

一、

數據集名稱: metal_nut

- Number of defect classes: 四種
- Types of defects classes: bent, color, scratch, flip
- Number of images used in your dataset: 220
- Distribution of training and test data:
 - Training: 220 good images. Test: 115 (good: 22, bent: 25, color: 22, scratch: 23, flip: 23)
- Image dimensions: 700 × 700 pixels

二、參數固定使用：(batch_size: 8, LR: 1e-2, epochs: 30)

做了四種更動並用準確率%當比較標準：為了提升模型效能，我嘗試了四種不同策略首先，對 ResNet18 增加 MLP 頭（兩層全連接層）後，準確率為 30%。第二次改用輕量化架構 MobileNetV2，表現提升至 50%。第三次採用 EfficientNet-B0，得益於其深度與寬度比例優化，也達到 50%。最後，我保留 ResNet18 架構，對少數類別進行更強的資料增強（如旋轉、顏色抖動、透視變形），模型準確率上升至 60%。相較於單純更換模型架構，**針對性資料增強**對提升少數類別的辨識能力效果最顯著。這個方法透過針對少數類別進行旋轉、裁剪、亮度改變、顏色擾動等操作，可以模擬真實世界中不同條件下的缺陷樣貌，讓模型看到更多「變化形式」，進而更穩健地學會缺陷特徵。而且資料增強僅針對缺陷類別執行，並避免鏡像或水平翻轉等會改變缺陷位置語意的操作，因此保留了原始的關鍵特徵。因此，在資料量有限且類別不均的情況下，透過精心設計的資料增強，能大幅改善模型泛化能力與辨識效果，這也解釋了為何它是本實驗中最有效的改進策略。

三、

1. 在資料分布中，長尾分布（long-tail distribution）是指部分類別擁有大量樣本，而其他多數類別樣本數量稀少。這種分布容易出現在現實世界的資料集中。在 MVTEC AD 中，「good」類別擁有大量圖像，而每種缺陷類別僅有少量圖像，形成嚴重的資料不平衡，導致模型傾向於學習 good 類別而無法學習到其他類別。

2. Long-Tailed Recognition via Weight Balancing

https://openaccess.thecvf.com/content/CVPR2022/html/Alshammari_Long-Tailed_Recognition_via_Weight_Balancing_CVPR_2022_paper.html

我找到了一篇很有參考價值的 paper，CVPR 2022 發表了一篇名為《Long-Tailed Recognition via Weight Balancing》的 paper 在實際應用中，資料分布常常不是平均的，反而會呈現所謂的「長尾分布」：少數類別（擁有大量樣本，而多數類別則缺乏資料。在我們這次使用的 MVTEC AD 資料集中就很明顯，像是 metal_nut 的訓練集中只有 good 圖片。這樣的情況會讓模型在訓練時偏向於 good 類別，進而忽略掉少數但重要的異常類別，造成實際檢測時誤判率偏高。

在這篇 paper 中他們針對分類模型在長尾資料下的偏差問題，提出了「權重平衡（Weight Balancing）」的方法。他們發現，在長尾情況下，模型學到的分類器會傾向給主流類別較大的權重範數（magnitude），讓模型更容易判斷那些常見類別，反而忽略少數類別。為了改善這個現象，他們設計了一些技巧，例如使用 L2 正則化、限制分類器權重大小（MaxNorm），甚至對尾部類別的權重新初始化等，來調整學習過程中的重心。

如果應用到我們的異常檢測任務中，像是在 fine-tune PatchCore 或 YOLO 的階段，就可以利用這種權重調整方式，讓模型不要太偏向 good 類別，而能更敏銳地察覺各種少見但重要的缺陷。這樣的做法不需要太複雜的修改，卻能實際改善模型對不平衡資料的辨識效果。

四、

根據我在網路上找到的文獻資料、面對這種情況大概有幾種方法可以處理

1. 自編碼器（Autoencoder, AE）：

用 good 圖訓練重建模型 → 異常圖的重建會失敗 → 利用誤差判斷異常

2. GAN 或 VAE-based 模型（e.g., AnoGAN, f-AnoGAN, CutPaste）：

生成模型學習正常分佈，異常會與生成圖差異大

3. One-class 分類模型（如 One-Class SVM, Deep SVDD）

4. 特徵比較方法（如 PatchCore, PaDiM）：

利用預訓練模型提取 good 圖特徵，測試時比較與正常特徵的距離

5. 資料增強模擬缺陷（如 CutPaste）：

只用 good 圖人工製造「假異常」，轉為二元分類問題

MVTec AD 的訓練資料主要是正常圖像，幾乎沒有缺陷樣本，這種情況下適合採用無監督或自監督的異常檢測策略。例如 Autoencoder 或 VAE 可以訓練模型去重建正常圖像，異常圖像則會出現較大的重建誤差，也能使用像 CutPaste 這類方法透過資料增強產生假異常，讓模型學會區分異常特徵。不過，在這些方法中，我認為目前最有效的策略是使用 PatchCore、PaDiM 或 FastFlow 等特徵比對式方法，因為這類方法利用預訓練模型擷取高階語意特徵，再與正常樣本建立的特徵記憶庫比對，可以更精準地捕捉細微但關鍵的異常變化，並在多數 MVTec 產品類型上都有很高的偵測準確率。

五、

1.

若要使用物件偵測模型（如 YOLO-World）進行訓練，需準備包含缺陷位置的 bounding box 標註，每張圖需標示出 defect 的位置與類別，例如使用 COCO 格式的 .json 標註檔。若使用影像分割模型（如 SAM），則需要像素級的 mask 標註，通常為每張原圖對應一張二值 mask，標示出缺陷區域的位置，格式可為 PNG 或 numpy array。這些資料都可以由 MVTec 提供的 ground_truth 裡進行轉換與整理。

2.

YOLO-World 和 SAM 分別在物件偵測與影像分割領域表現突出，並且都已在大規模資料集上預訓練過，因此具備良好的泛化能力。這對 MVTec AD 這種缺陷樣本稀少的資料集來說特別有幫助，因為我們不需要從頭訓練模型，只要進行少量微調（fine-tuning）就能快速上手。

以我在本次作業中使用的 metal_nut 為例，常見缺陷像是 bent 和 scratch 都具有明顯的空間特徵。像是 YOLO-World 可以學習在圖像中快速框出 bent 缺陷的位置，適合用在即時檢測場景。而對於像 scratch 這種線狀、分佈不規則的缺陷，SAM 就很適合用來分割出精確的區域範圍，甚至可以生成像素級的 mask，供後續分析或可視化使用。此外，SAM 還具備互動式標註功能，可以幫助標註者快速建立初始 segmentation mask，降低人工成本。這些優勢都讓它們非常適合用來解決 MVTec 類型的缺陷檢測任務。

