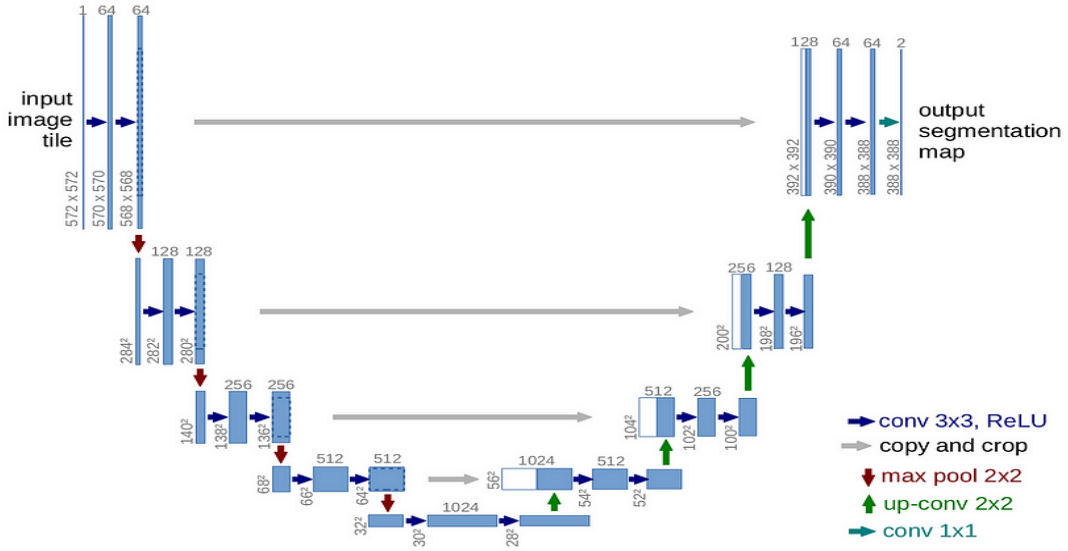
Weld Defect Detection Using U-Net

Project Overview

The goal of this project is to develop an automated system capable of detecting and segmenting weld defects in industrial settings using U-Net architecture. This application will benefit industries that rely on welding for manufacturing, such as automotive, aerospace, and construction, by reducing the need for manual inspection, improving accuracy, and increasing efficiency in quality control processes. We developed a U-Net Model specially structured for defect detection task with grayscale X-ray images and fixed input patch sizes by training it on the images obtained from the GDXRAY dataset, which includes X ray images of welds procured from an industrial setting. We were able to obtain binary masks for the same images which highlighted the defective regions in the weld as the output. We have attempted to develop the U-Net model mostly from scratch using certain online references wherever needed. We improvised on the dataset by designing a patch-based training pipeline with the Albumentations library. We used Dice + BCE loss for better class imbalance handling. Overall, we were able to achieve decent results, with the U-Net model being able to detect most weld defects with some false positives.

Approach

Algorithm:
We used the U-Net model architecture for this project; this is a special type of Convolutional Neural Network (CNN) which specializes in image segmentation. The U-Net was first designed for biomedical applications, but its applications have become widespread over time, even being used in self-driving cars, industrial inspection and the automation industry. Its name comes from its U-shaped architecture that includes a contracting path (encoder) and an expanding path (decoder), allowing it to perform precise, pixel-wise segmentation even with limited training data. It consists of a contracting path known as the encoder and an expanding path called the decoder. The encoder in U-Net compresses the input image through convolution and pooling layers to capture high-level features, while the decoder progressively up-samples these features to reconstruct the segmentation map. To preserve spatial details lost during down-sampling, skip connections directly link encoder layers to their corresponding decoder layers, ensuring accurate localization of features. Below is a representation of a conventional U-Net architecture.



Independent implementation and online resource usage:

This project involved end-to-end implementation of data handling, model training, and inference for weld defect defection. We worked with the GDXray dataset, which had a limited number of images and annotated masks, posing a challenge for training a deep learning model. To address this, we developed a custom data pipeline that extracted fixed-size patches (320×480) from the X-ray images and applied augmentations such as horizontal/vertical flips and rotations using the Albumentations library. This helped expand the dataset and expose the model to more variations of defect shapes and locations.

The core U-Net architecture, training loop, custom Dice loss integration, and validation strategy were implemented independently in PyTorch. For inference, we also created a CLI-based script that loads images and displays predicted masks. However, we referred to this GitHub repository from Burak Kılıç for architectural inspiration and modified the standard U-Net to include BatchNorm2d layers for better training convergence and performance. These changes were integrated and tested independently within our own training pipeline.

Experimental Protocol

System Configuration:

- Python: 3.8+
- Installed libraries:
    - torch==1.13+cu117
    - albumentations==1.4.18
    - opencv-python, matplotlib, Pillow
    - All dependencies were managed via pip in a virtual environment.

Training Setup:

- Trained on a laptop with NVIDIA RTX 4050 GPU.
- Frameworks: PyTorch, Albumentations, OpenCV.
- 30 epochs, batch size 8.
- Combined BCE + Dice loss used.

Dataset:

- Used the GDXray weld X-ray dataset with limited annotated masks.
- Cropped full images into overlapping patches of 320×480 (step size 160).

Augmentation & Preprocessing:

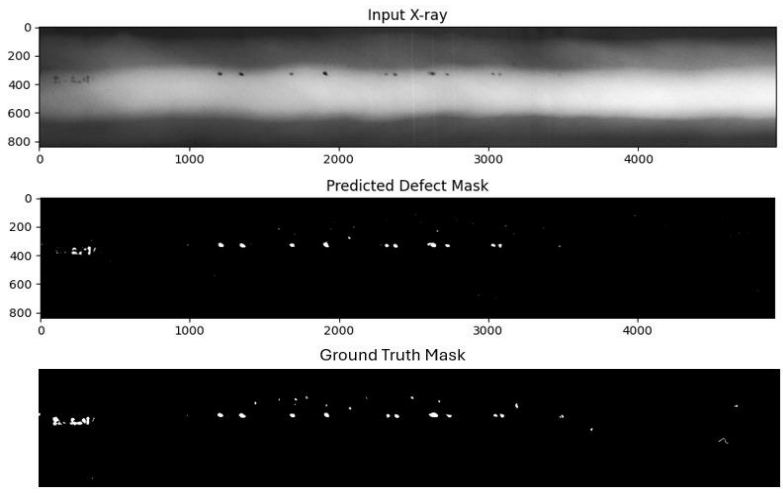
- Applied Albumentations: horizontal/vertical flip, ±15° rotation.
- Masks binarized; images normalized and converted to tensors.
- Data was split into 80% training, 20% validation using augmented patches.

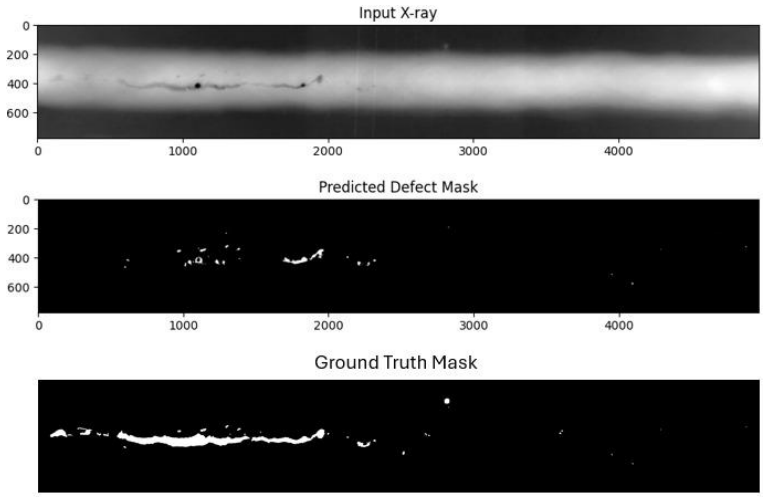
Evaluation:

- Metrics: Accuracy and Dice score on validation set.
- Visual results inspected for mask quality.

Results

After training our U-Net model for 30 epochs on the augmented GDXray weld defect dataset, we saw clear and steady improvement in its ability to detect defects. Starting with a Dice score of 0.3458, the model improved significantly over time and reached a final Dice score of 0.6907, along with an impressive 99.24% accuracy. These results suggest that the model effectively learned how to identify and segment weld defects, even when they were subtle or irregular in shape. Visual inspections of the outputs showed good alignment with the ground truth masks, particularly for medium sized defects. Some errors remain mainly in noisy regions but overall, the model produced consistent and encouraging results. This outcome supports the effectiveness of our data augmentation strategy and patch wise training pipeline.





Analysis

The results show that our model is quite capable of identifying weld defects from X-ray images, with a final Dice score of 0.6907. That said, there's still room to improve. While the model performed well overall, we noticed it sometimes struggled in areas where the image had a lot of background noise or low contrast. These are tricky situations even for more complex models. On the positive side, using augmentation really helped the model generalize better, especially since we had a limited dataset to start with. In the future, we could explore adding more advanced components like attention layers or post-processing techniques to reduce false positives and sharpen the masks. Overall, the project gave us useful insights into both the strengths and the limitations of using U-Net for industrial defect detection.

One key strength of the approach lies in its simplicity—using patch-wise segmentation allowed the model to focus on localized features and perform well even on small or subtle defects. The training process was efficient, running smoothly on a personal laptop equipped with an NVIDIA RTX 4050 GPU.

However, some limitations were also observed. Due to the patch-wise nature of the data, the model occasionally failed to capture the broader context of the weld, which is important for interpreting more complex or elongated defects. Additionally, occasional false positives were seen in visually noisy areas, possibly due to artifacts in the X-ray scans or the limited variation in the training dataset.

Despite these challenges, the project successfully demonstrates that a relatively lightweight and accessible deep learning pipeline can be used for industrial defect detection. With more diverse training data and some refinement (such as post-processing or full-image inference), this system could be further improved and adapted for deployment in real-world quality control applications.

Discussion and Lessons Learned

Working on this project gave us hands-on experience with real-world challenges in deep learning, particularly in the context of weld defect detection. One of the key takeaways was understanding how limited data can significantly affect model performance. Since the GDXray dataset had only a few annotated masks, we relied heavily on data augmentation using Albumentations to create a more diverse training set. This helped the model generalize better and improved prediction accuracy.

We also realized how critical patch size and proper loss functions are. Early results were inconsistent until we standardized patch dimensions and combined BCE with Dice loss, which gave much better Dice scores. Managing file structure, automation through CLI tools, and maintaining clean, readable code also turned out to be more important than expected, especially for reproducibility and evaluation.

Areas to Improve:

- Introduce cross-validation for more reliable evaluation
- Try pre-trained backbones for better performance
- Explore architecture like Attention U-Net
- Use simple post-processing to clean predicted masks

Bibliography

- Ronneberger, O., Fischer, P., & Brox, T. (2015). U-Net: Convolutional Networks for Biomedical Image Segmentation. In Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015 (pp. 234–241). Springer.
- burakai. Weld-Defect-Detection-with-U-Net. GitHub repository.

- Domingoery, C. GDXray: The Database of X-ray Images for Nondestructive Testing. Pontificia Universidad Católica de Chile

- Ito-Aramendía, A. Decoding the U-Net: A Complete Guide. Medium.

- Buslaev, A., Iglovikov, V. I., Khvedchenya, E., Parinov, A., Druzhinin, M., & Kalinin, A. A. (2020). Albumentations: Fast and Flexible Image Augmentations. Information, 11(2), 125