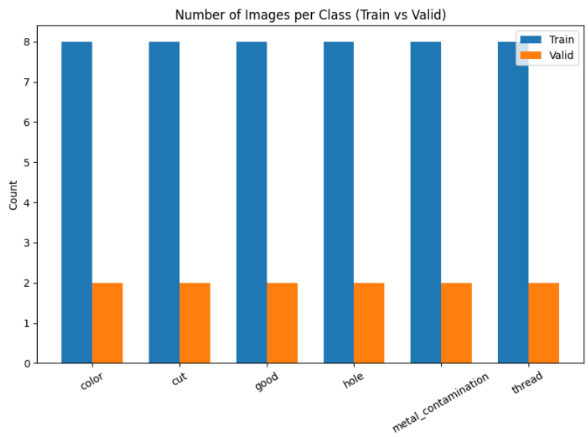
Deep Learning HW3

113034507 龔良軒

Q1. I chose the type “carpet” as my target for the experiment. There are some details about this dataset:

- Number of defect classes: 5 defect classes and 1 good condition.
- Types of defect classes: color, cut, hole, metal_contamination, thread.
- Number of images used in your dataset: In the baseline experiment, I used 10 images for each type of defect. (48 images for training and 12 images for testing)
- Distribution of training and test data: Both followed the same distribution as below:



- Image dimensions: Each image was resized from (1024, 1024) into (32,32)

Q2. I did a baseline experiment and other 4 experiments based on “carpet” dataset. All the experiments are implemented with “early stop”. The result and other details are as below:

experiment	model	Epoch	Batch Size	Optimizer (learning rate)	Training Accuracy	Training Loss	Validation Accuracy	Validation Loss
Baseline	resnet18	50	32	Adam(0.001)	100.00%	0.22	41.67%	2.54
Exp. 1	resnet18	50	32	Adam(0.001)	97.70%	0.29	72.73%	1.23
Exp. 2	resnet34	50	32	Adam(0.001)	100.00%	0.21	90.91%	0.95
Exp. 3	resnet50	50	32	Adam(0.001)	97.70%	0.19	77.27%	1.01
Exp. 4	resnet34	50	32	Adam(0.01)	97.70%	0.04	77.27%	1.72

*** The row in orange background is baseline experiment**

*** The row in blue background is the experiment with the best performance**

The following is the description for each experiment:

- Experiment1. Increase # of data:
    - # of train set changes from 48 to 87
    - # of test set changes from 12 to 22.
- Experiment2. Follow Exp. 1 and use resnet34 as a substitution of resnet18.
- Experiment3. Follow Exp. 1 and use resnet50 as a substitution of resnet18
- Experiment4. Follow Exp. 2 and set initial learning rate = 0.01

There are two key factors contributing to the strong performance in Experiment 2:

1. Using ResNet34 instead of ResNet18: ResNet34 is a deeper network, which allows it to capture more detailed features from the images. This can lead to better representation and classification of subtle defects.
2. Increasing the training data: More training data improves the model's ability to generalize and reduces the risk of overfitting. This makes the model more stable and effective during evaluation.

Q3. (i). A long-tail distribution refers to a statistical pattern where a few categories have many samples, while most categories have relatively few. In the context of image classification or anomaly detection, this means that certain classes ("good" class) have significantly more training data compared to the defective classes.

Q3. (ii).

***Reference Paper:** An oversampling method for wafer map defect pattern classification considering small and imbalanced data (2021)

***Summary:** The paper proposes an improved oversampling method tailored to wafer map defect pattern classification, especially for small and highly imbalanced datasets (like our case), which are common in semiconductor manufacturing. The core of the paper is the **LRO method**, which aims to improve classification accuracy by generating synthetic samples more effectively than traditional oversampling techniques like SMOTE.

***LRO framework:**

- **Clustering minority class samples** using DBSCAN.
- **Learning intra-class feature distributions** using an autoencoder.
- **Sampling new data** based on label-wise reconstruction loss.

- **Balancing the dataset** by generating more samples for rare patterns.

Their method can be used to improve the model in our case because the dataset in our case is also **small and imbalanced.** We can take this method to oversample the data of the minority of categories. Furthermore, the task mentioned in this paper is also **Defect Classification** which is the same as our case.

Ref: Kim, E. S., Choi, S. H., Lee, D. H., Kim, K. J., Bae, Y. M., & Oh, Y. C. (2021). An oversampling method for wafer map defect pattern classification considering small and imbalanced data. Computers & Industrial Engineering, 162, 107767.

Q4. There are a few strategies for developing an anomaly detection model while facing an imbalance problem:

- Implement Oversampling to images of minority classes
- Apply data augmentation techniques to simulate defective images, which will help the model generalize better to unseen anomalies during training.
- using a one-class SVM trained only on the normal images. It will create a boundary around the normal data and classify anything outside this boundary as an anomaly. Then, we can focus on "anomalies" to classify their defect type.

Q5. (i). The requirements of data for both models are listed below:

- Data prepared for object detection: The dataset should include images along with **bounding box annotations** around the defects. These annotations specify the coordinates of the rectangular regions where defects appear and should be labeled with the appropriate defect type.
- Data prepared for segmentation: The dataset must contain images paired with **pixel-level masks**, where each pixel in the image is labeled according to the defect category or background. This allows the model to learn the exact shape and location of the defects at a finer level than bounding boxes.

Q5. (ii). Models like YOLO-World and SAM are trained on tons of diverse data, so they're really flexible when it comes to picking up new tasks. With transfer learning, we can fine-tune them using just a small amount of our own data, and they'll still do a great job at spotting or segmenting defects. Since they're good at generalizing and even support open-vocabulary or zero-shot learning, they're especially handy for catching defects.