1. (10 points) Download the MVTec Anomaly Detection Dataset from Kaggle (here). Select one type of product from the dataset. Document the following details about your dataset:
   - Number of defect classes.
   - Types of defect classes.
   - Number of images used in your dataset.
   - Distribution of training and test data.
   - Image dimensions.

Dataset name: bottle

Number of defect classes: 4

Types of defect classes: good, broken_large, broken_small, contamination

Number of images used in dataset: 4 * 20(num_image of each classes)

Distribution of training and test data (0.8, 0.2)

Image dimensions: 3 * 900 * 900

2. (30 points) Implement **4** different attempts to improve the model's performance trained on the dataset you choose in previous question. Ensure that at least one approach involves modifying the pre-trained model from TorchVision. Summarize the outcomes of each attempt, highlighting the best performing model and the key factors contributing to its success. You may also need to describe other hyperparameters you use in your experiment, like epochs, learning rate, and optimizer. (Approximately 150 words.)

id	resize	epoch	lr	optimizer	batch_size	resnet	weights
1	32*32	50	0.001	adam	32	resnet18	IMAGENET1K_V1'
2	64*64	50	0.001	adam	32	resnet18	IMAGENET1K_V1'
3	32*32	200	0.001	adam	32	resnet18	IMAGENET1K_V1'
4	32*32	100	0.001	adam	16	resnet18	'IMAGENET1K_V1'
5	32*32	100	0.001	adam	32	resnet50	'IMAGENET1K_V1'
6	32*32	100	0.001	adam	32	resnet50	IMAGENET1K_V2'
7	32*32	300	0.001	adam	32	resnet18	IMAGENET1K_V1'
8	128*128	150	0.001	adam	32	resnet18	IMAGENET1K_V1'

result_id	train loss	train acc	val loss	val acc	test acc
1	1.2119	45.31%	1.2948	37.50%	50%
2	0.7748	73.44%	1.6478	25%	37.50%
3	0.9935	65.63%	0.9746	75%	87.50%
4	1.2198	46.88%	1.0356	56.25%	75%
5	1.0238	54.69%	1.0549	68.75%	75%
6	1.1779	60.94%	1.2166	37.50%	69%
7	0.9904	59.38%	0.9522	68.75%	75%
8	0.9844	60.88%	1.0425	68.75%	75%

The key factors of id(3) I think is epoch, which set up from 50 to 200.

3. (20 points) In real-world datasets, we often encounter long-tail distribution (or data imbalance). In MVTec AD dataset, you may observe that there are more images categorized under the 'Good' class compared to images for each defect class. (Approximately 150 words.)
   (i) (5 points) Define what is 'long-tail distribution.'
   (ii) (15 points) Identify and summarize a paper published after 2020 that proposes a solution to data imbalance. Explain how their method could be applied to our case.

(i) A long-tail distribution is a probability distribution that exhibits a large number of occurrences far from the "head" or central part of the distribution.

(ii) DeepSMOTE: Fusing Deep Learning and SMOTE for Imbalanced Data. DeepSMOTE builds on the concept of Synthetic Minority Over-sampling Technique (SMOTE) but extends its application to deep learning models, particularly focusing on generating high-quality, sharp, and information-rich outputs without requiring a discriminator network. The method operates end-to-end, learns representations of the raw data to embed into a lower-dimensional feature space for oversampling, and generates output that can be visually inspected
https://ar5iv.labs.arxiv.org/html/2105.02340

4. (20 points) The MVTec AD dataset's training set primarily consists of 'good' images, lacking examples of defects. Discuss strategies for developing an anomaly detection model under these conditions. (Approximately 100 words.)

One of the strategies is like just training the model focus on only the 'good' images, and justify whether it is good or not, but this kind of solution can't recognize what kind of defects occured. Or we can go with some other method like Semi-Supervised Anomaly Detection which leverages a small set of labeled data (if available) along with a larger set of unlabeled data, and can be particularly effective when you have a limited amount of 'defective' samples.

(Adapted from chatgpt)

5. For the task of anomaly detection, it may be advantageous to employ more sophisticated computer vision techniques such as object detection or segmentation. This approach will aid in identifying defects within the images more accurately. Furthermore, there are numerous open-source models designed for general applications that can be utilized for this purpose, including YOLO-World (website) and SAM (website). (Approximately 150 words.)
   (i) (10 points) To leverage these powerful models and fine-tune them using our dataset, it is necessary to prepare specific types of datasets. What kind of data should be prepared for object detection and for segmentation.
   (ii) (10 points) Why are these models suitable for fine-tuning for our custom dataset?

(i) Object detection: You should prepare a dataset that includes images with annotations for each object of interest in the images. (e.g., the x and y coordinates of the top left corner, width, and height, and the class label for each object detected in the image).

Segmentation: Segmentation models require a dataset where each pixel in the image is labeled with the class of the object it belongs to.

(ii) There have some reasons:

Transfer Learning: models like YOLO and SAM have been pre-trained on large, diverse datasets.

Hight Performance: these models are designed to achieve high performance in terms of accuracy and speed.

Customizability: Both models are highly customizable.

By fine-tuning these models on a dataset prepared specifically for your application, you can achieve high precision in detecting anomalies within images, leveraging the advanced capabilities of these models to identify defects more accurately than with general anomaly detection approaches.

(Adapted from chatgpt)