1. Bottle Dataset
    - Number of defect classes : 3
    - Types of defect classes : broken_large, broken_small, contamination
    - Number of images used in dataset : 292
    - Distribution of training and test data :

	Training data	Test data
Good	209	20
broken_large	0	20
broken_small	0	22
contamination	0	21

    - Image dimensions : 900*900*3

2.

	Model	Accuracy
Attempt 1	Pretrained Resnet 18	91.57%
Attempt 2	**Fine-tune Pretrained Resnet 18**	**97.59%**
Attempt 3	Pretrained VGG 16	92.77%
Attempt 4	Fine-tune Pretrained VGG 16	96.39%

All the attempts are resized to 224*224, and the batch size equals 32. For Attempt 2 and Attempt 4, epoch=10, learning rate=5e-5, optimizer is Adam.

Initially, I tried rotations and flips for data augmentation for all attempts but discovered that performance was better without them. This may be because the good sample images of the bottle are highly symmetrical, where excessive rotation or flipping is not beneficial.

Moreover, I observed that pretrained VGG16 outperformed pretrained ResNet18, and the fine-tuned models consistently surpassed their non-fine-tuned counterparts. This might be because **fine-tuning** allows the model to adjust its weights so that the extracted features become more aligned with the data distribution of the target dataset, thereby enhancing sensitivity to tiny changes during anomaly detection.

Setting the epoch too high during fine-tuning led to worse performance, which is consistent with overfitting risks when dealing with a small set of standard samples. The best result came from fine-tuned ResNet18 (Attempt 2), possibly because ResNet's **skip connections** can help stabilize gradient

flow, and with the proper hyperparameters, it adapts well to the dataset.

3.

(i) Long-Tail Distribution

A long-tail distribution refers to a situation where a few classes (head classes) have many samples while most classes (tail classes) have only a few samples. This imbalance can lead to biased model training, as the model may overfit the head classes and underperform in the tail classes.

(ii) A solution to data imbalance

Dablain et al. (2022) propose a novel oversampling method for deep learning models and imbalanced image data. The core idea is to use an encoder-decoder architecture. An encoder maps an input image into a compact, low-dimensional feature space that captures its essential characteristics, while a decoder reconstructs the original image from this condensed representation. Once the images are represented in this lower-dimensional feature space, a SMOTE-inspired technique is applied. Traditional SMOTE generates synthetic minority samples by interpolating between existing ones; this concept is extended to encoded features. New synthetic features are created by mixing the features of minority class examples, and then the decoder converts these synthetic features back into high-quality artificial images.

While our application involves training only on normal samples, the ideas in DeepSMOTE can still be helpful. The encoder-decoder framework could be adapted to learn robust representations of our normal data, and a modified oversampling strategy might be used to simulate subtle anomalies. This could enrich our training set and improve the sensitivity of our model.

4. When training an anomaly detection model on the MVTec AD dataset where the training set consists almost entirely of defect-free images, the focus shifts to learning a robust representation of what constitutes normality. For instance, an autoencoder can be trained to reconstruct images of flawless products. Since the network becomes proficient at reproducing these normal images, any input containing defects will likely

result in a noticeably higher reconstruction error, signaling an anomaly. Similarly, a generative adversarial network may be employed, where the generator learns to create realistic images of non-defective items, and the discriminator becomes adept at identifying subtle discrepancies when defective images are encountered. Another approach involves extracting features using a pre-trained network and computing distances in the feature space; images far from the established distribution of normal samples can be marked as anomalous.

5.

(i) Data for object detection and for segmentation

For object detection, preparing a dataset with images annotated using bounding boxes that clearly mark the location of potential defects is essential. Each defect should be labeled with its corresponding class or type, which helps the model learn where and what to look for.

For segmentation, the dataset should include images with pixel-level annotations, where every pixel is labeled according to whether it belongs to a defect or the normal background. This level of detail enables the model to capture each anomaly's exact shape and boundary.

(ii) Why are these models suitable for fine-tuning?

These models are suitable for fine-tuning our custom dataset because they come pre-trained on extensive, diverse datasets, which gives them robust feature extraction capabilities. Their architectures are designed to manage complex visual tasks, allowing them to effectively learn and generalize the nuanced patterns of defects from a limited set of annotated examples. Fine-tuning such models helps adapt these learned features to the anomalies present in our dataset, leading to more precise defect identification and segmentation.

Reference

Dablain, D., Krawczyk, B., and Chawla, N. V. (2022), "DeepSMOTE: Fusing deep learning and SMOTE for imbalanced data," *IEEE transactions on neural networks and learning systems*, Vol. 34, No. 9, pp. 6390-6404.