1.Dataset Details (Cable)

- Number of defect classes: 4
- Types of defect classes:
    1. Bent_wire
    2. Cable_swap
    3. Missing_wire
    4. Mouse_bite
- Number of images used in the dataset:
    - Total: 392 images
    - Normal images: 224
    - Defective images: 168
- Distribution of training and test data:
    - Training: 224 normal images (no defects)
    - Testing: 168 images (40 normal, 128 defective)
- Image dimensions: 1024 × 1024 pixel

2.**Attempt 1**

This experiment fine-tunes a pre-trained ResNet-18 by replacing its final layer with a custom classifier incorporating regularization, and applies adaptive average pooling to standardize the feature map dimensions across samples.

Hyperparameters:

- Batch size: 48, Epochs: 60
- Loss: CrossEntropyLoss(label_smoothing=0.1)
- Optimizer: AdamW(lr=7e-4, weight_decay=1e-3)
- Scheduler: CosineAnnealingLR, eta_min = 5e-6

Attempt 2

This variant enhances ResNet-18 by integrating self-attention modules after layers 1 and 2, coupled with dropout for improved regularization. The goal is to boost spatial feature extraction tailored to the task.

Hyperparameters:

- Batch size: 32, Epochs: 60
- Loss: CrossEntropyLoss(label_smoothing=0.13)
- Optimizer: AdamW(lr=1.2e-3, weight_decay=3.1e-4)
- Scheduler: CosineAnnealingLR, eta_min = 3e-6

Attempt 3

This approach adapts a pre-trained ResNet-18 by freezing its earlier layers and inserting DropBlock2D regularization after layers 3 and 4. It also leverages adaptive average pooling and a redesigned classifier, modifying the standard TorchVision

implementation.

Hyperparameters:

- Batch size: 48, Epochs: 70
- Loss: CrossEntropyLoss(label_smoothing=0.05)
- Optimizer: AdamW(lr=3e-3, weight_decay=3e-4)
- Scheduler: CosineAnnealingLR, eta_min = 1e-6

Attempt 4

In this experiment, the default average pooling in ResNet-18 is replaced by learnable GeM pooling to enable more adaptive feature aggregation. The model fine-tunes layers 3, 4, and the final classifier (comprising dropout and a linear layer), while earlier layers are frozen.

Hyperparameters:

- Batch size: 48, Epochs: 50
- Loss: CrossEntropyLoss(label_smoothing=0.11)
- Optimizer: AdamW(lr=8.1e-4, weight_decay=1.6e-3)
- Scheduler: CosineAnnealingLR ,eta_min = 5e-6

Result:

Attempt 4 achieves the best generalization performance, with a validation accuracy of approximately 93.75%. Its success is attributed to the use of **learnable GeM pooling** for refined feature aggregation and **targeted fine-tuning** of higher-level layers, which together foster more efficient and effective representation learning.

3.(i) **Long-tail distribution** describes a probability distribution characterized by a small number of high-frequency categories (the "head" of the distribution) and a significantly larger number of low-frequency categories (the "long tail"). In the context of datasets, this manifests as a few classes possessing a disproportionately large number of samples, while the vast majority of classes have considerably fewer instances. This inherent imbalance can pose substantial challenges for machine learning models, often leading to biased learning and suboptimal generalization, particularly for the underrepresented tail classes.

(ii) A notable recent work addressing data imbalance in long-tailed recognition is "Constructing Balance from Imbalance for Long-tailed Image Recognition" by Yue Xu et al. (ECCV 2022). This paper introduces a novel approach that aims to construct more balanced feature representations despite the initial data imbalance. Their method involves generating synthetic feature representations for the tail classes through interpolation within the learned feature space. By creating these virtual samples, the model is exposed to a more balanced distribution of features across all classes, thereby mitigating the bias towards head classes and enhancing the learning

of discriminative features for the less frequent tail classes.Applying this methodology to the MVTec AD dataset, where the 'Good' class represents the head and the various defect classes form the tail, could involve the following: After training an initial feature embedding network, the proposed method could be used to generate synthetic feature vectors for each of the defect classes. These generated features, along with the original imbalanced data, would then be used to train a classifier. This augmentation of the tail classes with synthetically generated features could lead to a more balanced training process.

3.

To address the challenge of anomaly detection with predominantly 'good' images in the MVTec AD dataset, unsupervised and self-supervised methods are essential. One common approach is to train reconstructive models like autoencoders, GANs, or VAEs on normal data; anomalies are detected by their higher reconstruction errors. Another strategy uses pre-trained CNNs to extract features from 'good' images, followed by one-class classification or clustering in feature space to identify deviations. Additionally, self-supervised contrastive learning can enhance representation learning by distinguishing subtle differences without requiring anomaly labels, improving robustness in detecting defects.

4.

(i) Dataset Preparation for Fine-tuning

To fine-tune object detection models like YOLO-World, the dataset must include bounding box annotations, where each defect is labeled with a class and its corresponding coordinates in the image (e.g., in COCO or Pascal VOC format). For segmentation models such as SAM (Segment Anything Model), pixel-level masks are required—each defect should be precisely delineated within a binary or instance segmentation mask. In both cases, high-quality, human-verified annotations are critical, as these models rely heavily on spatial precision to learn effective visual representations.

(ii) Suitability for Fine-tuning

YOLO-World and SAM are pre-trained on large, diverse datasets, making them highly transferable. YOLO-World offers fast and accurate object localization across categories, ideal for detecting various defect types. SAM, with its zero-shot segmentation capability, is particularly powerful for learning fine-grained structures from limited data. Their strong generalization ability allows effective adaptation to the MVTec AD dataset, improving defect detection and localization performance under class imbalance.