

Steel Surface Defect Detection using Deep Learning

Vira Fitriza Fadli
Department of Informatics
Widyatama University
Bandung, Indonesia

Iwa Ovyawan Herlistiono
Department of Informatics
Widyatama University
Bandung, Indonesia

Abstract:- Steel defects are a frequent problem in steel companies. Proper quality control can reduce quality problems arising from steel defects. Nowadays, steel defects can detect by automation methods that utilize certain algorithms. Deep learning can help the steel defect detection algorithm become more sophisticated. In this study, we use deep learning CNN with Xception architecture to detect steel defects from images taken from high-frequency and high-resolution cameras. There are two techniques used, and both produce respectively 0.94% and 0.85% accuracy. The Xception architecture used in this case shows optimal and stable performance in the process and its results.

Keywords:- Defect Detection, Steel Defect, Deep Learning, Xception.

I. INTRODUCTION

In the steel industry, quality control during the production process becomes very important. Steel quality control performed with detecting defects on the steel surface. Quality degradation will occur if steel defects are not detected correctly. Detection of steel defects at the right time can help in handling quality problems of steel to be produced. Directly the quality of steel will determine the durability and lifetime of the steel itself.

Nowadays, automation has utilized in carrying out quality control. One method in detecting defects can help by the presence of images from high-frequency and high-resolution cameras with the application of algorithms in them [1]. This detection requires a sophisticated learning algorithm that can improve the quality of detecting over time. Deep Learning technology supports the technique because of its ability to learn its computing.

Deep learning has revolutionized various industries because of excellent performance in computer vision. Unlike its predecessor machine learning, deep learning can work without instructions from its creator to produce fast and accurate predictions [2] so that it can help the workload of engineers in the steel industry.

The Deep Learning Model that commonly used for image recognition is the Convolutional Neural Network (CNN). CNN will help find defective objects contained in steel surface images. Therefore, deep learning with the CNN model will use in this study to detect steel defects.

II. RELATED WORKS

There are many previous algorithms applied to detect steel defects. The algorithms used are various, namely SIFT and Voting Strategy [3], YOLO network [4], ANN, and DAN algorithms [5], etc. S. Tian and K. Xu [6] detected steel defects with genetic algorithms and extreme machine learning. This ELM is combined with genetic algorithms to improve the results obtained. The results of experiments with steel defect samples show that the G-ELM algorithm effectively increases the ELM algorithm's identification accuracy.

M. S. Kim et al. [7] applied deep learning to classify steel defects using the CNN model combined with Siamese neural network types. In their research, CNN used in feature extraction and Siamese used as a network structure with a scheme using a convolutional layer with a small filter size. Of all the algorithms used, it produces good accuracy in classifying steel defects. S. Y. Lee et al. [8] and J. L. Greece et al. [9] have also used deep learning and CNN in their research to detect steel defects.

In previous studies that applied deep learning with the CNN model, we reviewed the primary ability of CNN can be affected by the use of architecture (pre-trained models). CNN has a variety of architectures, and generally, these architectures have several different layers. Three CNN architectures are the most known and commonly used, namely VGG16 [10], Inception V3 [11], and Xception [12]. Based on ImageNet Keras [13], Table 1 describes the performances of each architecture.

Models	Size	Top-1 Acc	Top-5 Acc	Depth
VGG-16	528 MB	0.713	0.901	23
InceptionV3	92 MB	0.779	0.937	159
Xception	88 MB	0.790	0.945	126

Table 1:- Performances pre-trained CNN

In the table, the top-1 and top-5 accuracy refers to the performance of the model in the ImageNet validation dataset. Xception architecture has the highest accuracy results of top-1 and top-5 accuracy and uses smaller memory compared to VGG-16 and InceptionV3 architectures.

Some studies discuss the comparison of the three architectures in terms of performance. Comparison of Xception architecture and Inception V3 in F. Chollet's research [12] proves that compared to InceptionV3, Xception shows a small advantage in its classification

performance in the ImageNet dataset however Xception has a significant advantage in JFT. E. Ayan and H. M. Unver [14], who compared between Xception and VGG16, proved that Xception outperforms with VGG16 in diagnosing pneumonia. On the other hand, VGG16 shows better performance in diagnosing normal cases. For a separate comparison of InceptionV3 and VGG16, we have not found research to discuss the topic. But the investigation by C. A. Mamani Diaz et al. [15] compared the architecture of VGG-16, Inception V3, and Xception in classifying agricultural precision using deep learning. Several experiments conducted to support the selection of architecture with satisfactory performance. In the testing process, experiments using different epochs in all three architectures prove that at 300 cycles, the training behavior of VGG16 and InceptionV3 stagnated. At the same time, Xception continued to improve because of its active layers and faster classification capabilities than other architectures. This research proves that the performance of the Xception architecture is more prominent than the others.

From previous studies discussing three architectural comparisons, we conclude that Xception will use as an architecture or pre-trained model to detect steel defects in this study. But in the final results, we will explain the accuracy and classification results of the VGG16 and InceptionV3 architectures for this case.

III. DEEP LEARNING

Deep learning is a sub-field of machine learning that emphasizes a multi-layered learning process. Deep learning can solve complex problems due to features that can extract automatically. The ability of deep learning proven in image recognition, audio, video, text classification, etc. Deep learning related to an algorithm that mimics how the basic system of the human brain works, which is commonly called the Artificial Neural Network.

A deep neural network, Convolutional Neural Network, Recurrent Neural Network, is neural network methods commonly used in deep learning. Convolutional neural networks are methods specifically designed for image recognition and using certain architectures. Unlike other types of neural networks, CNN can hold all the information contained in the image and can produce a more accurate classification of objects.

There are two main block structures of CNN, namely feature extraction and classification. Feature extraction consists of convolutional layer operations and the pooling layer (sub-sampling layer). The classification consists of fully connected layers equipped with activation functions (softmax, sigmoid, etc.). In feature extraction, the convolutional layer is the first layer whose operation is to take an image from the input layer and then extract the image features according to the specified filter. When the convolutional layer calculation operation performed, the resulting value may be negative; therefore, the RELU activation function will use in here. The RELU activation function performs its action by changing the negative value

of the feature map to zero. The next step of feature extraction is the pooling layer, which works by reducing the resulting convolutional layer's dimensions. The pooling layer consists of max pooling and average pooling, but generally, what used is max pooling. After the feature extraction stage, it will be forward to the layer whose function is as a classification. The fully connected layer is the last layer that predicts output. The FC layer has an activation function that used to find the probability value in forecasting the classification results after the feature map has finished operating.

IV. METHOD

In this study, two techniques will use to detect steel defects. The first technique is a binary classification that recognizes images that have defects or no defects. Besides that, the function of this binary classification for filtering defect images to the next stage. In the second technique, multilabel classification performs, which detects images that have 1 class of defects or multi-class of defects.

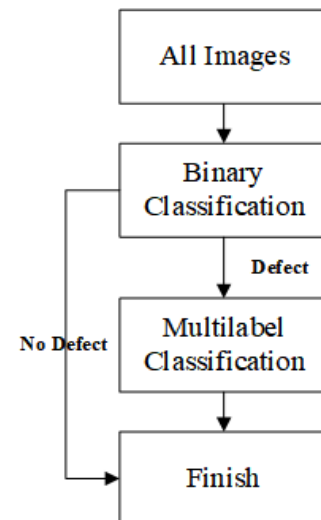


Fig 1:- Techniques of detection

A. Dataset

Dataset takes from [1], with a total of 12,568 pictures of training data. Dataset consists of a collection of grayscale steel surface images measuring 1600 x 256 pixels, which have 1 class defects, multi-class defects, and no defects. Steel surface image takes from a high-frequency and high-resolution camera, which consists of four classes, namely 1, 2, 3, and 4 [1]. From the results of Exploratory Data Analysis (EDA), 6666 images have defects and 5902 images without defect. In the images that have defects, there are 897 images defect class 1, 247 images defect class 2, 5150 images defect class 3, and 801 images defect class 4. In addition to the number of each class, the number of images with one label is 6293, with two label is 425 and 2 images with three labels. From our analysis through information obtained from Northeastern University (NEU) database web, each defect label has its name. Type of defects 1 (Pitted surface), defects 2 (Inclusion), defects 3 (Scratches), and defects 4 (Patches).

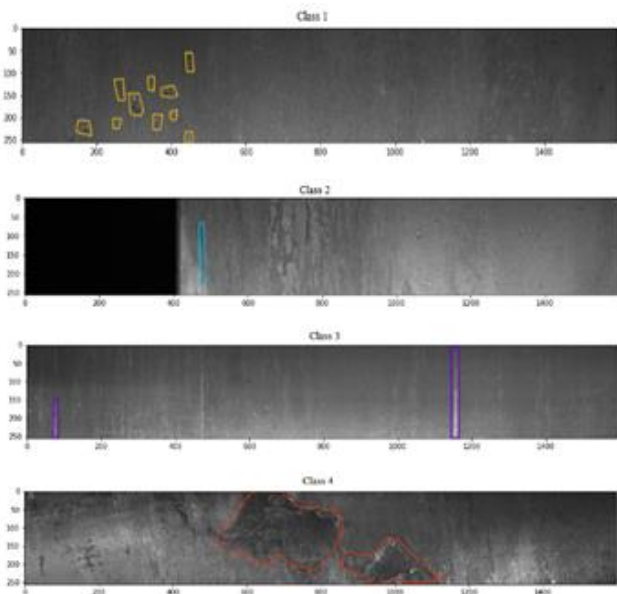


Fig 2:- Image with a defect of a single class

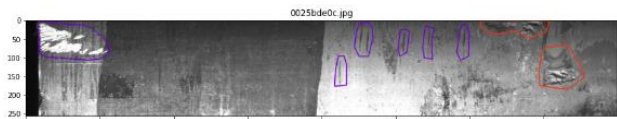


Fig 3:- Image with multiple defect classes

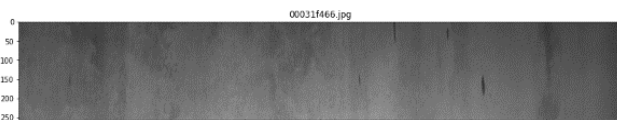


Fig 4:- Image without defect

B. Preprocessing

Deep learning in the process of image recognition requires image preprocessing. Preprocessing is influential in extracting features in the image. The method that will use in this preprocessing is data augmentation. Data augmentation is a technique of modifying data without removing the core of the data. Due to a large amount of data, this data augmentation helps speed up the preprocessing stage. The augmentation methods used in this study are (a) Rescale with 1./255, (b) Shear range with 0.1, (c) Zoom range with 0.1, (d) Brightness range with [0.6, 1.0], (e) fill mode "constant", (f) Cval with 0, (g) horizontal flip and (h) vertical flip. In the use of architecture, the image size will be resized to 256 x 256 pixels. Resize the image functions to speed up the training process.

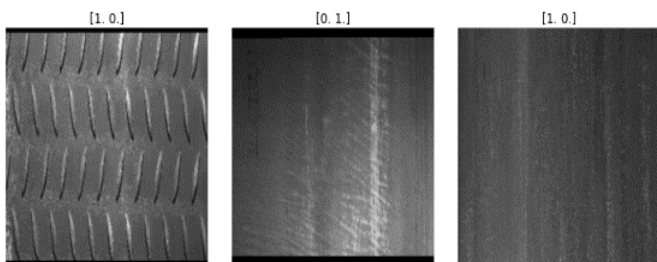


Fig 5:- Result of image augmentation in Binary Classification

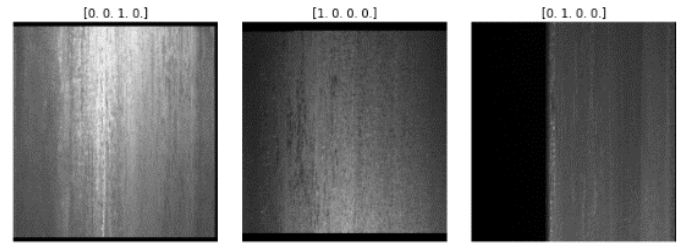


Fig 6:- Result of image augmentation in Multilabel Classification

In addition to preprocessing images, preprocessing on data will also be done with a percentage split. The percentage split ratio for training is 80%: 20%. Preprocessing images and preprocessing data for training are applied equally to binary classification and multilabel classification techniques.

C. Xception

In this study, the CNN architecture used is Xception [11] created by Francois Chollet. Xception is an extreme version of the Inception model whose architecture based on depthwise separable convolution layers. Inception with this extreme version is a strong hypothesis that illustrates 1 x 1 convolution, which can map the convolution correlations in n x n whose numbers are likely to be numerous. In deep learning, depthwise separable convolution is called separable convolution with depthwise convolution, which is a spatial convolution that can stand alone in each input channel and is followed by pointwise convolution, which projects the output channel.

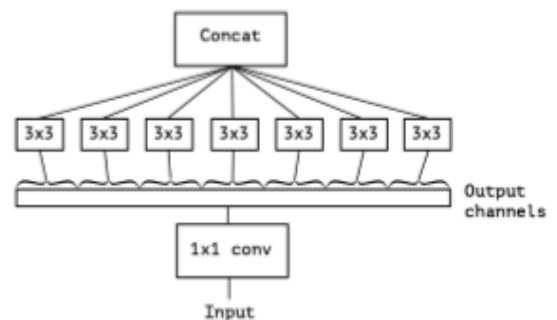


Fig 7:- Xception Module [11]

Xception consists of 36 layers that form a network for feature extraction. The Xception architecture feature map consists of 3, namely entry, middle, and exit. Each groove represents several blocks of layers supporting feature extraction. After the feature extraction block ends with the global average pooling layer, there is a fully connected layer whose usage is optional and ends with a logistic regression layer. In this case, the logistic regression layer not used, but what used is a fully connected layer consisting of global average pooling 2D layers and Dense, which shows the number of classification classes.

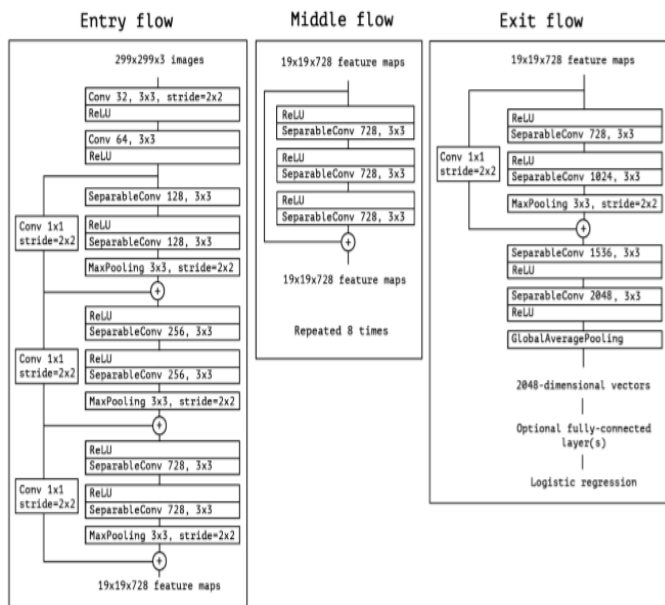


Fig 8:- Xception Architecture [11]

D. Performance Evaluation

For presenting the analysis in this study, we will use several metrics, graphs, and a confusion matrix. Then, the results of the analysis used to determine the ability of the model in defect detection.

V. EXPERIMENTAL RESULT

As explained in a method, there are two detection techniques in this case. Binary classification trained with 10054 training data and 2514 validation data, while multilabel classification trained with 5676 training data images and 1419 validation data identified four classes. Both techniques are training with the same parameters and optimizers. The parameters used in this training include a batch size of 16, the optimizer set is Adam, a loss function that is binary cross-entropy, learning rate with sizes 1e-4, and 15 epochs. Due to infrastructure limitations, we conducted this training using Google Colab with the Nvidia Tesla K80 GPU. The use of a GPU can help the execution speed compared to the CPU.

The execution time of the training present in Table 2, which shows binary classification requires a longer training time than multilabel. The average estimated time needed for each process is 0.939s / step in binary and 0.617s / step in multilabel. From this matter proves that the execution process does not use a long time even though the number of trained images is large.

Techniques	Inference Time
Binary Classification	162 minutes
Multilabel Classification	52 minutes

Table 2:- Training execution time

For more details, following visualization of the performance of both techniques that show training accuracy and training loss in each graph.

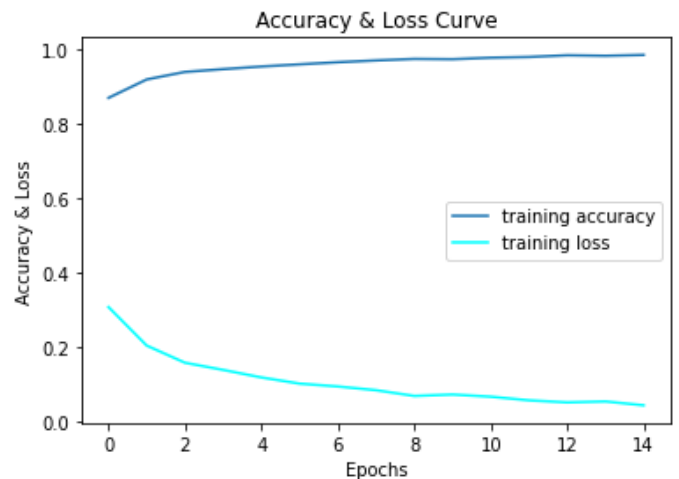


Fig 9:- Accuracy and Loss graph in Binary Classification

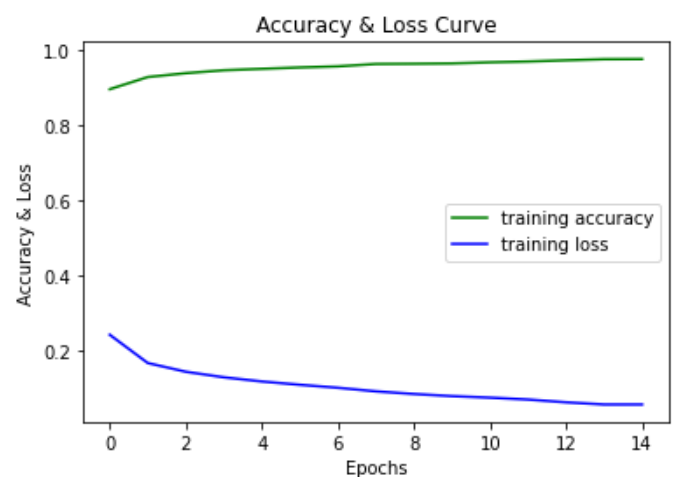


Fig 10:- Accuracy and Loss graph in Multilabel Classification

As shown in Figures 9 and 10, the accuracy produced in binary classification and multilabel classification is increasing. In the beginning epoch, an accuracy estimated to yield 0.80% to above 0.90% besides the loss value decreasing in each time. This matter shows that the interpretation is good at the model used. After performing training in both processes, the Xception model tested for the stability of its performance. The performance stability measured with the percentage split technique from 80%: 20% to 60%: 40%, which will produce accuracy and loss using the multilabel classification technique.

	80:20	75:25	70:30	65:35	60:40
Training accuracy	0,9762	0,9773	0,9775	0,9799	0,9823
Training loss	0,0562	0,0551	0,0490	0,0483	0,0445

Table 3:- Results of accuracy and loss in each percentage split

Based on Table 3, the accuracy and loss values of the training results don't have wide ranges. The higher percentage split in validation shows the accuracy increases, and loss decreases. It's a proven stable Xception model used in the training process.

VI. RESULT

This section will explain the results of the overall detection of steel defects. The Xception architecture that used to detect steel evaluated using several metrics. The metrics are accuracy, recall, precision, and f1 with the following formula:

$$\text{Accuracy} = \frac{TP+TN}{TP+FP+TN+FN} \tag{1}$$

$$\text{Recall} = \frac{TP}{TP+FN} \tag{2}$$

$$\text{Precision} = \frac{TP}{TP+FP} \tag{3}$$

$$F1 = \frac{2 \times (\text{Recall} \times \text{Precision})}{\text{Recall} + \text{Precision}} \tag{4}$$

With:

TP = True Positive

TN = True Negative

FP = False Positive

FN = False Negative

Some of these metrics will calculate the success of the predictions made. The data used to measure predictions are validation data with each data amounting to 2514 images and 1419 images. Accuracy results obtained from the binary classification process are 0.94% and 0.85% in multilabel classification. A comparison of other metrics describes in Table 4 and Table 5.

Binary	Precision	Recall	F1
No Defect (0)	0,90%	0,97%	0,93%
Defect (1)	0,97%	0,91%	0,94%

Table 4:- Results of Precision, Recall, F1 in Binary Classification

Multilabel	Precision	Recall	F1
Pitted (1)	0,75%	0,79%	0,77%
Inclusion (2)	0,50%	0,84%	0,63%
Scratches (3)	0,93%	0,88%	0,90%
Patches (4)	0,67%	0,69%	0,68%

Table 5:- Results of Precision, Recall, F1 in Multilabel Classification

In addition to several metrics, a confusion matrix also be presented in this section. The confusion matrix summarizes the results of the complete classification based on true and false objects.

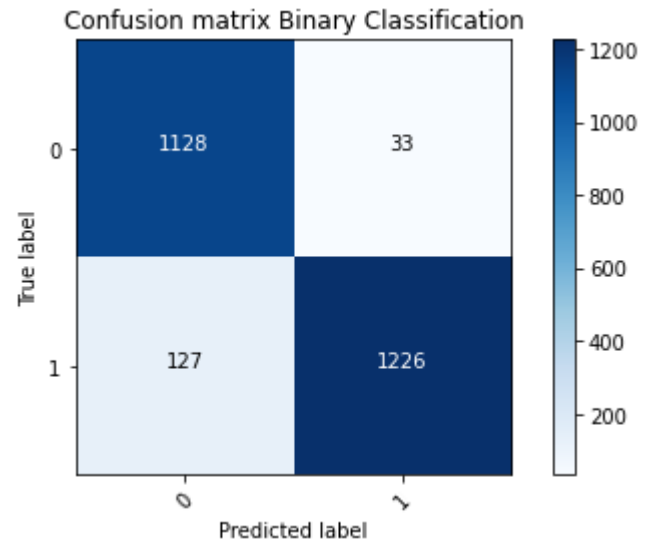


Fig 11:- Confusion Matrix in Binary Classification

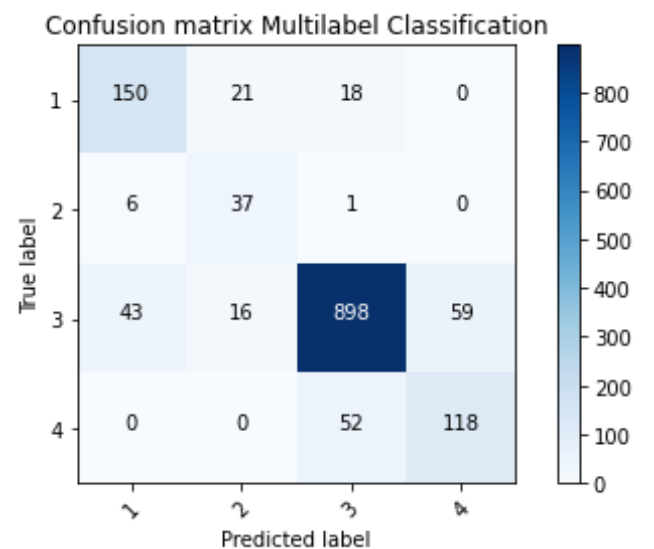


Fig 12:- Confusion Matrix in Multilabel Classification

In binary classification, it can see that class 1 (defect) has more classification errors than class 0 (no defect) while the correct classification results are 2354 from 2514 data. Multilabel predicts the correct classification of 1203 from 1419 data, and classification errors occur mostly in class 3 (Scratches).

In the multilabel case, results can present with a multilabel confusion matrix. This Confusion matrix identifies whether the image is multilabel in each class or not. Based on Figure. 13, the classification of non-multi labelled images is marked with label 0, while the correct classification of multilabel images is label 1.

Class 1		
1	1181	49
0	39	150
	1	0

Class 2		
1	1338	37
0	7	37
	1	0

Class 3		
1	332	71
0	118	898
	1	0

Class 4		
1	1190	59
0	52	118
	1	0

Fig 13:- Multilabel Confusion Matrix

VII. DISCUSSION

Besides, to perform a detailed analysis of Xception, we compared the Xception architecture with Inception V3 and VGG16 to detect steel defects using both techniques, as previously done in the Xception. From the training process, accuracy in both architectures obtained. InceptionV3 obtaining accuracy 0.91% in binary and 0.83% in multilabel. Whereas VGG16 accuracy in binary 0.91% in binary and 0.78% in multilabel. The following Table 6 and Table 7 will present the overall classification results of the three architecture.

VIII. CONCLUSION

In this study, the detection of steel defects using two techniques resulted in good classification. The Xception model that used shows excellent performance in both methods. Accuracy results obtained to classify the image is defective or does not reach 0.94% while in the classification of images that have 1 class or multi-class defects achieve an accuracy of 0.85%. Our test in comparing the InceptionV3 and VGG16 with Xception architecture proves that performance Xception more stands out. For further work, we will consider adding several data augmentation methods to improve accuracy and reduce the percentage of errors.

REFERENCES

- [1]. Severstal, "Severstal: Steel Defect Detection," 2019. [Online]. Available: <https://www.kaggle.com/c/severstal-steel-defect-detection>. [Accessed: 27-Apr-2020].
- [2]. I. den Bakker, Python Deep Learning Cookbook. Birmingham: Packt Publishing, 2017.
- [3]. B. Suvdaa, J. Ahn, and J. Ko, "Steel surface defects detection and classification using SIFT and voting strategy," Int. J. Softw. Eng. its Appl., vol. 6, no. 2, pp. 161–166, 2012.
- [4]. J. Li, Z. Su, J. Geng, and Y. Yin, "Real-time Detection of Steel Strip Surface Defects Based on Improved YOLO Detection Network," IFAC-PapersOnLine, vol. 51, no. 21, pp. 76–81, 2018, doi: 10.1016/j.ifacol.2018.09.412.
- [5]. E. A. Kholief, S. H. Darwish, and M. N. Fors, "Detection of steel surface defect based on machine learning using deep auto-encoder network," Proc. Int. Conf. Ind. Eng. Oper. Manag., no. April, pp. 218–229, 2017.
- [6]. S. Tian and K. Xu, "An algorithm for surface defect identification of steel plates based on genetic algorithm and extreme learning machine," Metals, vol. 7, no. 8, 2017, doi: 10.3390/met7080311.
- [7]. M. S. Kim, T. Park, and P. Park, "Classification of Steel Surface Defect Using Convolutional Neural Network with Few Images," 2019 12th Asian Control Conf. ASCC 2019, pp. 1398–1401, 2019.
- [8]. S. Y. Lee, B. A. Tama, S. J. Moon, and S. Lee, "Steel surface defect diagnostics using deep convolutional neural network and class activation map," Appl. Sci., vol. 9, no. 24, 2019, doi: 10.3390/app9245449.
- [9]. J. L. Yunani and M. V. Medvedev, "Steel Defects Recognition Using Deep Learning," 2019 Int. Multi-Conference Ind. Eng. Mod. Technol. FarEastCon 2019, pp. 1–4, 2019, doi: 10.1109/FarEastCon.2019.8933868.
- [10]. K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 3rd Int. Conf. Learn. Represent. ICLR 2015 - Conf. Track Proc., pp. 1–14, 2015.

Model	Binary Classification		True Sample
	No Defect (0)	Defect (1)	
Xception	1128	1226	2354
InceptionV3	1126	1151	2277
VGG16	1009	1284	2293
Total sample	1161	1353	2514

Table 6:- Results of all classification in Binary Classification

Model	Multilabel Classification				True Sample
	Pitted (1)	Inclusion (2)	Scratches (3)	Patches (4)	
Xception	150	37	898	118	1203
InceptionV3	78	32	941	131	1182
VGG16	121	34	929	29	1113
Total sample	189	44	1016	170	1419

Table 7:- Results of all classification in Multilabel Classification

- [11]. C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the Inception Architecture for Computer Vision," Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., vol. 2016-Decem, pp. 2818–2826, 2016, doi: 10.1109/CVPR.2016.308.
- [12]. F. Chollet, "Xception: Deep learning with depthwise separable convolutions," Proc. - 30th IEEE Conf. Comput. Vis. Pattern Recognition, CVPR 2017, vol. 2017-Janua, pp. 1800–1807, 2017, doi: 10.1109/CVPR.2017.195.
- [13]. Keras, "Applications." [Online]. Available: <https://keras.io/applications/>. [Accessed: 04-Feb-2020].
- [14]. E. Ayan and H. M. Ünver, "Diagnosis of pneumonia from chest X-ray images using deep learning," 2019 Sci. Meet. Electr. Biomed. Eng. Comput. Sci. EBBT 2019, pp. 0–4, 2019, doi: 10.1109/EBBT.2019.8741582.
- [15]. C. A. Mamani Diaz, E. E. Medina Castaneda, and C. A. Mugruza Vassallo, "Deep Learning for Plant Classification in Precision Agriculture," 2019 Int. Conf. Comput. Control. Informatics its Appl. Emerg. Trends Big Data Artif. Intell. IC3INA 2019, pp. 9–13, 2019, doi: 10.1109/IC3INA48034.2019.8949612.