National Tsing Hua University

11320IEEM 513600

Deep Learning and Industrial Applications

Homework 3

Name: 黃蘇珊 Student ID: 113034421

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:

Bottle

```
📊 Dataset Summary for 'bottle':
• Number of defect classes: 3
• Types of defect classes: ['broken_small', 'broken_large', 'contamination']
• Total images used: 292
    - Training images (only 'good'): 209
    - Test images (good + defective): 83
• Image dimensions: 900 x 900 x 3
```

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.)

Hyperparameters for All Attempts: 50 epochs, batch size = 32, Adam optimizer, learning rate = 0.001, CrossEntropyLoss, and CosineAnnealingLR as the learning rate scheduler.

Attempt	Model	Pre-trained model from TorchVision?	Augmentation	Training Accuracy	Final Test Accuracy	Overfit	Notes
1	ResNet18 + fine-tuned FC	Yes	Basic (Resized +Normalization)	Consistently >95%	92.77%	No	Loss Curves: Stable and smooth over epochs
2	CNN Baseline + Dropout (p=0.5)	No	AutoAugment, Horizontal Flip	98.33%	83.13%	Moderate	Improved with Dropout, but val accuracy lower than ResNet
3	ResNet18 + CutMix augmentation	Yes	CutMix + AutoAugment	90–95%	80.72% Validation Result : fluctuating	Slight	Performance affected by small dataset and noisy mixed labels
4	MobileNet V2 + Fine-tuned Classifier	Yes	No	99.58%	89.16%	Slight but controlled	Validation Loss: Stabilized around 0.33 in later epochs

ResNet18 model with frozen layers and a fine-tuned classifier gave the best result, even though it only used simple preprocessing like resizing and normalization. This result was surprising because models with more advanced techniques (like CutMix) are expected to do better. However, since the dataset was small, **using strong augmentations may have added too much noise and made it harder for the model to learn the important details**. ResNet18 worked well because it used pre-trained features from ImageNet, which are already very good at capturing patterns. **Overall, this shows that sometimes, using a simple, pre-trained model is better than using complex methods—especially when working with limited data.**

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.**

   A long-tail distribution means that in a dataset, a few classes have a lot of data (these are the "head"), while most other classes have only a little (these are the "tail"). In our case with the MVTec AD dataset, there are many "Good" images but only a few for each type of defect. This imbalance makes it harder for the model to learn how to detect the rare defect types.

   (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**.

   A paper from 2021 called "Distribution-Balanced Loss for Long-Tailed Recognition" by Wu et al. offers a solution. They created a special loss function that helps the model pay more attention to the underrepresented (rare) classes without overfitting. Applied to our MVTec AD case, this method could help the model learn better defect representations, improving detection on rare defect types like 'broken_large' or 'contamination', even with fewer examples compared to the 'Good' class.

   Paper Link :
   https://openaccess.thecvf.com/content/CVPR2021/html/Wu_Adversarial_Robustness_Under_Long-Tailed_Distribution_CVPR_2021_paper.html

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.)

   When the training set contains only 'good' images, we can treat anomaly detection as an **unsupervised or one-class classification problem.** The main idea is to train a model to learn the normal patterns, so that anything that looks different (a defect) during testing can be flagged as an **anomaly**. Common strategies include using autoencoders or feature extractors (like pretrained CNNs) to learn good image representations. During testing, images with high reconstruction error or feature deviation are considered defective. Techniques like patch-level comparison, embedding distance, or score thresholding are often used to detect subtle and rare defects.

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.

If we want to train a model like YOLO-World to find defects in images, we need to **draw boxes** around the defect areas in each picture and **label** what kind of defect it is — for example, "broken_small". These boxes help the model learn where to look and what to identify.

For a model like SAM (Segment Anything Model), we go a step further. Instead of just drawing boxes, we need to **mark the exact shape** of each defect — kind of like coloring inside the lines. This helps the model learn precisely which pixels belong to the defect. These "masks" are saved as special images where the defect area is clearly marked.

(ii) (10 points) Why are these models suitable for fine-tuning for our custom dataset?

Models like YOLO-World and SAM are already trained on tons of images from many different situations. This means they're really good at understanding shapes, objects, and patterns. With transfer learning, they can quickly adapt to detect defects in our dataset, even when we have limited training data. Their strong generalization, speed, and precision make them ideal for improving anomaly detection beyond simple classification methods.