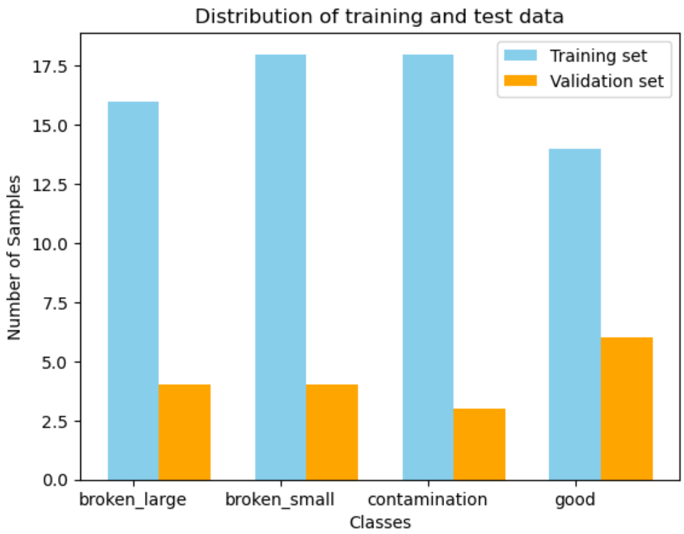
1. • **Number of defect classes:** 3 (exclude normal).

   • **Types of defect classes:** broken large, broken small and contamination.

   • **Number of images used in your dataset:** 83 (20 broken large, 22 broken small, 21 contamination and 20 normal).

   • **Distribution of training and test data:**



   • **Image dimensions:** *all_data.shape* = (83, 900, 900, 3) -> height = 900, width = 900, 3 channels.

2. **Original:**

   input size = 32x32,

   batch size = 128, epochs = 50

   learning rate = 1e-3

   optimizer: Adam optimization

   learning rate scheduler: Cosine Annealing

   pre-trained model = ResNet18

   Method 1:

   train less epochs to avoid overfitting. (epoch = 20)

   Method 2:

   change the input size to 224x224.

   Method 3-1:

   change the pre-trained model to ResNet50.

   Method 3-2:

   change the pre-trained model to Vision Transformer.

Method 4:

change the pre-trained model to Vision Transformer and also change the batch size from 128 to 36.

(**Note:** the parameters that are not mentioned in 'Method 1~4' remain the same as those of 'Original'.)

Conclusion: Changing the input size to 224x224 improves the performance significantly, and performance of transformer-based pre-trained model seems to be better than that of CNN-based (ResNets) pre-trained model.

	Training accuracy	Validation accuracy	Test accuracy
Original	93.94 %	58.82 %	23.53 %
Method 1	69.70 %	58.82 %	47.06 %
Method 2	87.88 %	76.47 %	76.47 %
Method 3-1	95.45 %	76.47 %	76.47 %
Method 3-2	96.97 %	88.24 %	88.24 %
Method 4	90.91 %	**94.12 %**	**88.24 %**

3. (1) In statistics and business, a long tail of some distributions of numbers is the portion of the distribution having many occurrences far from the "head" or central part of the distribution. (***Wikipedia***)

(2) **UnderXGBoost** is an ensemble learning algorithm that integrates the strengths of bagging and XGBoost. It uses an under-sampling technique to sample sub-dataset (size should equal to that of the minority class) from the majority class. In this case, the size of each defect dataset is approximately 20 and that of good dataset is 229, so we randomly choose about 20 samples from good dataset with replacement. Next, use the subset to train a classifier (may be a CNN model). Repeat sampling and training several times we will get many classifiers. Finally, the prediction is made from a majority voting mechanism among all trained classifiers.

(https://doi.org/10.1016/j.compbiomed.2024.108244)

4. There are some techniques such as *representation-based* which extract feature vectors from images and project them into a high-dimensional feature space (e.g. **PatchCore**), *data augmentation-based* by generating synthetic anomalies (e.g. **CutPaste**), and *reconstruction-based* (e.g. **Autoencoder**). Reconstruction-based methods are on the basis that if the model is trained only with normal data, when it encounters abnormal data at testing stage it may not be able to reconstruct those images accurately. Recently, Deng and Li proposed a framework called **reverse distillation (RD)** composed of a pre-trained teacher encoder, a one-class bottleneck embedding module and a student decoder. RD has been proven outperformed other methods mentioned above in anomaly detection.

5. (1) For object detection, we should prepare the dataset that contains position and labels (which category it belongs to) of objects on the image. For segmentation, we should prepare the dataset that contains labels (which category it belongs to) of each pixel on the image, such as the mask in MVTec AD Dataset.
(2) Since models designed for object detection or segmentation tasks are capable of identifying objects or locating regions within images, which is helpful to identify the position of defects for anomaly detection. Moreover, those models are open-source and have been pre-trained on large datasets, and thus have great performance and generalization ability (by fine-tuning on datasets of downstream task).