Efficiency-Accuracy Trade-offs in Quantised Deep Transfer Learning for Radiant Coil Inspection

Sit Paing Htun
Department of Control Systems and Instrumentation Engineering
King Mongkut's University of Technology Thonburi
Bangkok, Thailand
sitpaing.htun@kmutt.ac.th

Sarawan Wongsa
Department of Control Systems and Instrumentation Engineering
King Mongkut's University of Technology Thonburi
Bangkok, Thailand
sarawan.won@kmutt.ac.th

Isaratat Phung-on
Maintenance Technology Centre ISTRS
King Mongkut's University of Technology Thonburi
Bangkok, Thailand
isaratat.phu@kmutt.ac.th

***Abstract*—Radiant coils used in steam crackers for the pyrolysis process deteriorate over time owing to extreme operating conditions. Regular inspection of these coils is required to optimise maintenance scheduling and maximise product yield. The objective of this research is to enable on-site inspection by artificial intelligence (AI) using edge devices where computational efficiency is limited. Five state-of-the-art models were trained using transfer learning, followed by two different post-training quantisation methods. Model sizes were reduced by over 70% in both methods, and inference speed was significantly faster, but most models experienced a measurable decline in predictive performance. Architectural analysis reveals that heavier models combined with PTQ can outperform purpose-built lightweight models on constrained hardware in both accuracy and quantisation robustness, a finding with direct implications for edge-deployment model selection. These results demonstrate that AI-assisted on-site inspection using quantised models is a feasible and practical complementary tool to established inspection techniques.**

***Index Terms*—radiant coils, inspection, transfer learning, post-training quantisation**

I. Introduction

Radiant coils are the core heat-transfer component of ethylene steam-cracking furnaces, operating continuously at temperatures between $950\,^{\circ}\mathrm{C}$ and $1100\,^{\circ}\mathrm{C}$ whilst exposed to hydrocarbon media over extended service periods [1]. These extreme conditions drive a range of degradation mechanisms, including carburisation, oxidation, high-temperature creep cracking, thermal fatigue, overheating, and bending. Among these, carburisation is particularly damaging, accounting for $40\,\%$ of failures in alumina-former radiant coils [2]. The process involves carbon diffusion into the tube wall, promoting the formation of carbide and oxide layers that reduce creep strength, alter magnetic properties, and increase hardness. Routine decoking operations, which gasify accumulated coke by injecting air and water vapour, can themselves strip the protective alumina oxide layer, further accelerating deterioration [3]. At present, degradation is typically identified only after a critical failure has occurred, and no reliable non-destructive method exists to quantify the extent of material deterioration before that point [2]. This reactive maintenance paradigm results in unplanned shutdowns, reduced product yield, and elevated safety risk, motivating the need for an accurate, timely, and practical condition-assessment solution.

Artificial Intelligence (AI), and Deep Learning (DL) in particular, has gained considerable traction in materials science as a means of building predictive models from image and spectroscopic data [4]. Convolutional neural networks (CNNs) are especially well-suited to visual inspection tasks, owing to their ability to extract hierarchical features directly from raw images. Because labelled materials datasets are inherently limited in size, transfer learning (TL) is a natural strategy: a model pre-trained on a large corpus is fine-tuned on the target domain, reducing both the data requirement and training time whilst preserving generalisation ability [5], [6]. For radiant coil inspection, where image acquisition is costly and time-consuming, TL provides a viable path from laboratory microscopy to an automated classification tool.

Prior work on edge deployment for industrial inspection has explored several complementary strategies. Lightweight architectures — ShuffleNetV2 [7], MobileNetV2 [8], and MobileViT [9] — reduce parameter count through efficient convolution and attention designs, achieving low-power operation at the cost of representational capacity. Knowledge distillation transfers a teacher model's soft predictions to a compact student but requires retraining [6]. Quantisation-aware training (QAT) incorporates precision constraints during training to pre-condition weights for low-bit inference but adds training complexity. Post-training quantisation (PTQ) is an architecture-agnostic alternative: weights and activations are mapped from FP32 to INT8 after training, with no retraining required [10], [11]. PTQ has been evaluated on general vision benchmarks such as ImageNet, where INT8 quantisation of large models typically incurs less than 1% accuracy loss [11]. However, its interaction with domain-adapted transfer-learned models on small, metallurgical datasets remains underexplored, and no prior study has compared PTQ schemes across architectural families in the context of industrial inspection.

Heavier models such as EfficientNetV2 [12] and ResNeXt50 [13] achieve higher accuracy but at a greater computational cost, motivating the use of PTQ to bridge efficiency and accuracy for field deployment. This paper addresses that gap by benchmarking five state-of-the-art architectures

— spanning lightweight, hybrid, and heavy design families — on a radiant coil degradation classification task, each trained via transfer learning and subsequently compressed using static and dynamic INT8 PTQ. The specific contributions are: (i) a comparative analysis of efficiency-accuracy trade-offs of five architectures under two PTQ methods; (ii) an architectural explanation of differential quantisation robustness across CNN and hybrid model families; (iii) a characterisation of the practical trade-offs between static and dynamic PTQ on ARM-based edge hardware; and (iv) a demonstration of the feasibility of deploying quantised transfer-learned models on resource-constrained hardware, providing a foundation for future portable, expert-assisted inspection tools.

II. Hardware and Software Configurations

This section describes the specifications of the training environment, the quantisation environment, and the inference environment on Raspberry Pi.

A. Training Environment

All models were trained using transfer learning with PyTorch on the Google Colab Tesla T4 GPU, and the determinism and reproducibility of the training pipeline were ensured by global seeding, fixed deterministic algorithms, and the following library versions:

- Python: 3.12
- PyTorch compiled CUDA version: 12.6
- PyTorch: 2.7.1
- Torchvision: 0.22.1

B. Quantisation Environment

The trained models were quantised on a local workstation equipped with an AMD Ryzen 7 5700U processor and 8 GB RAM. The models were quantised in a virtual environment with the following library versions:

- Python: 3.13
- torch (CPU): 2.11.0
- torchvision (CPU): 0.26.0
- ONNX: 1.20.1
- ONNX script: 0.6.2
- ONNX Runtime: 1.24.4

C. Inference Environment

The inference of the models was done on Raspberry Pi 4 model B 4 GB version. Inference was performed in a virtual environment with the following library versions:

- Python: 3.11
- torch (CPU): 2.0.1
- torchvision (CPU): 0.15.2
- ONNX Runtime: 1.24.4

III. Methodology

Building on the experimental environments described above, this section details the end-to-end methodology, encompassing data acquisition, model selection, the transfer learning pipeline, and post-training quantisation.

A. Data Acquisition

The microscopic image data used to train the models was collected from radiant coils fabricated from HT E alloy, comprising nickel (45%), chromium (30%), aluminium (4%), with trace amounts of carbon and niobium, and a balance of iron. Specimens with service lives of one to seven years were subjected to a preprocessing sequence prior to imaging. Each tube was first screened by eddy-current testing to determine the degradation class; only after this screening were small sections removed for further preparation. Sections were ground using MD grinding discs with diamond grits of 120, 220 and 6µm, then diamond-polished with 3µm and 1µm suspensions to produce a scratch-free surface. Polished surfaces were chemically etched in a solution of hydrochloric and nitric acids diluted with water for approximately one minute. The prepared specimens were imaged using a Dino-Eye Edge AM7025X digital eyepiece camera (5 MP) mounted on a metallurgical microscope; images were acquired with DinoCapture 2.0 (v1.5.49 C) at 50× magnification and a resolution of 2592 × 1944 pixels. Sample images from all five classes are shown in Fig. 1. A total of 1,499 images were captured across the five classes; the class distribution is given in Table I.

TABLE I
Distribution of Microscopic Images

Class	Training Set	Test Set	Total
A (New)	270	68	338
B	238	60	298
C	239	60	299
D	236	60	296
E	214	54	268
Total	**1,197**	**302**	**1,499**

B. Model Selection

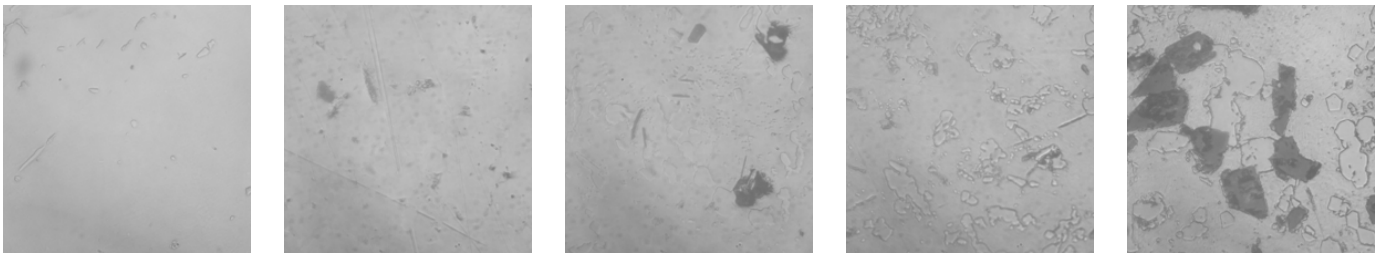
To classify the collected microscopic images, five state-of-the-art models, introduced in Section I, were selected. They can be categorised into three groups: lightweight, hybrid, and heavy. The lightweight model group includes two CNN models: ShuffleNetV2 and MobileNetV2, whilst the hybrid model group comprised one vision transformer model: MobileViT. The heavy model group consists of two CNN models: EfficientNetV2 and ResNeXt50. Trainable parameters of each model are listed in Table II.

TABLE II
Summary of Model Parameters

Model	Number of Parameters
MobileNetV2	2,230,277
ResNeXt50 (32x4d)	22,990,149
ShuffleNetV2 (x1.0)	1,258,729
EfficientNetV2-S	20,183,893
MobileViT-S	4,940,873

C. Transfer Learning

The five models were trained using the same transfer learning pipeline with deterministic and reproducible measures.



(a) Class A (b) Class B (c) Class C (d) Class D (e) Class E

Fig. 1. Comparison of microscopic images for all five classes.

They were trained on a Google Colab notebook with Tesla T4 GPU with pinned versions listed in section II and deterministic algorithm control of the training environment. Before training, the data was stratified into 80/20 training and test split. Therefore, 1,197 images were used as the training set and 302 as the test set. The training set was trained using 5-fold cross validation. The images were resized and were augmented with horizontal flips and random rotations before being passed to the models as input. The models underwent two-stage training: head-only training, in which only the classifier layer is updated and fine-tuning which trains all layers of the model. The hyperparameter settings for each training stage are summarised in Table III. During each fold, the parameters of the epoch with the highest macro F1 score were saved as the best model, and after 5 folds, the overall best model with the highest F1 score was selected. The detailed flow chart of the training process is shown in Fig. 2. The best-performing model from this process was subsequently exported for quantisation, as described in Section III-D.

TABLE III
HYPERPARAMETER SETTINGS FOR MICROSCOPIC IMAGE TRAINING

Category	Hyperparameter	Value
General Settings	Input Image Size	224×224 (256×256 for MobileViT)
	Batch Size	32
	Metric Strategy	Macro
	LR Scheduler	Cosine Annealing
Stage 1	Max Epochs	5 epochs
	Learning Rate	0.001
	Weight Decay	0.0001
	Patience	3 epochs
Stage 2	Max Epochs	30 epochs
	Learning Rate	0.0001
	Weight Decay	0.0001
	Patience	5 epochs

D. Post-training Quantisation (PTQ)

Two PTQ schemes were applied using the ONNX Runtime (ORT) library, reducing precision from FP32 to INT8. The first convolutional and final classifier layers were excluded from quantisation, as boundary layers are most sensitive to quantisation error and the error is magnified across layers [14]. Excluding the input layer preserves pixel-domain statistics that differ from intermediate feature distributions; excluding the

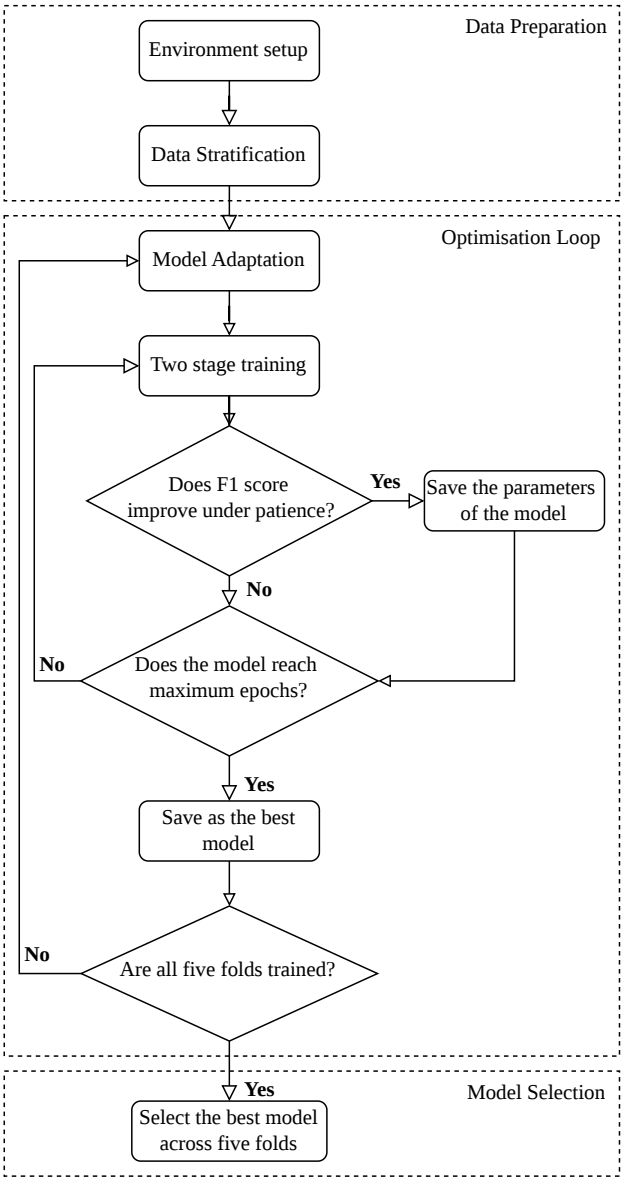


Fig. 2. Transfer Learning Framework.

classifier preserves the logit-mapping precision that determines the final class decision.

1) Static Post-training Quantisation (Static PTQ): A calibration set of 150 images (30 per class), drawn from the training data and subjected to identical preprocessing, was used to compute fixed per-layer quantisation parameters (scales and zero points) for both weights and activations via quantize_static().

2) Dynamic Post-training Quantisation (Dynamic PTQ): Only weights are quantised offline via quantize_dynamic(). Activation statistics are computed on-the-fly at runtime: ORT quantises activations to INT8, performs INT8 matrix multiplication, and dequantises back to FP32 before the next layer. This eliminates the calibration step but introduces a per-layer quantise-dequantise overhead during inference.

IV. Experimental Results

A. Evaluation Metrics

Performance was evaluated using macro-averaged Accuracy, Precision, Recall, and F1 score — standard multi-class metrics computed over all five classes using TP, TN, FP, and FN counts — together with Inference Speed (ms per image) measured on the Raspberry Pi 4B. These metrics are reported for all three model variants (baseline, static PTQ, dynamic PTQ) in the following subsection.

B. Results Comparison

Table IV shows that all five baseline models achieve strong performance, with EfficientNetV2 leading at 97.68 % accuracy and 97.69 % F1 score. ShuffleNetV2 is the fastest at 130 ms inference on the Raspberry Pi 4B, reflecting its lightweight design.

Under static PTQ (Table V), all models incur an accuracy penalty owing to the reduction from FP32 to INT8 precision. MobileNetV2 and MobileViT are most severely affected, dropping to 26.82 % and 40.40 % accuracy respectively, rendering them unsuitable for practical use. ShuffleNetV2, EfficientNetV2, and ResNeXt50 are more resilient; ResNeXt50 in particular retains 96.36 % accuracy. Static PTQ yields the greatest efficiency gains: all models achieve inference at least 82 % faster and storage at least 72 % smaller than their baselines.

Dynamic PTQ (Table VI) produces a more favourable accuracy–efficiency balance for most architectures. Because only weights are quantised offline whilst activations are computed at runtime, storage is marginally lower than static PTQ but inference is slower. MobileViT remains impaired (41.72 %), whilst ShuffleNetV2 recovers to 91.06 %. The standout result is ResNeXt50, which matches its baseline performance exactly under dynamic PTQ whilst running nearly 80 % faster and occupying 75 % less storage. The efficiency–accuracy trade-off across all conditions is visualised in Fig. 3, and storage comparisons are shown in Fig. 4. The following section interprets these findings in the context of quantisation robustness and the practical requirements of field-deployable inspection systems.

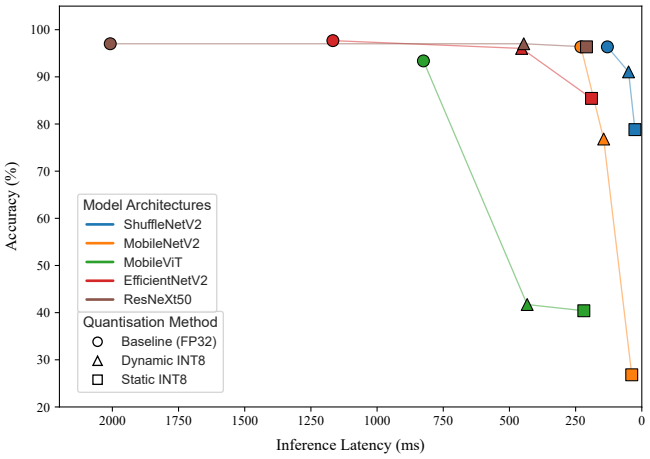


Fig. 3. Efficiency-Accuracy Trade-off.

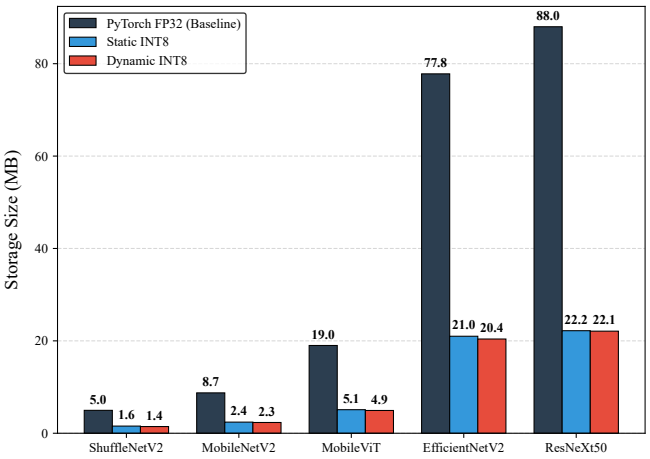


Fig. 4. Storage Comparison of the models.

V. Discussion

A. Quantisation Robustness and Architecture Selection

The results demonstrate that quantisation robustness and architectural compactness are not equivalent properties. Lightweight models (ShuffleNetV2, MobileNetV2) and the hybrid MobileViT were designed to minimise parameter count, yet all three suffer disproportionate accuracy loss under both PTQ schemes. Their limited representational capacity renders them more susceptible to the information loss incurred by reducing precision from FP32 to INT8. By contrast, ResNeXt50 and EfficientNetV2 adapt well, with ResNeXt50 achieving no measurable accuracy drop under dynamic PTQ.

The divergence between ResNeXt50 and ShuffleNetV2 — both CNN families — warrants architectural explanation. ResNeXt50 employs aggregated residual transformations with 32 parallel groups of 4-dimensional filters [13]. This multi-path design creates inherent redundancy: quantisation noise corrupting one branch is compensated by the remaining branches and the residual skip connection, which passes

TABLE IV
BASELINE MODEL PERFORMANCE COMPARISON

Models	Accuracy (%)	Precision (%)	Recall (%)	F1 score (%)	Inference (ms)	Size (MB)
ShuffleNetV2	96.36	96.44	96.45	96.38	**130**	**4.96**
MobileNetV2	96.36	96.50	96.34	96.35	230	8.74
MobileViT-S	93.38	94.58	93.37	93.46	825	19.00
EfficientNetV2	**97.68**	**97.69**	**97.71**	**97.69**	1167	77.80
ResNeXt50	97.02	97.06	97.04	97.03	2008	88.00

TABLE V
STATIC PTQ MODELS PERFORMANCE COMPARISON

Models	Accuracy (%)	Precision (%)	Recall (%)	F1 score (%)	Inference (ms)	Size (MB)
ShuffleNetV2	78.81	83.86	78.67	77.82	**26**	**1.55**
MobileNetV2	26.82	17.50	24.33	14.28	38	2.41
MobileViT-S	40.40	31.90	40.08	25.01	219	5.10
EfficientNetV2	85.43	86.68	84.71	84.92	190	21.00
ResNeXt50	**96.36**	**96.54**	**96.37**	**96.37**	209	22.20

TABLE VI
DYNAMIC PTQ MODELS PERFORMANCE COMPARISON

Models	Accuracy (%)	Precision (%)	Recall (%)	F1 score (%)	Inference (ms)	Size (MB)
ShuffleNetV2	91.06	92.00	91.12	90.47	**50**	**1.45**
MobileNetV2	76.82	81.96	76.67	72.13	144	2.33
MobileViT-S	41.72	48.83	41.34	27.24	433	4.93
EfficientNetV2	96.03	96.12	96.08	96.02	454	20.40
ResNeXt50	**97.02**	**97.06**	**97.04**	**97.03**	446	22.10

FP32-scale gradients unchanged. ShuffleNetV2, by contrast, achieves efficiency through channel-splitting and a depth-wise convolution scheme with minimal intra-layer redundancy — precisely the property that makes it compact also makes it intolerant of INT8 noise. MobileNetV2's inverted residuals and linear bottlenecks [8] constrain intermediate feature dimensionality, leaving little headroom to absorb quantisation error. MobileViT's cross-attention layers exhibit broader and more variable activation distributions than convolutional layers, making fixed-range INT8 quantisation (as applied in static PTQ) systematically lossy. This architectural redundancy advantage is not apparent from parameter count alone, underscoring the need for domain-specific PTQ benchmarking rather than reliance on general ImageNet results.

Compared with published INT8 PTQ results on ImageNet-scale datasets, where accuracy drops below 1% are routinely reported for ResNet variants [11], the larger degradation observed for lightweight models here likely reflects the sharper loss minima induced by fine-tuning on a small (1,499-image) domain-specific dataset. Heavier models, with more distributed weight representations, appear to occupy flatter minima more tolerant of weight perturbation — an advantage amplified in the small-data regime and not predictable from ImageNet benchmarks alone. The 150-image calibration set may further contribute to MobileViT's poor static PTQ accuracy: transformer attention activations are more input-dependent and longer-tailed than convolutional feature maps, so a small calibration set likely underestimates the true activation range, causing systematic clipping. Per-channel quantisation for attention layers and sensitivity analysis of calibration set size are recommended for future hybrid-architecture deployments.

The counter-intuitive inference slowdown of dynamic PTQ relative to the FP32 baseline on the Raspberry Pi 4B is explained by the runtime activation overhead. Without a dedicated INT8 accelerator, the ARM Cortex-A72 CPU must compute activation statistics, apply scale and zero-point mapping, perform INT8 matrix multiplication, and dequantise to FP32 at every layer — a cycle that can outweigh the savings from reduced weight access. Static PTQ eliminates this overhead by fixing all quantisation parameters at calibration time, enabling fully fused INT8 kernels. Consequently, on ARM hardware without an NPU, static PTQ delivers larger inference speedups, whilst dynamic PTQ is preferable when calibration data is unavailable or when serving multiple input domains. Future work should also investigate flatness-aware stopping criteria (rather than peak macro F1) and quantisation-aware training (QAT) as routes to improved PTQ tolerance in lightweight models.

B. Towards Portable, Expert-Assisted Inspection

The longer-term goal of this work is an AI-powered platform that supports materials engineers in the field, complementing established techniques such as eddy-current testing rather than replacing expert judgement. The five-class degradation taxonomy maps directly onto service-life severity levels, enabling model outputs to inform maintenance decisions without further manual interpretation.

The Raspberry Pi 4B was chosen as a reproducible low-power benchmark rather than the intended production device. Sub-500 ms inference on such hardware implies that

comparable or superior latency is achievable on a modern mobile processor or dedicated NPU in a rugged handheld device. At 446 ms per image (∼2.2 fps), ResNeXt50 with dynamic PTQ is adequate for the anticipated field workflow — positioning a microscope, capturing a still image, and reviewing the classification result — since specimen scanning is deliberate and continuous live-preview is not required. Interactive live-preview would require below 100 ms (10 fps), a target achievable on NPU-equipped mobile SoCs, which represent the intended production platform. The Raspberry Pi results therefore establish a conservative lower bound on expected field performance.

The primary limitations of this study are the modest dataset size (1,499 images from a single alloy grade, HT E) and the use of PTQ alone. Different alloy grades develop distinct carbide morphologies; the quantised weights would require recalibration or retraining before deployment on specimens from different alloy families or furnace operating histories. The architectural robustness findings reflect structural, not domain-specific, properties and are expected to transfer across metallurgical contexts. Future work should expand the dataset to additional alloy grades and investigate QAT to recover accuracy in lightweight and hybrid models.

VI. Conclusion

This study has demonstrated that post-training quantisation is a viable strategy for adapting deep transfer learning models to resource-constrained inspection hardware, with storage reductions exceeding 72% and inference speed improvements of up to 82% across all five architectures tested. Lightweight and hybrid models (ShuffleNetV2, MobileNetV2, MobileViT) suffer substantial accuracy drops, whilst complex models (EfficientNetV2, ResNeXt50) adapt well, with ResNeXt50 preserving its full baseline performance under dynamic PTQ. Architectural analysis shows that representational redundancy — exemplified by ResNeXt50's aggregated residual transformations — confers natural robustness to INT8 noise, whereas parameter efficiency leaves insufficient headroom to absorb quantisation error. On ARM hardware without a dedicated INT8 accelerator, static PTQ delivers larger inference speedups through fused kernel execution, whilst dynamic PTQ is preferred when calibration data is unavailable. The combination of high accuracy, compact storage, and sub-500 ms inference makes ResNeXt50 and EfficientNetV2 well-suited for integration into an AI-assisted inspection platform for radiant coil degradation assessment, providing a proof of concept for field-deployable tools that can augment materials engineers and contribute to safer, more cost-effective operation of ethylene cracking furnaces.

Acknowledgement

The authors would like to thank the research team at the Maintenance Technology Centre, King Mongkut's University of Technology Thonburi, for preparing the specimens and for providing the microscopic images of the radiant coils.

References

[1] C. Du, C. Liu, X. Li, B. Liu, J. Lu, X. Tan, and C. Song, "Application of magnetic analyzers for detecting carburization of pyrolysis furnace tubes," *Frontiers in Materials*, vol. 10, p. 1147125, 2023.

[2] C. Chiablam, B. Poopat, M. Noipitak, and S. Heyrman, "Eddy current analysis for predicting deterioration stages in alumina former radiant coils," *Engineering Failure Analysis*, vol. 158, p. 107943, 2024.

[3] P. M. B. de Santana, C. A. Della Rovere, E. C. d. M. C. Albuquerque, J. G. Dessi, and C. A. C. de Souza, "Failure analysis in pyrolysis furnaces: Impact of carburization and thermal cycles on tube properties," *Engineering Failure Analysis*, vol. 167, p. 109000, 2025.

[4] V. Gupta, K. Choudhary, F. Tavazza, C. Campbell, W.-k. Liao, A. Choudhary, and A. Agrawal, "Cross-property deep transfer learning framework for enhanced predictive analytics on small materials data," *Nature communications*, vol. 12, no. 1, p. 6595, 2021.

[5] S. Feng, H. Zhou, and H. Dong, "Application of deep transfer learning to predicting crystal structures of inorganic substances," *Computational Materials Science*, vol. 195, p. 110476, 2021.

[6] P. V. Dantas, W. Sabino da Silva Jr, L. C. Cordeiro, and C. B. Carvalho, "A comprehensive review of model compression techniques in machine learning: Pv dantas et al.," *Applied Intelligence*, vol. 54, no. 22, pp. 11804–11844, 2024.

[7] N. Ma, X. Zhang, H.-T. Zheng, and J. Sun, "Shufflenet v2: Practical guidelines for efficient cnn architecture design," in *Proceedings of the European conference on computer vision (ECCV)*, pp. 116–131, 2018.

[8] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4510–4520, 2018.

[9] S. Mehta and M. Rastegari, "Mobilevit: light-weight, general-purpose, and mobile-friendly vision transformer," *arXiv preprint arXiv:2110.02178*, 2021.

[10] D. Liu, Y. Zhu, Z. Liu, Y. Liu, C. Han, J. Tian, R. Li, and W. Yi, "A survey of model compression techniques: Past, present, and future," *Frontiers in Robotics and AI*, vol. 12, p. 1518965, 2025.

[11] B. Rokh, A. Azarpeyvand, and A. Khanteymoori, "A comprehensive survey on model quantization for deep neural networks in image classification," *ACM Transactions on Intelligent Systems and Technology*, vol. 14, no. 6, pp. 1–50, 2023.

[12] M. Tan and Q. Le, "Efficientnetv2: Smaller models and faster training," in *International conference on machine learning*, pp. 10096–10106, PMLR, 2021.

[13] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1492–1500, 2017.

[14] S. Wang, C. Li, Y. Kang, J. Fan, Z. Ou, and A. Yao, "Sliderquant: Accurate post-training quantization for llms," *arXiv preprint arXiv:2603.25284*, 2026.