Defect identification on metallic surfaces with machine learning techniques

Assignment on Multimodal Machine Learning

Msc AI

University Of Piraeus NCSR DEMOKRITOS

Sidiropoulos Michail
MTN2320

Papadopoulos Dimitrios Iason
MTN2315

***Abstract*— In this study machine learning techniques are implemented in order to optimize the defect identification problem in metallic surfaces. Initially, the physical problem and its importance are discussed and why traditional defect identification techniques have to be defied. Then the techniques chosen are explained and their advantages in the specific problem. Last the writers give their opinions on the results of each technique and their conclusions overall.**

Keywords—YOLO, VGG16, Defects, Object detection, Metal surfaces, CNN, classification

I. Introduction

Undoubtedly, a machine's failure can lead to catastrophic results. Either in small scale such as a small mechanism or in large scale like a whole manufacturing plant it is crucial that the condition of the important components is monitored. When the condition of the important aspects is known by the quality and production engineers, their remaining useful life can be calculated in order to avoid unwanted breakdowns. This can be achieved by implementing nondestructive testing techniques, in which the part undergoes certain processes in order to monitor its health without destroying it or altering its functionality. In the case of metallic surfaces, the testing process can last a long time because the surface has to be captured with a high-speed camera and then be meticulously examined by the engineer in charge. It is obvious that, in cases where the metallic surfaces that have to be examined are thousands, the whole process might take weeks. Artificial Intelligence and machine learning can help optimize the process of identifying defects in metallic surfaces. In more detail by implementing machine learning or deep learning algorithms, the possible failures in pictures of metallic surfaces can be automatically identified. This study implements two object detection and classification algorithms in order to find the location of the defect and also classify it according to a certain range of defects.

II. Data

The dataset that was chosen is C10-DET because opposed to other datasets it resembles realistic situations. The data collection process consists of a set of CCD cameras and high-speed CCD cameras in order to achieve high detection speed and resolution. Then the images are transferred to a server for defect detection. [1] The dataset consists of 2300 images with 3563 labeled objects, each object is a defect that belongs to one of the ten different

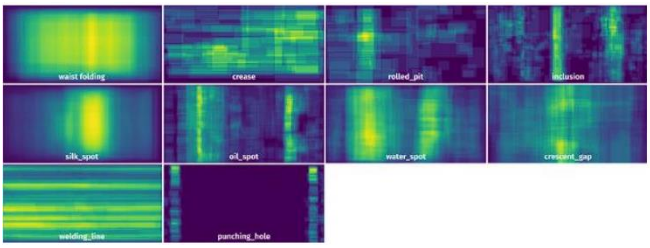

Fig.1. Heatmap of the spatial distribution of each defect

defect classes. The dataset is heavily imbalanced, and some images contain more than one defect. Also, each defect has a different shape and size and might occur in different locations of the surface. In Fig.1 a heatmap of the spatial distribution of each defect is shown. It visualizes the probability of each defect's location and also gives an insight into their shape and size.[1]

Class	Images	Objects	Count on image (average)	Area on image (average)
silk_spot (rectangle)	734	884	1.2	16.7%
welding_line (rectangle)	512	513	1	5.91%
punching_hole (rectangle)	329	329	1	0.31%
water_spot (rectangle)	310	354	1.14	7.39%
crescent_gap (rectangle)	264	265	1	6.37%
oil_spot (rectangle)	250	569	2.28	3.41%
inclusion (rectangle)	201	347	1.73	1.57%
waist folding (rectangle)	140	143	1.02	40.47%
crease (rectangle)	53	74	1.4	8.45%
rolled_pit (rectangle)	46	85	1.85	5.55%

Fig.2. Defect reference count

Fig.2 displays the balances of every class. The image count refers to the number of images that contain a defect whereas the object count refers to the number of each defect occurrence, because each defect might occur more than once in the same picture. The imbalance of the dataset is obvious, for example there are total of 53 images of creases while the number of images with silk spots is 734. It is also physically expected, because each defect occurs under different circumstances and some defects are more prone to happen than others. Thus, a balanced dataset would not correspond to a realistic situation and by trying to balance the specific dataset creates the risk of the model overlearning certain features. It is expected that the models will not perform adequately with the classes that are underrepresented and also with silk spots. Because of the heavy imbalance it is possible that it will overlearn that feature and confuse other

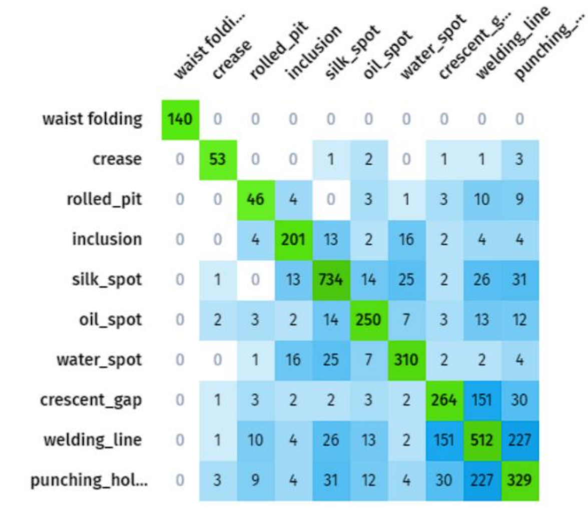

Fig.3 Co-occurrence matrix of every defect

similar features as silk spots. [1] During the manufacturing process it is expected that certain defects take place together or that a defect might be the cause of another, as displayed in the co-occurrence matrix in Fig.3. For example, silk spots, oil spots and water spots can be accompanied by lots of other defects while waist foldings are always alone. Punching holes and crescent gaps usually occur with welding lines. [1]

The defect types that are studied are:

Punching, which occurs because the metallic surfaces are punched according to the product specifications and in some cases unwanted punches can cause punching defects.[1] Welding line, which technically is not a defect, occurs when metallic strips are changed in the production line and the coils of the new strip are welded together. [1] Crescent gap is a half-circle defect that occurs in metallic strips when they are not cut correctly. [1] Water spots are spots that occur in the drying process. [1] Oils spots occur when the lubricant used is contaminated. [1] Silk spots are a wave like pattern that occur in strips when uneven temperature and pressure is applied. [1] Inclusions are small spots that may fall off or be pressed into the surface. [1] Rolled pits are pits or bulges on the surface and they are caused by work roll or tension roll damage. [1] Crease is a vertical fold that forms across or at the edges of a strip. [1] Waist folding is a defect caused by folds that indicate large deformation because of low carbon. [1]

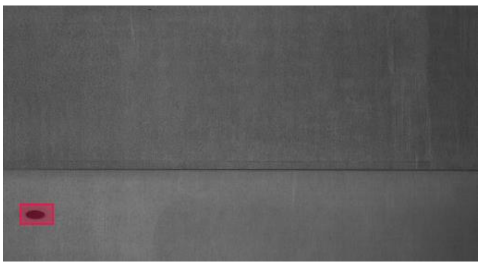

Fig.4 Punching

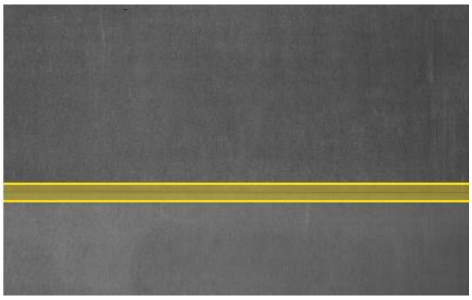

Fig.5 Welding Line

Fig.6 Crescent Gap

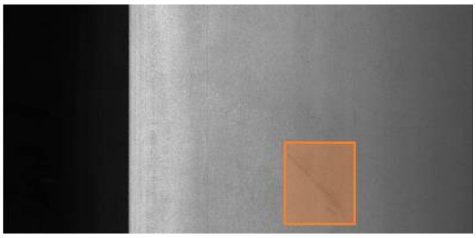

Fig.7 water Spots

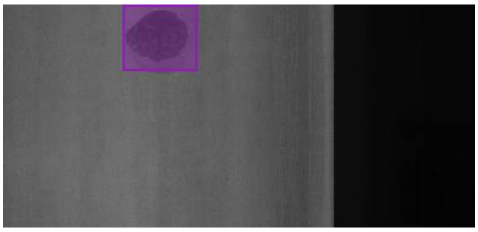

Fig.8 Oil Spots

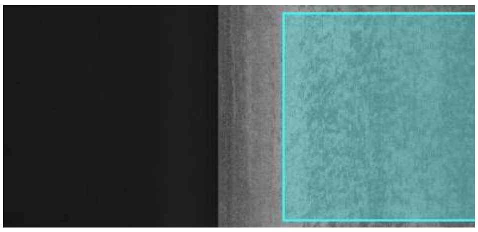

Fig.8 Silk Spots

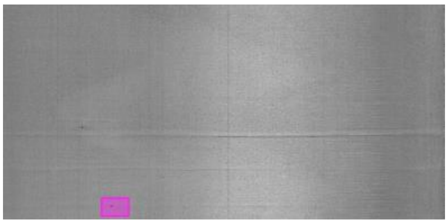

Fig.9 Inclusions

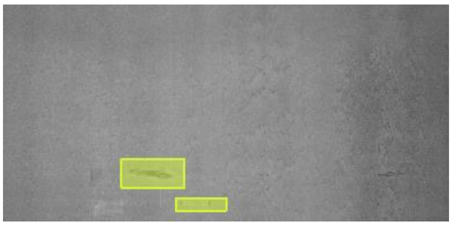

Fig.10 Rolled Pits

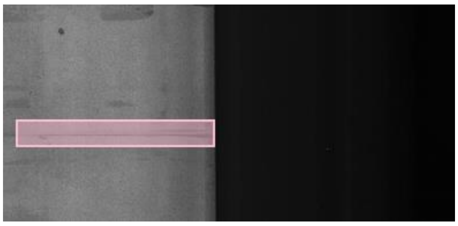

Fig.11 Crease

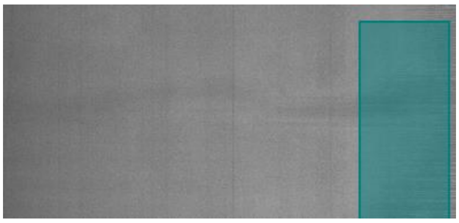

Fig.12 Waist Foldings

III. Techniques

The goal of this study is to identify the location of each defect and also classify each sample according to their defects. The chosen techniques for defect identification are YOLO and VGG16 Yolo is a widely used computer vision model for object detection that uses a Convolutional Neural Network (CNN). The concept of this model is that it predicts both the bounding boxes of the objects and their classes with a single CNN. The model is trained with a large, annotated dataset in order to learn patterns and properties related to different kinds of objects. [2] As it was shown in the previous chapter, these kinds of defects can have a variety of different shapes and sizes, thus yolo was chosen, because it can make accurate predictions about varying sizes of objects. Also, the bounding boxes that are generated are precise, which is important because in certain cases more than one defect can take place on the same surface. [2] YOLO consists of 24 convolutional layers, 4 max pooling layers, and 2 fully connected layers. In all layers, the activation functions used are ReLU and the final linear uses a linear activation function. The regularization techniques that are used are batch normalization and dropout. [3] The model works with four different techniques, residual blocks, bounding boxes, Intersection over unions and non-maximum suppression.[3] First the image is split into a grid with cells of equal size, the residual blocks. The objects inside each block are localized and their classes are predicted. Then the bounding boxes that correspond to each object in the image are created. A regressor is used to calculate the attributes of the bounding boxes and the coordinates with respect to the object and the grid as well as the probability of the box containing objects and their corresponding classes. Continuing, in order to distinguish the boxes that contain relevant information and discard the others intersection over unions is used. Intersection over unions is a metric that calculates the quantity of the area of an object inside a box and if that quantity is less than a certain threshold then that box is discarded. Last Non max suppression is the technique that helps reduce noise, because it keeps only the boxes with the highest probability of containing an object.

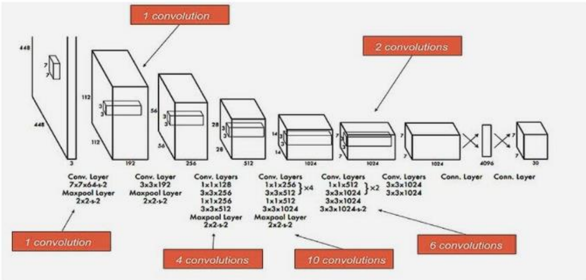

Fig. 13 Yolo v1 reimplementation architecture [4]

VGG16 is a popular computer vision model that implements CNNs and is used for image classification and object detection. The basic architecture of VGG-16 consists of 16 layers, 13 convolutional layers and 3 fully connected layers and its popularity derives from its simplicity and effectiveness. The convolutional layers use the ReLU activation function, and they are followed by max-pooling layers with progressively increasing depth, thus enabling the model to learn complex visual features. The fully connected layers use the ReLU activation function, and the last layer uses the softmax activation function because the model is implemented for classification. For the needs of this study, we froze the dense layers and divided the output into two. The regression output predicts the coordinates of the bounding boxes, and the classification output predicts the class of each object. Even though its architecture is rather simple, it is a robust and versatile technique that produces accurate results. [5]

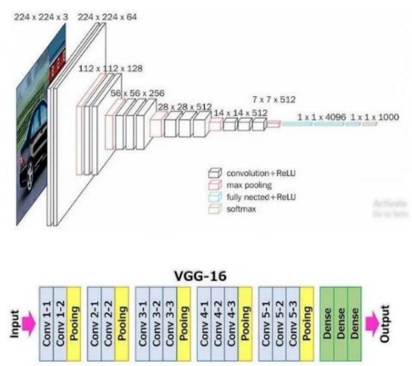

Fig. 14 VGG16 architecture [6]

IV. Training results

A. YOLO

Fig. 15 displays the losses in training and validation using YOLO. The plots on the left represent the losses for the coordinates of the rectangles that are used for the object detection process. The model was trained with 100 epochs and it is able to generate bounding boxes with relatively high accuracy. The loss metric for the classification task is binary cross entropy and it is suitable for this task because YOLO treats each class independently. The performance in validation is similar to that in training and it is satisfying for the specific task. The plots on the right represent the losses for the object classification task. The two plots indicate that the model performs well in the classification task because in both cases their results are similar, and the losses are very small.

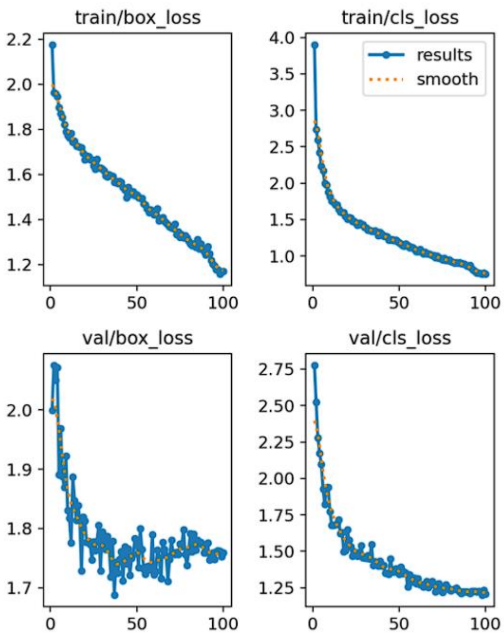

Fig. 15 Losses in training and validation using YOLO

The confusion matrix for the validation set in fig.16 shows an adequate performance. First, as it was mentioned earlier the dataset is heavily imbalanced and the sample of silk spots are a lot more than the samples of creases. The cases in which the performance was worse originate from either the datasets imbalance or the properties of the defect. For example, certain classes were underrepresented, thus the model could not train properly in order to accurately predict them. Also, some defects because of their small size and in addition, because of their shape, might resemble the metallic surface, so they are hard for the model to distinguish.

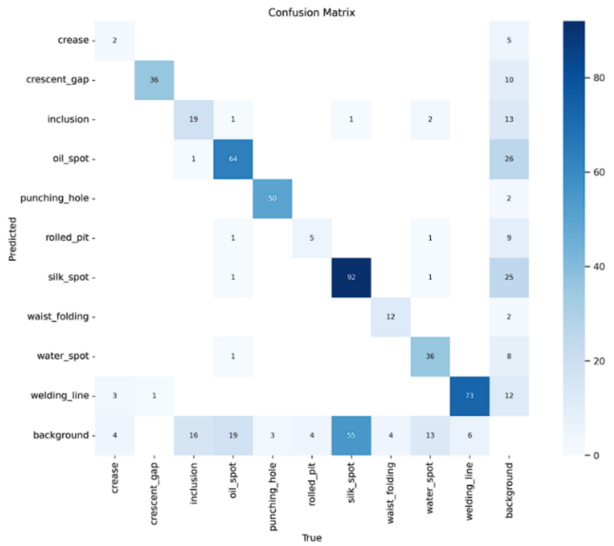

Fig. 16 Confusion matrix for the validation set using YOLO

B. VGG16

The plot in fig.21 shows the mean absolute error of the coordinates of the bounding boxes. It is obvious that the model was trained well because the errors are very small, indicating that the model will be able to accurately find the bounds of each object.

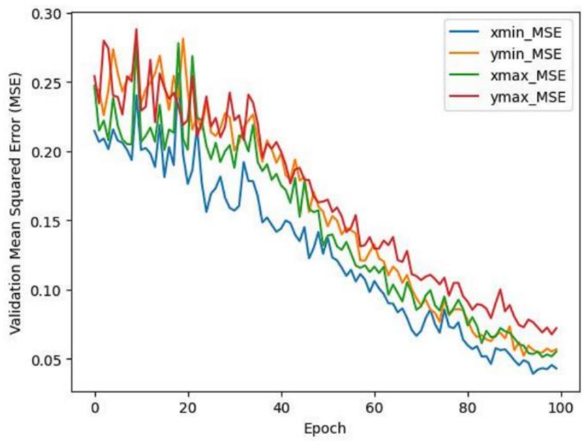

Fig.21 Mean absolute error of the coordinates of the bounding boxes in the training set using VGG16

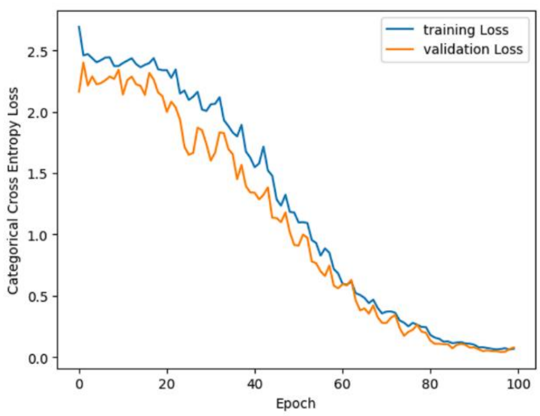

Fig.22 Categorical cross entropy loss of the predicted classes in the training set using VGG16

The plot in fig.22 shows the mean squared error of the average of the probabilities for each class's correct prediction. The plot shows that the model both during training and in validation performs very well because the mean squared error is very small and almost zero.

Fig.23 shows the accuracy of the model for the classification task. Even though the results seem very good, this metric does not accurately depict the model's performance. Since the dataset is heavily imbalanced, accuracy does not give an accurate description of the model, thus this metric might be misleading. It is expected that the predicted classes are not that accurate which will be shown with the confusion matrix.

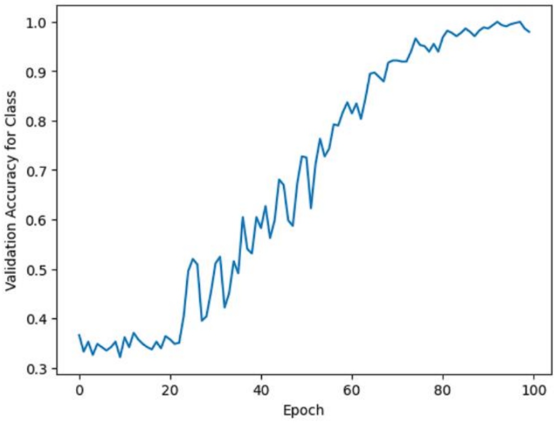

Fig.23 Accuracy score of the predicted classes in the validation set using VGG16

V. Testing results

A.YOLO

From the confusion matrix in fig.17 it is obvious that the model gives relatively satisfying results. For certain features the model performed inadequately but it was expected. For example, creases, because of their shape, can be confused with the topology of the surface and their samples were limited so the model could not train very good. Inclusions in some samples are very small thus they cannot be distinguished by the model, or they might resemble a low-resolution picture. The pictures of rolled pits were very few so the model was not trained properly and also the test samples were limited, so the model cannot predict accurately that defect. Even though the model was trained with enough samples of silk spots, it could not distinguish them accurately because their appearance is very subtle, and they are similar to the background. Most importantly though, they resemble water spots and oil spots, and the model could not distinguish them. The bad performance with creases and rolled pits could be solved by using more samples so that the model can train better. In order to enhance the model's performance with silk spots a solution would be to use higher resolution cameras that could capture with greater detail the difference between the defect and the rest of the surface and also capture better their differences compared to similar defects.

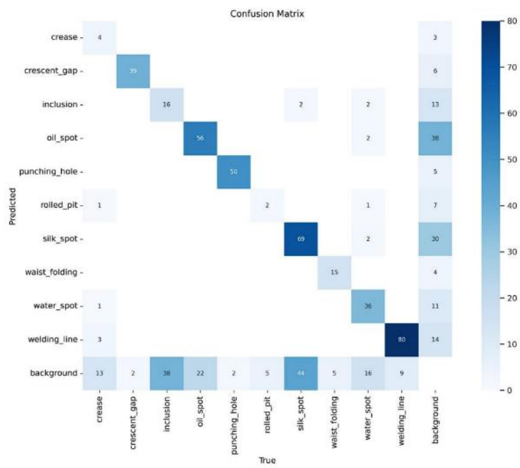

Fig.17 Confusion matrix for the test set in YOLO

In fig.18 and fig.19 a random instance of the object detection and defect classification tasks are shown. Fig.18 shows the true annotations and fig.19 shows the predicted ones. In the bottom right corner in fig.18 two defects can be found but the model predicted only one. In the bottom left corner in fig.18, the model does not predict the welding line because it is very similar to the rest of the

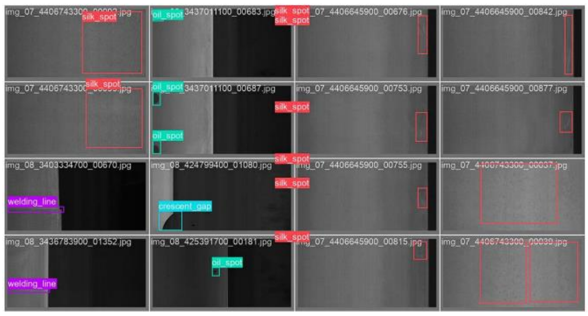

Fig.18 Example of object detection and defect classification using YOLO – correct annotation

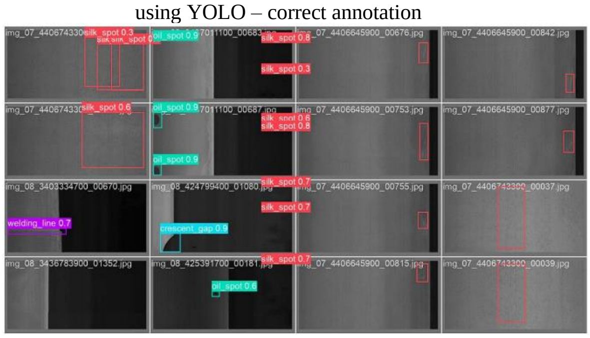

Fig.19 Example of object detection and defect classification using YOLO – predicted annotation

metallic surface. Reviewing the precision recall curve is important in this case because it gives an accurate estimate about the model's results. Precision refers to the ratio of the true positives divided by the predicted positives and recall is the ratio of the true positives divided by the actual positives. Precision refers to the quality of the predictions whereas recall refers to the quantity of the predictions. By examining the curve in fig.20, it is obvious that the model did not perform adequately with certain classes, but it performed better with others. As mentioned before, the model was not trained properly with rolled pits, thus making it hard to predict that class and the performance for that class was the worst overall. Also because of both the limited samples and the hard to distinguish geometry, creases and inclusions are the two classes that performed better than rolled pits but not good compared to the others.

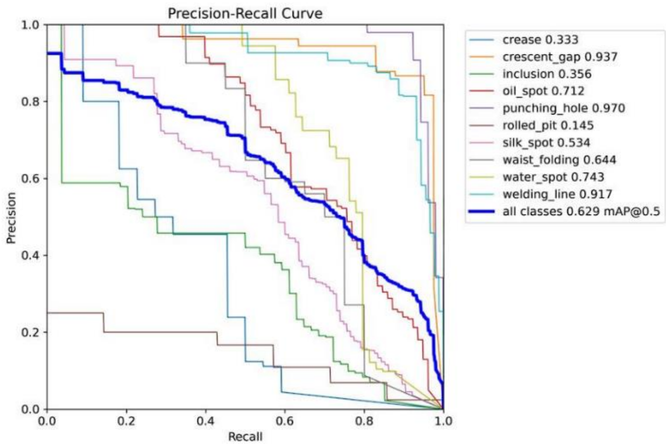

Fig.20 precision-recall curve of the test set usin YOLO

B.VGG16

The confusion matrix of the test set reveals that the model performed relatively well and also reveals for another time that the classes are imbalanced. As it was mentioned in the confusion matrix of the other model, the performance was reduced in cases of classes with fewer samples as the model did not train well. Also, the physical characteristics of the defects play a vital role in the identification, because in cases where the size is small or the defect itself is similar to the background of the surface, the model cannot distinguish them accurately.

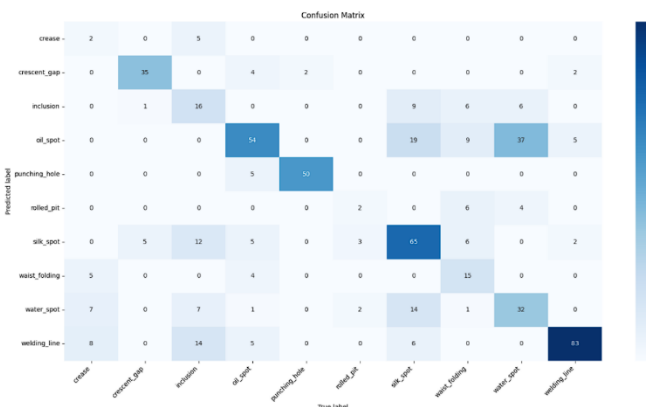

Fig.24 Confusion matrix for the test set in VGG16

VI. CONCLUSIONS

In conclusion, the two techniques YOLO and VGG16 showed in some cases promising results, and they can solve the problem of defect detection adequately. The two models were trained and tested with a heavily imbalanced dataset, in which the most presented class was ten times larger than the smallest. Not only that but the physical properties of certain defects burdened the model even more because some have peculiar size and shape that are hard to locate. The metric that can accurately depict the model's performance is the confusion matrix. Even though other metrics showed better results, confusion matrices give the correct estimate about the model. Once again, the confusion matrices show the imbalance of the dataset because the samples of some defects in the test set were few and the model did not learn their features good enough. Continuing, some defects look very similar to others like oil spots, water spots and silk spots and they were confused with each other in some instances. Moreover, since on many occasions different defects coexist, it is possible that it is difficult for the model to distinguish them individually. Last, in order to optimize the solution of the defect detection problem, better datasets have to be created. The new datasets will train better algorithms if they contain a wider range of defects with more balanced classes. They should also be better annotated so that the model does not learn wrong features or in certain cases where different defects co-occur information is not lost.

References

[1] https://datasetninja.com/gc10-det#introduction

[2] M. Raza, "Yolo v8: A Deep Dive Into Its Advanced Functions and New Features" Medium, October 2023

[3] Z. Kelta, "YOLO Object Detection Explained" datacamp, September 2021

[4] https://www.researchgate.net/figure/YOLO-Full-Model-Architecture_fig4_371136193

[5] https://www.geeksforgeeks.org/vgg-16-cnn-model/

[6] Great Learning, "Everythinh you need to know about VGG16", medium September 2021