Contents

[Binary Classification: 1](#_Toc41186995)

[Result: 3](#_Toc41186996)

[Conclusion: 4](#_Toc41186997)

**List of Tables:**

[Table 1 Confusion Matrix 4](#_Toc41187002)

[Table 2 Precision-Recall Matrix 4](#_Toc41187003)

## **List of Figures:**

[Figure 1 (a) Non-Defect Steel (b) Defect Steel 1](#_Toc41187077)

[Figure 2 Resizing the image as a part of preprocessing to the model 2](#_Toc41187078)

[Figure 3 CNN model architecture used to train the model 2](#_Toc41187079)

[Figure 4 The CNN model architecture summary 3](#_Toc41187080)

[Figure 5 (a) Loss Curve (b) Accuracy Curve 3](#_Toc41187081)

## Binary Classification:

The Severestal Steel has defect and non-defect steels. The first part of this work is to identify the defect steel from the non-defect steels. Identification of defect steel is the first and most important task to ensure the quality and reputation of the company. Later, from the defected steel, the second part comes to identify the location of the defect and then do the required modification or replacement.

The defect and non-defect steels are shown in Figure 1. The sign of irregularity, corroded surface can be easily seen on the steel image with defect label.



Figure (a) Non-Defect Steel (b) Defect Steel

To achieve this task of Binary Classification, first, preprocessing has been done. The original image size is 1600x256. It’s resized to 256x256 to input the uniform image size to the model. The Figure 2 shows the size transformation of image.

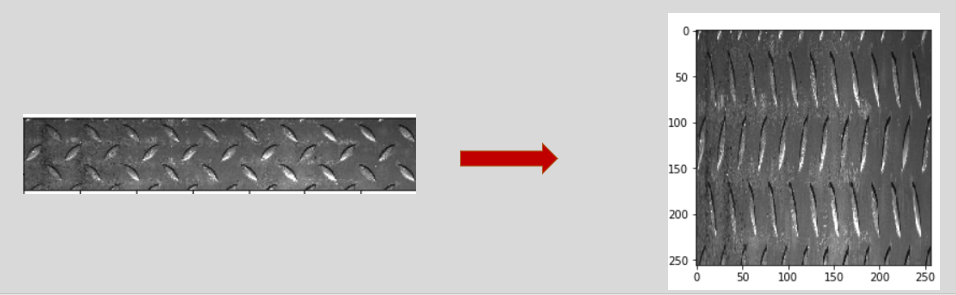


Figure Resizing the image as a part of preprocessing to the model

To do the Binary Classification, Convolutional Neural Network (CNN) model has been used, as it outperforms all other models in context of images. The architecture of CNN model has been presented in Figure 3. The CNN model starts with 3x3 convolution, with 36 filters in the first layer followed by 2x2 Max Pooling layer, reducing the image size by half. The image with size of 256 is reduced to size of 128 by Max Pooling layer. In the second layer, again there is convolution layer with 3x3 kernel, with 64 filters and so on. At the end of four layers of convolution and max pooling, the final layer is flattened, and feed to the dense layer. The model summary in details has been presented in Figure 4, showing the number of parameters for each layer.

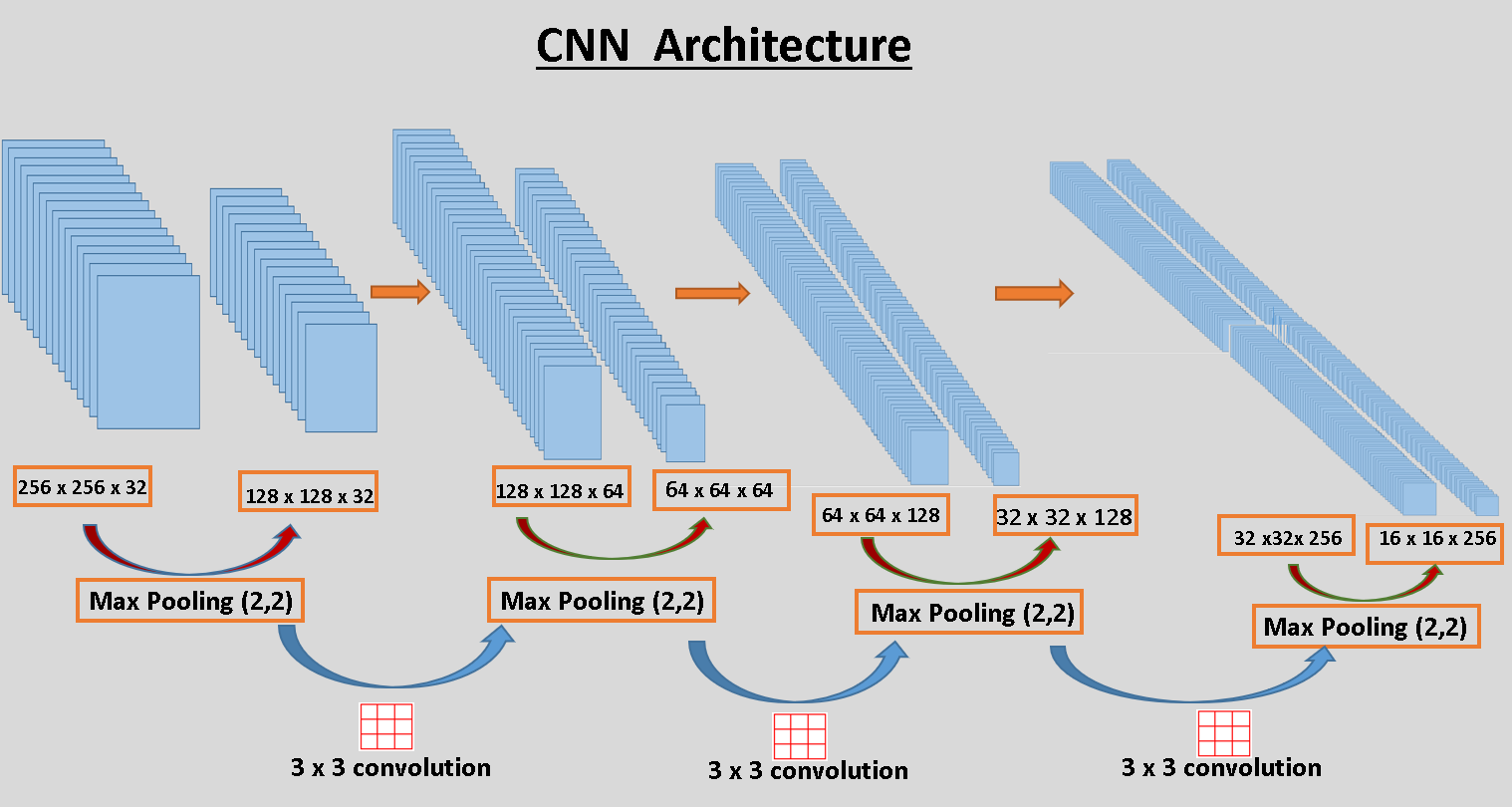


Figure CNN model architecture used to train the model

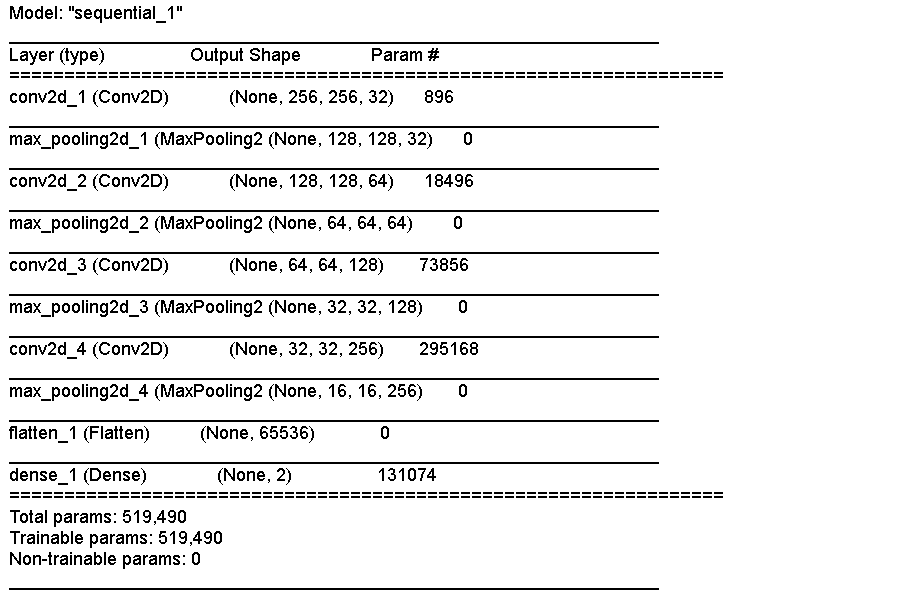


Figure The CNN model architecture summary

## Result:

Although the model is small with just four convolution layers and one dense layers, it gives quite good accuracy. In the figure 5 (a) , the loss keeps on decreasing with the increase in number of epochs. The validation accuracy keeps on increasing upto ten epochs, then starts decreasing. The training accuracy keeps on increasing, which was expected with the increase in the number of epochs. The overfitting can be seen here after ten epochs, however a very good accuracy has been achieved till then. The validation accuracy of 85% has been achieved which is quite good. The model can be said to generalizing well.

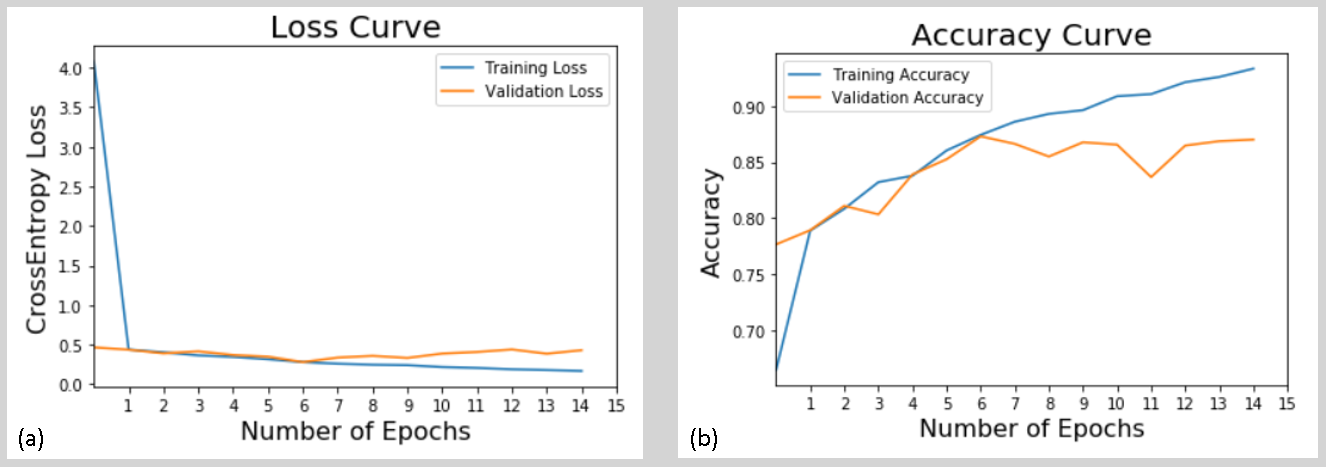


Figure (a) Loss Curve (b) Accuracy Curve

**Confusion Matrix**

Table 1 shows the confusion matrix obtained by the model on test-data. It’s working quite well, as it is able to identify most of the examples correctly. However, there are some wrong prediction, but they are quite low in number. For these types of problems of quality insurance, the problem comes when a defected steel is classified as non-defect steel and is sent to the market. This number should be kept as low as possible. In our model, it’s identifying 142 defect cases as non-defect cases. It’s around six percent of total cases used to create the confusion matrix.

Table Confusion Matrix

|  |  |  |
| --- | --- | --- |
|  | **True No-Defect** | **True Defect** |
| **Predicted No-Defect** | 1039 | 142 |
| **Predicted Defect** | 181 | 1152 |

**Precision-Recall Matrix**

Table 2 shows the precision recall matrix. The prediction precision is quite high around 85% for non-defect examples, and 89% for defect examples. But recall is of more interest, as we wants to see how accurately, it’s able to recall the steel type as defect or non-defect. The recall percentage for non-defect and defect cases are 88% and 86% respectively.

Table Precision-Recall Matrix

|  |  |  |
| --- | --- | --- |
|  | **Precision** | **Recall** |
| **No-Defect** | 0.85 | 0.88 |
| **Defect** | 0.89 | 0.86 |

## Conclusion:

The CNN model is generalizing quite well. It’s able to identify the defect steel with a very good accuracy. However, the performance could be increased with regularization and more tuning the model. For the time being, the model performance is quite well.