocks follow, consisting of depthwise separable convolutions and inverted residuals similar to MobileNetV2. These blocks enable efficient information processing while maintaining model expressiveness.

Depthwise separable convolutions, a fundamental building block, separate spatial and channel-wise convolutions, reducing parameters and computations for increased efficiency. Inverted residuals with linear bottlenecks aid in feature propagation within the network.

EfficientNetB0 integrates a Feature Pyramid Network for efficient multi-scale feature capture, enhancing the model's ability to understand both local and global context. Squeeze-and-Excitation (SE) blocks recalibrate channel-wise feature responses, focusing on informative channels for improved discriminative power.

The architecture concludes with global average pooling and fully connected layers for final classification. Global average pooling aggregates feature information globally, reducing spatial dimensions for increased robustness to input variations.

EfficientNetB0's systematic scaling, efficient building blocks, and incorporation of mechanisms like SE blocks and Feature Pyramid Network contribute to its effectiveness. Its ability to balance model size and computational efficiency makes it widely adopted in resource-constrained environments, such as mobile and edge devices. In the context of this study, the base model of EfficientNetB0 will be integrated, excluding the top layer, to harness its scalability and efficiency for vehicle type classification.

ResNet50 [13]. ResNet50 is a renowned deep convolutional neural network architecture that addresses challenges associated with training very deep networks. The architecture introduces residual blocks, featuring skip connections to facilitate the learning of residuals or differences between input and output. This innovation alleviates the vanishing gradient problem, enabling the training of deep networks effectively.

The original documentation for ResNet50 [13] did not provide a detailed architectural diagram either. Consequently, we are utilizing a diagram from a study conducted by Hindawi [14] to illustrate the model structure.

ResNet50's architecture involves stacking multiple residual blocks, resulting in a total of 50 layers. The bottleneck architecture within each residual block incorporates 1x1, 3x3, and 1x1 convolutions. This design reduces the computational load while preserving crucial features.

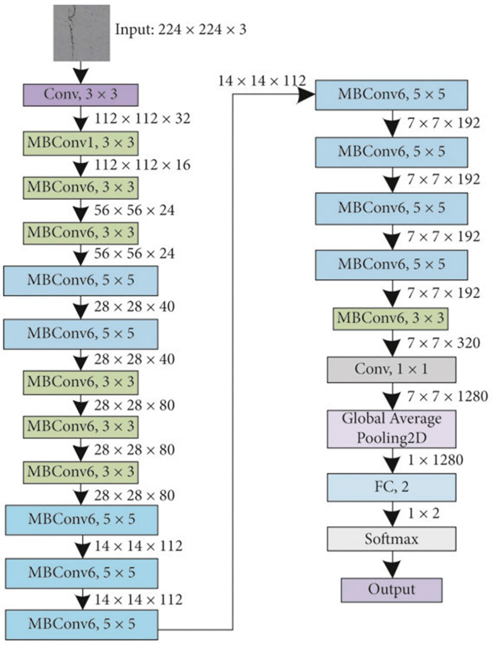


Figure 3. EfficientNetB0 Architecture [12]

Skip connections play a vital role in ResNet50 by allowing the direct flow of information between layers. These connections mitigate the vanishing gradient problem during backpropagation, enhancing the efficiency of learning.

The architecture includes a global average pooling layer towards the end, which computes the average of each feature map, reducing spatial dimensions before the final fully connected layer for classification.

ResNet50's structural design, particularly the use of residual blocks and skip connections, has made it a benchmark for training deep neural networks. Its effectiveness in image classification and other computer vision tasks, along with its application in transfer learning with pre-trained models on datasets like ImageNet, contributes to its widespread adoption in various domains.

Knowledge Distillation [15]. Knowledge Distillation is a model compression technique involving a teacher model, soft targets, and a student model. The teacher model, a well-trained and complex neural network, generates soft targets—probability distributions over classes—instead of hard labels. The student model, smaller and less complex, aims to replicate the knowledge embedded in the teacher. The training process minimizes the difference between the student's predictions and the soft targets, encouraging the student to capture nuanced knowledge.

A temperature parameter scales the soft targets, influencing the level of confidence in predictions.

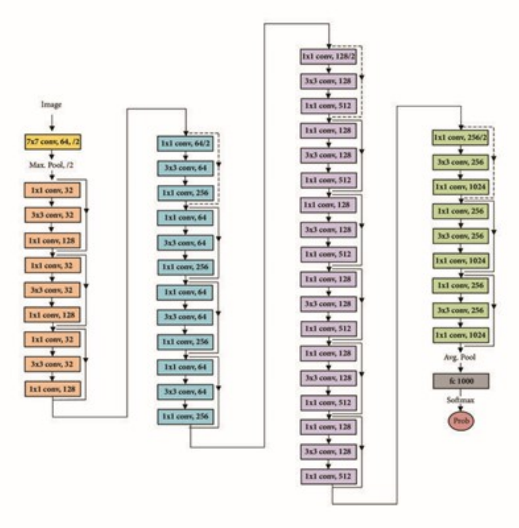


Figure 4. ResNet50 Architecture [14]

Higher temperatures yield softer probability distributions, allowing the student to explore a broader solution space during training.

The loss function in training includes traditional hard label cross-entropy loss and knowledge distillation loss, penalizing deviations from the soft targets provided by the teacher. This encourages the student to mimic the teacher's behavior.

Optionally, the trained student model may undergo fine-tuning on the original training data without the teacher's guidance, refining the model and adapting it to specific dataset characteristics.

Knowledge Distillation, with its focus on transferring rich knowledge from a complex teacher to a simpler student, is valuable for compressing models without compromising performance. This makes it suitable for deployment on edge devices and in scenarios with limited computational resources.

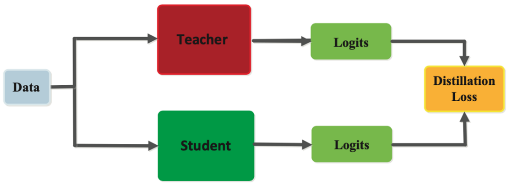


Figure 5. Response-based knowledge distillation [16]

In this study, the application of Knowledge Distillation proves successful in compressing ResNet50 into smaller models like MobileNetV2 and EfficientNetB0. ResNet50, with its complexity, may be impractical for resource-constrained devices. Knowledge Distillation transfers its knowledge to lightweight models, ensuring a balance between size and performance. The chosen student models, MobileNetV2 and EfficientNetB0, designed for efficiency, inherit ResNet50's knowledge, tailored to match their requirements. The process includes learning from the original data and soft targets from ResNet50. Fine-tuning adapts the student models to the target dataset, maintaining high performance in a computationally efficient manner. Knowledge Distillation proves versatile, optimizing neural networks for real-world applications with limited resources.

Gradually Unfreeze Layers. In the fine-tuning phase of our deep learning approach, we employ a carefully structured strategy known as 'gradually unfreeze layers.' Following an initial pre-training on a large-scale dataset, where the model acquires general features, all layers are frozen to preserve the knowledge gained. The subsequent unfreezing process is conducted incrementally, starting from the last layers and progressing towards the initial layers. This stepwise unfreezing strategy is designed to exploit the high-level representations encoded in deeper layers while preventing overfitting, particularly when dealing with smaller, more specialized datasets.

The rationale behind the gradual unfreezing methodology is rooted in the balance between leveraging the richness of information in deep layers and maintaining model generalization. Unfreezing too many layers at once may lead to overfitting, making the model less adaptable to the nuances of the target dataset. The gradual release of constraints allows the model to fine-tune its learned features judiciously, adapting to the intricacies of the specific task. This approach aligns with our empirical observations, where the controlled adaptation of layers has proven effective in enhancing model performance while preserving its ability to generalize effectively across diverse datasets. The details of this methodology are meticulously outlined in the subsequent sections, providing clarity on our experimental setup and guiding reproducibility in future studies.

2.2. Solution limitations

Acknowledging the limitations posed by the dataset's size, comprising approximately 400 images, the study anticipates potential challenges related to model performance. Given the predominance of augmented data within the training set, the study recognizes the possibility of models exhibiting tendencies toward overfitting or underfitting.

Additionally, it is essential to underscore the significance of parameter tuning as a critical component in addressing these limitations. The study will place particular emphasis on the systematic fine-tuning of model parameters and hyperparameters. This iterative process is expected to facilitate the optimization of model complexity, thereby enhancing the model's ability to generalize effectively while minimizing the risk of overfitting or underfitting.

2.3. Design approach and implementation methodology

In the pursuit of this research's design approach, pre-trained models MobileNetV2, ResNet50, and EfficientNetB0 are reconstructed via the TensorFlow framework, followed by tailored adaptations for the vehicle type classification task. The fundamental implementation steps encompass feature extraction, model customization, and the subsequent training phase. To address the potential of overfitting, a proportion of the dataset (50%) is allocated for testing purposes, with flexibility for potential adjustments to optimize validation accuracy. This allocation exhibits flexibility in accommodating potential refinements for the optimization of validation accuracy.

Knowledge distillation was also incorporated into the research, with a specific focus on distilling knowledge from ResNet50 to EfficientNetB0. This knowledge distillation approach is intended not for performance enhancement but rather for comparative analysis against other fine-tuned models. The iterative and systematic nature of this approach emphasizes the research's dedication to establishing a solution that is resilient and dependable, especially when operating within the constraints imposed by a limited dataset.

3. Results and Discussion

3.1. Experimental and Result

This section delineates the experimental design and methodology employed to scrutinize various training techniques within the constraints of limited datasets. Table 1 presents a detailed overview of the utilized training methods, encompassing base, pre-trained, fine-tuned, and distillation-based approaches. The primary focus is on MobileNetV2 and EfficientNetB0, each subjected to distinctive training strategies.

Experimental Framework

- **Base training:** Models are trained from scratch without pre-existing weights, serving as the baseline for comparison.

- **Pre-trained training:** Models are initialized with pre-trained weights from the ImageNet dataset, freezing all layers except the final classification layer during the initial training.

- **Fine-tuned training:** The last 10 layers of the model are unfrozen after 40 epochs to refine learned features based on dataset characteristics.

- **Fine-tuned all layers training:** A variant of the aforementioned fine-tuned training methodology involves the comprehensive unfreezing of all layers as opposed to only the last 10 layers.

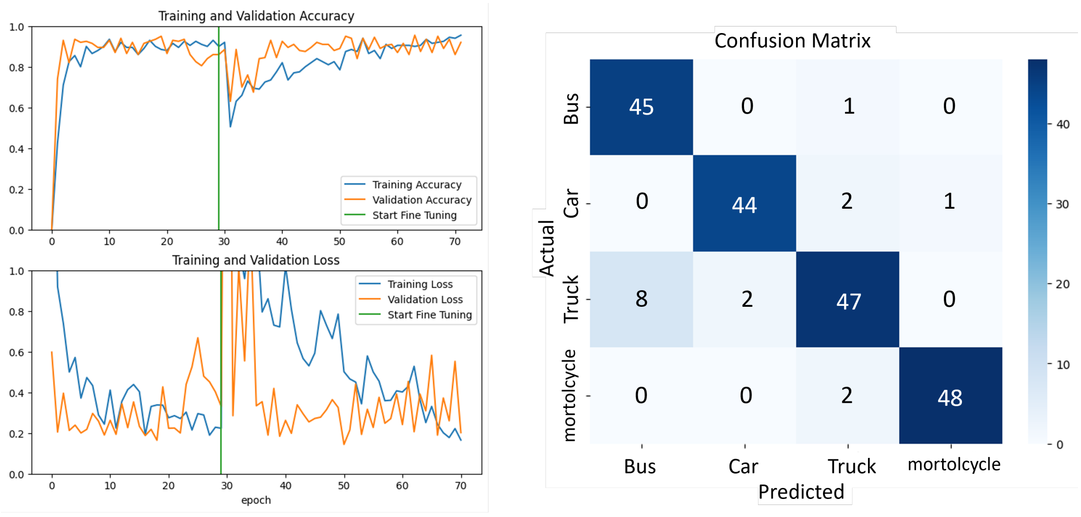
- **Knowledge Distillation** Knowledge distillation transfers knowledge from the larger pre-trained ResNet50 to MobileNetV2 and EfficientNetB0. In this investigation, the distillation process underwent multiple refinements. Distillation-base involves building models without pre-trained weights and subsequently applying distillation. Distillation1 and Distillation2 are variations applied to pre-trained models, each with distinct parameter tweaks—Distillation1 with a temperature of 10 and alpha of 0.01, while Distillation2 adopts a temperature of 5 and alpha of 0.05. Notably, in knowledge distillation, all layers are unfrozen to enhance adaptability.

- **Gradually Unfreeze** A systematic unfreezing of the last 10 layers was implemented. Following an initial 20 epochs of pre-training with all layers frozen for stability, a strategic unfreezing schedule was adopted. Specifically, 2 layers were unfrozen after the 20th epoch, followed by an additional 3 layers every 40 epochs, culminating in the gradual release of the final 2 layers after the subsequent 40 epochs.

Table 1. Models' performance comparision

Models		Accuracy	
		Train	Validation
MobileNetV2	Pre-trained	92.00%	88.50%
	Fine-tuned	95.50%	92.00%
	Distillation1	90.00%	32.00%
	Distillation2	91.00%	52.50%
ResNet50	Pre-trained	90.42%	91.25%
	Fine-tuned	96.25%	72.50%
EfficientNetB0	Base	29.50%	20.50%
	Distillation-base	61.00%	42.00%
	Pre-trained	96.50%	91.50%
	Fine-tuned	99.00%	95.50%
	Fine-tuned all layers	54.00%	45.50%
	Distillation1	99.50%	76.00%
	Distillation2	93.50%	82.00%

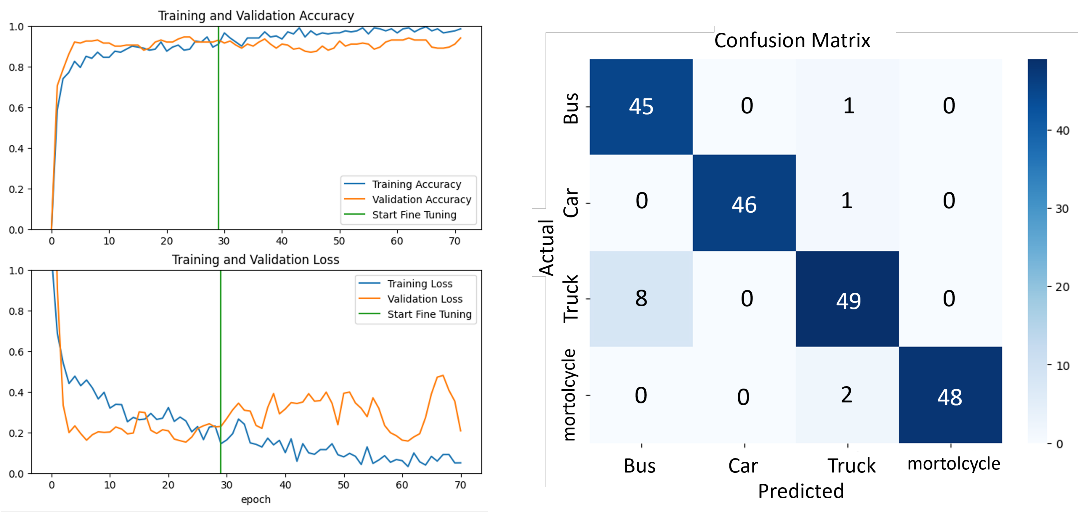
Performance of EfficientNetB0 and ResNet50. In the comprehensive analysis of the results, it becomes evident that EfficientNetB0 exhibits superior performance, yielding training and validation dataset accuracies of 99% and 95.5%, respectively. In contrast, for intricate models such as ResNet50, preserving the frozen state of layers contributes to enhanced data generalization, consequently leading to improved validation accuracy. This persistent trend endures despite the recognition that unfreezing layers may afford superior performance



a) Training performance

b) Confusion Matrix

Figure 6. MobileNetV2 Performance



a) Training performance

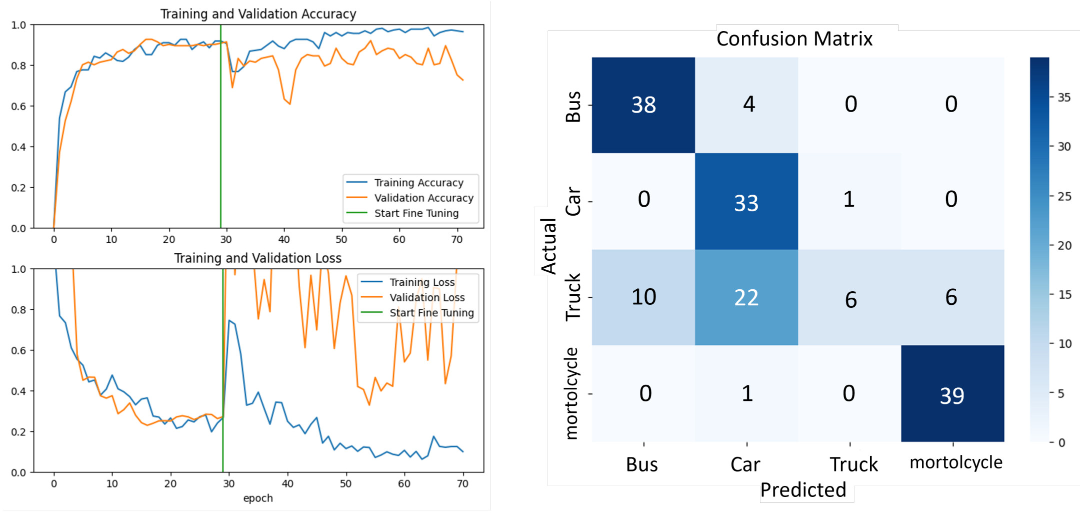
b) Confusion Matrix

Figure 7. EfficientNetB0 Performance

on the training dataset, albeit at the cost of a substantial decline in accuracy on the validation dataset.

Knowledge Distillation Strategies. The efficacy of ResNet50 as a source for knowledge distillation to models like EfficientNetB0 is grounded in its inherent complexity and capacity to capture intricate features. EfficientNetB0 and ResNet50 share a similar structure, both employing residual connections, which facilitates the knowledge transfer between the two models. As a deep and well-established architecture, ResNet50 possesses a wealth of learned knowledge from its extensive training on large-scale datasets, making it a potent teacher model. The transfer of knowledge from ResNet50 to more lightweight models like EfficientNetB0 stands to benefit from the comprehensive and nuanced features encapsulated within ResNet50's architecture. This strategic knowledge distillation process thus facilitates the enhancement of the targeted models' performance by leveraging the wealth of information encoded in ResNet50.

Conversely, MobileNetV2 exhibits a distinct structure compared to ResNet50 and EfficientNetB0. MobileNetV2 utilizes "inverted residuals" blocks instead of the "bottleneck" blocks found in ResNet50 and EfficientNetB0. This could create a context mismatch between the features of the teacher and student models, making knowledge transfer more



a) Training performance

b) Confusion Matrix

Figure 8. ResNet50 Performance

challenging. This results in favorable outcomes, akin to the effectiveness observed when transferring knowledge to EfficientNetB0. However, with MobileNetV2, the validation accuracy experiences a drastic decline, underscoring the challenges associated with knowledge distillation in models with divergent architectures.

Impact of Distillation Parameters. Distillation with a temperature of 5 and alpha of 0.05 consistently yields superior performance across experiments. These carefully chosen parameters, demonstrating their effectiveness, are adopted for subsequent experiments to ensure methodological consistency. The influence of distillation parameters proves to be a critical factor in achieving optimal results, emphasizing the need for meticulous parameter selection in distillation processes.

Enhancement through Distillation. Distillation emerges as a valuable enhancement strategy, significantly boosting accuracy in both training and validation datasets. The substantial improvement from 29% to 61% in training accuracy and from 20.5% to 42% in validation accuracy underlines the efficacy of knowledge distillation as a means of refining model performance. This finding underscores the potential of distillation to enhance model capabilities even in scenarios where fine-tuning all layers might lead to substantial accuracy degradation.

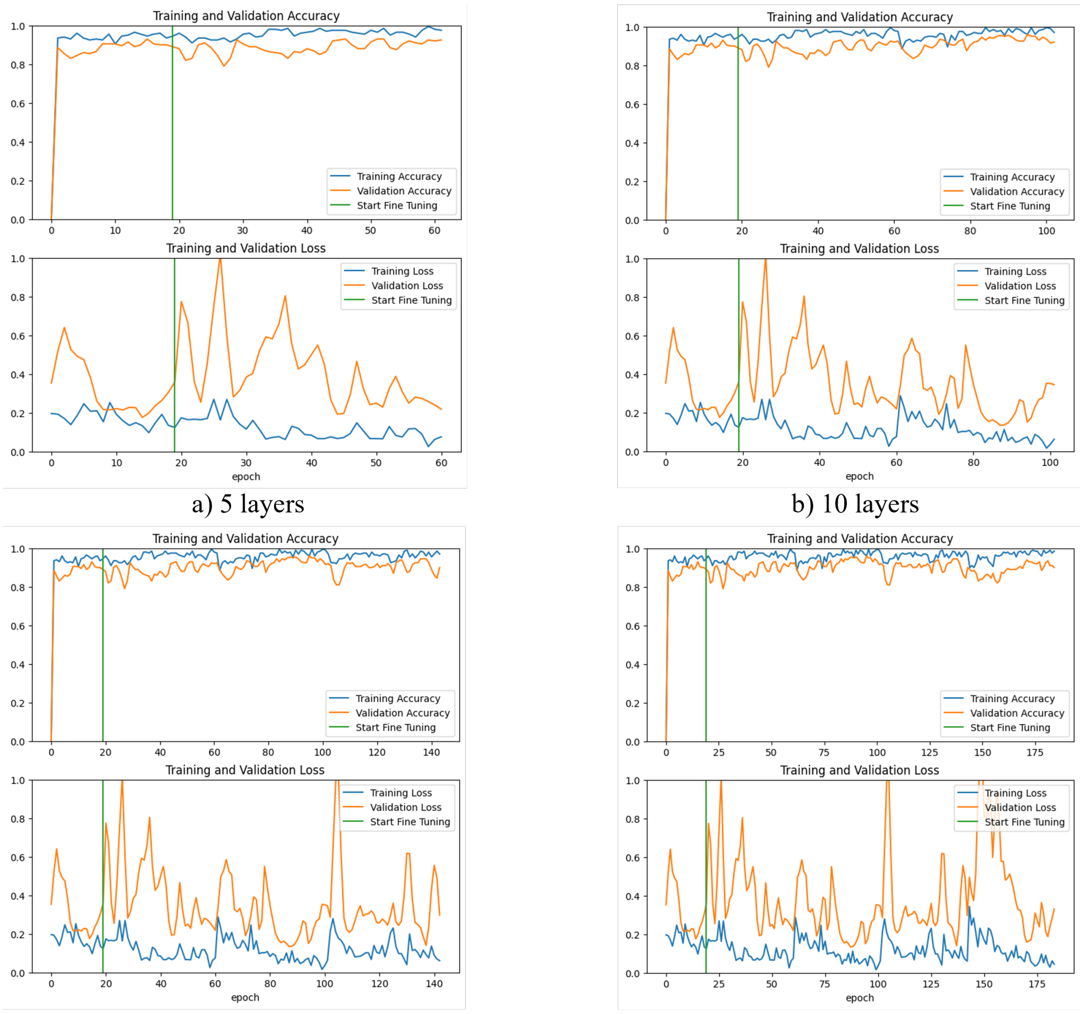
Fine-tuning Strategies. When all layers are unfrozen for fine-tuning, a drastic drop in accuracy is observed, declining from 96.5% and 95.5% in the training and validation datasets to 54% and 45.5%, respectively. However, distillation proves instrumental in mitigating this drop in accuracy, substantially improving it to 93.5% and 82%. While not reaching the levels of pre-trained models, this improvement suggests that with a slightly larger dataset, distillation could potentially outperform pre-trained frozen models. This observation underscores the potential robustness of distillation, particularly in scenarios where fine-tuning all layers might lead to significant accuracy degradation.

Given that EfficientNetB0, with transfer learning, has demonstrated the best performance among the models considered, further experiments will be conducted to explore various transfer learning parameters. Initial focus will be on investigating the impact of the number of layers to freeze after pre-training the model, with the subsequent results presented in Table 2.

Table 2. Number of layers unfrozen performance comparision

No of layers unfrozen	Train dataset	Validation Dataset
5 layers	98.50%	93.00%
10 layers	99.00%	95.5%
15 layers	97.00%	94.00%
20 layers	98.50%	94.50%
Random 10 layers	98.50%	94.50%

Optimal Layer Unfreezing. As indicated by the results in Table 2 and Figure 9, the model attains its peak performance when approximately 10 layers are unfrozen. Below this range, specifically at 5 layers unfrozen, or beyond, at 15 layers unfrozen, the model's accuracy experiences a slight decrease. This observation suggests that unfreezing 10 layers represents the optimal number of layers to unfreeze in this particular scenario, effectively preventing both underfitting and overfitting of the model. Subsequently, additional



a) 5 layers

b) 10 layers

c) 15 layers

d) 20 layers

Figure 9. Models' performance through number of unfreeze layers.

experiments were conducted with this optimal number of unfreezed layers. Specifically, a random selection of 10 layers was unfrozen, and a gradual unfreezing approach was employed. The corresponding numbers of unfreezed layers were systematically varied in the sequence of 2, 3, 3, and 2. The results of these experiments provide valuable insights into the nuanced impact of layer unfreezing on model performance and are presented for further analysis.

Random Selection of Unfrozen Layers. Table 2 also presents the outcomes of experiments involving the random selection of 10 layers for unfreezing. The results indicate a modest decline in performance compared to exclusively unfreezing the last 10 layers. After conducting this random selection process 10 times, the highest observed performance stands at 98.50% and 94.50% for the training and validation datasets, respectively. Notably, the majority of instances where the highest performance was achieved involved unfreezing layers from the last half of the model. This observation underscores a noteworthy trend – unfreezing layers from the latter portion of the model tends to yield more favorable results.

This trend aligns with the intrinsic characteristics of deep neural networks, where later layers often capture more abstract and complex features. Unfreezing layers towards the end of the model allows the network to fine-tune and adapt these high-level features to the specific task at hand. The substantial performance improvement in this scenario suggests that the latter layers encapsulate information crucial for the dataset under consideration.

Moreover, unfreezing only the last layers, as opposed to unfreezing too many layers, serves as a strategic approach to prevent the model from becoming overly complex and overfitting to the training data. This carefully maintained balance between model accuracy and generalization capability is instrumental in ensuring the model's robustness across diverse datasets. The empirical success of this strategy reinforces the importance of informed layer selection, as it not only enhances performance but also guards against potential overfitting issues.

Gradual Unfreezing Approach. In contrast, Table 3 and Figure 10 illustrates the results of experiments involving the gradual unfreezing of layers, with the total number summing up to ten (2, 3, 3, 2). Comparatively, this approach yields better performance than simply unfreezing the last 10 layers in a single step. The results depicted in Figure 10 show a slight drop in validation accuracy when unfreezing up to 5 layers, followed by a rapid ascent, surpassing the performance achieved by exclusively unfreezing the last 10 layers. Although there is a minor fluctuation when the number of unfreezed layers reaches 10, the gradual unfreezing strategy ultimately achieves a peak performance of 99.50% and 97.00% for the training and validation sets, respectively. This outcome slightly surpasses the performance achieved by simply unfreezing the last 10 layers in a single step. The observed fluctuation emphasizes the dynamic nature of the training process, while the eventual peak performance highlights the effectiveness of the gradual unfreezing strategy in optimizing model accuracy. This dynamic adaptation process contributes to mitigating issues of overfitting and further enhances the model's ability to yield superior results during training.

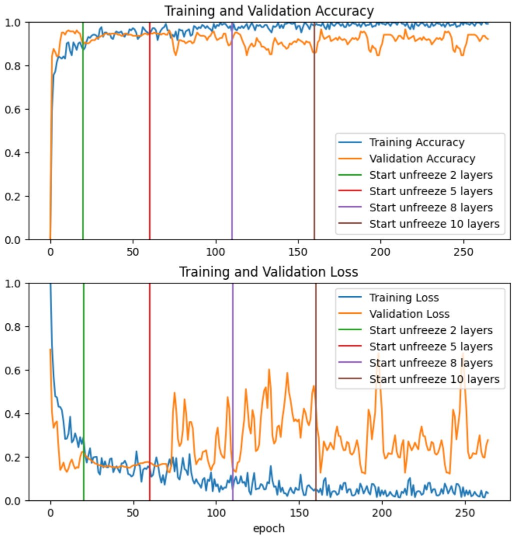


Figure 10. Gradually unfreezing layers performance

Table 3. Gradually unfreezing layers performance

Total layers unfrozen	Train dataset	Validation Dataset
2 layers	96.00%	94.50%
5 layers	97.00%	90.00%
8 layers	99.00%	93.50%
10 layers	99.50%	97.00%

3.2. Discussion

Results from the experimental evaluation underscore EfficientNetB0's consistent superiority over MobileNetV2 and ResNet50, achieving training and validation accuracies of 99% and 95.5%, respectively. This highlights the pivotal role of model architecture in obtaining robust outcomes. However, the constraints imposed by the dataset size pose challenges, particularly for deeper models such as ResNet50. The intricate architecture of ResNet50, although powerful, struggles to effectively generalize the dataset, resulting in lower accuracy on the validation dataset. Notably, freezing the layers of complex models like ResNet50 yields benefits in terms of enhanced generalization, albeit at the potential cost of lower accuracy on the training dataset. This trade-off significantly improves accuracy on the validation dataset.

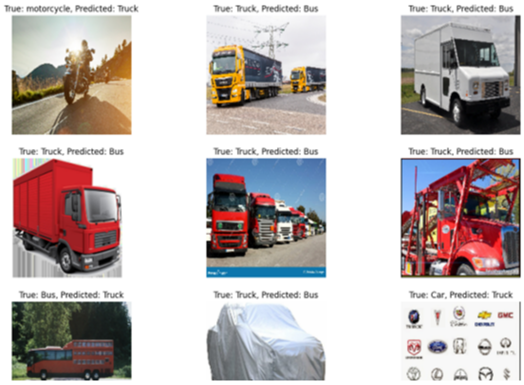

True: motorcycle, Predicted: Truck
True: Truck, Predicted: Bus
True: Truck, Predicted: Bus
True: Truck, Predicted: Bus
True: Truck, Predicted: Bus
True: Truck, Predicted: Bus
True: Bus, Predicted: Truck
True: Truck, Predicted: Bus
True: Car, Predicted: Truck

Figure 11. EfficientNetB0 mislabels

The task of categorizing trucks within the dataset poses a formidable challenge, characterized by a pronounced incidence of mislabeling. This prevalent misclassification tendency often results in the erroneous assignment of instances representing trucks to categories such as buses or cars, as discerned across various models. The manifestation of this misclassification phenomenon is graphically depicted in the confusion matrices presented in Figures 6, 7, 8, and **??**. Particularly noteworthy is the elucidation provided by Figure 11, where a more detailed illustration of the mislabeled instances is presented. This detailed examination brings to light the model's proclivity to misclassify instances, particularly evident due to the intrinsic similarities among the labels assigned to trucks, buses, and cars.

The efficacy of knowledge distillation, exemplified in the transfer from ResNet50 to EfficientNetB0, showcases the potency of ResNet50 as a teacher model.

Structural disparities, notably in MobileNetV2, pose challenges, emphasizing the critical role of model compatibility and structural similarity in distillation effectiveness.

Optimal distillation outcomes are achieved with a temperature of 5 and alpha of 0.05, emphasizing the importance of fine-tuning these parameters. Distillation proves valuable, substantially enhancing accuracy compared to the base model. It mitigates accuracy drop during fine-tuning, particularly with all layers unfrozen, showcasing robustness in challenging scenarios.

EfficientNetB0, utilizing transfer learning, stands out as the top-performing model. Exploration of transfer learning parameters identifies approximately 10 layers as optimal, balancing underfitting and overfitting. Random layer selection introduces variability, with a preference for layers from the latter portion. Gradual unfreezing surpasses single-step unfreezing, contributing to dynamic adaptation and mitigating overfitting issues for superior training results.

4. Conclusions

In conclusion, this study not only provides insights into effective training techniques but also underscores the impact of dataset constraints on model performance. The challenges associated with intricate model architectures and specific class difficulties, such as truck classification, highlight the need for continuous refinement and adaptation in deep learning approaches. As the field progresses, addressing these challenges will be essential for the development of models that perform optimally within resource-constrained environments, paving the way for advancements in real-world applications.

Acknowledgment

Hoang-Dieu Vu was funded by the PhD Scholarship Programme of Vingroup Innovation Foundation (VINIF), code VINIF.2023.TS.021. We would like to thank Dr. Khoat Than (HUST).

References

[1] Keiron O'Shea and Ryan Nash. An introduction to convolutional neural networks. *arXiv preprint arXiv:1511.08458*, 2015.

[2] Stephen Karungaru, Lyu Dongyang, and Kenji Terada. Vehicle detection and type classification based on cnn-svm. *International Journal of Machine Learning and Computing*, 11:304–310, 08 2021. doi:10.18178/ijmlc.2021.11.4.1052.

[3] Hicham Bensedik, Ahmed Azough, and Meknasssi Mohammed. Vehicle type classification using convolutional neural network. pages 313–316, 10 2018. doi:10.1109/CIST.2018.8596500.

[4] Xinchen Wang, Weiwei Zhang, Xuncheng Wu, Lingyun Xiao, Yubin Qian, and Zhi Fang. Real-time vehicle type classification with deep convolutional neural networks. *Journal of Real-Time Image Processing*, 16:1–10, 02 2019. doi:10.1007/s11554-017-0712-5.

[5] Shuo Feng, Huiyu Zhou, and H.B. Dong. Using deep neural network with small dataset to predict material defects. *Materials Design*, 162, 11 2018. doi:10.1016/j.matdes.2018.11.060.

[6] Miguel Romero, Yannet Interian, Timothy Solberg, and Gilmer Valdes. Training deep learning models with small datasets, 12 2019.

[7] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009. doi:10.1109/CVPR.2009.5206848.

[8] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4510–4520, 2018.

[9] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

[10] Antonios Tragoudaras, Pavlos Stoikos, Konstantinos Fanaras, Athanasios Tziouvaras, George Floros, Georgios Dimitriou, Kostas Kolomvatsos, and Georgios Stamoulis. Design space exploration of a sparse mobilenetv2 using high-level synthesis and sparse matrix techniques on fpgas. *Sensors*, 22:4318, 06 2022. doi:10.3390/s22124318.

[11] Mingxing Tan and Quoc V. Le. Efficientnet: Rethinking model scaling for convolutional neural networks. *arXiv*, (1905.11946), 2020.

[12] Chao Su and Wenjun Wang. Concrete cracks detection using convolutional neuralnetwork based on transfer learning. *Mathematical Problems in Engineering*, 2020:1–10, 10 2020. doi:10.1155/2020/7240129.

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[14] Ali Bhatti, Muhammad Umer, Syed Adil, Mansoor Ebrahim, Daniyal Nawaz, and Faizan Ahmed. Explicit content detection system: An approach towards a safe and ethical environment. *Applied Computational Intelligence and Soft Computing*, 2018, 07 2018. doi:10.1155/2018/1463546.

[15] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv*, (1503.02531), 2015.

[16] Jianping Gou, Baosheng Yu, Stephen J. Maybank, and Dacheng Tao. Knowledge distillation: A survey. *International Journal of Computer Vision*, 129(6):1789–1819, March 2021. ISSN 1573-1405. doi:10.1007/s11263-021-01453-z. URL `http://dx.doi.org/10.1007/s11263-021-01453-z`.