Weld Defect Detection

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

Abstract:

This project develops an image classification system to identify different types of weld defects using convolutional neural networks (CNNs) in PyTorch. The system is trained and validated on a YOLO annotated dataset, achieving up to 98% training accuracy and ~77% validation accuracy. Evaluation includes confusion matrix and per-class accuracy metrics.

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 = non fulfillment defects
- Preprocessing: Resized to 224x224, normalized
- Splitting: 80% training, 20% validation using stratified sampling

Preprocessing & Augmentation:

- Transformation: transforms.Resize, transforms.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: 98.15%
- Final Validation Accuracy: 77.49%

Evaluation Metrics:

- Confusion Matrix
- Accuracy per class:
  - Class 0: 56.82%
  - Class 1: 90.48%
  - Class 2: 82.52%
  - Class 3: 12.50%
  - Class 4: 7.69%

Visualizations:

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

Conclusion:
The trained model shows high training accuracy and decent generalization. Improving per class performance on underrepresented defect types is a key area for future work.

References:

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