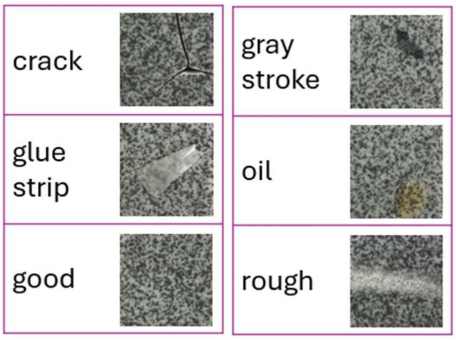
NTHU 11220IEEM 513600 **Homework #3** YuWei **112003852**

Q1. (10 points) - Dataset: Tile

- Number of classes: Total 6 classes: 5 defects + 1 good
- Types of defect classes: crack, glue_strip, gray_stroke, oil and rough.
- Number of images used in your dataset: 60 images.
- Distribution of training and test data.:

# of image	Training	Test	Sub-total
Crack	8	2	**10**
glue_strip	8	2	**10**
good	8	2	**10**
gray_stroke	8	2	**10**
oil	8	2	**10**
rough	8	2	**10**
Sub-total	**48**	**12**	**60**

image examples:



- Image dimensions.: 840 x 840 pixels

Q2. (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.)

Trial	TorchVision model	optimizer	epoch	learning rate	Batch	test accuracy%
0	Resnet18	Adam	50	1E-03	32	16.67%
1	Resnet18	Adam	100	1E-05	32	41.67%
2	Resnet18	SGD, momentum=0.9	50	1E-03	16	58.33%
3	Resnet50	SGD, momentum=0.99	100	5E-04	8	75.00%
4	Densenet121	SGD, momentum=0.99	100	5E-04	8	91.67%

Trial1 Strategy: Concentrated on rectifying underfitting by modifying epochs and learning rates for optimal convergence.

Trial2 Adjustments: Adopted momentum-based SGD and a smaller batch size, introducing variability to overcome potential optimization pitfalls.

Trial3 Refinements: Transitioned to the deeper Resnet50, capitalizing on its ability to learn nuanced features, with fine-tuned momentum and learning rate for balanced training progression.

Trial4 Optimization: Utilized the Densenet121 with pretraining, exploiting its dense feature integration to substantially improve anomaly detection capabilities.

Q3. (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.'

'Long-tail distribution': The imbalance such that the majority classes ('Good' in this case) can dominate the learning process, causing the model to perform poorly on the minority classes ('Defects'), which are underrepresented.

(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.

Theodoropoulos, S., et al. (2024). On-the-fly image-level oversampling for imbalanced datasets of manufacturing defects. Machine Learning. https://doi.org/10.1007/s10994-023-06498-4

This paper introduces a novel, on-the-fly image-level oversampling technique to address class imbalance in datasets of manufacturing defects, significantly enhancing the performance of CNNs in defect detection.

Suggested methodology:

- Pre-train a CNN on the imbalanced dataset to learn initial class boundaries.
- Assess prediction confidence to identify images close to the classification boundary.
- Use low-confidence images as seeds for generating new, similar images.
- Targeted oversampling focuses on hard-to-classify regions for image generation.
- Incorporate generated images into training on-the-fly for dynamic learning.
- Fine-tune the CNN with this augmented, balanced dataset for improved minority class recall.

Q4. (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.)

In cases like MVTec AD datasets with predominantly 'good' samples, the strategic application of one-class classification becomes crucial. Specifically, employing Support Vector Machines (SVM), a powerful method for anomaly detection, alongside techniques like autoencoders, can be particularly effective. By training these models solely on 'good' samples, they learn the normal distribution of features, enabling them to detect anomalies through significant deviations in reconstruction error. Enhancing the model's robustness and adaptability involves dataset augmentation strategies, such as rotation, scaling, and adding noise, which broaden the model's understanding of 'normal' conditions. Couple examples of SVM application include facial recognition, for identifying individuals; and spam detection, to filter unwanted emails. This enriched approach facilitates the identification of outliers as anomalies.

Q5. 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.

a) For object detection, compile images with precise bounding boxes around objects, tagged with their class labels, to teach spatial recognition.

b) For segmentation, particularly semantic, create a dataset of images with pixel-level annotations, using masks to define objects' shapes and class-specific colors.

This granularity aids models in accurately identifying and delineating object boundaries for precise analysis.

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

Leveraging pre-trained models like YOLO (*You Only Look Once*) and SAM (*Segment Anything Model*) maximizes the benefits of their extensive initial training on a wide array of datasets. These models excel in adapting to unique features of custom datasets, thereby enhancing task-specific precision and efficiency.

YOLO: Unmatched in <u>speed</u>, it's the go-to for real-time detection, seamlessly processing video streams or live data to deliver instant results, crucial for security and surveillance applications.

SAM: Its <u>accuracy in segmentation</u> is unparalleled, offering pixel-perfect outlines that enable detailed analysis and anomaly detection in images, vital for medical imaging and quality control.

Utilizing these models streamlines development and conserves resources, providing a robust foundation for developing advanced, customized solutions.