

A Few-shot Learning Based Light-weight Deep Learning Model for Detecting Bearing Crack Size Using Vibration Sensor Data

Farhan Md. Siraj ¹, Syed Tasnimul Karim Ayon ², Mahe Zabin ², and Jia Uddin ^{3,*}

¹ Department of Computer Science and Engineering, BRAC University, Dhaka-1212, Bangladesh

² Human and Digital Interface Department, JW Kim College, Woosong University, Daejeon, Republic of Korea

³ Artificial Intelligence and Big Data Department, Endicott College, Woosong University, Daejeon, Republic of Korea

Email: {farhan.md.siraj, syed.tasnimul.karim.ayon}@g.bracu.ac.bd (F.M.S. & S.T.K.A.);

mahezabin@wsu.ac.kr (M.Z.); jia.uddin@wsu.ac.kr (J.U.)

*Corresponding author

Abstract—This paper investigates lightweight Vision Transformer architectures for few-shot bearing fault diagnosis under a leakage-free, held-out-bearing evaluation protocol. Using the Paderborn University bearing dataset at a fixed operating condition (900 rpm, 0.7 Nm, 1000 N), we convert raw 64 kHz vibration signals into filter-bank spectrograms and compare three transformer feature extractors—SimpleViT, ParallelViT, and MobileViT—within a Prototypical Network in a one-shot setting. To avoid the optimistic bias common in small-sample studies, the support and query images are disjoint and the test set consists of bearings never seen during training; accuracy is reported as the mean over 2000 episodes with 95% confidence intervals. Under this protocol, the lightweight MobileViT attains $80.5\% \pm 0.7\%$ accuracy using only 2.0M parameters, statistically matching SimpleViT ($79.9\% \pm 0.6\%$, 53.5M parameters) while training about five times faster than ParallelViT (76.7M parameters), which generalizes least well ($67.5\% \pm 0.8\%$). We further probe the three models under test-time perturbation and find that clean accuracy alone is a poor guide to deployment: all three remain usable under $\pm 10\%$ time jitter, with MobileViT losing 5.7 points and the two pure transformers statistically unchanged, but the MobileViT embedding contracts to a nearly constant vector under additive sensor noise and its accuracy falls to chance, whereas the two pure transformers degrade gracefully. We trace this to the normalization layers rather than to model size or to the hybrid design: MobileViT is the only one of the three that uses BatchNorm, whose running statistics are invalidated by the input shift, and re-estimating those statistics on unlabelled noisy data recovers 26.5 ± 2.4 accuracy points without retraining. Parameter count is therefore not the primary driver of generalization, and normalization choice, not compactness, governs robustness in this setting.

Keywords—vision transformers, few-shot learning, bearing fault detection, machine learning, prototypical networks, model efficiency, robustness, normalization layers

I. INTRODUCTION

Bearing fault detection is a critical component of industrial maintenance systems, with significant implications for operational reliability, safety, and cost reduction [1]. Early and accurate detection of bearing faults can prevent catastrophic failures, reduce downtime, and extend the lifespan of industrial machinery [2–7]. Traditional approaches to bearing fault diagnosis often rely on extensive datasets containing numerous examples of each fault type, typically collected through vibration analysis, acoustic emissions, or thermal imaging. Nevertheless, gathering complete fault data poses substantial difficulties in numerous practical industrial environments. Certain fault types may occur infrequently, making it difficult to accumulate sufficient examples for training conventional deep learning models. Additionally, the cost and disruption associated with deliberately inducing faults in operational equipment for data collection purposes can be prohibitive [8]. These constraints have motivated research into Few-Shot Learning (FSL) approaches, which aim to recognize patterns from very limited examples in extreme cases, just a single example per class. Vision Transformers (ViT) have emerged as powerful architectures in computer vision tasks, demonstrating remarkable performance across various domains. First adapted from natural language processing to visual applications by Dosovitskiy *et al.* [9], transformers have demonstrated remarkable image classification performance by applying their architecture directly to sequences of image patches. Their self-attention mechanism captures global relationships and distant dependencies, potentially identifying subtle bearing fault patterns that might escape detection by conventional Convolutional Neural Networks (CNNs). In this study, we investigate the application of three different ViT

architectures integrated with PN and FSL for fault detection on bearing crack size:

SimpleViT: A standard implementation of the ViT that processes sequences of image patches through multiple trans- former encoder layers [10]. **ParallelViT:** An architecture that employs multiple parallel branches of transformer encoders, allowing for diverse feature representations and potentially enhanced performance [11]. **MobileViT:** A lightweight variant designed for mobile and edge devices, combining the efficiency of MobileNet convolutions with the representational power of transformers [12].

For Few-Shot Learning (FSL), we implement a Prototypical Network (PN) framework [13], utilizing each architecture as a feature extractor. The PN generates prototype representations for each class from limited support examples and identifies query examples by calculating their distance to these prototypes in feature space. This method is especially valuable for industrial settings where gathering extensive fault data is difficult. This paper offers several key contributions: We present a thorough comparison of three different ViT architectures for few-shot bearing fault detection, assessing their effectiveness, computational demands, and industrial applicability. We show that transformer-based models can successfully perform one-shot learning for bearing fault diagnosis, effectively learning from just a single example of each fault type. We examine the balance between model complexity and performance, providing guidance on which transformer variants best suit resource-limited industrial monitoring systems. We explore misclassification patterns across different fault types, offering insights into which bearing fault conditions are more challenging to distinguish. The novelty of this work is threefold: (i) we provide a controlled, leakage-free comparison of lightweight and large ViT feature extractors for few-shot bearing fault diagnosis under a held-out-bearing protocol that prevents the over-optimistic results common in small-sample studies; (ii) we show that, under this protocol, a 2.0M-parameter hybrid CNN–transformer matches the accuracy of models with an order of magnitude more parameters; and (iii) we show that this clean-data equivalence does not survive perturbation, and trace the resulting embedding collapse to the use of batch-estimated normalization statistics rather than to model size, demonstrating that recalibrating those statistics on unlabelled noisy data recovers a substantial part of the loss without retraining. The paper continues as follows: Section II examines relevant research in bearing fault detection, FSL, and ViTs. Section III outlines our methodology, covering dataset preparation, model architectures, and the PN approach. Section IV delivers our experimental findings and comparative analysis. Section V concludes the paper.

II. LITERATURE REVIEW

Recent developments have marked a fundamental transformation in computer vision through the emergence of Transformer architectures [14–16]. Though originally conceived for natural language processing [17], the self-

attention mechanism has been adapted for visual contexts, leading to the development of the Vision Transformer (ViT) [9].

Simultaneously, the challenge of learning from limited data has driven significant research in few-shot learning [13, 18, 19]. Our work builds on these streams by integrating efficient transformer architectures with few-shot learning principles, and by examining how front-end architecture interacts with robustness and with deployment cost. FSL addresses the scarcity of labeled data by enabling models to generalize from a few examples. FSL proves to be the optimal approach when confronted with insufficient training data, a common scenario in nascent or emerging industries and occupations demanding arduous or costly data collection, such as industrial defect detection [20].

Early approaches such as Matching Networks [18] and PNs [13] framed the problem as a metric learning task, where similarities between query and support samples were explicitly modeled. Later, Relation Networks [19] and optimization-based methods like Model-Agnostic Meta-Learning (MAML) [21] and its variants [22, 23] extended this idea by learning an initialization or optimization strategy that could rapidly adapt to new tasks. Surveys in this domain [24] further summarize a range of techniques from embedding based to graph neural network methods [25, 26] highlighting both the promise and limitations of existing approaches. Relative to these non-transformer metric-learning approaches, which can be data-efficient but unstable under distribution shift, and to specialized CNN diagnosis models that excel at local feature extraction but capture limited global context, the self-attention of ViTs is attractive for modelling long-range spectral dependencies; our contribution is to study this in a strictly leakage-free, cross-bearing setting.

Within bearing diagnosis specifically, several recent lines of work do not use a pure transformer, and it is worth stating explicitly what they offer relative to the approach studied here. Lightweight convolutional models with self-attention ensembles [27] and convolutional-embedding transformers such as CLFormer [28] report strong accuracy under limited-sample conditions at a small fraction of the cost of a full ViT, and the latter in particular treats robustness to noise as an explicit design goal, which is directly relevant to the noise behaviour we report in Section IV.E. Optimization-based meta-learning has also been applied to this problem: generalized MAML formulations handle heterogeneous signals across domains [29], prior-knowledge-embedded meta-transfer learning targets variable operating conditions [30], and meta-learning has been used specifically for few-shot bearing diagnosis under variable working conditions [31]. On the metric side, ensembles of transformer features with Mahalanobis distance learning [32] and transformer-based few-shot methods robust to noisy labels [33] extend the prototypical formulation used here. Our contribution is complementary rather than competing: we do not propose a new architecture, and we instead hold the few- shot head fixed in order to ask what a leakage-free, held-out-bearing protocol reveals about the comparative behaviour of

transformer front ends of very different sizes, including under perturbation.

The Vision Transformer model [9] established that pure transformer architectures applied directly to discrete image patches could achieve performance comparable to leading Convolutional Neural Networks (CNNs). Building on this, works such as Swin Transformer [34] introduced hierarchical representations with shifted windows to better capture local features. Meanwhile, early explorations of augmenting convolutional architectures with attention mechanisms [35] paved the way for full transformer-based designs. Several surveys [36] have since chronicled the evolution of transformer-based vision models and their growing importance in diverse tasks. For many practical applications, especially on mobile devices, efficiency is as important as accuracy. MobileViT [12] represents a notable effort in creating transformer models that are both lightweight and general-purpose. In parallel, mobile CNN architectures such as MobileNets [37] and MobileNetV2 [38] have set a precedent for designing efficient networks. More recent works like MobileFormer [39] and neural architecture search techniques such as MnasNet [40] have further advanced the state of mobile vision systems. For deployment, model compression through quantization, pruning, and knowledge distillation [41, 42] provides complementary avenues to further reduce the footprint of an already-compact model such as MobileViT, and structural reparameterization [43] and generative compression for health monitoring [44] indicate further routes to reducing the deployed footprint. We report in Section IV.I that these gains are not distributed evenly across architectures. Recent work has explored simplified transformer designs that maintain competitive performance with reduced computational overhead [41, 45]. By balancing model simplicity with the inherent power of self-attention, our approach aims to provide robust performance even when training data is scarce. The literature reveals a clear trend toward unifying the strengths of transformers and FSL. While early works focused on either robust feature extraction or efficient adaptation, recent research is bridging these areas. Lightweight transformer models [12, 46] and efficient convolutional backbones [37, 38, 47] both reduce computational cost and open up applications in mobile and edge computing scenarios.

III. PROPOSED METHODOLOGY

Our approach follows a structured pipeline comprising data preprocessing, feature extraction using Vision Transformers (ViTs), and Few-Shot Learning (FSL) with a Prototypical Network (PN). Fig. 1 illustrates the data preprocessing steps, where raw vibration signals are transformed into spectrograms through filtering and normalization techniques.

These spectrograms serve as input to our ViT-based feature extraction framework, as depicted in Fig. 2. We implement three ViT variants such as SimpleViT, ParallelViT, and MobileViT to encode discriminative features before passing them through the FSL training

process. The extracted representations are used within the PN framework to compute class prototypes and predict query samples based on distance metrics. This pipeline ensures efficient fault classification even under limited labeled data conditions.

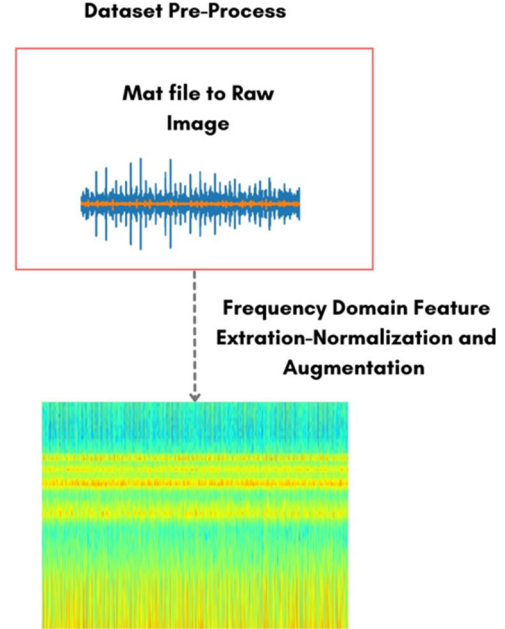


Fig. 1. Dataset preprocessing.

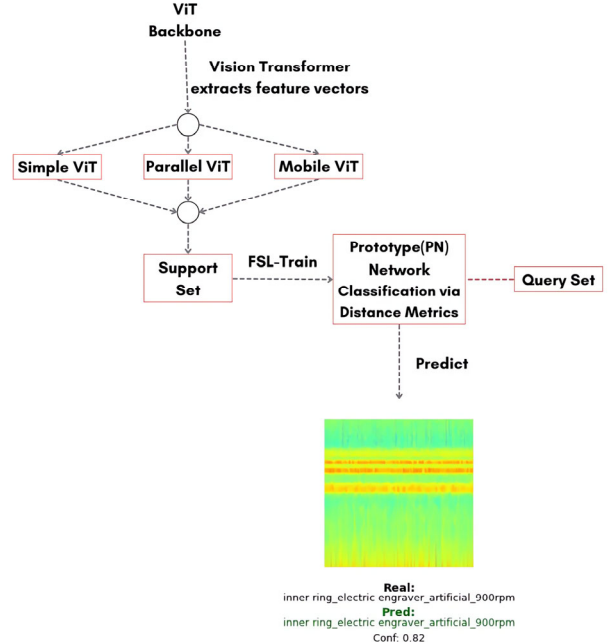


Fig. 2. ViT-based feature extraction pipeline: input spectrograms are encoded by the SimpleViT, ParallelViT, or MobileViT backbone, producing feature embeddings that are passed to the Prototypical Network for few-shot classification.

A. Dataset Overview

Our work utilizes the Paderborn University bearing dataset [48], acquired on a modular electromechanical

drive-train test rig at the Chair of Design and Drive Technology, Paderborn University. The rig comprises an electric drive motor, a torque-measurement shaft, a rolling-bearing test module, a flywheel, and a load motor. Damaged 6203 ball bearings are mounted in the test module, and the vibration signal is recorded synchronously with the motor current signals at a sampling rate of 64 kHz. In this study we use a single operating condition (setting N09 M07 F10: 900 rpm, 0.7 Nm, 1000 N), for which 20 measurements of four seconds each are available per bearing. The dataset includes healthy bearings, bearings

with artificially induced damage (electric engraver and drilling), and bearings with real damage produced by accelerated lifetime tests (fatigue pitting); representative damage patterns are shown in Fig. 3. To enable modern computer-vision approaches, we convert these raw temporal signals into time-frequency image representations (spectrograms) that capture spectral patterns and temporal fault evolution. The dataset is organized into training and test subsets with five distinct classes (Table I), with the split performed at the bearing level so that test bearings are never seen during training.

TABLE I. CLASS STRUCTURE OF THE PREPROCESSED BEARING DATASET

Class Name	Fault Location	Fault Type	Origin	RPM	Samples (Train/Test)
outer ring fatigue pitting real 900 rpm	Outer ring	Fatigue pitting	Natural	900	319 / 160
outer ring drill artificial 900 rpm	Outer ring	Drill crack	Artificial	900	320 / 159
inner ring fatigue pitting real 900 rpm	Inner ring	Fatigue pitting	Natural	900	639 / 320
inner ring electric engraver artificial 900 rpm	Inner ring	Electric engraver	Artificial	900	158 / 159
healthy bearings 900 rpm	N/A	N/A	Healthy	900	160 / 159



Fig. 3. Artificial bearing damage patterns documented in the Paderborn dataset [48]. Controlled defects include drilled holes and engraved cracks at predetermined locations.

The Paderborn dataset [48] systematically documents various failure modes through controlled experiments. Fig. 4 illustrates a specific outer ring defect created through indentation processes, demonstrating how localized stress concentrations can initiate crack propagation under operational loads.

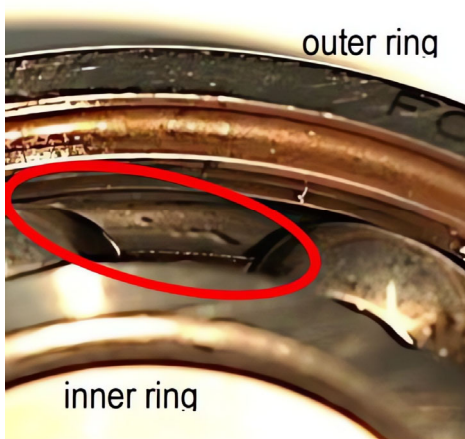


Fig. 4. Outer ring indentation defect generated through controlled mechanical stress [48]. This artificial damage mechanism simulates real-world impact-induced failures.

Complementing the indentation flaws, Fig. 5 showcases inner ring pitting damage resulting from accelerated fatigue testing. This failure mode differs fundamentally from mechanical indentation as it emerges through material degradation under cyclic loading conditions [48].

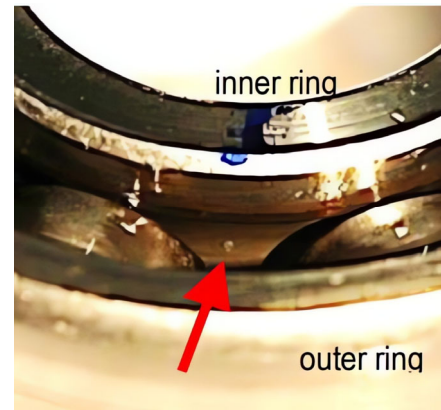


Fig. 5. Inner raceway pitting damage from natural fatigue processes [48]. Such defects develop gradually under continuous operation, mimicking real-world degradation patterns.

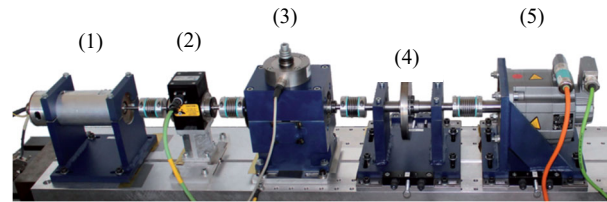


Fig. 6. Modular Paderborn University test rig used for dataset collection [48]. The platform enables controlled bearing degradation studies at the operating condition used in this work (900 rpm, 0.7 Nm, 1000 N).

The controlled degradation studies captured in these figures were conducted using the modular Paderborn University test rig shown in Fig. 6. This experimental rig enables precise manipulation of operational parameters while monitoring bearing health indicators [48].

The experimental apparatus consists of multiple elements: an electric motor (1), a torque measurement shaft (2), a rolling bearing test module (3), a flywheel (4), and a load motor (5) as shown in Fig. 6 [48]. Different defective ball bearings are installed in the bearing test module to produce the experimental measurements.

Table II lists the individual bearings assigned to each split. The split is made at the bearing level: no bearing code appears on both sides, so every test episode is drawn from a physically different bearing than any seen in training. Each bearing contributes approximately 160 spectrogram images. Three of the five classes are represented by a single held-out test bearing, which is a limitation we return to in Section V.

TABLE II. BEARING-LEVEL COMPOSITION OF THE TRAINING AND TEST SPLITS

Class	Training bearings	Test bearings
Healthy	K001	K002
Inner ring, electric engraver	KI03	KI05
Inner ring, fatigue pitting	KI04, KI16, KI18, KI21	KI14, KI17
Outer ring, drill	KA07, KA08	KA09
Outer ring, fatigue pitting	KA04, KA22	KA16

Note: Codes follow the Paderborn naming convention; no code is shared between splits.

This multimodal representation combines sensor data, visual fault documentation Fig. 3, and experimental context Fig. 6 to support comprehensive bearing health analysis. The image transformation strategy preserves the time–frequency characteristics essential for Prognostics and Health Management (PHM) applications while making the data directly usable by vision architectures. All three backbones are trained from random initialization; no pre-trained weights are used, so the comparison isolates architecture from pre-training.

B. Data Preprocessing

The raw vibration signals undergo several preprocessing steps to prepare the dataset for training, transforming them into frequency-domain representations suitable for vision-based learning models [49, 50].

1) Signal processing

The original vibration signals are extracted from MATLAB (.mat) files containing multi-channel sensor data recorded under different bearing conditions. The vibration data is sampled at 64 kHz, and only the vibration 1 channel is selected for further analysis. The signal is segmented into fixed-length frames of 50 ms, with a 50% overlap between consecutive frames, ensuring adequate temporal resolution.

2) Frequency-domain feature extraction

To extract informative fault features, the segmented signals are converted into spectrograms using a filter bank-based approach: A 64-band logarithmically spaced filter bank is applied, covering a frequency range from approximately 20 Hz up to just below the 32 kHz Nyquist frequency (at the 64 kHz sampling rate). Each band is designed on a logarithmic Equivalent Rectangular Bandwidth (ERB) scale, ensuring better separation of

fault-related frequency components. A 2nd-order Butterworth band-pass filter is applied to each frequency band, capturing energy variations across different fault conditions. The energy of each filtered signal is calculated and organized into a spectrogram matrix, with rows representing frequency bands and columns corresponding to time segments.

The spectrogram is then converted to decibels (dB scale) to enhance the visibility of subtle patterns.

The parameters of this front end were chosen to match the physics of the measurement rather than tuned for accuracy. At the studied operating condition the shaft rotates at 15 Hz, so a 50 ms frame spans slightly less than one revolution and the segment represented by one image spans several revolutions, which is sufficient for the periodic impact train of a localized defect to appear in more than one frame; the 50% overlap prevents an impact falling on a frame boundary from being attenuated in every frame. The logarithmic ERB spacing allocates proportionally more of the 64 bands to the low-frequency region containing the shaft harmonics and the ball-pass frequencies, while still covering the high-frequency structural resonances that the impacts excite, up to just below the Nyquist limit. A 2nd-order Butterworth response was preferred over a sharper filter because its flatter passband distorts the relative band energies less, which matters when the resulting energies are compared across bands within a single image. Section IV.H reports an occlusion analysis that verifies empirically which parts of this filter bank the trained models actually rely on.

3) Data normalization and input pipeline

The spectrograms undergo normalization through subtraction of the dataset mean and division by the dataset standard deviation, ensuring uniform feature distributions across samples. Processed spectrograms are resized to 256×256 pixels and normalized with the standard ImageNet channel statistics (means 0.485/0.456/0.406, standard deviations 0.229/0.224/0.225) before entering the network.

We deliberately report results without data augmentation. Episodic training already resamples support and query images at every step, and adding stochastic augmentation on top of it would confound the robustness study of Section IV.E: a model trained on noise-augmented spectrograms cannot be used to measure how an unaugmented model responds to noise it has never encountered. All models therefore share the identical deterministic input pipeline described above, at training and at test time, and every perturbation reported in Sections IV.E–IV.H is applied only at test time. Noise-augmented training is a natural mitigation for the failure mode we identify, and we return to it in Section V. Fig. 7 shows preprocessed samples from all five classes:

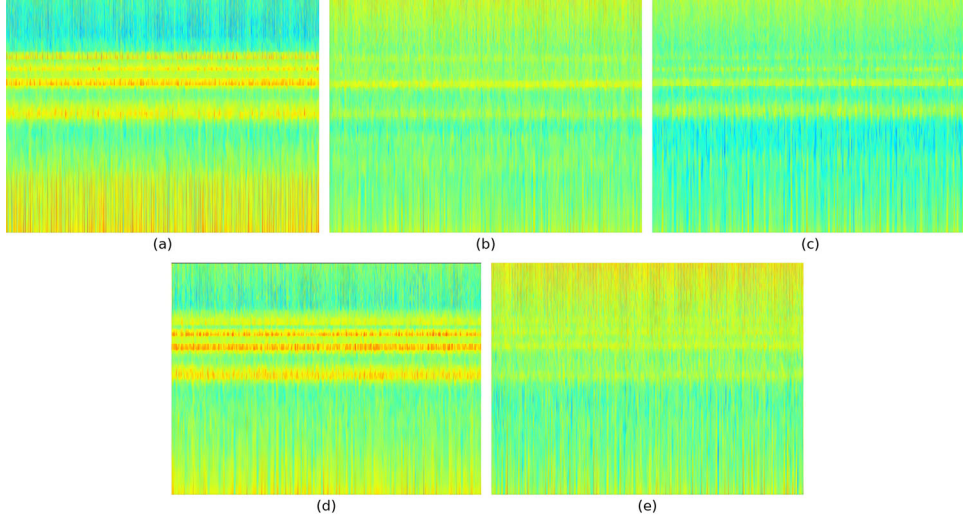


Fig. 7. Bearing image samples for each class. (a) Class 1: Outer ring fatigue pitting (real); (b) Class 2: Outer ring drill (artificial); (c) Class 3: Inner ring fatigue pitting (real); (d) Class 4: Inner ring electric engraver (artificial); (e) Class 5: Healthy bearings.

C. Feature Extraction Using ViTs

We utilize a ViT backbone to derive high-level, discriminative features from the processed images. We experimented with several ViT variants (MobileViT, ParallelViT, and SimpleViT). We wrap the SimpleViT model in a custom module that extracts 1024-dimensional feature vectors from input images. Alternative ViT variants (e.g., MobileViT or ParallelViT) are implemented similarly, allowing us to compare different transformer architectures under the FSL setting.

1) ViT Architectures

We implement three ViT variants as feature extractors, each designed with distinct computational and representational trade-offs:

a) MobileViT

MobileViT is a hybrid CNN-Transformer architecture optimized for mobile and edge devices [12]. This variant combines MobileNet-style CNN blocks for local feature extraction with lightweight multi-head self-attention to integrate global context. Its channel dimensions scale efficiently from 16 to 384, producing compact 384-dimensional embeddings. One of its primary strengths is its computational efficiency, which supports real-time inference on hardware with limited resources while still achieving competitive accuracy.

b) Parallel ViT

Parallel ViT architecture enhances feature diversity through two independent self-attention branches operating in parallel [11]. Each branch processes input patches separately, enabling the capture of complementary spatial patterns. With 16×16 pixel patches, giving 256 tokens against the 64 tokens of SimpleViT, 6 transformer layers, 8 attention heads, and 1024-dimensional embeddings, this variant prioritizes rich feature representation. While its increased parameter count raises computational demands, the parallel design aggregates multi-scale contextual information more explicitly than single-branch counterparts.

c) Simple ViT

Our simplified ViT implementation adapts the standard architecture for general-purpose few-shot learning [10]. It employs larger 32×32 pixel patches to reduce computational overhead while retaining multi-scale learning capabilities through 16 attention heads. The 6-layer transformer backbone features 2048-dimensional hidden layers in its MLP blocks, generating 1024-dimensional output embeddings. This configuration balances model complexity and performance, offering flexibility across diverse tasks without excessive specialization.

D. PN Framework

Few-Shot Learning (FSL) denotes a model's capacity to learn effectively from only a small set of labeled examples, in contrast to conventional machine learning methods that depend on large volumes of labeled data for training [18]. Its growing appeal is driven by its applicability in real-world situations where acquiring labeled data is either costly or time-intensive [51].

Within deep learning, FSL generally employs meta-learning techniques that train models to generalize across diverse tasks with only a few examples. Approaches like MAML and PN have been pivotal in advancing FSL methodologies [13].

Given the challenges associated with limited labeled data, we adopt a PN approach a meta learning strategy that effectively learns from a few examples. The core ideas include:

a) Prototype computation

For each class in the support set, the feature vectors are averaged to produce a class prototype:

$$pc = \frac{1}{|S_c|} \sum_{x_i \in S_c} f_{\theta}(x_i) \quad (1)$$

where $f_\theta(x_i)$ denotes the feature embedding of the support image x_i using the ViT encoder, and S_c represents the support set for class c .

b) *Classification via distance metrics*

Query images are classified by computing the Euclidean distance between their features and the computed prototypes. The negative distances serve as logits so that closer features yield higher probabilities.

c) *Loss and optimization*

The cross-entropy loss computed from the predicted logits is used to update model parameters via the Adam optimizer.

E. *Training and Testing Procedure*

1) *Training workflow*

Task Sampling: The dataset contains exactly five classes, so every episode is a 5-way task over all of them; the episodic sampling therefore varies which images, and hence which bearings, form the support and query sets rather than which classes appear. From each class, we sample 1 support image and 1 query image, implementing a 1-shot scenario ($k = 1$). The support and query images within an episode are always disjoint, and at evaluation the episodes are drawn exclusively from held-out test bearings that are never seen during training. This simulates the FSL challenge, where the model must learn to classify new classes with only a limited number of examples.

- **Prototype Computation:** To represent each class, we compute a class prototype, which is the average of the feature embeddings of all support images in that class, which is denoted in 1.
- **Query Classification:** After calculating the class prototypes, we classify the query images by measuring the distance between the query's embedding and each class prototype. The feature embedding $f_\theta(q)$ for each query image q is obtained, and we compute the Euclidean distance between the query's embedding and the prototypes. The query is then assigned to the class with the closest prototype.
- **Loss Function:** Following prototype computation, we employ a cross-entropy loss function for model optimization. This loss evaluates the difference between predicted probability distributions across classes and the actual class labels. The loss function is defined as:

$$L = -\sum_{q \in Q} \log \frac{\exp(-d(q, p_{y_q}))}{\sum_{c=1}^n \exp(-d(q, p_c))} \quad (2)$$

Here, Q represents the query set, and y_q is the true label of the query image. The distance $d(q, p_c)$ is the Euclidean distance between the query embedding $f_\theta(q)$ and the class prototype p_c . This formulation ensures the model minimizes the distance to the correct prototype while distinguishing between other prototypes.

- **Feature Normalization:** To stabilize the distance computations and improve model performance, we apply L2 normalization to the feature embeddings

output by the ViT encoder. This normalization ensures that all feature vectors are of unit length, preventing the scale of the embeddings from affecting the distance metric.

- **Time Profiling:** To evaluate the model's computational performance, we measure the training duration per episode. This assessment is vital for determining the model's practical utility, particularly in industrial implementations where temporal efficiency is a critical factor.
- **Model Update:** After calculating the loss, the model's parameters, particularly those of the ViT encoder, are updated through backpropagation. This iterative process allows the model to improve its ability to classify query images by learning better class representations from the support set.

F. *Evaluation Framework*

Our evaluation framework combines quantitative metrics and qualitative analysis to comprehensively assess model performance. Quantitatively, we measure standard classification metrics including accuracy (percentage of correct predictions on query sets), precision, recall, and F1-Score. Accuracy is reported as the mean over 2000 evaluation episodes together with 95% confidence intervals. To gain deeper insights into model behavior, we analyze confusion matrices to detect class-specific biases—particularly misclassifications between similar fault types—and calculate class-wise accuracy to identify underperforming categories. Qualitatively, we conduct visual comparisons of predicted versus ground truth labels across query set images, supplemented by examining model confidence scores to understand prediction certainty patterns.

G. *Challenges and Solutions*

Two key challenges emerged during development: overfitting and class imbalance. We address overfitting through episodic training—which mimics real-world few-shot learning scenarios during optimization—combined with early stopping based on validation performance. Class distribution imbalances are mitigated through our uniform task sampling strategy, ensuring equal probability for all classes during episode construction to prevent model bias toward frequent categories. This dual approach maintains generalization capability while promoting balanced feature learning across all fault types.

IV. EXPERIMENTAL RESULT

Our study's experimental approach and findings center on applying FSL with three distinct architectures: Simple ViT, Parallel ViT, and Mobile ViT. For the Paderborn vibration dataset, we implemented a three-stage processing method. Initially, we imported the raw MAT file data. Then, we generated a custom spectrogram by filtering the signal frame by frame using a bank of band-pass filters with log-spaced center frequencies. Lastly, we visualized the spectrogram to analyze the vibration characteristics over time.

A. Experimental Setup

All experiments were conducted on a single NVIDIA T4 GPU (16 GB VRAM). The implementation is based on Python and PyTorch, using the vit-pytorch implementations of SimpleViT, ParallelViT, and MobileViT as feature extractors. Models are trained with the Adam optimizer (base learning rate 3×10^{-5} , a 50-episode linear warm-up, and gradient-norm clipping at 1.0) for 800 episodes, with a fixed random seed of 42 and best-checkpoint selection on a held-out validation split. Feature embeddings are L2-normalized before the Euclidean distances to the class prototypes are computed. All perturbation experiments reported in Sections IV.E–IV.H reuse these trained models without retraining; only the test episodes are altered.

Unless stated otherwise, accuracies in the perturbation studies are means over 400 episodes with 95% confidence intervals obtained from the normal approximation to the mean of the per-episode accuracies. All conditions within a study are evaluated on the same episodes, drawn once from a fixed seed, so that two conditions can be compared as a paired difference; we report the mean paired difference with its own confidence interval, which is substantially tighter than the interval of either condition and is the correct basis for judging small effects. Each study draws its own independent episode sample, so the clean reference accuracy varies by one to two points between tables; these differences are within the reported intervals and carry no meaning.

Perturbations are applied to the resized image in the $[0, 1]$ range, before ImageNet normalization, we quote both figures throughout (see Table III). For additive noise, the nominal SNR is defined on that image as $10 \log_{10}(x^2/\sigma^2)$. Normalization scales signal and noise equally, so the division by the channel standard deviation cancels from the ratio; the mean subtraction does not, and because it removes a channel mean of approximately 0.485 from spectrograms whose own mean is appreciably higher, the signal term is reduced while the noise term is unchanged. The SNR seen at the network input is therefore lower than the nominal figure by a fixed offset: we measured 13.1, 3.1 and -6.9 dB for nominal settings of 20, 10 and 0 dB respectively, and we quote both figures throughout. Time jitter is a circular shift along the image width, expressed as a percentage of that width.

TABLE III. PERFORMANCE COMPARISON OF ViT VARIANTS (HELD-OUT-BEARING TEST SET; MEAN $\pm$ 95% CI OVER 2000 EPISODES)

Model	Params	Time (s)	Train Acc	Test Acc
Simple ViT [10]	53.5M	344.8	100.0%	79.9% $\pm$ 0.6%
Parallel ViT [11]	76.7M	1510.5	80.0%	67.5% $\pm$ 0.8%
Mobile ViT [12]	2.0M	298.7	100.0%	80.5% $\pm$ 0.7%

B. Evaluation Metrics

Below are the Evaluation Metrics we employed to validate our findings: Accuracy: For a 5-way episodic task, accuracy is the fraction of query images assigned to their correct class, averaged over episodes:

$$Accuracy = \frac{1}{|E|} \sum_{e \in E} \frac{1}{|Q_e|} \sum_{q \in Q_e} \mathbb{1}_{\hat{y}_q = y_q} \quad (3)$$

Precision and Recall: These are computed per class c from the pooled episode predictions, where TP_c , FP_c and FN_c count query images of class c classified correctly, images of other classes assigned to c , and images of c assigned elsewhere:

$$Precision_c = \frac{TP_c}{TP_c + FP_c}, Recall_c = \frac{TP_c}{TP_c + FN_c} \quad (4)$$

F1-Score: The F1-Score, computed as the harmonic mean of precision and recall, offers a balanced evaluation of the model's performance:

$$F1 - Score = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (5)$$

We implemented the following parameters for all experiments, as summarized in Table IV.

TABLE IV. EXPERIMENTAL PARAMETERS

Parameter	Value
Number of classes included in each task (N WAY)	5
Quantity of support set images allocated per class (N SHOT)	1
Quantity of query set images allocated per class (N QUERY)	1
Total number of tasks used for evaluation (N EVALUATION TASKS)	2000
Total number of episodes employed during training (N TRAINING EPISODES)	800

C. Result Comparison: Simple ViT, Parallel ViT, Mobile ViT

The experimental evaluation of the three ViT variants reveals a clear accuracy-efficiency trade-off. ParallelViT, the most parameter-intensive model (76.7M parameters), generalizes least well to unseen bearings, attaining $67.5\% \pm 0.8\%$ accuracy, and its multi-branch design with 16×16 patches also incurs the highest training and inference cost. Simple-ViT (53.5M parameters) attains $79.9\% \pm 0.6\%$. The most compact model, MobileViT (2.0M parameters), attains $80.5\% \pm 0.7\%$, statistically matching SimpleViT while using under 4% of its parameters and training about five times faster than ParallelViT. These results indicate that, under a leakage-free held-out-bearing protocol, parameter count is not the primary driver of generalization on clean data. The realistic accuracies obtained are consistent with the cross-bearing difficulty reported for this benchmark [48]. Table III reports the full set of metrics. Sections IV.E–IV.I show that this clean-data ranking does not by itself determine which model should be deployed.

On an NVIDIA T4 GPU, training required 298.7 s for MobileViT, 344.8 s for SimpleViT, and 1510.5 s for ParallelViT, while the average inference time per episode was 33.9 ms, 33.0 ms, and 176.3 ms, respectively. The gap between training accuracy and held-out-bearing accuracy indicates that the protocol measures generalization to unseen bearings rather than memorization. We note that MobileViT is marginally slower per episode on the GPU than SimpleViT (33.9 against 33.0 ms) despite requiring 3.7 times fewer multiply-accumulate operations; Section IV.I shows that this is the same arithmetic-intensity effect that governs its behaviour under

compression. Fig. 8(a) illustrates the comparison between the true labels and the predictions from the simple ViT experiment. Fig. 8(b) presents a similar comparison, but for the parallel ViT experiment. Fig. 8(c) highlights the real versus predicted labels for the mobile ViT experiment.

The per-class behaviour of the three models is summarized by the confusion matrices in Fig. 9. MobileViT and SimpleViT produce strong, clean diagonals, whereas ParallelViT exhibits substantially more inter-class confusion, consistent with its lower overall accuracy.

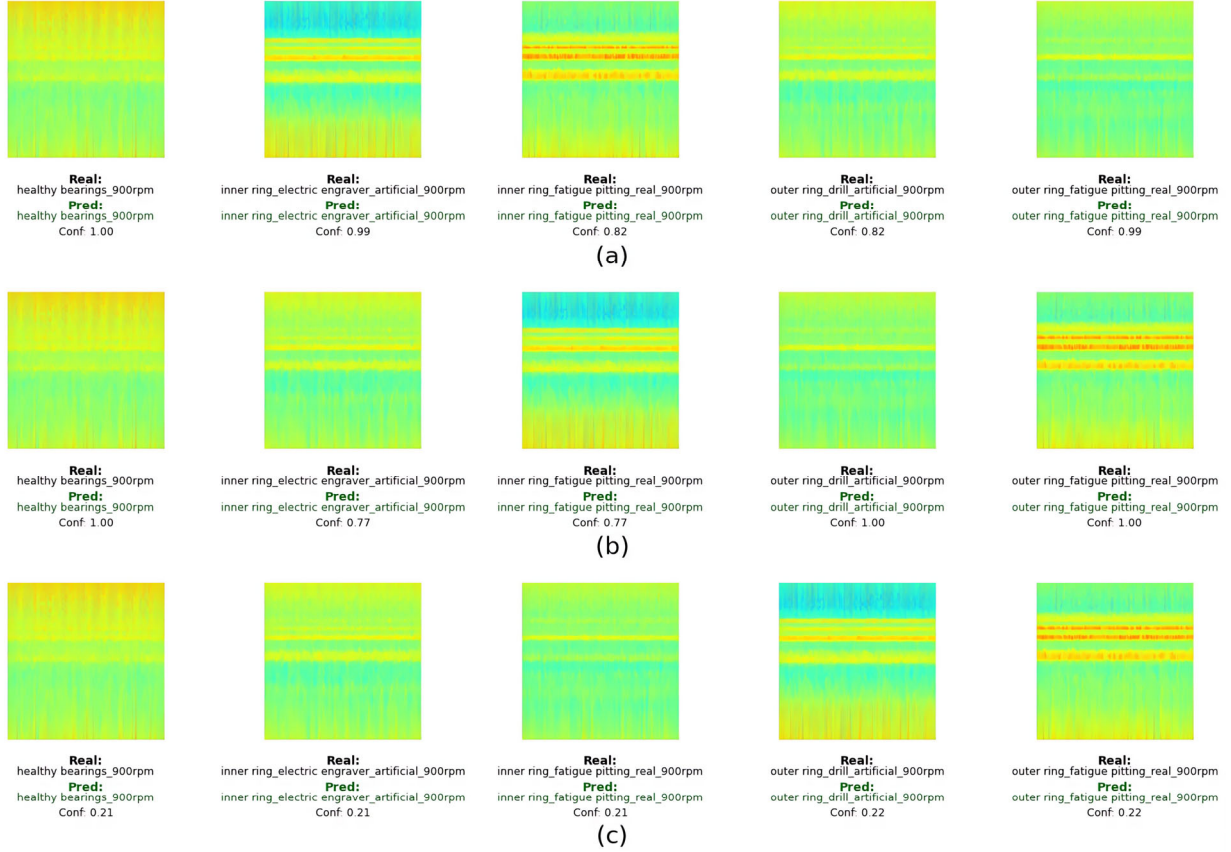


Fig. 8. True labels vs. predictions for (a) the Simple ViT experiment; (b) the Parallel ViT experiment; (c) the Mobile ViT experiment.

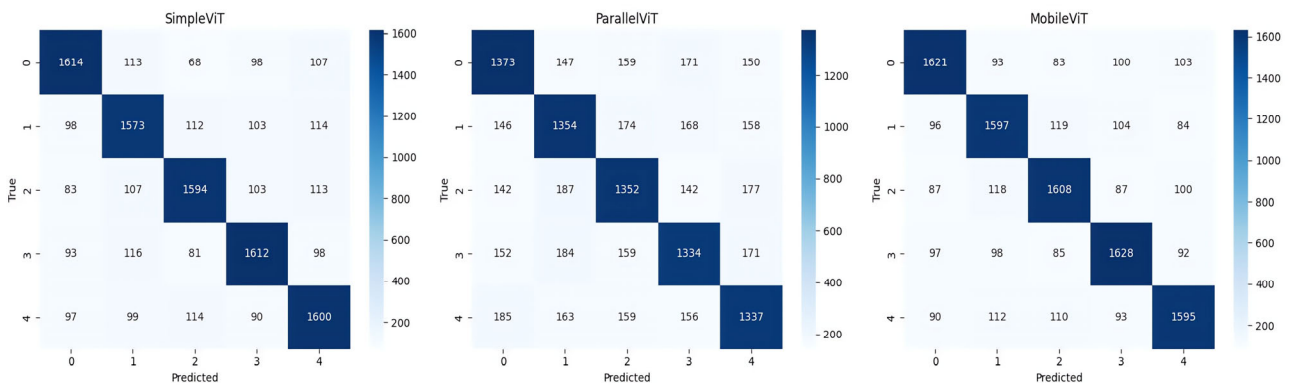


Fig. 9. Confusion matrices for SimpleViT, ParallelViT, and MobileViT on the held-out-bearing test set.

D. Per-class Behaviour

Table V reports precision, recall and F1 for each of the five classes. Performance is close to uniform: for SimpleViT and MobileViT every class lies between 0.78 and 0.81 F1, and for ParallelViT between 0.67 and 0.69. No fault type is systematically harder than the others under the uniform episodic sampling used here, and in particular

the two artificially induced damage types are not easier to recognise than the two real fatigue-pitting classes. This uniformity is a consequence of the sampling strategy rather than of the raw class frequencies, and it indicates that the class imbalance present in the underlying image counts (Table I) does not translate into a class bias at evaluation time.

TABLE V. PER-CLASS PRECISION, RECALL AND F1 ON THE HELD-OUT-BEARING TEST SET (1-SHOT, 5-WAY, 2000 EPISODES)

Class	SimpleViT			ParallelViT			MobileViT		
	P	R	F1	P	R	F1	P	R	F1
Healthy	0.81	0.81	0.81	0.69	0.69	0.69	0.81	0.81	0.81
Inner ring, electric engraver	0.78	0.79	0.78	0.67	0.68	0.67	0.79	0.80	0.79
Inner ring, fatigue pitting	0.81	0.80	0.80	0.67	0.68	0.68	0.80	0.80	0.80
Outer ring, drill	0.80	0.81	0.80	0.68	0.67	0.67	0.81	0.81	0.81
Outer ring, fatigue pitting	0.79	0.80	0.79	0.67	0.67	0.67	0.81	0.80	0.80

E. Robustness to Sensor Noise and Temporal Misalignment

Because a deployed monitoring system never observes laboratory-quality signals, we re-evaluated the three trained models under two test-time perturbations: additive white Gaussian noise at a controlled SNR, and a circular shift along the time axis emulating misalignment between the acquisition window and the shaft phase. No model was retrained, no augmentation was used at training time, and all conditions share the same 400 episodes. Table VI reports the results.

Temporal misalignment is benign. At $\pm 10\%$ jitter SimpleViT and ParallelViT are statistically unchanged and MobileViT loses 5.7 points, which is consistent with the periodic impact train being approximately shift-invariant within the analysis window. Additive noise separates the three models sharply. SimpleViT does not degrade at the mildest setting; it scores 2.4 points above its clean result, a difference at the edge of the reported intervals which we attribute to mild noise smoothing high-frequency nuisance structure in the spectrogram rather than to any genuine benefit. It then degrades steadily, falling by 48.4% points to 30.3% at the strongest setting. ParallelViT is the most noise-tolerant, statistically unchanged down to 3.1 dB at the input and still reaching 48.3% at -6.9 dB. MobileViT, the model with the highest clean accuracy in this sample, falls to 22.0% at the mildest setting, and remains at chance for a 5-way task at every stronger setting.

TABLE VI. HELD-OUT-BEARING ACCURACY (%) UNDER TEST-TIME PERTURBATION, MEAN $\pm$ 95% CI OVER 400 SHARED EPISODES. CHANCE IS 20%

Test condition	SimpleViT	ParallelViT	MobileViT
Clean	78.7 $\pm$ 1.4	68.6 $\pm$ 1.8	81.9 $\pm$ 1.6
SNR 20 dB nominal (13.1 dB input)	81.1 $\pm$ 1.3	68.7 $\pm$ 1.9	22.0 $\pm$ 1.6
SNR 10 dB nominal (3.1 dB input)	58.1 $\pm$ 1.9	68.3 $\pm$ 1.6	20.7 $\pm$ 1.6
SNR 0 dB nominal (-6.9 dB input)	30.3 $\pm$ 1.9	48.3 $\pm$ 2.1	19.9 $\pm$ 1.7
Time jitter $\pm 5\%$	79.7 $\pm$ 1.4	68.2 $\pm$ 1.8	78.6 $\pm$ 1.6
Time jitter $\pm 10\%$	79.5 $\pm$ 1.4	67.8 $\pm$ 1.8	76.2 $\pm$ 1.7

The shape of the MobileViT failure is diagnostic. A graded sensitivity would produce a gradient; an abrupt fall to chance at the mildest perturbation, flat thereafter, indicates that something in the network stops transmitting information rather than transmitting it less accurately. Measuring the embeddings confirms this: the standard deviation of the MobileViT embedding across samples

falls from 0.0177 on clean images to 0.0024, and the mean pairwise cosine similarity between query embeddings rises to 0.997. The embedding contracts to an approximately constant vector, all query-prototype distances become nearly equal, and the arg-max decision becomes uninformative. SimpleViT (0.0050 to 0.0067, cosine 0.948) and ParallelViT (0.0053 to 0.0057, cosine 0.963) show no such contraction.

F. Normalization Layers Explain the Collapse

The three architectures differ in one respect that predicts exactly this behaviour. MobileViT contains 35 BatchNorm layers in its convolutional stem and inverted-residual blocks; SimpleViT and ParallelViT contain none, using LayerNorm throughout (Table VII). BatchNorm normalizes with running statistics estimated during training. Those statistics are a property of the training input distribution, not of the individual sample, so a shift in the input distribution invalidates them for every layer simultaneously and the resulting degradation is abrupt rather than graded. LayerNorm computes its statistics per sample at inference and is therefore insensitive to this particular shift. This mechanism is established in the corruption-robustness literature, where re-estimating BatchNorm statistics on the shifted distribution at prediction time is a known and effective correction [52, 53], and where adapting the normalization layers is the basis of fully test-time adaptation methods [54]. To our knowledge it has not previously been identified as the dominant failure mode of lightweight hybrid backbones in few-shot bearing diagnosis, where compact CNN-transformer models are routinely recommended for edge deployment on the basis of clean accuracy alone. Recovery is reported as the paired change relative to the corresponding row of Table VI.

TABLE VII. NORMALIZATION LAYERS BY ARCHITECTURE AND THE EFFECT OF BATCHNORM RECALIBRATION ON UNLABELLED PERTURBED DATA

Model	Layers		Accuracy at nominal SNR (%)		
	BN	LN	20 dB	recalibrated	recovery
SimpleViT	0	15	81.1 $\pm$ 1.3	—	—
ParallelViT	0	24	68.7 $\pm$ 1.9	—	—
MobileViT	35	18	22.0 $\pm$ 1.6	48.5 $\pm$ 1.9	+26.5 $\pm$ 2.4

We tested this directly. Holding the weights fixed, we reset the BatchNorm running statistics of MobileViT and re-estimated them from 320 unlabelled perturbed images, then re-evaluated on the same episodes. Accuracy rises from 22.0% to 48.5% at the mildest setting, a paired gain of $+26.5 \pm 2.4$ points, and from 20.7% to 36.8% at the

intermediate setting, a gain of $+16.1 \pm 2.4$ points. Recalibration therefore recovers a substantial fraction of the loss without a single labelled example and without touching the learned weights, which both confirms the mechanism and supplies a practical mitigation. The procedure is the prediction-time BatchNorm adaptation of [52, 53]; our contribution here is not the technique but the diagnosis, and the observation that the model most often recommended for resource-constrained bearing monitoring is precisely the one that requires it. That recovery is partial indicates that stale normalization statistics are the dominant but not the sole factor; noise-augmented training, or test-time adaptation of the affine parameters as in [54], would be the natural next step.

The practical consequence is more useful than a bare robustness ranking. The relevant property is not that MobileViT is a hybrid or that it is small, but that it is the only one of the three carrying batch-estimated normalization statistics. Any architecture with this property should be expected to fail in the same abrupt way under a distribution shift, and should be deployed either with recalibration on data from the target installation or with the normalization replaced by a per-sample scheme.

G. Support-Set Size, Prototype Estimation and Distance Metric

The one-shot setting studied above is the hardest case, because a single support image determines the entire class representation. Table VIII reports accuracy as the number of shots increases. All three models improve monotonically from one to five shots, and the gain is largest for the two models furthest apart on clean data: MobileViT rises from 81.9% to 86.7% and ParallelViT from 68.6% to 74.7%. On a clean support set the mean prototype is at least as good as the medoid everywhere, though the paired advantage is small and reaches significance only for SimpleViT and ParallelViT at three shots ($+1.30 \pm 1.19$ and $+1.70 \pm 1.47$ points) and for MobileViT at five ($+1.50 \pm 0.87$). Averaging reduces variance; with uncontaminated supports there is nothing for an order statistic to protect against.

TABLE VIII. EFFECT OF SUPPORT-SET SIZE AND PROTOTYPE ESTIMATOR, MEAN $\pm$ 95% CI OVER 400 SHARED EPISODES PER ROW

Shots	Prototype	SimpleViT	ParallelViT	MobileViT
1-shot	mean	78.7 ± 1.4	68.6 ± 1.8	81.9 ± 1.6
3-shot	mean	82.7 ± 1.4	73.6 ± 1.8	85.4 ± 1.5
3-shot	medoid	81.4 ± 1.4	71.9 ± 1.8	84.9 ± 1.5
5-shot	mean	82.8 ± 1.4	74.7 ± 1.9	86.7 ± 1.3
5-shot	medoid	82.3 ± 1.4	73.9 ± 1.9	85.2 ± 1.4

Contamination changes the size of the problem but not the ordering of the estimators, and the reason is instructive. We replaced each support image with a heavily degraded version (0 dB nominal SNR plus $\pm 10\%$ jitter) with probability p , emulating a support sample recorded during a transient. Table IX shows that at $p = 0.4$ the one-shot configuration loses between 16.4 and 25.3 points, since a corrupted support image becomes the prototype outright. Additional shots recover much of this: at five shots with a

mean prototype the loss relative to clean falls to 10.3 points for SimpleViT and 12.2 for MobileViT.

TABLE IX. SUPPORT-SET CONTAMINATION: EACH SUPPORT IMAGE IS REPLACED BY A DEGRADED VERSION WITH PROBABILITY P

Setting	p	SimpleViT	ParallelViT	MobileViT
1-shot, mean	0.0	78.7 ± 1.4	68.6 ± 1.8	81.9 ± 1.6
1-shot, mean	0.4	62.3 ± 1.8	50.1 ± 1.7	56.6 ± 1.9
3-shot, mean	0.4	70.8 ± 1.3	51.0 ± 1.4	72.6 ± 1.5
3-shot, medoid	0.4	66.9 ± 1.6	52.5 ± 1.6	61.0 ± 1.7
5-shot, mean	0.4	72.5 ± 1.2	51.4 ± 1.4	74.5 ± 1.5
5-shot, medoid	0.4	70.3 ± 1.6	55.1 ± 1.6	63.9 ± 1.7

Note: Mean $\pm$ 95% CI over 400 shared episodes

The medoid, however, does not help, and for MobileViT it actively hurts: at $p = 0.4$ the paired difference between medoid and mean is -11.6 ± 1.5 points at three shots and -10.6 ± 1.6 at five. Only ParallelViT benefits, and only at five shots ($+3.65 \pm 1.53$). The explanation follows directly from Section IV-F: corrupted supports do not scatter, they contract toward the same degenerate region of the embedding space. A corrupted sample is therefore closer to the other embeddings than a clean one is, so the medoid criterion preferentially selects precisely the sample it was intended to reject. The embedding collapse that produces the noise failure thus also disables the standard robust-prototype remedy in the model most affected by it. The practical conclusion is that robustness here comes from acquiring more support samples and from averaging, not from an order-statistic estimator.

TABLE X. PROTOTYPE DISTANCE METRICS, WITH THE PAIRED DIFFERENCE AGAINST THE METRIC USED IN THE PAPER

Metric	acc.	Δ	acc.	Δ	acc.	Δ
Euclidean, L2-normalised	78.7	—	68.6	—	81.9	—
Cosine similarity	78.7	0.00	68.6	0.00	81.9	0.00
Euclidean, unnormalised	78.8	+0.05	74.7	+6.15	81.7	-0.20
Transductive shift	80.9	+2.20	69.9	+1.35	84.5	+2.65

Note: mean $\pm$ 95% CI over 400 shared episodes

Finally, Table X examines the choice of distance. The concern that the Euclidean distance may be too simple for high-dimensional embeddings does not apply in the form usually stated: because the embeddings are L2-normalised before the distance is computed, $\|q - p\|^2 = 2 - 2q^T p$ for unit vectors, so the Euclidean distance is a strictly decreasing function of the cosine similarity and the two metrics induce identical rankings and therefore identical predictions. The table confirms this to the last reported digit. Removing the normalisation leaves SimpleViT and MobileViT unchanged but improves ParallelViT by 6.15 ± 1.50 points, indicating that part of its discriminative information is carried in the embedding norm and discarded by normalisation. We retain the normalised metric for all reported results because it is the standard prototypical formulation and because adopting a per-model metric would confound architecture with head design, which is the comparison this study is built to isolate; the ParallelViT figure is better read as evidence that its embedding geometry is poorly conditioned than as a

recommended change. A transductive correction, shifting the support embeddings by the difference between the query and support means, gives a small but significant gain to SimpleViT ($+2.20 \pm 1.01$) and MobileViT ($+2.65 \pm 0.98$). Related metric-learning strategies, including Mahalanobis distances estimated from multiscale features, are explored in [32].

H. Frequency-Band Sensitivity of the Filter Bank

To identify which parts of the filter bank the trained models rely on, we divided the 64 bands into eight contiguous blocks, zeroed one block at a time at test time, and re-measured accuracy on the same episodes. Reporting the paired change against the clean condition rather than the raw accuracy is essential here, because several effects are smaller than the width of an unpaired interval.

The three models behave differently. SimpleViT is essentially unaffected: no block costs it more than 2.4 points, and four blocks produce a small increase, significant for blocks 0, 3, and 7, which is consistent with the occluded bands carrying more nuisance variation than fault information for this model. ParallelViT and MobileViT depend heavily on the mid-frequency region: removing block 3 costs ParallelViT 17.0 ± 2.4 points and removing block 2 costs MobileViT 15.2 ± 2.2 . No single block is indispensable for any model, so none relies on a narrow-band shortcut, but the evidence for the two weaker-generalising models is concentrated in the mid-frequency resonance region excited by localized impacts, whereas SimpleViT distributes it across the spectrum.

I. Deployment Profile and Feature Attribution

Parameter count is an incomplete proxy for deployability, so we profiled the three models directly. On the same host, MobileViT occupies 8.1 MB in FP32 against 214.1 MB for SimpleViT and 306.6 MB for ParallelViT, and requires

0.93 GMACs against 3.43 and 19.65 respectively. Its single-sample CPU latency, 76.7 ms, is the lowest of the

three but is not lower in proportion to its arithmetic cost, because depthwise-separable convolutions achieve lower arithmetic in intensity than the dense matrix products that dominate a pure transformer.

This mismatch is more pronounced after compression. Applying INT8 dynamic quantization to the linear layers reduces SimpleViT from 118.6 ms to 58.6 ms and from 214.1 MB to 53.7 MB, whereas MobileViT becomes slower, from 76.7 ms to 100.1 ms, even though its footprint falls to 4.6 MB. The reason is structural: MobileViT's cost is dominated by convolutions, which this form of quantization does not convert, so the quantize–dequantize overhead at each linear layer is paid without a corresponding saving. The practical guidance is that a lightweight architecture and a quantization-friendly architecture are not the same thing, and that a deployment target should be profiled rather than inferred from parameter count. We were not able to measure energy consumption directly, as no instrumented edge device was available; size, MAC count and single-sample CPU latency are reported as proxies, and direct measurement on embedded hardware remains future work, as it does in comparable resource-constrained monitoring applications [55].

To examine which parts of the spectrogram drive the decision, we computed a gradient-weighted activation map of the last convolutional stage of MobileViT with respect to the negative distance between the query embedding and the correct class prototype (Fig. 10). The attribution is concentrated in horizontal structures, that is, in specific frequency bands sustained across time, rather than in isolated time–frequency points, which is consistent with the band-occlusion analysis and with the physical expectation that a localized defect excites persistent resonance bands. Fig. 11 shows a two-dimensional projection of the test embeddings for the three models: SimpleViT and MobileViT produce five compact, largely separated clusters, while the ParallelViT clusters overlap substantially, which accounts for its uniformly lower per-class F1 in Table V.

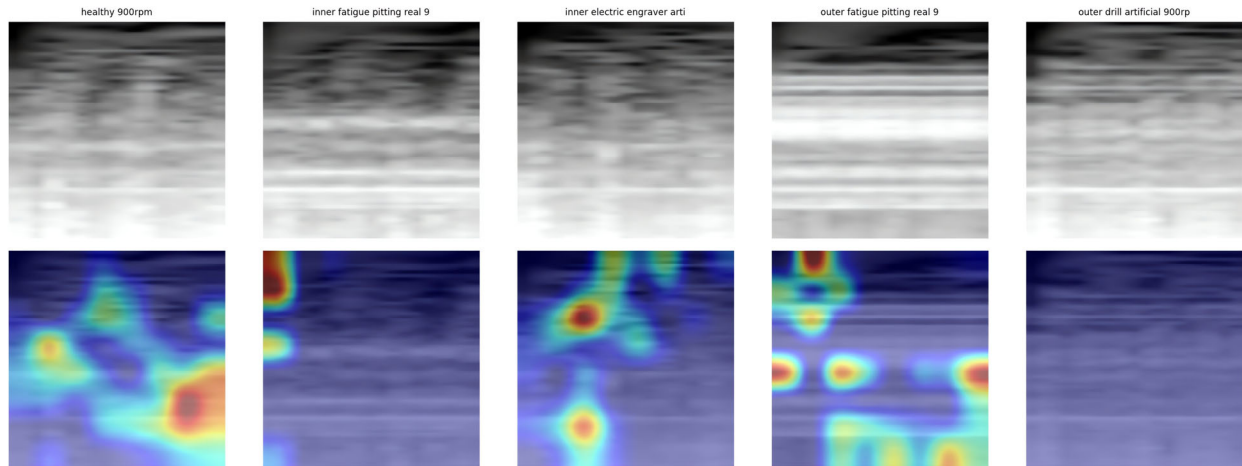


Fig. 10. Gradient-weighted attribution for MobileViT on one held-out test episode. Top row: query spectrograms. Bottom row: attribution with respect to the negative distance to the correct class prototype.

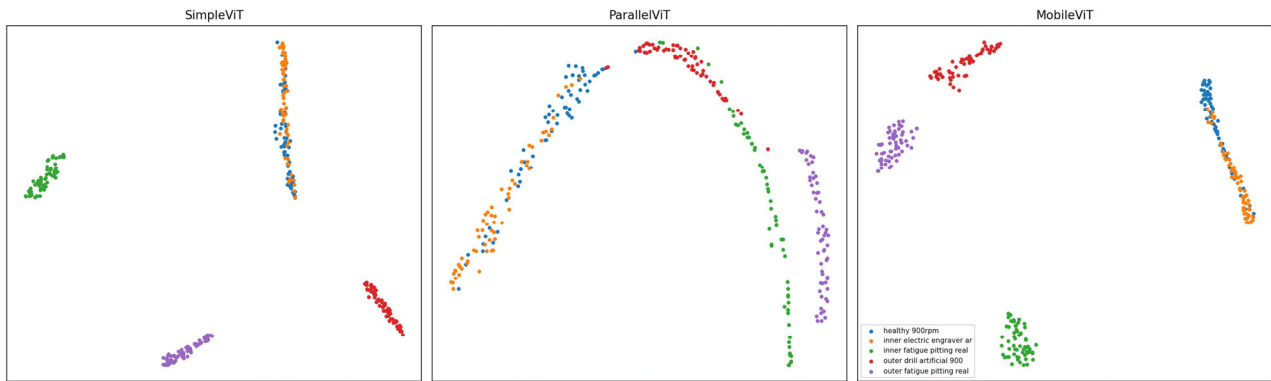


Fig. 11. t-SNE projection (perplexity 30, PCA initialisation, random seed 42) of the held-out-bearing test embeddings for SimpleViT, ParallelViT and MobileViT, using up to 60 images per class. Classes follow Table I.

The study's broader implications extend beyond mere performance metrics, highlighting the evolving landscape of ViT architectures. Each model variant Parallel ViT with its feature extraction diversity, Simple ViT with its balanced performance, and Mobile ViT with its efficiency represents a distinct approach to addressing fundamental challenges in deep learning design. These findings underscore the critical importance of aligning model architecture with specific operational constraints, whether prioritizing accuracy, robustness, deployment efficiency, or feature extraction capacity.

V. CONCLUSION

This study has demonstrated that ViT architectures, integrated with a Prototypical Network framework, can address few-shot bearing fault diagnosis under a rigorous, leakage-free held-out-bearing evaluation protocol. On clean data a compact 2.0M-parameter hybrid transformer (MobileViT) matches the accuracy of pure-transformer models with an order of magnitude more parameters, while the largest model generalizes least well; parameter count is not the primary driver of cross-bearing generalization.

Under test-time perturbation this ranking inverts, and the reason is specific rather than architectural folklore. MobileViT collapses to chance under additive sensor noise, and the collapse is caused by its BatchNorm layers, whose running statistics are estimated on the clean training distribution and are invalidated the moment that distribution shifts. Re-estimating those statistics on unlabelled perturbed data, with the learned weights untouched, recovers $+26.5 \pm 2.4$ accuracy points. The same embedding contraction also explains why the medoid prototype, the conventional remedy for a contaminated support set, fails for this model: corrupted supports contract toward a common region and are therefore selected preferentially by a centrality criterion. The actionable finding is that normalization scheme, not compactness, governs robustness here.

These results translate into a small set of practical guidelines. First, check the normalization scheme before selecting a compact backbone for a noisy installation, and if it uses batch statistics, recalibrate them on data from the target installation; this requires no labels and no retraining.

Second, acquire three to five support samples per class where possible and average them; this is the single most effective measure against a flawed prototype, and more effective than any robust estimator we tested. Third, profile the target hardware rather than extrapolating from parameter count, since INT8 quantization halved the latency of the largest transformer while making the smallest model slower. Fourth, plan for periodic re-acquisition of the support set as wear progresses; in the prototypical formulation this needs only new support images and no retraining, which is an operational advantage of metric learning over a retrained classifier.

Several limitations should be read alongside these conclusions. The evaluation uses a single operating condition (900 rpm, 0.7 Nm, 1000 N), so generalization across speeds and loads remains open. More importantly, the held-out test set contains one test bearing for three of the five classes and two for the fourth (Table II). The confidence intervals reported throughout are episode-level: they quantify sampling variability over images drawn from this fixed set of bearings, and they should not be read as intervals over bearing-to-bearing variation, which would require many more physically distinct bearings per class and which we regard as the most valuable single extension of this work. The validation split used for checkpoint selection is carved from the training images and therefore shares bearings with training; it is disjoint from the test bearings, so it cannot leak test information, but it is not an independent estimate of cross-bearing performance. Models are trained from random initialization without augmentation, which isolates architecture but leaves noise-augmented training unexamined as a mitigation for the failure mode we identify.

Several directions remain open. Extending the framework from damage-type classification to remaining-useful-life regression would let the same embedding support maintenance scheduling, and the approach should be tested on acoustic-emission and motor-current signals. Continual adaptation as damage evolves and evaluation across operating conditions are natural next steps, as is a systematic comparison of the filter-bank front end against wavelet and decomposition-based time-frequency representations; we note that a controlled comparison of front ends would require regenerating the image dataset and retraining every architecture under each representation,

so that the transform is not confounded with the training conditions, and we regard this as a study of comparable scope to the present one rather than an extension of it. Test-time adaptation of normalization statistics, of which our recalibration experiment is the simplest instance, is a further avenue suggested directly by our results. Finally, deployment in a safety-critical setting carries obligations beyond accuracy: a system with roughly 80% one-shot accuracy is a decision aid rather than an autonomous authority, its confidence should be exposed to the operator, and the failure mode documented in Section IV-F argues for an explicit input-distribution check ahead of the classifier rather than reliance on graceful degradation.

CONFLICT OF INTEREST

The authors declare no conflict of interest.

AUTHOR CONTRIBUTIONS

F.M.S.: Conceptualization, Methodology, Software, Formal analysis, Investigation, Data curation, Writing. S.T.K.A.: Conceptualization, Methodology, Software, Formal analysis, Investigation, Data curation, Writing. M.Z.: Writing—review and editing. J.U.: Supervision, Project administration, Funding acquisition. All authors have read and approved the final version.

FUNDING

This research is funded by Woosong University Academic Research 2026.

REFERENCES

- [1] M. E. Haque, M. N. Absur, F. Al Farid, M. K. Siam, A. Uddin, and H. A. Karim, "Bearfusionnet: A multi-stream attention-based deep learning framework with explainable AI for accurate detection of bearing casting defects," *Comput. Mater. Contin.*, vol. 86, no. 3, 33, 2026.
- [2] B. Sun and X. Liu, "Bearing early fault detection and degradation tracking based on support tensor data description with feature tensor," *Appl. Acoust.*, vol. 188, 108530, 2022.
- [3] A. Chennana, A. C. Megherbi, N. Bessous *et al.*, "Vibration signal analysis for rolling bearings faults diagnosis based on deep-shallow features fusion," *Sci. Rep.*, vol. 15, 9270, 2025.
- [4] A. Nabhan, "Vibration analysis of adding contaminants particles and carbon nanotubes to lithium grease of ball bearing," *Vibroengineering PROCEDIA*, vol. 8, pp. 28–32, Oct. 2016.
- [5] A. Nabhan, M. Nouby, A. M. Samy, and M. O. Mousa, "Vibration analysis of deep groove ball bearing with outer race defect using Abaqus," *J. Low Freq. Noise Vib. Act. Control*, vol. 35, no. 4, pp. 312–325, 2016.
- [6] A. Nabhan, M. Nouby, and A. M. Samy, "Multiple defects detection in outer race of gearbox ball bearing using time domain statistical parameters," *Int. J. Veh. Struct. Syst.*, vol. 8, no. 3, pp. 167–174, 2016.
- [7] A. Nabhan and A. Rashed, "Experimental and numerical investigation of defect-size estimation in taper roller bearing," *Noise Vib. Worldwide*, vol. 49, no. 11, pp. 345–354, 2018.
- [8] M. E. Haque, M. Zabin, and J. Uddin, "EnsembleX-AIMotor: A lightweight framework for fault classification in electric vehicle drive motors using feature selection, ensemble learning, and explainable AI," *Machines*, vol. 13, no. 4, 2025.
- [9] A. Dosovitskiy, L. Beyer, A. Kolesnikov *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," arXiv preprint, arXiv:2010.11929, 2020.
- [10] L. Beyer, X. Zhai, and A. Kolesnikov, "Better plain vit baselines for imagenet-1k," arXiv preprint, arXiv:2205.01580, 2022.
- [11] H. Touvron, M. Cord, A. El-Nouby, J. Verbeek, and Hervé Jégou, "Three things everyone should know about vision transformers," arXiv preprint, arXiv:2203.09795, 2022.
- [12] S. Mehta and M. Rastegari, "Mobilevit: Light-weight, general-purpose, and mobile-friendly vision transformer," in *Proc. International Conference on Learning Representations*, April 25–29, 2022.
- [13] J. Snell, K. Swersky, and R. Zemel, "Prototypical networks for few-shot learning," in *Proc. the 31st International Conference on Neural Information Processing Systems*, NIPS'17, 2017, pp. 4080–4090.
- [14] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2020, pp. 213–229.
- [15] Y. Li, K. Zhang, J. Cao *et al.*, "Localvit: Analyzing locality in vision transformers," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, 2023, pp. 9598–9605.
- [16] M. S. Ryoo, A. Piergiovanni, A. Arnab, M. Dehghani, and A. Angelova, "Tokenlearner: What can 8 learned tokens do for images and videos?" arXiv preprint, arXiv:2106.11297, 2022.
- [17] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in Neural Information Processing Systems*, vol. 30, 2017.
- [18] O. Vinyals, C. Blundell, T. Lillicrap, D. Wierstra *et al.*, "Matching networks for one shot learning," *Advances in Neural Information Processing Systems*, vol. 29, pp. 3637–3645, 2016.
- [19] F. Sung, Y. Yang, L. Zhang, T. Xiang, P. H. S. Torr, and T. M. Hospedales, "Learning to compare: Relation network for few-shot learning," in *Proc. the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 1199–1208.
- [20] F. M. Siraj, S. T. K. Ayon, M. A. Samad, J. Uddin, and A. Choi, "Few-shot lightweight SqueezeNet architecture for induction motor fault diagnosis using limited thermal image dataset," *IEEE Access*, vol. 12, pp. 50986–50997, 2024.
- [21] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *Proc. International Conference on Machine Learning*, PMLR, 2017, pp. 1126–1135.
- [22] S. Ravi and H. Larochelle, "Optimization as a model for few-shot learning," in *Proc. International Conference on Learning Representations*, April 24–26, 2017.
- [23] K. Lee, S. Maji, A. Ravichandran, and S. Soatto, "Meta-learning with differentiable convex optimization," in *Proc. 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 10649–10657.
- [24] Y. Wang, Q. Yao, J. T. Kwok, and L. M. Ni, "Generalizing from a few examples: A survey on few-shot learning," *ACM Comput. Surv.*, vol. 53, no. 3, 2020.
- [25] V. G. Satorras and J. B. Estrach, "Few-shot learning with graph neural networks," in *Proc. International Conference on Learning Representations*, April 30–May 3, 2018.
- [26] Y. Liu, J. Lee, M. Park, S. Kim, E. Yang, S. Hwang, and Y. Yang, "Learning to propagate labels: transductive propagation network for few-shot learning," in *Proc. International Conference on Learning Representations*, May 3–9, 2019.
- [27] H. Zhong, Y. Lv, R. Yuan, and D. Yang, "Bearing fault diagnosis using transfer learning and self-attention ensemble lightweight convolutional neural network," *Neurocomputing*, vol. 501, pp. 765–777, 2022.
- [28] H. Fang, J. Deng, Y. Bai, B. Feng, S. Li, S. Shao, and D. Chen, "CLFormer: A lightweight transformer based on convolutional embedding and linear self-attention with strong robustness for bearing fault diagnosis under limited sample conditions," *IEEE Transactions on Instrumentation and Measurement*, vol. 71, pp. 1–8, 2022.
- [29] J. Lin, H. Shao, X. Zhou, B. Cai, and B. Liu, "Generalized MAML for few-shot cross-domain fault diagnosis of bearing driven by heterogeneous signals," *Expert Systems with Applications*, vol. 230, 120696, 2023.
- [30] Z. Lei, P. Zhang, Y. Chen, K. Feng, G. Wen, Z. Liu, R. Yan, X. Chen, and C. Yang, "Prior knowledge-embedded meta-transfer learning for few-shot fault diagnosis under variable operating conditions," *Mechanical Systems and Signal Processing*, vol. 200, 110491, 2023.

- [31] L. Zeng, J. Jian, X. Chang, and S. Wang, "A meta-learning method for few-shot bearing fault diagnosis under variable working conditions," *Measurement Science and Technology*, vol. 35, no. 5, 056205, 2024.
- [32] M.-H. Vu, V.-Q. Nguyen, T.-T. Tran, V.-T. Pham, and M.-T. Lo, "Few-shot bearing fault diagnosis via ensembling transformer-based model with mahalanobis distance metric learning from multiscale features," *IEEE Transactions on Instrumentation and Measurement*, vol. 73, pp. 1–18, 2024.
- [33] H. Wang, C. Li, P. Ding, S. Li, T. Li, C. Liu, X. Zhang, and Z. Hong, "A novel transformer-based few-shot learning method for intelligent fault diagnosis with noisy labels under varying working conditions," *Reliability Engineering & System Safety*, vol. 251, 110400, 2024.
- [34] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proc. 2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, IEEE Computer Society, 2021, pp. 9992–10002.
- [35] I. Bello, B. Zoph, Q. Le, A. Vaswani, and J. Shlens, "Attention augmented convolutional networks," in *Proc. 2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019, pp. 3285–3294.
- [36] S. Khan, M. Naseer, M. Hayat, S. W. Zamir, F. S. Khan, and M. Shah, "Transformers in vision: A survey," *ACM Computing Surveys*, vol. 54, no. 10, 2022.
- [37] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "MobileNets: Efficient convolutional neural networks for mobile vision applications," arXiv preprint, arXiv:1704.04861, 2017.
- [38] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.C. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," in *Proc. the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4510–4520.
- [39] Y. Chen, X. Dai, D. Chen, M. Liu, X. Dong, L. Yuan, and Z. Liu, "Mobile-former: Bridging MobileNet and transformer," in *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 5260–5269.
- [40] M. Tan, B. Chen, R. Pang, V. Vasudevan, M. Sandler, A. Howard, and Q. V. Le, "MnasNet: Platform-aware neural architecture search for mobile," in *Proc. 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 2815–2823.
- [41] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. Jégou, "Training data-efficient image transformers & distillation through attention," in *Proc. 38th Int. Conf. Mach. Learn. (ICML)*, 2021, vol. 139, pp. 10347–10357.
- [42] S. Han, H. Mao, and W. J. Dally, "Deep compression: Compressing deep neural networks with pruning, trained quantization and Huffman coding," in *Proc. International Conference on Learning Representations*, May 2–4, 2016.
- [43] P. K. A. Vasu, J. Gabriel, J. Zhu, O. Tuzel, and A. Ranjan, "Fastvit: A fast hybrid vision transformer using structural reparameterization," in *Proc. the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 5762–5772.
- [44] N. Costa and L. Sánchez, "Few-shot generative compression approach for system health monitoring," *Reliab. Eng. Syst. Saf.*, vol. 256, 110749, 2025.
- [45] Z. Lu, S. He, X. Zhu, L. Zhang, Y.-Z. Song, and T. Xiang, "Simpler is better: Few-shot semantic segmentation with classifier weight transformer," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, 2021, pp. 8721–8730.
- [46] S. T. K. Ayon, F. M. Siraj, and J. Uddin, "Steel surface defect detection using learnable memory vision transformer," *Computers, Materials & Continua*, vol. 82, no. 1, pp. 499–520, 2025.
- [47] M. Tan and Q. V. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," in *Proc. the 36th International Conference on Machine Learning*, 2019, pp. 6105–6114.
- [48] C. Lessmeier, J. K. Kimotho, D. Zimmer, and W. Sextro, "Condition monitoring of bearing damage in electromechanical drive systems by using motor current signals of electric motors: A benchmark data set for data-driven classification," in *Proc. PHM Soc. Eur. Conf.*, 2016, vol. 3.
- [49] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2020, pp. 9726–9735.
- [50] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *Proc. 37th Int. Conf. Mach. Learn. (ICML)*, 2020, pp. 1597–1607.
- [51] F. M. Siraj, S. T. K. Ayon, and J. Uddin, "A few-shot learning based fault diagnosis model using sensors data from industrial machineries," *Vibration*, vol. 6, no. 4, pp. 1004–1029, 2023.
- [52] S. Schneider, E. Rusak, L. Eck, O. Bringmann, W. Brendel, and M. Bethge, "Improving robustness against common corruptions by covariate shift adaptation," in *Proc. 34th Int. Conf. Neural Inf. Process. Syst. (NeurIPS)*, 2020, 968, pp. 11539–1155.
- [53] Z. Nado, S. Padhy, D. Sculley, A. D'Amour, B. Lakshminarayanan, and J. Snoek, "Evaluating prediction-time batch normalization for robustness under covariate shift," arXiv preprint, arXiv:2006.10963, 2021.
- [54] D. Wang, E. Shelhamer, S. Liu, B. Olshausen, and T. Darrell, "Tent: Fully test-time adaptation by entropy minimization," in *Proc. International Conference on Learning Representations*, May 3–7, 2021.
- [55] J. Millar, H. Haddadi, and A. Madhavapeddy, "Energy-aware deep learning on resource-constrained hardware," arXiv preprint, arXiv:2505.12523, 2025.

Copyright © 2026 by the authors. This is an open access article distributed under the Creative Commons Attribution License which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited ([CC BY 4.0](https://creativecommons.org/licenses/by/4.0/)).