HW3 Deep Learning and Industrial Applications

1.

Number of defect classes	7
Types of defect classes	broken_teeth、combined、fabric_border、fabric_interior、rough、split_teeth、squeezed_teeth
Number of images used in your dataset	train：good：240 test ：good：32 broken_teeth：30 combined：29 fabric_border：30 fabric_interior：30 rough：30 split_teeth：30 squeezed_teeth：30
Distribution of training and test data	train：good：240(100%) test ：good：32(13.2%) broken：209(86.8%)
Image dimensions	1024 x 1024 pixels

2.

Method	Model	Optimizer	Data Augmentation	Epochs	LR Scheduler	Test accuracy
1	ResNet50	Adam	Standard	50	CosineAnnealing	25%
2	ResNet18	Adam	Strong	50	CosineAnnealing	18.75%
3	ResNet18	SGD	Standard	50	StepLR	31.25%
4	ResNet18 (fine-tune)	Adam	Standard	50	CosineAnnealing	25%

Based on test accuracy, Method 3 (ResNet18 + SGD + StepLR) shows the best performance with a test accuracy of 31.25%. Although Method 1 (ResNet50 + Adam + CosineAnnealing) and Method 4 (ResNet18 + Adam + Fine-tuning) both achieved a test accuracy of 25%, Method 3 outperforms with its learning rate scheduler and optimizer choice. Method 2 (ResNet18 + Adam + Strong Augmentation) performs the worst with only 18.75% accuracy. Therefore, Method 3 is the most effective model.

3. (i) Long-tail distribution refers to a situation in a dataset where most samples are

concentrated in a few classes, while the majority of classes have very few samples, creating a "long tail". This means that some classes have a significantly larger number of samples than others, and the rare classes have a notably lower number of samples.

(ii) Paper Summary on Data Imbalance Solution:

- Title: "Class-Balanced Loss Based on Effective Number of Samples"
- Authors: Cui, Y., Jia, M., Lin, T.-Y., Song, Y., & Belongie, S. (2019)

This paper proposes a class-balanced loss function based on the effective number of samples, which adjusts class weights according to the number of samples to address the data imbalance problem. This method can effectively improve the detection of minority classes, preventing the model from being biased towards the majority classes. For the MVTec AD dataset, this approach can reduce the dominance of the "Good" class in training and improve the model's ability to identify defect classes.

4. Since the MVTec AD dataset's training set primarily consists of "good" images and lacks examples of defects, the following strategies can be used to develop an anomaly detection model:
   1. Autoencoders: Used to learn the reconstruction of "good" images and identify anomalies based on reconstruction errors.
   2. Generative Adversarial Networks (GANs): Generate synthetic "good" images and compare them with real images.
   3. Self-Supervised Learning: Learn data structure from unlabeled data to detect anomalies.

5. (i) Dataset Preparation:
   1. Object Detection: This requires a dataset with bounding boxes annotated for each object, specifying its position and the class of the object.
   2. Segmentation: This requires a dataset with pixel-level annotations, where each pixel is labeled according to the class of the object it belongs to, typically with a class mask.

   (ii) Models like YOLO-World and SAM have been pre-trained on large-scale datasets and possess strong feature extraction capabilities. These models can be adapted for various vision tasks, and due to their pre-learned rich features, fine-tuning them on our custom MVTec AD dataset allows us to leverage the features they have already learned, making them highly effective at detecting anomalies or defects within the images, thus improving anomaly detection accuracy.