Journal of Electrical Engineering, Vol. 77, No. 2, 2026, pp. 123-129
https://reference-global.com/journal/JEE

Multi-stage fine-tuning of EfficientNetV2-S for material recognition on edge devices

Stefan Tomić[1*], Dhabia Aldhuhoori[1], Turker Turker[1], Jelena Nikolić[2], Zoran Perić[2], Riccardo Zese[3]

Accurate material recognition with low computational overhead is critical for edge applications such as autonomous drones, mobile robots, and smart manufacturing systems. Direct fine-tuning of deep backbones often leads to early saturation in validation accuracy due to overfitting on small, domain-specific datasets. To address this, we propose a structured multi-phase fine-tuning strategy for EfficientNetV2-S, progressively unfreezing layers over four stages with adaptive learning rate scheduling. The approach also incorporates label smoothing, dropout, and data augmentation to enhance generalization. We evaluated the method on a curated dataset of 1,730 images across four material classes: glass, metal, paper, and plastic. The resulting model achieves a validation accuracy of 95.66%, demonstrating that the proposed pipeline effectively balances accuracy and computational efficiency, making it suitable for real-time deployment on resource-constrained edge devices.

Keywords: EfficientNetV2S, material classification, multi-phase fine-tuning, transfer learning, model optimization

1 Introduction

Automated material classification using computer vision has become increasingly important in applications such as recycling, quality control, and smart construction [1]. The ability to reliably distinguish materials such as plastic, glass, metal, and paper in real time enables efficient sorting, supports sustainable practices, and enhances the autonomy of modern systems. However, accurate material recognition remains challenging due to the high visual similarity between classes, varying illumination conditions, and the limited availability of annotated data. These issues are often compounded by class imbalance and the computational constraints typical for edge devices [2].

While convolutional neural networks (CNNs) have significantly advanced visual recognition, many high-performing architectures are computationally demanding and thus impractical for real-time deployment on resource-constrained platforms. Recent lightweight models, such as EfficientNetV2-S [3], offer a promising balance between accuracy and efficiency. The ongoing relevance and state-of-the-art performance of the EfficientNetV2-S architecture are further evidenced by very recent studies. For instance, Zhao et al. [4] demonstrated its effectiveness through progressive optimization for complex image classification tasks, while Njoroge et al. [5] highlighted its superior balance in edge-hybrid modelling compared to other lightweight architectures like MobileNetV2. These advancements underscore the suitability of EfficientNetV2-S for resource-constrained material recognition tasks. However, achieving strong generalization on domain-specific tasks like material classification requires not only efficient architectures but also well-structured training procedures. Moreover, dataset quality and domain relevance continue to pose critical challenges. While public datasets are widely used, they often lack task-specific context, prompting many studies to rely on custom or hybrid datasets.

Preliminary experiments on our curated material dataset indicated that directly fine-tuning the complete backbone often led to early convergence at suboptimal validation accuracy, likely due to overfitting given the limited number of training samples. This observation motivated the development of a structured, multi-phase fine-tuning strategy that progressively adapts the backbone parameters while maintaining stable convergence and avoiding catastrophic forgetting.

In this paper, we present a structured multi-stage fine-tuning strategy tailored for EfficientNetV2-S to enhance material classification performance in scenarios with limited data and computational resources. Our approach combines progressive layer unfreezing, adaptive learning rate scheduling, and regularization

[1]Deptartment of Electromechanical Eng. and Tech., Abu Dhabi Polytechnic, Al Nasr Street, MBZ Z23, Abu Dhabi, UAE
[2]Faculty of Electronic Engineering, University of Nis, Aleksandra Medvedeva 14, 18000 Nis, Serbia
[3]Department of Chemical, Pharmaceutical and Agricultural Sciences, University of Ferrara, Ferrara, Italy
emails: stefan.tomic@actvet.gov.ae, A00056988@actvet.gov.ae, turker.turker@actvet.gov.ae, jelena.nikolic@elfak.ni.ac.rs, zoran.peric@elfak.ni.ac.rs, riccardo.zese@unife.it

https://doi.org/10.2478/jee-2026-0013, Print (till 2015) ISSN 1335-3632, On-line ISSN 1339-309X

techniques such as label smoothing, dropout, and data augmentation to address these challenges effectively. In contrast to the heavy computational demands and large-scale data requirements highlighted recently in [1] and [2], we show that a compact EfficientNetV2-S model trained on a curated dataset of 1,730 images can accurately classify four material classes: glass, metal, paper, and plastic, demonstrating that high performance is achievable without relying on massive datasets or oversized models. The proposed strategy achieves 95.66% validation accuracy, confirming its effectiveness while maintaining a lightweight and efficient model design. Our main contributions are:

1. A progressive multi-stage fine-tuning framework that systematically increases the number of trainable parameters to enhance feature adaptation while minimizing catastrophic forgetting.

2. Demonstration of high accuracy with a relatively small dataset of 1730 images, highlighting the effectiveness of careful regularization and data augmentation.

3. Preparation for edge deployment, including support for Quantization-Aware Training (QAT), enabling efficient inference without compromising training fidelity.

4. Comprehensive evaluation and analysis of parameter scaling, training dynamics, and performance trade-offs, providing practical guidance for resource-efficient model deployment.

The rest of the paper is organized as follows. Section 2 reviews related work and emphasizes that achieving accurate and efficient material classification in real-world images remains challenging. Section 3 provides a detailed specification of our model architecture and training strategy. Section 4 reports the results, compares them with findings from the literature, and explains the rationale behind the chosen strategy. Finally, Section 5 summarizes the key findings and concludes the paper.

2 Related work

Numerous studies have validated the effectiveness of transfer learning in material classification. Mengiste et al. [6] achieved 95% accuracy using a pre-trained InceptionV3 model with texture features on just 208 construction material images. Early work by Wieschollek and Lensch [7] applied CNNs to this domain, followed by Sticlaru [8], who explored neural architectures tailored to material recognition. Zhang et al. [9] further enhanced performance by integrating deep feature representations.

Building on these foundational works, recent studies have achieved impressive results on waste and recycling classification. Rayhan and Rifai [10] reported 95.2% accuracy using DenseNet121 on a hybrid waste classification dataset of 4586 images. Moreover, Khan et al. [11] achieved 99% accuracy using ResNet50 for classifying four metal types from microstructure images, demonstrating that even with modest dataset sizes, transfer learning can yield high accuracy.

Larger datasets also benefit from deep learning methods. For instance, Malik et al. [12] applied neural network models to classify waste to promote sustainable development, while authors in [13] used VGG-16 on a recycling dataset of 12873 images, achieving 84.6% accuracy. These outcomes underscore the versatility of pre-trained architectures such as ResNet, DenseNet, and VGG across applications in recycling, construction, and waste management. However, dataset quality and domain relevance remain critical challenges. As noted by Sharan et al. [14], accurate and efficient material classification in real-world images remains difficult. Public datasets are commonly used but often lack task-specific context, leading many studies to adopt custom or hybrid datasets. For instance, Mengiste et al. [6] developed a domain-specific dataset focused on condition-dependent material properties, while Rayhan and Rifai [10] enriched their dataset with locally sourced images to enhance contextual relevance.

A persistent challenge in material classification is differentiating visually similar materials, e.g., glass vs. plastic or paper vs. cardboard, under varying lighting and background conditions. Several techniques have been proposed to address this, including texture-based descriptors (e.g., GLCM – Gray-Level Co-occurrence Matrix features [6]) and ensemble learning. Rayhan and Rifai [10] combined MobileNetV2 and DenseNet121, while Malik et al. [12] used region-specific fine-tuning with EfficientNet-B0 to improve accuracy in geographically localized waste classification. Table 1 summarizes key references, outlining their challenges, solution strategies, and achieved accuracies. The findings demonstrate that transfer learning, particularly when combined with ensemble modeling and texture encoding, effectively mitigates visual ambiguity – achieving accuracies ranging from 84% to 99%, depending on specific task, dataset size, and model architecture.

Unlike prior work using large datasets, our approach trains a compact model on 1,730 domain-specific images across four classes (glass, metal, paper, plastic), achieving 95.66% validation accuracy. While slightly below some benchmarks, our model is optimized for edge deployment, balancing accuracy, size, and inference speed for real-time mobile applications.

Table 1. Material classification solutions and their performance

Material domain	Solution approach	Study reference	Accuracy (%)
Construction materials	InceptionV3 + Texture features	Mengiste et al. [6]	95.0
Waste materials	Multi-model ensemble	Rayhan and Rifai [10]	95.2
Metal microstructures	ResNet50 transfer learning	Khan et al. [11]	99.0
Recyclable materials	VGG-16 transfer learning	Liu and Liu [13]	84.6

3 Model architecture and training strategy

3.1 Dataset preparation and preprocessing

Starting with the publicly available Flickr Material Database [14], the dataset was further enlarged by collecting additional images from diverse online sources and relevant material datasets, producing a set of more than 4,000 images to enhance visual diversity. However, a large portion of these images did not reflect realistic single-object material recognition scenarios, as they frequently contained multiple materials, dominant background regions, or visually ambiguous targets. To address this limitation, we designed a custom automated data curation pipeline based on object detection. Using a YOLOv8 detector [15], candidate material bearing objects were identified in each image, and only samples containing a clearly dominant object were retained. The largest detected region was then used to extract an object-focused crop, thereby suppressing background clutter and enforcing object-level consistency across the dataset. This automated filtering process reduced the dataset to 1,730 high-quality images spanning four material classes: paper, plastic, metal, and glass. As a result, the final dataset is not only compact but also closely aligned with the intended material classification task, providing a more reliable training and evaluation benchmark than the original unfiltered collection. The dataset was split into 80% for training and 20% for validation. This split remained fixed and consistent across all epochs to ensure stable and comparable evaluation during the multi-stage fine-tuning process.

3.2 Model architecture

We adopt EfficientNetV2-S as the backbone due to its strong balance between accuracy and computational efficiency on resource-constrained platforms [3]. The network is initialized with ImageNet pretrained weights to leverage robust general-purpose feature representations, which is particularly important given the limited size of the material dataset [3, 5]. All input images are resized to 299 by 299 pixels with three color channels before being passed to the network, ensuring a consistent input resolution that matches the configuration shown in Fig. 1 and provides sufficient spatial detail for material discrimination.

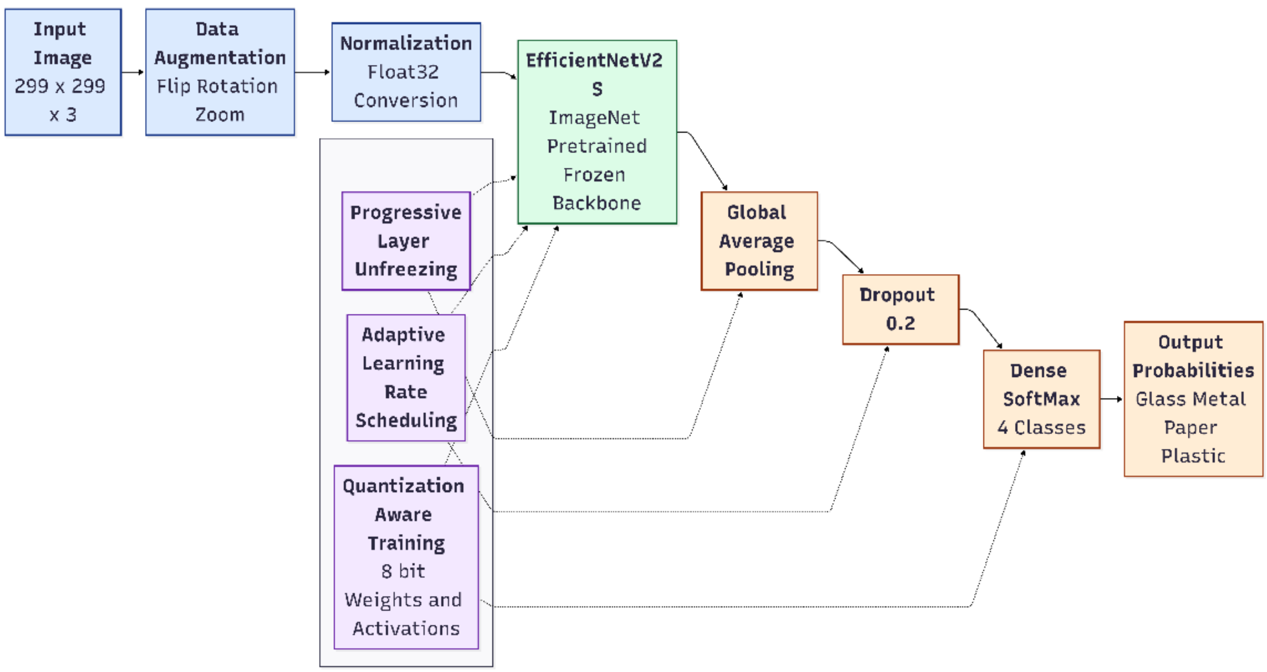

Fig. 1. Overview of the multi-phase fine-tuning strategy for EfficientNetV2-S

To improve generalization, data augmentation is applied during training to simulate variations in scale, orientation, and illumination, following strategies similar to those reported in [6]. This process effectively enlarged the training set by generating on-the-fly variations of the original images, ensuring a diverse input stream for the model during training which supports the prevention of overfitting to the training set. The augmented images are processed by the EfficientNetV2S convolutional backbone, which extracts hierarchical visual features through a compound scaled architecture optimized for parameter efficiency and representational capacity [3]. As illustrated in Fig. 1, a global average pooling layer is applied to the final convolutional feature maps, compressing the spatial dimensions into a fixed-length feature vector while reducing overfitting and computational cost. This is followed by dropout regularization with a rate of 0.2 to enhance generalization further.

The classification head consists of a fully connected SoftMax layer with four output units corresponding to the target material classes, glass, metal, paper, and plastic. Model adaptation to the material recognition task is achieved through a multi-stage fine-tuning process in which progressively deeper layers of the backbone are unfrozen and updated, allowing task-specific features to be learned while preserving the stability of low-level representations. In addition, QAT was applied using TensorFlow Model Optimization Toolkit by simulating 8-bit integer quantization of both weights and activations during training. Simulated quantization operators were inserted throughout the trainable layers of the network during fine-tuning, allowing the model to adapt to quantization-induced noise while preserving accuracy. This facilitates compatibility with low-precision inference, resulting in minimal performance degradation and reducing the need for post-training quantization. In contrast, Post-Training Quantization often lacks direct control over the resulting accuracy drop. As suggested in [16], one solution for more sophisticated control of accuracy loss involves the selection of varying bit-widths for different layers during the quantization process, which further emphasizes the advantages of integrating quantization awareness directly into the training phase.

3.3 Progressive multi-stage fine-tuning strategy

To optimize both predictive performance and parameter efficiency, we adopt a structured four-phase fine-tuning strategy leveraging the EfficientNetV2-S backbone pretrained on ImageNet [3]. Unlike con-ventional two-phase transfer learning [17], this multi-stage approach gradually unfreezes layers of the backbone, enabling stable adaptation while minimizing the risk of overfitting and excessive computational cost, similar to what has been demonstrated in [18] and [19].

Phase 1: Feature extraction with frozen backbone

Training began with the EfficientNetV2-S backbone frozen while only a lightweight classification head, consisting of global average pooling, dropout, and a dense SoftMax output layer, was optimized. This design reduced the number of trainable parameters and provided a stable initialization for transfer learning. A learning rate of 0.001 was used, along with label smoothing and early stopping, to improve generalization and reduce overfitting, following established best practices reported in [20] and [21].

Phase 2: Progressive unfreezing of backbone layers

After stabilizing the classification head, we progressively unfroze the backbone in four stages to enable controlled domain adaptation. In Stage 1, the top 30 layers were unfrozen to adjust mid-level features. Stage 2 increased the number of trainable layers to 100, facilitating higher-level feature refinement, while Stage 3 expanded the number of trainable layers to 200 for more comprehensive adaptation. Stage 4 involved full end-to-end fine-tuning to consolidate all learned representations. Each stage was run for 3-4 epochs, with learning rates periodically reduced to prevent catastrophic forgetting, as highlighted in [22]. The gradual, staged unfreezing was designed to balance stability and plasticity: early adaptation focused on higher-level features to leverage pre-trained knowledge, while later stages allowed fine-grained adjustments across the network. During full unfreezing, ReduceLROnPlateau scheduling was applied to dynamically adjust the learning rate, ensuring smooth convergence without destabilizing earlier learned features.

3.4 Training configuration and callbacks

Keras callbacks were employed to monitor performance and adaptively manage the training process. ModelCheckpoint preserved weights corresponding to the highest validation accuracy, while early stopping [21] terminated training when improvements plateaued, preventing overfitting. During the final fine-tuning phases, ReduceLROnPlateau [23] was applied to dynamically adjust learning rates, enabling smooth convergence and improved generalization. Training was conducted in Google Colab with GPU acceleration, with Google Drive integration for logging, checkpointing, and fault tolerance [24], ensuring reproducibility and safe resumption of experiments after interruptions.

The modular codebase facilitated rapid experimentation across architectures and fine-tuning strategies. The final model achieved 95.66% validation accuracy and required 77.58 MB of storage, making it optimized for deployment on resource-constrained edge devices such as the Raspberry Pi 5.

4 Performance evaluation and training analysis

This section analyzes the performance of EfficientNetV2S on the curated material dataset, focusing on the effectiveness of the proposed multi-phase fine-tuning strategy and its impact on accuracy and convergence.

4.1 Training progression and fine-tuning strategy

A staged training approach was employed to unfreeze layers of the EfficientNetV2-S backbone progressively. During Phase 1, only the classification head (≈5,124 trainable parameters) was optimized, providing a stable initialization before fine-tuning the backbone. Phase 2 consisted of four stages in which deeper layers were incrementally unfrozen. The number of trainable parameters increased from approximately 2.5 million in Stage 1 to 8M, 15M, and finally 20.3M in Stage 4, enabling gradual domain adaptation while minimizing the risk of catastrophic forgetting.

Each fine-tuning stage was trained until convergence using validation-based early stopping rather than a fixed epoch budget. A patience of five epochs was employed to avoid premature termination and ensure stable adaptation of newly unfrozen layers. In practice, earlier stages converged rapidly within three epochs due to limited parameter updates, while the fully unfrozen stage required slightly longer training (four epochs) to stabilize the substantially larger parameter space. This adaptive training duration ensured efficient optimization while preventing overfitting and unnecessary computation.

Figures 2 and 3 show the training and validation loss and accuracy over epochs, demonstrating convergence and generalization. Figure 4 shows the learning rate schedule, while Fig. 5 depicts the relationship between validation accuracy and the number of trainable parameters. During Phase 1 (Epochs 1-5), the model rapidly learned discriminative features, achieving 92.20% validation accuracy with a loss of 0.5492. As deeper layers were progressively unfrozen in Phase 2 (Epochs 6 to 18), validation accuracy steadily increased, reaching 95.66%, with loss decreasing to 0.3386. Training and validation losses consistently declined across successive stages before plateauing, as shown in Fig. 6. The scheduled learning rate, reduced from 10^{-3} to $5{\times}10^{-6}$, was critical for stable convergence and effective optimization without introducing instability.

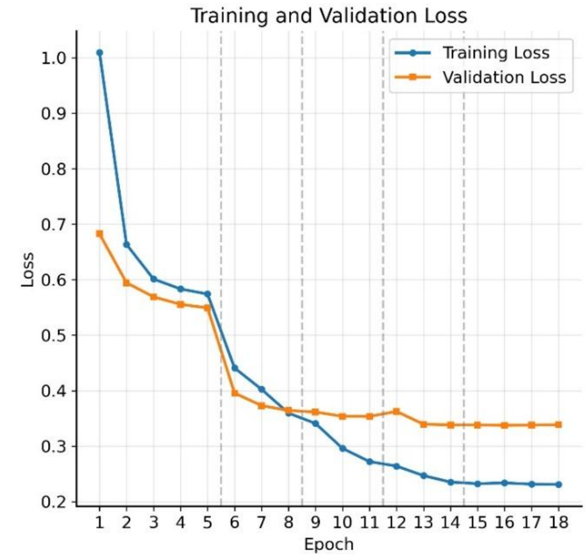

Fig. 2. Training and validation loss over epochs

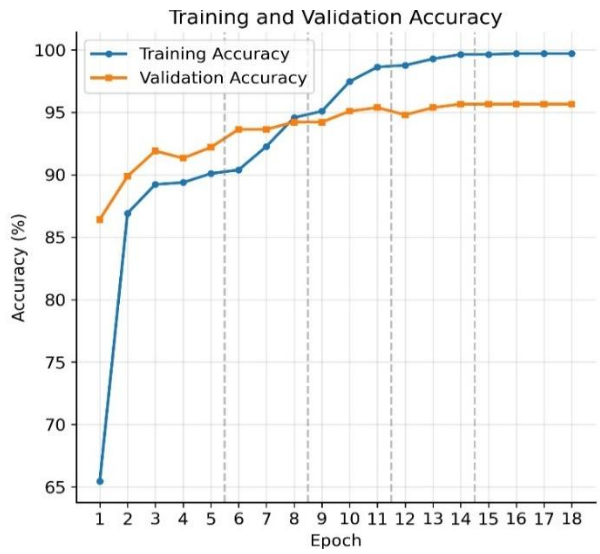

Fig. 3. Training and validation accuracy over epochs

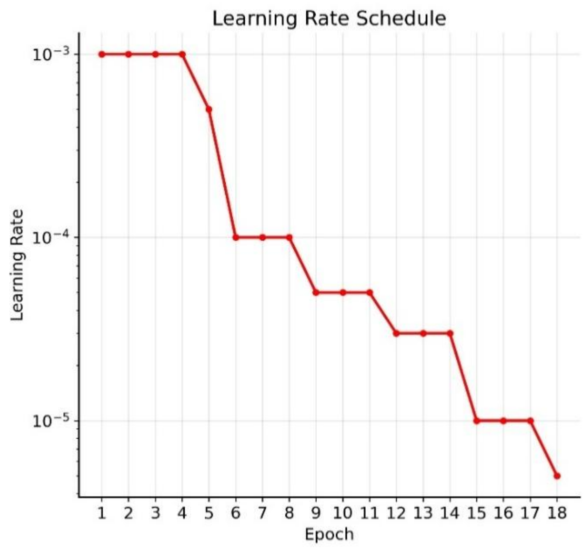

Fig. 4. Learning rate over epochs

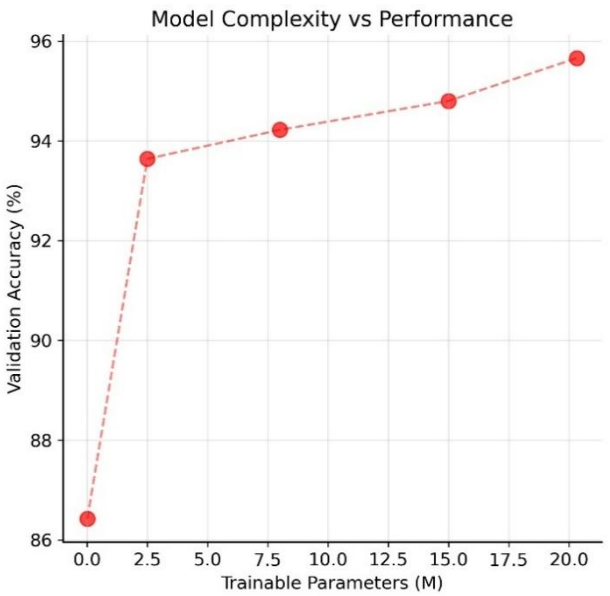

Fig. 5. Dependence of validation accuracy on the number of trainable parameters

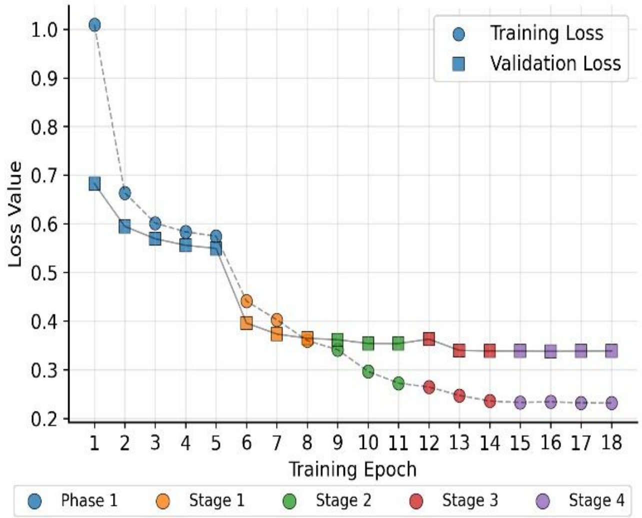

Fig. 6. Training loss progression across tuning stages

4.2 Parameter scaling and performance correlation

The systematic scaling of trainable parameters across training phases had a profound impact on model performance. Table 2 summarizes the progression of trainable parameter counts, corresponding validation accuracies, incremental accuracy improvements per phase, and the associated number of training epochs required for convergence. One can notice that the substantial increase in trainable parameters, representing an approximate 4,000-fold expansion from Phase 1 (5124 parameters) to Stage 4 (20.3M parameters), enabled the model to progressively capture more sophisticated and nuanced material feature representations. This parameter scaling strategy facilitated the transition from basic feature extraction to complex domain-specific pattern recognition capabilities.

Although the fine-tuning strategy was highly effective, Stage 4 showed a performance plateau, with complete backbone unfreezing yielding no further improvement beyond Stage 3. This suggests the model had reached its optimal capacity given the dataset's size and complexity, indicating a balanced trade-off between model expressiveness and available data. Convergence patterns and loss trends across all stages further validate the effectiveness of the proposed methodology in achieving near-optimal performance while avoiding overfitting.

Table 2. Progression of trainable parameters and validation accuracy across training phases

Phase / Stage	No. of parameters	Final accuracy	Accuracy improvement	No. of epochs
Phase 1	5.1K	92.20%	5.78%	5
Phase 2 Stage 1	2.5M	94.22%	0.58%	3
Phase 2 Stage 2	8.0M	95.38%	1.16%	3
Phase 2 Stage 3	15.0M	95.66%	0.86%	3
Phase 2 Stage 4	20.3M	95.66%	0.00%	4

5 Summary and conclusion

This paper presented a structured multi-phase fine-tuning strategy for EfficientNetV2-S to improve material classification accuracy while maintaining computational efficiency for deployment on edge devices. The method was evaluated on a curated dataset of 1,730 images across four material classes: plastic, glass, metal, and paper. By combining an initial feature extraction phase with progressive backbone unfreezing, adaptive learning rate scheduling, and regularization techniques, we achieved a high validation accuracy of 95.66%. Experimental results showed that the staged increase in trainable parameters significantly improved performance up to a saturation point, beyond which additional complexity yielded no further gains. These findings demonstrate that the proposed fine-tuning framework effectively balances model capacity with dataset limitations, avoids overfitting, and delivers near-optimal performance. The results validated the effectiveness of multi-phase fine-tuning for high-accuracy material recognition under resource constraints. The presented strategy enhances transfer learning performance while supporting practical deployment in real-time applications such as mobile robotics, smart manufacturing, and intelligent sorting systems. In the future, the curated dataset is planned to be made publicly available to support reproducibility and further research. For benchmarking, lightweight architectures such as MobileViT and EfficientNetV2 variants are particularly suitable, offering a balance between accuracy and computational efficiency for edge deployment. In addition, the future work will explore larger datasets, integrate lightweight object detection into end-to-end pipelines, and fully apply quantization-aware training to further optimize edge deployment, taking layer-wise weight statistics into account.

Acknowledgement

This work has been supported in part by the Horizon Europe Twinning project AIDA4Edge (Grant Agreement No. 101160293), and by the Ministry of Science, Technological Development and Innovation of the Republic of Serbia under Grant No. 451-03-34/2026-03/200102.

References

[1] P. Thielmann, Y. Zhou, B. Mirbach, D. Stricker and J. Rambach, "A review of computer vision for industrial-grade waste classification," *IEEE Access*, vol. 13, pp. 151934–151953, 2025.

[2] E. Adán and A. Adán, "Computer vision for glass waste: Technologies and sensors," *Sensors*, vol. 25, no. 21, pp. 6634, 2025.

[3] M. Tan and Q. Le, "EfficientNetV2: Smaller models and faster training," in *Proc. Int. Conf. Machine Learning (ICML)*, PMLR, Jul. 2021, pp. 10096–10106.

[4] Z. Zhao, E. B. A. Bakar, N. B. A. Razak, et al., "Progressive optimization of EfficientNetV2 for classification based on images of corroded objects," *Comp. Appl. Math*., vol. 44, p. 419, 2025.

[5] T. Njoroge, R. Kibuku, and K. Mugoye, "Comparative and edge-hybrid modeling of EfficientNetV2 and MobileNetV2 for multi-class crop disease classification with statistical validation," *J. Edge Comput*., vol. 4, no. 2, pp. 234–262, 2025.

[6] E. Mengiste, K. R. Mannem, S. A. Prieto, and B. García de Soto, "Transfer-learning and texture features for recognition of the conditions of construction materials with small data sets," *J. Comput. Civ. Eng.*, vol. 38, no. 1, pp. 04023036, 2024.

[7] P. Wieschollek and H. Lensch, "Transfer learning for material classification using convolutional networks," *arXiv preprint* arXiv:1609.06188, 2016. Available: https://arxiv.org/abs/1609.06188

[8] A. Sticlaru, "Material classification using neural networks," *arXiv preprint* arXiv:1710.06854, 2017. Available: https://arxiv.org/abs/1710.06854

[9] Y. Zhang, M. Ozay, X. Liu, and T. Okatani, "Integrating deep features for material recognition," in *Proc. 23rd Int. Conf. Pattern Recognit. (ICPR)*, IEEE, Dec. 2016, pp. 3697–3702.

[10] Y. Rayhan and A. P. Rifai, "Multi-class waste classification using convolutional neural network," *Appl. Environ. Res.*, vol. 46, no. 2, 2024.

[11] M. A. H. Khan, H. Sabnis, J. A. A. Jothi, J. Kanishkha, and A. D. Prasad, "Classification of microstructure images of metals using transfer learning," in *Proc. Int. Conf. Modelling and Development of Intelligent Systems*, Cham: Springer Nature Switzerland, Oct. 2022, pp. 136–147.

[12] M. Malik, S. Sharma, M. Uddin, C. L. Chen, C. M. Wu, P. Soni, and S. Chaudhary, "Waste classification for sustainable development using image recognition with deep learning neural network models," *Sustainability*, vol. 14, no. 12, p. 7222, Jun. 2022.

[13] K. Liu and X. Liu, "Recycling material classification using convolutional neural networks," in *Proc. 21st IEEE Int. Conf. Machine Learning and Applications (ICMLA)*, Dec. 2022, pp. 83–88.

[14] L. Sharan, R. Rosenholtz, and E. H. Adelson, "Accuracy and speed of material categorization in real-world images," *J. Vis.*, vol. 14, no. 9, pp. 12, 2014.

[15] Ultralytics, "YOLOv8 – You Only Look Once, Version 8," 2023. Available: https://github.com/ultralytics/ultralytics. [Accessed: Jun. 01, 2025].

[16] J. Nikolić, S. Tomić, Z. Perić, A. Jovanović, and D. Aleksić, "Accuracy degradation aware bit rate allocation for layer-wise uniform quantization of weights in neural network," *J. Electr. Eng.*, vol. 75, no. 6, pp. 425–434, 2024.

[17] Y. Bengio, "Deep learning of representations for unsupervised and transfer learning," in *Proc. ICML Workshop on Unsupervised and Transfer Learning*, JMLR W&CP, Jun. 2012, pp. 17–36.

[18] J. Howard and S. Gugger, "Fastai: A layered API for deep learning," *Information*, vol. 11, no. 2, p. 108, 2020.

[19] M. Kornblith, J. Shlens, and Q. Le, "Do better ImageNet models transfer better?," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2019, pp. 2661–2671.

[20] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the Inception Architecture for Computer Vision," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2016, pp. 2818–2826.

[21] L. Prechelt, "Early stopping – but when?," in *Neural Networks: Tricks of the Trade*, Berlin, Heidelberg: Springer, 2002, pp. 55–69.

[22] J. Kirkpatrick *et al.*, "Overcoming catastrophic forgetting in neural networks," *Proc. Natl. Acad. Sci.*, vol. 114, no. 13, pp. 3521–3526, 2017.

[23] L. N. Smith, "Cyclical learning rates for training neural networks," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Mar. 2017, pp. 464–472.

[24] M. Abadi *et al.*, "TensorFlow: A system for large-scale machine learning," in *Proc. USENIX Symp. Operating Systems Design and Implementation (OSDI)*, 2016, pp. 265–283.

Received 4th February 2026