International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 08 Issue: 06 | June 2021 www.irjet.net p-ISSN: 2395-0072

X-Ray Weld Defect Recognition Using Deep Learning Technique

Distun Stephen[1], Dr. Lalu P.P[2], Dr. Sudheesh R.S[3]

[1]PG Student, Dept. of Mechanical Engineering, Government Engineering College Thrissur, Kerala, India
[2,3]Assistant Professor, Dept. of Mechanical Engineering, Government Engineering College Thrissur, Kerala, India

---

Abstract - *X-ray weld defect detection is a significant task in the industry which requires trained human experts and enough specialists for performing timely inspections. This paper proposes a deep learning based approach to identify different weld defects automatically from x-ray images. To employ this a dataset containing 250 x-ray images labelled for five different classes: gas pore, cluster porosity, crack, tungsten inclusion and no defect is developed. Then a Convolutional Neural Network model is designed and used for its training purpose. In addition we apply data augmentation technique to get better generalization performance with small dataset. Using this method we got a better test accuracy, which authenticates the possibility that the proposed approach is promising for weld defect recognition from x-ray images.*

***Key Words*: Digital radiography, Deep learning, CNN, Data augmentation, welding defects, Feature maps**

1.INTRODUCTION

Digital radiography [1] is a type of non-destructive testing that uses radiation such as x-rays or gamma rays to produce digital images rather than traditional x-ray film. Both the surface and internal flaws can be detected with the help of x-ray density variation. Conventional digital x-ray weld testing involves significant training for human experts to identify the defects. It also cause bottlenecks in the production/time to market timelines. Further training is required to have enough specialists for performing timely inspections. Because of these drawbacks, it is necessary to develop an automatic inspection system to improve the objectivity, consistency, accuracy and efficiency of digital radiography. Digital radiography using machine learning [2] helps human make decisions or totally replace manual method. Such systems enables early defect detection. Machine learning is a subset of artificial intelligence which includes algorithms that can learn from data and improve on their own to produce the desired output. In traditional machine learning, there will be a separate feature extraction algorithm and a classification model. Deep learning (DL) is a subdivision of machine learning with a strong emphasis on teaching computers to learn like humans: by being presented with an example. In DL, feature learning and classification is done automatically in the same model. DL is the growing trend in artificial intelligence to abstract better results when data is large and complex. Deep learning-based defect inspections are particularly effective when it comes to assessing complex surfaces and detecting cosmetic defects such as scratches or dents. As well, such systems have the ability to inspect more precisely or classify features of certain items based on their defining characteristics even if those characteristics vary in subtle but acceptable ways.

In recent years, DL methods have been achieving good performance on image related tasks such as object recognition. Singla et al. build a deep learning framework for steel surface defects Classification. S Zhou et al. proposed a classification based on the convolutional neural network, which performed very well. However, it needs lots of training data sets. Zhu et al. proposed a modified convolutional neural network (CNN) to implement classification of weld surface defect images. Zapata et al. obtained a classification system of welding defects based on an artificial neural network and an adaptive network-based fuzzy inference system. Liu et al. achieved welding defect classification with VGG16-based neural network. The classification includes two types of defects and a no defect class. Yu et al. described a localization method for casting defect based on feature matching, but the resolution was too low to detect small defect in the castings. Zahran et al. extracted feature from the power density spectra (PDS) of the weld segmented areas and used artificial neural network (ANN) to match features in order to recognize different defects. Wang et al. presented a deep learning based approach to identify three types of welding defects and their locations in X-ray images by adopting a pre-trained RetinaNet-based CNN.

This paper is based on image classification using a deep learning architecture known as convolutional neural network (CNN) which is better than other neural networks since it has features parameter sharing and dimensionality reduction. This paper aims at development of a deep learning based method to process the x-ray weld defect images and achieve a relatively better performance on a smaller dataset with five classes: gas pore, cluster porosity, crack, tungsten inclusion and no defect.

2.METHODOLOGY

The overview of the proposed method is shown in Fig. 1. Supervised deep learning is adopted for training the dataset. Firstly the dataset is created by cropping the original radiographic images and divided into training set and validation set. Secondly a CNN model is built for training and after applying data augmentation and fine tuning strategies, a generalized trained model is formed that can classifies multiple weld defects from the radiographic images.

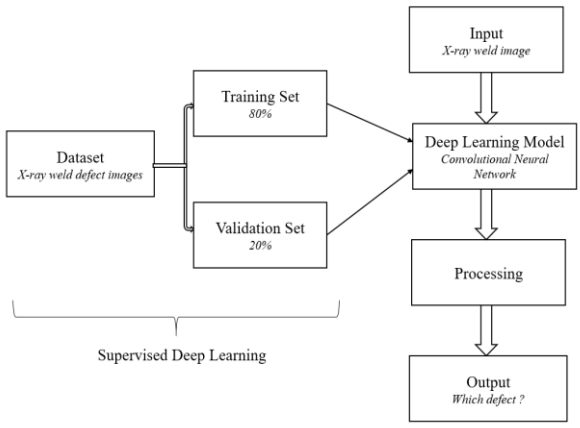

Fig -1: Methodology

3. DATASET PREPARATION

The radiographic weld defect image comes from [3], in which there are 63 images. We choose four main weld defects [4] gas pore, cluster porosity, crack and tungsten inclusion, as defects to be classified. These defects are manually cropped from the original images according to the labels and positions. After the crop, we get 250 images with 200 x 200 patch size which is then divided into 200 for training set and 50 for validation set. The classification accuracy of the supervised Deep Learning models is largely reliant on the amount and the diversity of data available during training. Deep Learning models trained to achieve high performance on complex tasks generally have a large number of hidden neurons. As the number of hidden neurons increases, the number of trainable parameters also increases. So the model requires a large amount of data to learn the values for a large number of parameters during the training phase. Data augmentation is a strategy that enables practitioners to significantly increase the diversity of data available for training models, without actually collecting new data. Data augmentation technique of rotation degree 45 and horizontal flip were applied during training process. Thus a total of 5000 images are used for training instead of 200 images. Fig. 2 shows the augmented images of a single input image.

Fig -2: Image augmentation

4. CONVOLUTIONAL NEURAL NETWORK

Convolutional neural networks are networks specialized for handling information that has a grid-like topology. CNNs architecture is composed of four different kinds of layers that are the input, convolutional, pooling and fully connected (also known as dense). It is the convolutional, pooling and fully connected layers, except output, that corresponds to the hidden layers in a CNN. These layers will be further explained in the following subsections. In Fig. 3, a general structure of a CNN is displayed.

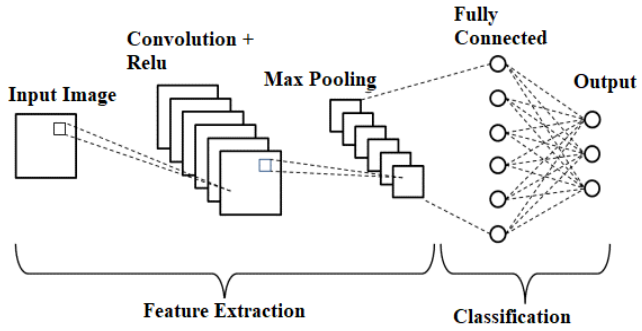

Fig -3: CNN Architecture

The programmer decides how many and what kind of layers that are needed to tackle the problem at hand. The pattern in the layered architecture can also differ between networks. It is possible to put multiple convolutional layers before a pooling layer. Multiple convolutional layers are well suited for larger and deeper networks when more detailed features need to be extracted from an image. A previous belief was that more layers provide better predictions. Recent studies have shown that just by adding layer upon layer does not improve accuracy. The study by He, K et al. show that around 150 layers are the limit for accuracy improvement on CNNs.

4.1 Convolution + ReLu layer

The input layer is where the data points serving as input is introduced to the model. In this study, the input data is images. The images will for the computer be seen as an array of pixel values. The purpose of the convolutional layer is that it applies a set amount of convolutional filters on the image and performs linear computations to the input array. The number of filters is up to the programmer to specify. The filters extract features from the image and create a corresponding feature map to each filter. The extracted features correspond to the largest changes in the images, which are gradients between different regions in the image. For example, if there is an image with white background and a straight black line in the middle, the largest gradient will be found at the interface between them. The convoluted data, i.e. the linear feature maps, is then processed by a transfer function called Rectified Linear Unit (ReLU). ReLU introduces non-linearity to the network since most of the data that CNNs process is non-linear. It corrects the feature maps with a threshold operation, any input value less than zero is set to zero thus ensuring that the feature maps are always positive. The feature maps, produced by the convolving and ReLU, are the input to the pooling layer. One way to visualize a convolutional layer is to think of the filter as a microscope that amplifies a specific region, called the receptive field, on the input data. The array of the image has

the dimensions W x H x D, where W is the width of the image, H is the height of the image and D is the number of colour channels. The filter is a matrix of numbers, called weights. The filter is smaller in spatial size than the input matrix but must have the same number of dimension. The filter will move over the image and perform element wise multiplications between the values in the filter with the original pixel values. The number obtained correspond to a high activation value in the original input. How the filters move over the input array is specified by what is called strides. Each stride moves the filter a specified length, often 2 pixels. The filter focuses on the specific region reached and calculates an activation value for each region. The specific numbers calculated create a new array of output, a feature map, which is smaller in size than the input array.

4.2 Pooling Layer

The pooling layer purpose is to down sample the output from the feature maps. The idea is to apply a filter, usually with the size 2 x 2 and a stride with the same length 2, on the output from the convolutional layer. By using a max pooling function, the highest value from the feature map is extracted. The idea is that once the specific feature from the original input is known, the relative location towards other features is more important than its exact location. Pooling significantly reduces the dimensions of the input volume. Pooling reduces the parameters by 75% and ideally obtains the most important features. The reduction decreases the computational cost of the model. One other aspect of pooling is that it decreases the problem of over fitting. In machine learning, over fitting can lead to a very good generalization on the training set. However, when it comes to validation and testing the model does not generalize properly. One issue with pooling can be that some information from the original input is lost. But the reduced risk of over fitting is often preferable compared to the possible reduction of important information. The name Fully Connected (FC) implies that every node in the previous layer is connected to every neuron in the current layer. To be able to connect every node on the first FC layer to the preceding layer, the outputs multidimensional arrays must be put in a single array. This is accomplished by applying vectorization to the matrices to perform a linear transformation into a single row vector. Every neuron in the FC layer uses all the output from the previous layer as input. All the connections lead to a great increase in parameters for the network to process. The input is weighted and an activation function, normally ReLU, is applied to determine the output. The first FC layer is often followed by additional FC layers, which have fewer neurons. This is to extract the highest activation output and to decrease the amount of parameters.

4.3 Output Layer

The output layer, or classification layer, is also an FC layer. It sorts the output from the final hidden layer into different categories using a different activation function than in the previous layers of the network. The activation function that is commonly used in this step is the softmax function. This function compresses the output from the last hidden layer into probability values in the different categories. Therefore, the sum of the outputs is always equal to 1. The output from the classification layer corresponds to the class with the highest probability. Fig. 4 shows the operation performed by the softmax function on the output layer.

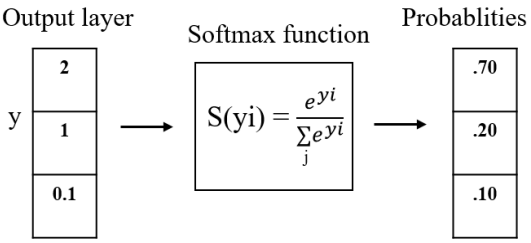

Fig -4: Output layer + Softmax function

5. MODEL CREATION AND TRAINING

The model was built using the open source neural network application programming interface (API) provided by Keras Documentation [5], which is written in Python programming language with tensor-flow as backend. The proposed model has four 2D convolutional layers, four ReLu activation layers, four pooling layers, one dropout layer. The dropout rate that is added to the model is an approach to regularization in neural network which helps reducing interdependent learning amongst the neurons. Dropout refers to ignoring units (i.e. neurons) during the training phase of certain set of neurons which is chosen at random. After the first convolution layer 32 feature maps is created and at next convolution layer 64 feature maps and finally 128 feature maps are generated at the last convolution layer. Python programming language is used for model creation with Keras – tensor-flow as backend. By applying fine tuning of the model parameters, the proposed model have four convolutional layers having convolution filter size 3X3, pooling filter size 2X2 and a dropout rate of 0.5.

Since deep learning model requires a lot of computational power to run on, the training was conducted at Nodal Centre for Robotics and Artificial Intelligence (NCRAI) lab [6]. The training process involves finding a set of weights in the network that proves to be good, or good enough, at solving the specific problem. This training process is iterative, meaning that it progresses step by step with small updates to the model weights each iteration and, in turn, a change in the performance of the model each iteration. The iterative training process of neural networks solves an optimization problem that finds for parameters (model weights) that result in a minimum error or loss when evaluating the examples in the training dataset. First the total loss at output layer is calculated as

$$L = -\Sigma i\, T(i) log P(i)$$

where,

T(i) = Target probability distribution

P(i) = Predicted probability distribution

Secondly the weight values are updated to minimize the total loss value. The optimizer root mean square propagation (RMS Prop) restricts the oscillations in the vertical direction. Therefore, we can increase our learning rate and our algorithm could take larger steps in the horizontal direction converging faster. The equation is as follows,

$$Wn = Wx - \eta^I\, \partial(L)/\partial Wx$$

$$\eta^I = \eta(Wavg + \varepsilon)^{1/2}$$

$$Wavg(t) = 0.9 * Wavg(t-1) + 0.1(\partial L/\partial Wt)^2$$

where,

η^I *= new learning rate*

η *= initial learning rate*

Wx = initial weight value

Wn = new weight value

$Wavg$ = weighted average at time t

Training consisted of 100 epochs with validation after each in order to supervise the improvement of the network. In every epoch a batch size of 4 was set, meaning that the program takes 4 images before each back propagation and weight update. Fig. 5 shows the feature maps obtained at each convolutional layers for the weld crack image. Feature maps are generated by applying filters or feature detectors to the input image. Feature map visualization will provide insight into the internal representations for specific input for each of the Convolutional layers in the model. Visualizing an inside story of how CNN learns to identify different features present in images provides a deeper insight into how the model works. It also help to understand why the model might be failing to classify some of the images correctly and hence fine-tuning the model for better accuracy and precision.

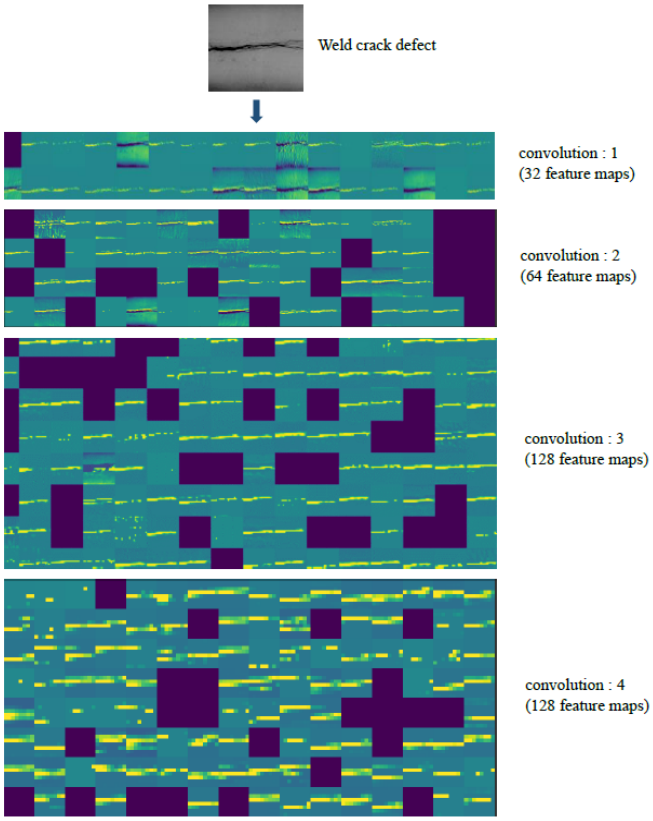

Fig -5: Feature maps for weld crack defect

6. RESULTS AND FINDINGS

The CNN model was trained for 100 epochs and 50 images were used for its cross validation. Each class consists 10 unknown images. Fig 6 represents the confusion matrix obtained for the test set.

Actual class	Cluster porosity	Crack	Gas pore	No Defect	Tungsten inclusion
Cluster porosity	**10**	0	0	0	0
Crack	0	**10**	0	0	0
Gas pore	0	0	**10**	0	0
No Defect	0	**7**	**2**	**1**	0
Tungsten inclusion	0	0	0	0	**10**
	Cluster porosity	Crack	Gas pore	No Defect	Tungsten inclusion

Predicted class

Fig -6: Confusion matrix for test set

Since CNN model uses edge detection method, no edges were detected from the no defect images and were wrongly classified. All other weld defects were correctly classified accordingly. Irrespective of input image size, the classifier done well on new images other than the test set. Fig 7 shows

some of the output predictions made by the x-ray weld defect classifier for new images.

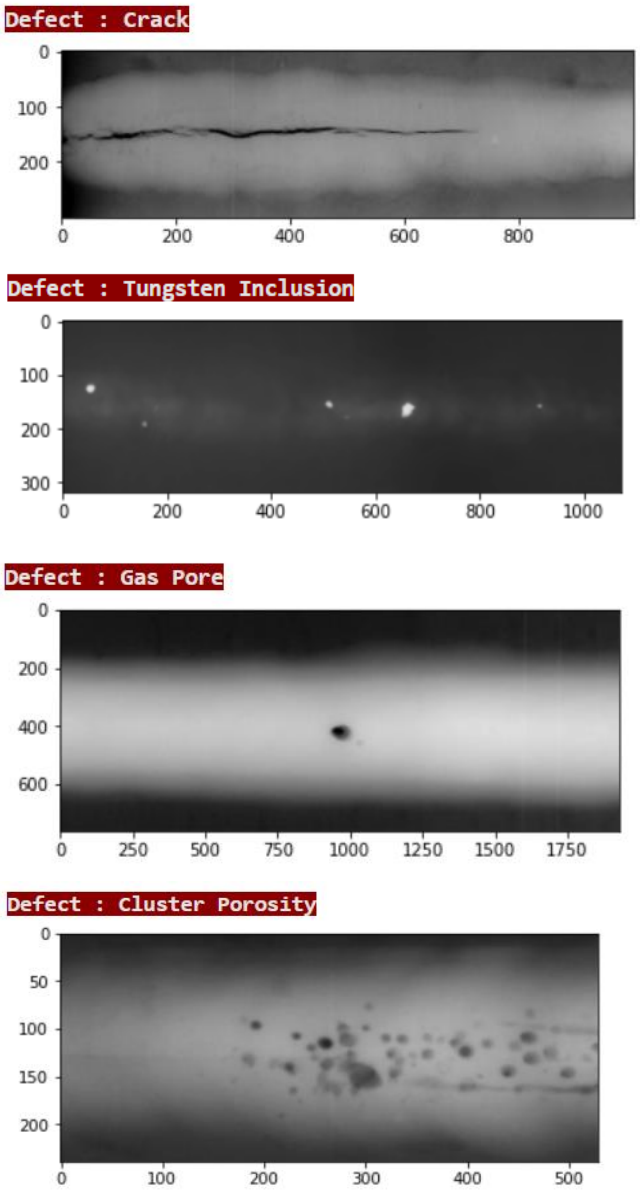

Fig -7: Output predictions made by the classifier

7. CONCLUSIONS

The implementation of deep learning model for the weld defect classification from x-ray images has been successfully employed. The model is developed with the help of GD x- ray image database that are publically available for use. As the database contains only minimum number of inputs we have extracted the sub-images and created our own dataset images which are used for the training and the testing process. The proposed method detects most of the defects accurately but not for no defect images. Data augmentation technology and fine tuning strategies helped to achieve better performance of the model. The present work can be extended to the classification of other welding defects.

ACKNOWLEDGEMENT

It is a great pleasure for me to acknowledgment all those who have assisted and supported me to lead my work to success. First of all, I would like to thank GOD almighty for blessing me with his grace and taking my endeavor to a successful culmination. I extend my sincere gratitude towards the second author Dr. Lalu P.P and the third author Dr. Sudheesh R.S who gave me immense support and knowledge for completing this work.

REFERENCES

[1] Edson Vasques Moreira, Heleno Ribeiro Simoes, Jose Maurıcio Barbosa Rabello, Marcelo dos Santos Pereira, “Digital radiography for the inspection of weld seams of pipelines – better sensitivity”, Welding International Vol. 24, No. 4, April 2010, 249–257.

[2] Nur Farhana Hordri, Siti Sophiayati Yuhaniz, Siti Mariyam Shamsuddin, “Deep Learning and Its Applications: A Review”, 2016.

[3] Mery D, Riffo V, Zscherpel U, Mondrag on G, Lillo I, Zuccar I, Lobel H, Carraso M: GDXray: the database of x-ray images for nondestructive testing. J. Nondest. Eval. 34, 42 (2015).

[4] Cory, W.Bill, “Quality assurance, inspection and performance certification”, Fans and Ventilations, 265-280, A practical guide, ELSEVIER, 2005.

[5] F.Chollet,“Keras,”2015, https://github.com/fchollet/keras. View at: Google Scholar.

[6] Nodal Center for Robotics and Artificial Intelligence (NCRAI) Lab, Government Engineering College Thrissur, https://github.com/Nodal-Centre-for-Robotics-and-AI-GECT.

[7] Karun Singla, Gangesh Chawla, Ranganath M. Singari, “Deep Learning Framework for Steel Surface Defects Classification”, IJAPIE-2019-01-135, Vol 4 (1), 25-32.

[8] Shiyang Zhou, Youping Chen, Dailin Zhang, Jingming Xie, and Yunfei Zhou. Classification of surface defects on steel sheet using convolutional neural networks. Mater. Technol, 51(1):123–131, 2017.

[9] Haixing Zhu, Weimin Ge, Zhenzhong Liu, “Deep Learning-Based Classification of Weld Surface Defects”, Appl. Sci. 2019, 9, 3312, MDPI.

[10] Juan Zapata, Rafael Vilar, Ramon Ruiz, “Performance evaluation of an automatic inspection system of weld defects in radiographic images based on neuro-classifiers,” EXPERT SYST APPL, 38(7), 8812-8824, 2011.

[11] Bin Liu, Xiaoyun Zhang, Zhiyong Gao, Li Chen, “Weld defect images classification with VGG16-based neural network”, IFTC 2017. CCIS, vol. 815, pp. 215–223. Springer, Singapore (2018).

[12] Y. Yu, L. Du, C. Zeng, J. Zhang, “Automatic localization method of small casting defect based on deep learning feature. Chin. J. Sci. Instrum. 37, 1364–1370 (2016).

[13] O. Zahran, H. Kasban, M. El-Kordy, F.E.Abd El-Samie, "Automatic weld defect identification from radiographic images", NDT E Int. 2013, 57, 26–35.

[14] Yuting wang, Fanhuai, Xuefeng Tong: "A Welding Defect Identification Approach in X-ray Images Based on Deep Convolutional Neural Networks "Springer Nature Switzerland AG 2019.

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun, "Deep Residual Learning for Image Recognition", arXiv: 1512.03385v1 [cs.CV] 10 Dec 2015.

[16] Kotme Nikita, Kulkarni Apurva, Lahane Aishwarya, Kare Komal, V. P. Mulik "Detection of Defects on Steel Surfaces Using Machine Learning Techniques" IJREAM, ISSN: 2454-9150 Vol-03, Issue 02, Apr 2017.

BIOGRAPHIES

Distun Stephen
PG Student
Dept. of Mechanical Engineering
Government Engineering College
Thrissur