Weld Defect Detection using a Small Dataset with U-Net

Burak Kılıç
İstanbul Technical University
Mechanical Engineering Department
Istanbul, Turkey

***Abstract*—In this paper, a modified U-Net model is used to detect defections on weld x-ray images.**

Keywords—welding, x-ray images, radiographic images, defect, flaw, imperfection, defect detection

I. Introduction

Welding processes should join materials flawlessly. However, weldments may contain flaws (defects, imperfections) for a variety of reasons. These flaws are visible as gaps, cracks, pores, and cavities. Lack of fusion, lack of penetration or excess penetration, porosity, inclusions, cracking, undercut, lamellar tearing, etc. Any of these flaws could result in unexpected failure below the design load or, in the case of cyclic loading, failure after fewer cycles than expected [1].

Because failure due to welding flaws usually results in brittle fracture, it is critical to control the entire welding process, including pre-welding, during-welding, and post-welding periods, to avoid failure due to a welding flaw. Pre-welding is the period during which the welder/manufacturer provides some quality documents demonstrating that the welder/welder/manufacturer is competent to execute the weldment and its mechanical properties. During-welding is mostly handled by a welder, but in robot welding it is sometimes handled by an operator. It is during this time that the majority of welding flaws occur. During the post-welding period, some tests are performed.

Post-welding tests are broadly classified as destructive testing (DT) and non-destructive testing (NDT). Destructive tests are accurate, but it is not acceptable to damage welded materials because they will lose their function. This method is applied to some specimens to gain an understanding of the mechanical properties of the weld, but in general, NDTs are performed on welded joints to demonstrate their flawlessness. Radiographic testing, ultrasonic testing, liquid penetrant testing, magnetic particle testing, eddy-current testing, and other NDT methods are common.

Radiographic testing is a popular NDT method for imaging weldments to detect and locate flaws. In the radiographic test method, weldment inhomogeneity causes variation in intensity in radiographic weld images. Lines visible in these radiographic images represent various flaws such as lack of fusion, lack of penetration, porosities, inclusions, cracks, undercuts, and so on. The thickness of the material in the welding process, weld type, weld position, and other factors influence radiographic testing methods such as exposure time, distance, and angle between the energy source and the material, all of which have an impact on image quality. Even though flaws on hot welds can be detected during the welding process, radiographic testing is most commonly used as a post-welding inspection method.

There are several approaches for inspecting radiographic images that use various image processing and computer vision methods such as fuzzy clustering methods, multiclass methods based on geometric texture features, neural networks, morphological watershed segmentation techniques, and so on. The implementation of flaw detection in radiographic weld images using a U-net-based approach is discussed in this paper [2,3].

II. Dataset

A. *Weld X-ray Datasets*

Weld defects can be classified in many ways. For example, in ISO 6520-1:2007 there are six main classes which are crack, cavity, solid inclusion, lack of fusion and penetration, imperfect shape and miscellaneous imperfections respectively. Furthermore, these classes have their own subclasses. On the other hand, these six main classes are sufficient for most cases. Sometimes it’s even possible to evaluate weld x-ray images under two classes as “good” or “bad” weld. Even it’s desired to make evaluation under two classes, it is still expected to be educated in radiographic testing for being capable to label weld x-ray data. In order to provide this proficiency it’s mostly expected to have some quality code certifications such as ISO, ASNT, AWS, etc., which takes time. Furthermore, in order to obtain accurate results when analysing radiographic images, the expert must draw on previous experience. However, results continue to be affected by the expert's mental state, with results for the same image varying from expert to expert [4].

On the other hand, labelling data for segmentation is inherently expensive. Taking these two cases into account, it becomes very difficult to create a proper dataset, and therefore it can be said that a proper public dataset for weld x-ray images does not exist. Best option is provided by Domingo Mary with the name of “GDXray+: The GRIMA X-ray database”[5]. It is normally for baggage x-ray images and weld images provided by Uwe Zscherpel at Institute for Materials Research and Testing (BAM), Berlin. In this paper, these weld images from GDXray dataset is used.

B. *GDXray+*

Weld subset of GDXray is consist of 67 unique weldment images. These 67 is stored in W0003 subfolder. In subfolder W0001, 10 of 67 are extracted as grayscale with a focus of weldment area. Furthermore, segmented versions of these 10 images are stored under W0002. In short, in this study there are only 10 grayscale source x-ray images from the GDXray dataset, and masked versions of them are used. List of the works uses GDXray dataset for weld defect detection are listed in Table 1.

TABLE I. STUEDIES USED GDXRAY DATASET

Paper Title	Authors
Automatic Detection of Welding Defects using Deep Neural Network [6]	Wenhui Hou et al.
Weld Defect Images Classification with VGG16-Based Neural Network [7]	Bin Liu et al.
Automatic detection of welding defects using the convolutional neural network [8]	Roman Sizyakin et al.
A Welding Defect Identification Approach in X-ray Images Based on Deep Convolutional Neural Networks [9]	Yuting Wang et al.
A New Image Recognition and Classification Method Combining Transfer Learning Algorithm and MobileNet Model for Welding Defects [10]	Haihong Pan et al.
Transfer Learning With CNN for Classification of Weld Defect [11]	Samuel Kumaresan et al.
An automatic welding defect location algorithm based on deep learning [12]	Lei Yang et al.
Recognition of weld defects from X-ray images based on improved convolutional neural network [13]	Ande Hu et al.

None of these works uses welds subset of GDXray dataset itself alone since it's not sufficient for CNN based works. They extended and created a bigger dataset with their own way. Some works are provided additional x-ray images from a domestic testing companies and some collected weld images from the internet and labelled with the help of experts. On the other hand, in these works, data augmentation is applied aggressively. Thus, related works became eligible for providing reliable computer vision tasks like classification, detection, and segmentation. Our motivation in this paper is to segmentation of weld defects only using of the subset of GDXray but no use of additional data since they're hard to achieve because of the reasons explained above.

III. METHOD

For the segmentation task U-Net architecture is used because of its advantages with the small datasets. U-net is originally developed for biomedical image segmentation which is similar to x-ray image segmentation. In the end if the segmentation would effectively the defects can be easily detected. It should not be confused with the object detection which is another challenging task in computer vision based on finding all the objects and their locations in an image.

A. Experimental Setup

Implementation are carried out both on Google Colab and a Windows 11 PC without graphics card but with an Intel i7 12700 CPU only. Python 3.10 environment created and PyTorch used as the main framework. NumPy and Pillow libraries is used for image reading, writing and manipulation operations. Data augmentations are carried with Albumentations library and detailed below.

B. Implementation of U-Net

Used architecture in this paper is the exact implementation of the original U-Net paper [14] with a few key differences. Basically, U-Net consists of encoding, pipeline, and decoding parts. In the beginning double convolution operation (convolution followed by convolution) is carried, and this followed by downsampling. Meanwhile, feature maps before downsampling applied are copied for the rest of the process. This double convolution followed by downsampling process is repeated four times and it's called encoding. After encoding, a double convolution carried without any additional operation, and this is called "pipeline". After pipeline, decoding part comes, and it is the same idea with the encoding but vice versa. It uses upsampling instead of downsampling to achieve dimensional consistency. At this point copied feature maps are concatenates with the corresponding feature maps. The schema of the whole process can be seen from Fig. 1 which is taken from the original paper [14]. Since the architecture looks like the letter "U", it's called U-Net.

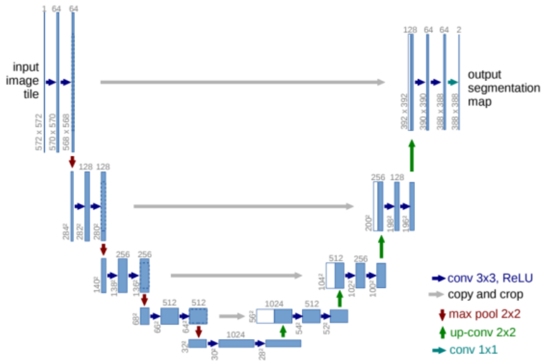

Fig. 1. U-Net architecture from original work

One difference from the original paper is the implementation of the Batch Normalization which is presented after a year from the U-Net. Batch Norm is applied after all convolution operations.

Another difference from the original paper is, in this work, only two output classes used instead of three since we only use one for black, and other for white pixels.

A subset consists of 250 images of a dataset called Carvana [15] is created for sanity check. After implementing our modified U-Net, the model trained with this created dataset to see some results if model works as expected. Only difference is from the main model is data augmentations are not applied in this version.

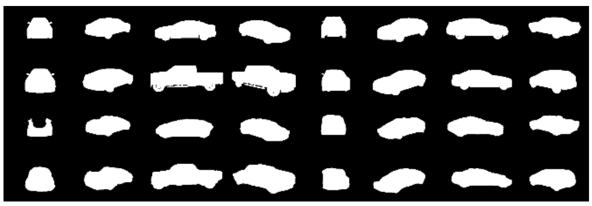

Fig. 2. Examples achieved with the Carvana dataset after 10 epochs. Ground truth (upper), predictions (down).

Saved predictions with the specified model and dataset, shows that model works fine. And it be improved with fine tuning, incrementing epoch size, etc. which is not the topic of this work.

C. Data Augmentation

Data augmentation is crucial since we have very least images. To handle this task a library called Albumentations is used.

First, a resize operation is executed. This simply divides a horizontal image vertically since the height of crops are set by 160 pixels. In this way we get 160 x 240 crops. Then these crops rotated with a limited angle. Horizontal flips are applied with a possibility of 0.5. And vertical flips are applied with a possibility of 0.1. This is because some imperfections depend on the shape, sone do not. For example, a crack may occur along the weldment. On the other hand, some defects may occur in any area without the dependency on the direction. Some inclusions and gas porosities can be given as examples to these kinds of defects. A normalization is applied to these crops to provide fast computing. Some experiments are also realized with combinations of the probabilities of these data augmentation operations.

D. Evaluations Metrics

Accuracy is not proper to use with the semantic segmentation because it's nature. Even for the worst scenarios accuracy would be higher than it reflects to reality since all images are consists of just black and white pixels. This is demonstrated explained in Fig 3.

Since accuracy is not an option for this model, "IoU" and "dice score" are the options for evaluation. "Dice score" (also known as Sørensen–Dice coefficient) is selected for this paper.

Dice score is explained with Venn schemas in Fig. 3 Main idea is the proportion of the intersected area over the whole are.

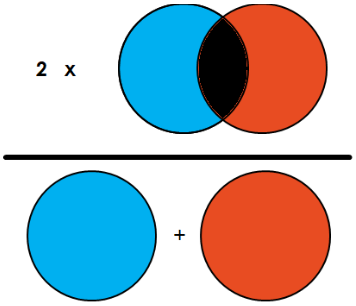

Fig. 3. Dice score explained with Venn Schemas

Loss scores is also calculated to measure system. Dice scores and loss scores are plotted and inspected via "tensorboard" module of the PyTorch.

IV. RESULTS AND DISCUSSION

It's expected to get higher dice scores and lower loss values over time. Unfortunately, in terms of GDXray weld subset, for this small dataset it's not possible to achieve a consistent model. Some results are shared below.

A. Results

In the beginning it starts with a 0 percent prediction as expected. An example is shown in Fig 4. The top images are ground truth images, while the bottom images are the corresponding predictions.

Fig. 4. Result with a dice score: 0 percent

It is possible to achieve higher scores over time during in the beginning of training. For example, Fig 5. belongs to 16 percent dice score value.

Fig. 5. Result with a dice score: 0.16

It is possible to achieve higher scores over time during training. The Fig 6. belongs to 40 percent dice score value.

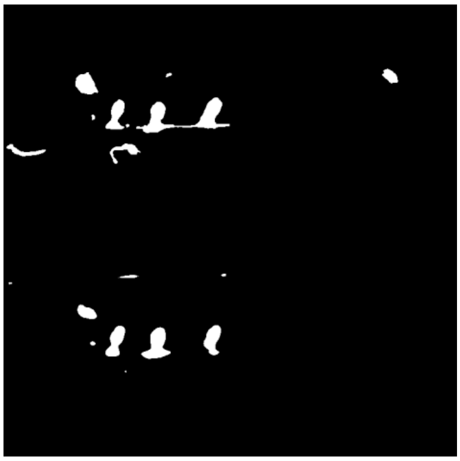

Fig. 6. Result with a dice score: 0.40

Some images can possibly get more than 0.5 dice score while it's not achieved consistently. It can be seen in Fig. 7 it's achieved more than 0.50 dice score.

Fig. 7. Result wit a dce score: 0.52

Unfortunately, model is not working consistently to get values more than 0.5. After a while dice score gets so small values and this causes a noise in the graph. It can be seen in Fig. 8.

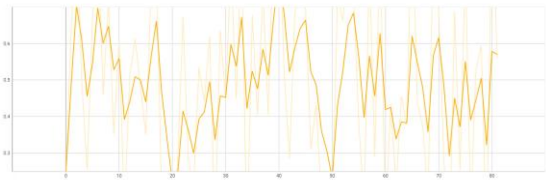

Fig. 8. Dice scores over time

On the other hand, model works over 0.90 dice score with a proper dataset. With same hyperparameters for a subset of the dataset called Lung Segmentation [16] with 173 images results are shared Fig 9.

Fig. 9. Results of the subset of the Lung Segmentation dataset

B. *Future Work*

It is possible to achieve consistent dice scores over time with transfer learning method. In this work, it was not possible to implement transfer learning to my model because of time constraints.

It's found that a proper dataset is crucial for the U-Net model even it's a popular choice for segmentation tasks with small datasets.

Acknowledgment

Thanks to Prof. Dr. H. K. Ekenel for his efforts and kindness during 2022 - 2023 Fall term. Also, thanks to Teaching Assistant of BLG506E Erdi Sarı for answering my silly questions and for his attitude.

References

[1] Nockemann, C., H. Heidt, and N. Thomsen. "Reliability in NDT: ROC study of radiographic weld inspections." NDT & E International 24.5 (1991): 235-245.

[2] Valavanis, Ioannis, and Dimitrios Kosmopoulos. "Multiclass defect detection and classification in weld radiographic images using geometric and texture features." Expert Systems with Applications 37.12 (2010): 7606-7614.

[3] Liao, T. Warren, and K. Tang. "Automated extraction of welds from digitized radiographic images based on MLP neural networks." Applied Artificial Intelligence 11.3 (1997): 197-218.

[4] Anand, Radhey Shyam, and Pradeep Kumar. "Flaw detection in radiographic weld images using morphological approach." Ndt & E International 39.1 (2006): 29-33.

[5] Mery, Domingo, et al. "GDXray: The database of X-ray images for nondestructive testing." Journal of Nondestructive Evaluation 34.4 (2015): 1-12.

[6] Hou, Wenhui, et al. "Automatic detection of welding defects using deep neural network." Journal of physics: Conference series. Vol. 933. No. 1. IOP Publishing, 2017.

[7] Liu, Bin, et al. "Weld defect images classification with vgg16-based neural network." International Forum on Digital TV and Wireless Multimedia Communications. Springer, Singapore, 2017.

[8] Sizyakin, Roman, et al. "Automatic detection of welding defects using the convolutional neural network." Automated Visual Inspection and Machine Vision III. Vol. 11061. SPIE, 2019.

[9] Wang, Yuting, Fanhuai Shi, and Xuefeng Tong. "A welding defect identification approach in X-ray images based on deep convolutional neural networks." International Conference on Intelligent Computing. Springer, Cham, 2019.

[10] A New Image Recognition and Classification Method Combining Transfer Learning Algorithm and MobileNet Model for Welding Defects

[11] Kumaresan, Samuel, et al. "Transfer learning with CNN for classification of weld defect." IEEE Access 9 (2021): 95097-95108.

[12] Yang, Lei, et al. "An automatic welding defect location algorithm based on deep learning." NDT & E International 120 (2021): 102435.

[13] Hu, Ande, et al. "Recognition of weld defects from X-ray images based on improved convolutional neural network." Multimedia Tools and Applications 81.11 (2022): 15085-15102.

[14] Ronneberger, Olaf, Philipp Fischer, and Thomas Brox. "U-net: Convolutional networks for biomedical image segmentation." International Conference on Medical image computing and computer-assisted intervention. Springer, Cham, 2015.

[15] Iglovikov, Vladimir, and Alexey Shvets. "Ternausnet: U-net with vgg11 encoder pre-trained on imagenet for image segmentation." arXiv preprint arXiv:1801.05746 (2018).

.