Steel Surface Defect Detection Using Convolutional Neural Network

KATEB Yousra(1)*, MEGLOULI Hocine(2),KHEBLI Abdelmalek3)
Automation and Control in Petrochemical Industries; University M'hamed Bougara of Boumerdes, ALGERIA
y.kateb@univ-boumerdes.dz
h.meglouli@univ-boumerdes.dz
a.khebli@univ.boumerdes.dz

Abstract: Steel is the most important engineering and construction material in the world. It is used in all aspects of our lives. But as every metal is can be defected and then will not be useful by the consumer Steel surface inspection has seen an important attention in relation with industrial quality of products. In addition, it has been studied in different methods based on image classification in the most of time, but these can detect only such kind of defects in very limited conditions such as illumination, obvious contours, contrast and noise...etc. In this paper, we aim to try a new method to detect steel defects this last depend on artificial intelligence and artificial neural networks. We will discuss the automatic detection of steel surface defects using the convolutional neural network, which can classify the images in their specific classes. The steel we are going to use will be well-classified weather the conditions of imaging are not the same, and this is the advantage of the convolutional neural network in our work. The accuracy and the robustness of the results are so satisfying.

Keywords: Steel surface, Defect detection, Image classification, Convolutional Neural Network, ResNet-50.

1. INTRODUCTION

Steel, as a valuable metal, has had the best effect on our humankind life. It is, by a long shot, the most widely used, multi-utilitarian characteristic material found and mined from the earth. Actually, many would venture to such an extreme as to state that the improvement of humankind itself would have been inconceivable without the development of steel [1], and has an almost unlimited variety of uses in today's world, which is a marker for its incredible adaptability. We can find out some of its powerful characteristics that include: (1) Resistant to deformation at both high and cold temperatures, (2) corrosion resistant, (3) it can be easily machined, (4) it is formable to both hot and cold and (5) it's hard wearing, tough and all-weather-proof. [1]

The inspection of steel quality is a well-known problem since the last two decades, which it is mostly done visually, where human inspectors classify steel defects using their naked eye. This last makes sometimes-incorrect classification due to human control conditions, such as tiredness, which makes the visual inspection unreliable, obvious and sometimes drudgery. The automatic visual inspection can lead to a better results that human cannot give it in such case because of its reliability, clarity and precision. We aim in this research to detect steel defects by using approach for steel defect classification using Artificial Neural Network ANN [30] and more precisely CNN model based on the pre-trained Residual Network architecture with 50 layers (ResNet-50) [3].

Artificial Neural Networks have been developed as generalizations of mathematical models of human cognition or neural biology. Researchers have characterized a neural network by its pattern of connections between the neurons (called architecture), its method of determining the weights on the connections (called training or learning algorithm) and its activation function. [30]

ANNs have attracted a great deal of attention because of their pattern recognition capabilities, and their ability to handle noisy data; however, its ability to perform well is greatly influenced by the weight adaptation algorithm and the amount of noise in the data. [30]

We have organized sections of this paper as follow: In section II, we will describe the types of steel defects and the techniques used for its inspection, in the section III, we will present the experimentations of our method. The fourth VI and last section is for results discussion and conclusion.

2. STEEL DEFECT TYPES AND DETECTION INSPECTION TECHNIQUES

A. Steel Defect Types

A study has shown that there are more than 200 types ofdefects, that were determined in the biggest steel manufacturer in IRAN in 2014 [2]. However, in this work, we will focus only on three types of steel: a- Not defective steel, b- rusty steel and c-cracked steel. These types are shown in **Fig.1, Fig, 2** and **Fig.3** respectively.

Fig. 1: Non-Defective Steel

In this research, we have taken a database of steel defect images from the Google Image Search, in order to train a classifier for images containing **cracks**, **rust** and **non-defective steel**. As we will optimize its retrieval performance.

Fig.2: Rusty steel

We will use in this work the MATLAB software to implement the algorithm, to plot the error curve, and to show the accuracy of our method.

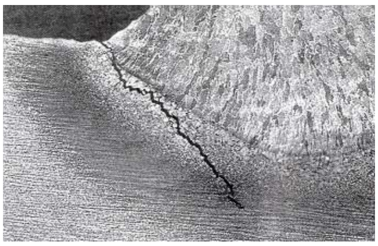

Fig.3: Steel crack

B. Steel inspection techniques

According to recent researches about steel inspection, A variety of techniques were found in the literature, they depend on steel category (slab, plate, billet…etc.). We can find in previous researches the Unsupervised learning where [4], [5] and [6] used Self-Organizing Map SOM method, [7] [8] used Learning Vector Quantization LVQ. As we can find thesupervised classifiers in [9]. [10] Where they used the k-Nearest Neighbor KNN, [11] [4] [12] used Neural Network-Back Propagation technique (NN-BP)andSupport Vector Machine SVM was used by [14] [15] [16] [17] [18] and [19]. Genetic Algorithm(GA) as announced in [31].

3. DATA PREPARATION AND EXPERIMENTATION PART

In this part, we will apply our method of classification using convolutional neural network **CNN** where we will use a pre-trained Residual Network **ResNet-50** [3], to gain time and accuracy at the same time. The reason to use such a method is explained in this section. The datasets training passes throw many steps, which are mainly:

. A. Image acquisition

Firstly, we have selected a group of images that are convenient to our work which we classified manually using our naked eye. We choose an average of **120** different pictures from different classes and in various conditions of illumination, contrast…etc. from the Google Search Images. Fig.4 shows three samples of steel images plotted in MATLAB:

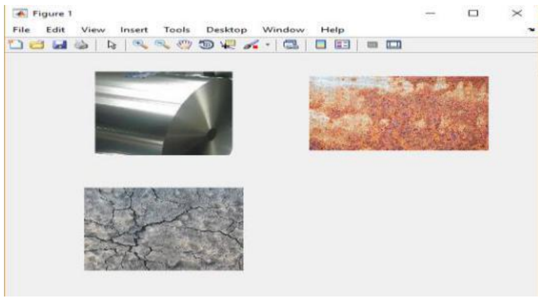

Fig.4: *the image shows non-defective, rusty, cracked steel, respectively plotted in MATLAB*

B. Preprocessing

The preprocessing step is very important and it facilitates the classification later, we re-sized the images to be conforming to **ResNet-50** input size, which is 224-by-224-by-3 as we converted all of them to **RGB** color.

We categorized the collected dataset into three different folders; each folder took a name as the class label of the steel image that it contains: not defective steel, rusty steel and cracked steel. We Split the sets into training and validation data. Pick 70% of images from each set for the training data and the remainder, 30%, for the validation data. We randomized the split to avoid biasing the results. The CNN model will process the training and test sets.

C. Features extraction

It involves reducing the quantity of resources needed to describe large set of information. In our net, the feature extraction step includes several convolution layers followed by max pooling and an activation function. In addition, a Softmax layer at the end of the network. [4]

Notice how the first layer of the network has learned filters for capturing blob and edge features. These "primitive" features are then processed by deeper network layers, which combine the early features to form higher-level image features. These higher-level features are better suited for recognition tasks because they combine all the primitive features into a richer image representation. [20]

D. Data training

We pre-trained our data by choosing the architecture of **ResNet-50** and this is to solve the problem of saturation and accuracy degradation [3]. This model was the winner of Image Net challenge in 2015 [21].

E. ResNet and the degradation problem

In general, in a deep convolutional neural network, several layers are stacked and are trained to the task. [22], the network learns several low/mid/high level features at the end of its layers. In residual learning, instead of trying to learn some features, we try to learn some residual Fig.5. Residual can be simply understood as subtraction of feature learned from input of that layer [22].

ResNet does this using shortcut connections (directly connecting input of (n)th layer to some (n+x)th layer. (Fig.5)

Even if there is vanishing gradient for the weight layers, we always still have the identity *x* to transfer back to earlier layers. (eq. 1).

Recall that a convolution operation typically shrinks the spatial resolution of an image, e.g. a 3x3 convolution on a 32 x 32 image results in a 30 x 30 image. The identity mapping is multiplied by a linear projection **W** to expand the channels of shortcut to match the residual (**Fig.5**). This allows for the input **x** and **F(x)** to be combined [23] as input to the next layer (Eq.2).

In the figure below, we can see the clear difference between a plain net and a residual net [24]:

$$H(x) = F(x) + x \qquad (1)$$

$$y = (x,\{Wi\}) + Ws * x \qquad (2)$$

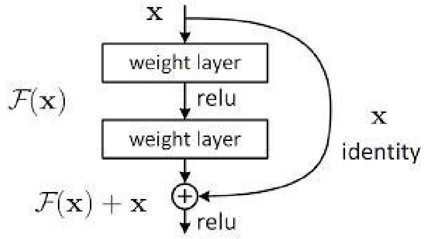

Fig.5: A simple model of ResNet [24]

In the figure below Fig.6, we can see the clear difference between a plain net and a residual net [24]:

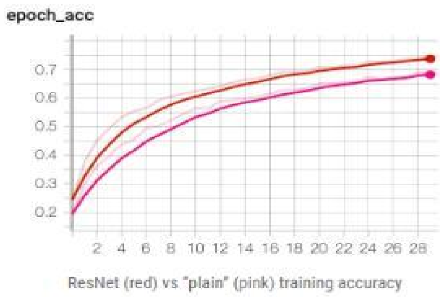

Fig. 6: the training accuracy between ResNet (red) and plain (pink)network

It has been proved that training this form of networks is easier than training simple deep convolutional neural networks and the problem of degrading accuracy is resolved [25]. Both Fig.7 and Fig.8 show the architecture of ResNet-50.

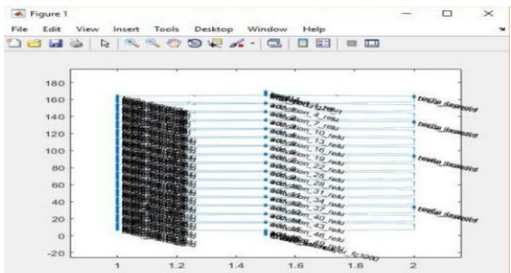

Fig.7 Architecture of ResNet-50

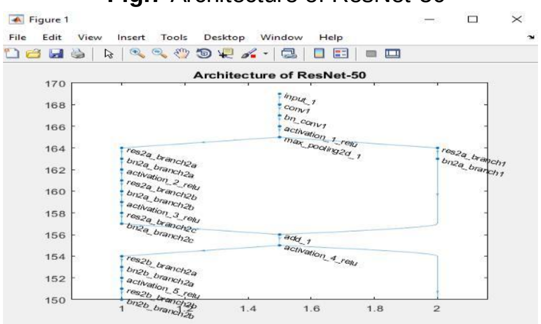

Fig.8: Architecture of ResNet-50 limited between [150 170]

The ResNet-50 model consists of five stages each with a convolution and Identity block. Each convolution block has three convolution layers and each identity block has three convolution layers. The ResNet-50 has over 23 million trainableparameters [26]. In the Fig.9 below, we can see the ResNet-50 blocks:

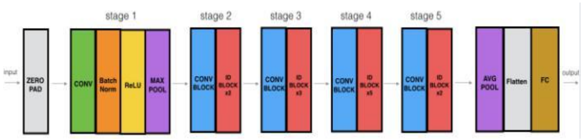

Fig.9: ResNet-50 Model [26]

F. Weight initialization

For a convolutional layer, we can write the response as in (3):

$$\mathrm{Yi} = \mathrm{Wi} * \mathrm{xi} + \mathrm{Bi} \qquad (3)$$

Where **Wi** is the weight for the i(th) neuron, **Yi** is the response, **Bi** is the bias, **xi** is the i(th) input neuron.

In addition, to initialize the weight, we follow the equation below:

$$\mathrm{W(new)}=\mathrm{W(old)}- \mu^* \qquad (4)$$

Where **C** is the cost function and is defined by (5):

$$C = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \qquad (5)$$

In addition, **μ** is the learning rate

The first convolutional layer weight is shown in the figure below:

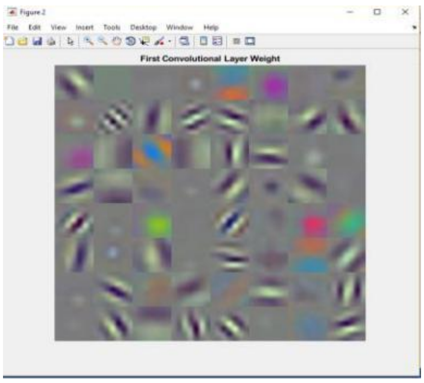

Fig.10: First convolutional layer weight

After preprocessing and training our network, we will classify the data as they are categorized in the folders. We used in this section the Support Vector Machine called **SVM** for the following reasons: 1) because it is relatively memory efficient, 2) it can classify features clearly, 3) it has simple decision boundary, and 4) it prohibits over-fitting. As it maximizes the margin between two classes [27].

We have chosen the Binary SVM because it is multi-classifier, and that what we need in our case. (Three classes will be recognized). An SVM, has generally the following [28] form: Eq (6)

$$Ys(W^*Xs +b) -1= 0 \qquad (6)$$

We used ***K*(*K*–1)/2** binary support vector machine **(SVM)** models using the one-versus-one **(OVA)** coding design, where *K* is the number of unique class labels (levels). An *error-correcting output codes* ***(ECOC)*** *model* reduces the problem of classification with three or more classes to a set of binary classifiers. For each binary learner, one class is positive, another is negative, and the rest are ignored. (Fig.11) This design exhausts all combinations of class pair assignments [29].

	Learner 1	Learner 2	Learner 3
Class 1	1	1	0
Class 2	−1	0	1
Class 3	0	−1	−1

Fig.11: A one-versus-one coding design

confMat =

19	3	1
0	23	0
2	1	20

confMat =

0.8261	0.1304	0.0435
0	1.0000	0
0.0870	0.0435	0.8696

Fig.12 : confusion matrix **fig.13 :** confusion matrix %

Now, we display the mean accuracy, which is 0.8986 or 89.86%. After ending with the classifier evaluation, we will try the newly trained classifier on new images. We furnished new images that is not seen before by the net, we Created augmented Image Data store IMDS to automatically resize the image when image features are extracted using activations. After that, we will extract the new image features using our trained CNN, and makethe prediction using our classifier.

The classification worked perfectly as we can see in the figures below (**Fig.14, Fig.15, and Fig.16**). Our classifier could recognize the image class correctly.

The Mini-Batch-Size is set 32 to ensure that the CNN and image data fit into GPU memory.

4. RESULT DISCUSSION

A. Prediction step and classifier evaluation

Now we repeat the procedure described earlier to extract image features from test Set. The test features can then passes to the classifier to measure the training accuracy.

In the prediction step, we used our classifier that contains the error-correcting output codes **(ECOC)** the training labels, the learner…etc. The parameters such as the activations, the network, feature layers, mini-batch-size-32...etc, were stocked in the test features. In the classifier evaluation test, we first extracted test features using the CNN, we passed CNN image features to trained classifier, we got the known labels, and we tabulated the results using a confusion matrix **fig.12**. We converted confusion matrix into percentage form as in **fig.13**

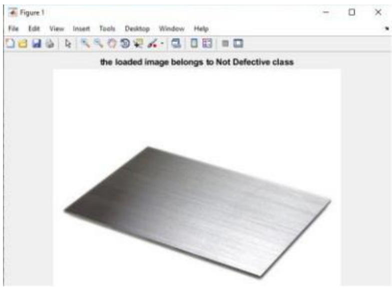

Fig.14: The loaded image belongs to **Not Defective** class

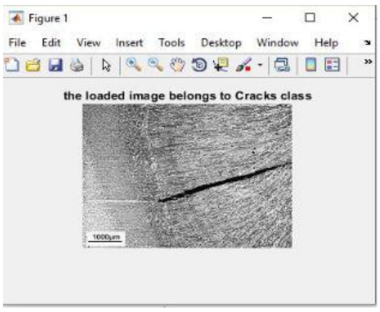

Fig.15: The loaded image belongs to **Cracks** class

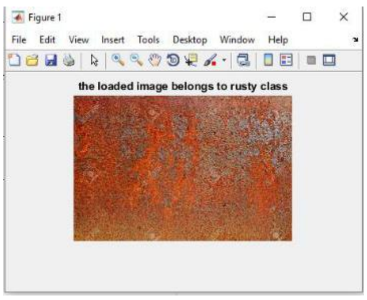

Fig.16: The loaded image belongs to **Rusty** class

V. CONCLUSION

In this study, we have shown how powerful tool is convolutional neural network. Especially when working on a vision classification task. We did not use a network that we trained ourselves, but we preferred using a pre-trained network ResNet-50, that helped us a lot to improve the classification to gain both time and accuracy, which are the factors to judge the performance of our network. We got a very efficient network that future researchers can utilize immediately to classify new images with different classes.

References

[1] K. Kyle, Importance of Steel: A History of Steel and Why It's Still Important Today, 2018

[2] M. Sharifzadeh, S. Alirezaee, R. Amirfattah and S. Sadri, Detectionof Steel Defect Using the Image Processing Algorithms M. Sharifzadeh Isfahan University of Technology, Isfahan, Iran 2014.

[3] Kaiming He, Xiangyu Zhang and ShaoqingRenJian Sun Deep Residual Learning for Image Recognition, Microsoft Research, 2015.

[4] P. Caleb, M. Steuer, Classification of surface defects on hot rolled steel using adaptive learning methods, KES'2000. Fourth international conference on knowledge-based intelligent engineering.

[5] T Maenpaa, Surface Quality Assessment with Advanced Texture Analysis Techniques (Proc. Of International Surface Inspection Summit, Luxembourg, 2006).

[6] LAO Martins, FLC Padua, PEM Almeida, Automatic detection of surface defects on rolled steel using Computer Vision and Artificial Neural Networks, IECON 2010 - 36th Annual Conference on IEEE Industrial Electronics Society, 2010, pp. 1081–1086

[7] G Wu, H Zhang, X Sun, J Xu, K Xu, A Bran-new Feature Extraction Method and its application to Surface Defect Recognition of Hot Rolled Strips (Proceedings of the IEEE International Conference on Automation and Logistics, Jinan, China, 2007), pp. 2069–2074. doi:10.1109/ICAL.2007.4338916

[8] J Olsson, S Gruber, Web process inspection using neural classification of scattering light. IEEE Transaction on IndustrialElectronics 40(2), 1443– 1448 (1993).

[9] F Dupont, C Odet, M Carton, Optimization of the recognition of defects in flat steel products with the cost matrices theory. NDT&E International 30(1), 3–10 (1997). URL:www.engr.wisc.edu/mpac/pdf/paper13.pdf, accessed July '12.

[10] C Ünsalan, A Erçil, Automated Inspection of Steel Structures, Recent Advances in Mechatronics (Springer-Verlag Ltd., Singapore, 1999), pp. 468–480.

[11] MR Yazdchi, AG Mahyari, A Nazeri, Detection and Classification of Surface Defects of Cold Rolling Mill Steel Using Morphology and Neural Network, pp. 1071–1076. IEEE CIMCA 2008, IAWTIC 2008, and ISE 2008.

[12] Dusmanta K. Mohanta1 and Pranab K Dutta, Review of vision-based steel surface inspection systems Nirbhar Neogi1*, 2014.

[13] B Suvdaa, J Ahn, J Ko, Surface defects detection and classification using SIFT and voting strategy. International Journal of Software Engineering and Its Applications 6(2), 168–170, (2012).

[14]] H Jia, YL Murphey, J Shi, T Chang, an Intelligent Real-time Vision System for Surface Defect Detection (IEEE-Proceedings of the 17th International Conference on Pattern Recognition, 2004), pp. 2–5.

[15] K Choi, K Koo, J Lee, Development of defect classification algorithm for POSCO rolling strip surface inspection system. SICE-ICASE International Joint Conference 10, 2499–2502 (2006).

[16] K Agarwal, R Shivpuri, Y Zhu, T Chang, H Huang, Process knowledge based multi-class support vector classification (PK-MSVM) approach for surface defects in hot rolling. ELSEVIER, Expert Systems with Applications 38, 7251–7262 (2011).

[17] S Ghorai, A Mukherjee, M Gangadaran, PK Dutta, Automatic defect detection on Hotrolled flat steel products. IEEETransInstrumMeas 62(3), 612–621 (2013).

[18] B Suvdaa, J Ahn, J Ko, Surface defects detection and classification using SIFT and voting strategy. International Journal of Software Engineering and Its Applications 6(2), 168–170 (2012).

[19] J Zhao, Y Yang, G Li, The Cold Rolling Strip Surface Defect On-Line Inspection System Based on Machine Vision (Second Pacific-Asia Conference on IEEE Circuits, Communications and System (PACCS), 2010), pp. 402–405.

[20] Donahue, Jeff, et al. "Decaf: A deep convolutional activation feature for generic visual recognition." arXiv preprint arXiv:1310.1531 (2013).

[21] Sik-Ho TSANG, ResNet- Winner of the ILSVRC 2015 (image classification, localization, detection, 2018.

[22] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2016.

[23] CONNOR Shorten, Introduction to ResNets, 2018.

[24] Andrew THOMAS, Introduction to the ResNet architecture; 2018.

[25] Kartikpudugo , What is the deep neural network known as "ResNet-50"? 2019.

[26] PRIYA Dwivedi , Understanding and coding a ResNet in Keras, 2019.

[27] Elena Marcher, Feature selection, SVM-based classification and application to mass spectrometry data analysis. Department of Computer ScienceVrijeUniversiteit Amsterdam, 2015.

[28] Edward Ma, Math behind SVM (Support Vector Machine) 2019.

[29] Fürnkranz, Johannes, "Round Robin Classification." J. Mach. Learn. Res., Vol. 2, pp. 721–747, 2002.

[30] FatmaZohra DEKHANDJI,Salim TALHAOUI and Youcef ARKAB, "Power Quality Detection, Classification and Monitoring Using LABVIEW", Algerian Journal of Signals andSystems, Vol. 4, Issue 2, December 2019. ISSN : 2543-3792-EISSN : 2676-1548. pp.: 101-111.

[31] Abdelmadjid RECIOUI, Mondher BENABID, Nabil DJILANI. "Rectangular Antenna Array Optimization using Wind Driven Optimization". Vol.1, Issue 2, December 2016| ISSN-2543-3792.pp.109