National Tsing Hua University

11320IEEM 513600 Deep Learning and Industrial Applications

Homework 3

Name: 董少霖 Student ID: 113034567

Due on 2025/04/10.
Note: DO NOT exceed 3 pages.

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.

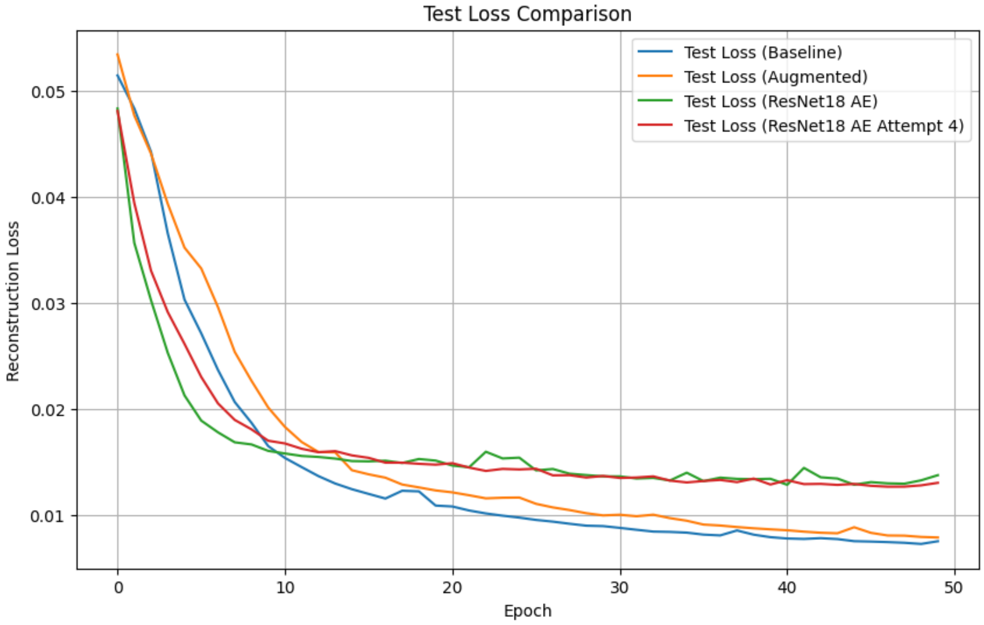
Ans:

- Number of defect classes: 7(An additional **combined** category appears in the test set, which includes multiple defect types in a single image, but is not considered an independent defect class.)
- Types of defect classes: bent_wire, cable_swap, cut_inner_insulation, cut_outer_insulation, missing_cable, missing_wire, poke_insulation(Not include combined)
- Number of images used in your dataset: 374(train: 224, test: 150)(Not include ground_truth data)
- Distribution of training and test data:
  - **Training data**: Contains only normal image (good class)
  - **Testing data**: Contains both normal images and the seven types of defect images (around 10–30 images per class)
- Image dimensions: All images are 1024 x 1024 pixels in size, RGB color images (3 channels)

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.)

Ans:

	Epochs	Learning Rate	Train Loss	Test Loss
1	50	0.001	0.0079	0.0075
2	50	0.001	0.0093	0.0079
3	50	0.001	0.0093	0.0138
4	50	0.0005	0.009	0.013



(i) We trained a basic convolutional **autoencoder** on resized train/good images(Use all 224 images to train). Reconstruction loss on test images was used as the **anomaly score.**

(ii) We applied mild **data augmentation** (horizontal flip and slight rotation) during training to enhance generalization. This improved robustness without overly distorting image structure.

(iii) We used a **pre-trained ResNet18** as the encoder and added a custom decoder. The model benefited from rich semantic features, leading to the best anomaly detection performance.

(iv) We reused the ResNet-based autoencoder and fine-tuned it using a higher learning rate (5e-4) to improve convergence. This resulted in slightly faster training and comparable performance.

The attempt 1 achieved the lowest test loss (0.0075), while the pre-trained ResNet18 (attempt 3) had the highest test loss (0.0138). This may be due to the mismatch between high-level semantic features extracted by ResNet and the pixel-level reconstruction required in autoencoders.

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.'

   Ans:

   A long-tail distribution refers to an imbalanced data pattern where a few classes have a large number of samples (head classes), while many others have very few samples (tail classes).

   In image datasets, this means most images belong to a dominant class (e.g., "Good"), while defect classes are rare and unevenly represented. This makes it challenging for models to learn features for minority classes, leading to poor generalization on rare defects.

   (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.

   Ans:

   One representative work is "**Balanced Meta-Softmax for Long-Tailed Visual Recognition**[1]" (CVPR 2021). The authors propose a class-balanced loss function that adjusts the softmax logits based on class frequency during training, reducing the dominance of head classes. In our case, this method could be adapted to fine-tune a classification head after feature extraction (e.g., using ResNet18).

   By adjusting the decision boundaries toward rare defect classes, it would help balance detection performance between "Good" and defect categories.

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.)

   Ans:

   Since the MVTec AD training set contains only 'good' images, a common strategy is to train an **unsupervised anomaly detection model**, such as an **autoencoder**.

[1] Ren, J., Yu, C., Ma, X., Zhao, H., & Yi, S. (2020). Balanced meta-softmax for long-tailed visual recognition. *Advances in neural information processing systems*, *33*, 4175-4186

The model learns to reconstruct normal patterns, and high reconstruction errors on test images indicate anomalies. Alternatively, using **pre-trained networks** (e.g., ResNet18) to extract features, followed by techniques like **k-NN**, **PCA**, or **one-class SVM**, allows detecting outliers without requiring defect samples. These methods rely on the assumption that anomalous patterns will deviate significantly from the learned normal feature distribution. (I didn’t use k-NN, PCA and one-class SVM.)

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.

Ans:

To fine-tune **object detection** models like YOLO-World, the dataset should include **bounding boxes** with corresponding **class labels** for each defect. This allows the model to localize and classify defects within images. For **segmentation** models like SAM (Segment Anything Model), the dataset must contain **pixel-wise masks** that highlight the exact shape and area of each defect region. These masks are typically stored in formats like .png or COCO-style annotations.

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

Ans:

YOLO-World and SAM are pre-trained on large, diverse datasets and excel at **generalization**. They can be fine-tuned efficiently on small custom datasets and adapted to domain-specific defects.

In our anomaly detection scenario, they enable **precise localization** and **visual interpretability**, helping identify not just whether an image is defective, but **where** and **how** the defect occurs — which is valuable for real-world inspection systems.