Weld Defect Detection

INFO-6147 – Deep Learning with PyTorch
Capstone Project Report
Abdallah Seyam
1340973

Abstract:
This project presents an image classification system for identifying different types of weld defects using convolutional neural networks (CNNs) implemented in PyTorch. The model is trained and validated on a YOLO-formatted dataset, achieving a final testing accuracy of 76.30%. The evaluation includes confusion matrices, per-class accuracy, and a classification report, offering insights into the model's strengths and its limitations in handling minority classes.

Introduction:
Detecting weld defects is critical in industrial inspection. Manual inspection is time consuming and prone to human errors. Deep learning models, particularly CNNs, can automate this process. This project applies CNN based image classification to detect weld defects from images using a real-world dataset formatted in YOLO style annotations.

Dataset:

- Source: weld defect detection dataset
- Classes: 5 weld defect types
    - 0 = adjacent defects
    - 1 = integrity defects
    - 2 = geometry defects
    - 3 = post processing defects
    - 4 = nonfulfillment defects
- Preprocessing: Resized to 224x224, normalized
- Splitting: 70% training, 15% validation, and 15% test using stratified sampling

Preprocessing & Augmentation:

- Transformation:

    Training: resize, RandomHorizontalFlip, ColorJitter, RandomRotation, ToTensor

    Validation/test: resize, ToTensor

- Label parsing: First entry in YOLO .txt file used as class ID

Model Architecture:

- Backbone: torchvision. models.resnet18 with modified fc layer to output 5 classes
- Loss: CrossEntropyLoss
- Optimizer: Adam with LR = 1e-3

Training Results:

- Epochs: 25
- Final Training Accuracy: 75.43%
- best Validation Accuracy: 69.36%

Evaluation Metrics:

- Confusion Matrix
- Accuracy per class:
    - Class 0: 24.24%
    - Class 1: 87.23%
    - Class 2: 83.12%
    - Class 3: 16.67%
    - Class 4: 40.00%

Visualizations:

- Accuracy over epochs
- Confusion matrix (sklearn)
- Bar plot of per-class accuracy

Conclusion:
The final testing accuracy reached 76.30%, indicating good generalization. However, performance on minority classes was weak, suggesting the need for techniques such as oversampling or class-weighted loss. Future improvements may include using transfer learning (e.g., pretrained ResNet) or ensemble models.

References:

- PyTorch documentation: https://pytorch.org
- YOLO label format guide: https://github.com/ultralytics/yolov5/wiki/Train-Custom-Data
- INFO-6147 course materials