International Journal of Computer Science and Engineering (IJCSE)
ISSN(P): 2278–9960; ISSN(E): 2278–9979
Vol. 10, Issue 1, Jan–Jun 2021; 55–64

International Academy of Science, Engineering and Technology
Connecting Researchers; Nurturing Innovations

DEFECT CLASSIFICATION IN SURFACE STRUCTURE OF STEELS USING DEEP LEARNING

Siddarth G[1], Sumitra Binu[2] & Pramila R M[3]

Research Scholar, CHRIST (Deemed to be University), Pune, India

ABSTRACT

For manufacturing industries, quality is the main concern, and to inspect the materials at a very faster rate to meet the production demands. In manufacturing, quality can be defined as the fitness of the product to use and it is an essential parameter in manufacturing industries. In order to attain that quality in any product or service, all the processes involved in that should be monitored carefully. It can be further elaborated into controlling/inspecting the processes to meet the standard form and this process of controlling is called quality control or quality inspection. The purpose of this project is to build a model that will quickly identify the presence of defects in the steel surfaces. The proposed yolov3 detector model attempts to improve the accuracy of prediction by using labelled images and different learning rates which takes into account the interaction effect of different parameters. The major outcome of this project is to build a suitable model that will identify defects faster and effectively when compared to other methods. In the future, the models can be tried with different activation functions and well-defined images on other architectures which may aid in building a model with better accuracy.

KEYWORDS: *Yolov3, Labelled Images, Activation Functions.*

Article History

Received: *24 Mar 2021* **|** ***Revised:*** *30 Mar 2021* **|** ***Accepted:*** *17 Apr 2021*

INTRODUCTION

In manufacturing, quality can be defined as the product's fitness to use and is an essential parameter in manufacturing industries. To attain that quality in any product or service, all the processes involved in that should be monitored carefully. It can be further elaborated into controlling/inspecting the operations to meet the standard form, and this process of governing is called quality control or quality inspection.

To meet industry standards, quality inspectors in manufacturing firms inspect product quality usually after the product is manufactured. It's a time-consuming manual effort and a rejected product results in wasted upstream factory capacity, consumables, labour, and cost. With the modern trend of Artificial Intelligence, industrial firms are looking to use deep learning-based computer vision technology during the production cycle itself to automate material quality inspection. The goal is to minimize human intervention, simultaneously achieving human-level accuracy or more as well as optimize factory capacity, labour cost, etc. The usage of deep learning is varied, from object detection in self-driving cars to disease detection with medical imaging. Deep learning has proved to achieve accuracy comparable with the levels achieved by human beings & even better.

Recent developments in the arena of artificial intelligence, machine learning algorithms, and sensor networks aids in addressing the challenges with respect to defect identification of parts by image recognition using convolutional neural networks (CNN). Today's increased level of automation in manufacturing also demands automation of material quality inspection with little human intervention. The trend is to reach human-level accuracy or more in quality inspection with automation. To stay competitive, modern Industrial firms strive to achieve both quantity and quality with automation without compromising one over the other.

REVIEW OF LITERATURE

Uji Iwahori et al. [1] discuss a new method called Dense SIFT to classify the defects in an electronic board. The dense SIFT does not use any reference images. Useful vital points are detected in the defect region because the dense sift captures all the relevant and irrelevant features and hence any new defect or irregularities in that specific region can also be detected. Evaluation of performance through the whole system, classification for each kind of defect, further improvement of accuracy, etc. can be considered as areas of further research.

Wuqin tang et al., [2] in their work, constructed a CNN-based neural network model to identify the defects present in the photovoltaic modules using electroluminescence images. In this, a CNN model is generated and is used to compare the accuracy and computational efficiency with the previous models like MobileNet, ResNet50, InceptionV3, VGG16. The model attained the highest accuracy. Even though on par with VGG16, the model is relatively simple in terms of computational accuracy because low numbers of parameters are required to train the model. Further accuracy improvement can be done by increasing the parameters.

Benjamin Staar et al. [3] discuss a CNN model for anomaly detection rather than defect classification. Practically in industries, the process is well optimized and the number of defects is very small. The anomaly detection is solved using deep metric learning with triplet networks. In the triplet network, 3 samples are provided as input in which the 2 samples belong to the same class and 1 is different. Class 1 & 2 and 1 & 3 are paired and their Euclidean distance is measured such that the same and various classes are trained. The future scope is to combine both texture images and object images for training and calculate accuracy. Sometimes the model fails on discriminating between the classes using Euclidean distance and hence opportunities are there for using other distance methods such as Manhattan distance.

Tian Wang and Yang Chen[4], in their work "A fast and robust convolutional neural network-based defect detection model in product quality control," focuses on building a deep learning model, particularly convolutional neural networks. The objective is to classify the defective and non-defective products using the images obtained from the computer vision system. Their two main architectures in CNN are global frame classification to classify the images according to the type of defects and the sub-frame detection part is to decide whether it is defective or not defective. The ReLu activation function is used for the non-linear transformation of features.

Yi-Fan Chen and Fu-Sheng Yang [5], in "Automatic Defect Detection System Based on Deep Convolutional Neural Networks" focuses on building an image classification model based on training the model by a small set of data. It also discusses the effect of light on the classification of defects and how lighting should be arranged when capturing an image. The image of resolution 2592*1944 pixel is captured and trained. The image is further broken down into 64*64 pixels and various combinations to train the different neural layers. Here ResNet containing 50 layers is used. The image is cut into 64 * 64 segment patches and if any of those patches correspond to the defect pattern, the image is normalized into

a greyscale map. Using row-column search the defected segment is matched to the colour segmentation and shown.

Gaps Identified

From the reviewed literature, it is observed that the techniques used for defect classification require a lot of computation power. Also, it is observed that there is a dearth of hybrid models that can improve the accuracy of classification. Defect classification models using efficient classification separation techniques like cross-entropy techniques are also not discussed in the literature. The majority of models discussed in the literature utilized a large number of training parameters. This research focuses on proposing a defect classification model that attempts to classify defects in steel with high accuracy and by using a minimum number of parameters and in the least possible time.

Objective

- The objective is to identify and develop a method/tool using deep learning models like CNN (Convolutional Neural Network) to identify the defects in material/product.
- The identified defect is then mapped into their respective classes.
- The primary objective is to develop a deep learning model using CNN (Convolutional Neural Network) to identify the defects in material/product and then map the products into their respective classes which include Defective/Non-defective.

Secondary Objectives Include

- To improve the accuracy of the model by using different activation functions such as ReLU, tanh, etc.
- To tune the model to achieve high accuracy and low computational cost/time when compared to the previous state of art methods.

Problem Statement

Quality inspection in industries by far is done by the traditional methods such as machine vision and computer vision systems and this affects the accuracy in determining the goodness of the product. This low accuracy is due to the restricted number of parameters that are set to identify the defects. The modern deep learning method does not restrict the number of parameters and uses a large number of parameters to train itself and produce the result effectively.

METHODOLOGY

Proposed System

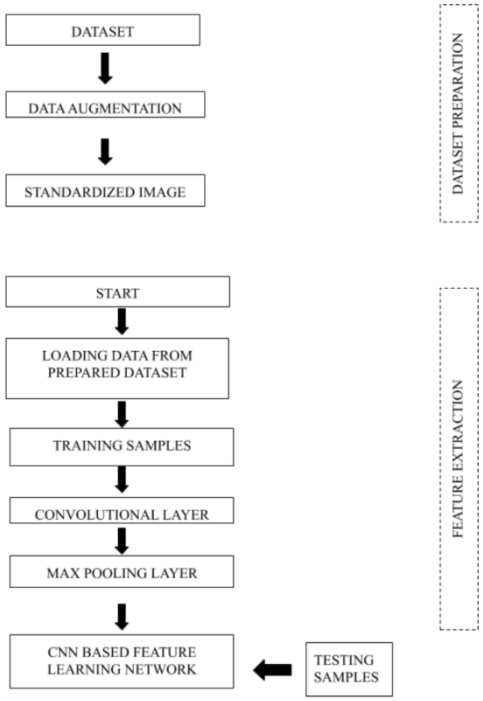

Figure 1: Block Diagram of Workflow for Model Building.

Fig. 1 represents the process involved in extracting the features from the image. Automatic identification of defects is a difficult task due to the limitation in the number of defective samples and different types of defects. Therefore, the image dataset requires some data preparation stages like data augmentation. It is the process of creating duplicate images for a better understanding of features in the images for various scales and orientations. This work focuses on building defect detection and classification using Convolutional Neural Network (CNN). The proposed system uses **Severstal: Steel** [6] defect detection dataset. The prepared data is then split into training samples and testing samples in the ratio of 80:20. The images are given as an input to the Convolutional Neural Network (CNN) and it will create a number of parameters for learning. During the initial phase, the horizontal and vertical edge features are detected and in subsequent convolutions, essential features are detected. The max-pooling is used to take the maximum element from the region of the feature map covered by the filter. Thus, the output after the max-pooling layer would be a feature map containing the most prominent features of the previous feature map. The obtained parameters are then checked using the testing samples for validating the model.

You Look Only Once (YOLO) [7] is the CNN architecture used in this work to solve the classification problem. The architecture was implemented in two stages. In the first stage, regions of interest in an image are selected. The next stage is to classify these regions using convolutional neural networks. Instead of selecting interesting parts of an image, they predict classes and bounding boxes for the whole image in one run of the algorithm. In the YOLO algorithm, it is necessary to establish what is actually being predicted. Ultimately, the aim is to predict a class of an object and the bounding box specifying object location. Each bounding box can be described using four descriptors: centre of a bounding box (bxby) width (bw) height (bh) value cis corresponding to a class of an object.

Class Prediction

The defects in the steel surfaces are identified and shown as bounding boxes. Each box predicts the type of defects present in it and may contain different types of defect classes in the bounding box. The softmax function is not used as it is unnecessary for good performance, independent logistic classifiers are used. During training, binary cross-entropy loss is used for the class predictions. This formulation helps when we move to more complex domains like the detection of defects on steel surfaces [6]. In this dataset, there are many overlapping labels (i.e. defects of different classes). Using a softmax imposes the assumption that each box has exactly one class which is often not the case. A multilabel approach better model of the data to have multiple defects in the same area.

Feature Extractor

A new network is used for performing feature extraction. A new network is a hybrid approach between the network used in YOLOv2, Darknet-19, and that newfangled residual network stuff. This network uses successive 3 × 3 and 1 × 1 convolutional layers but now has some shortcut connections as well and is significantly larger. It has 53 convolutional layers [8] so it is called Darknet-53.

Table 1: Darknet - 53

	Type	Filters	Size	Output
	Convolutional Convolutional	32 64	3 x 3 3 x 3/2	256 x 256 128 x 128
1 x	Convolutional Convolutional Residual	32 64	1 x 1 3 x 3	128 x 128
	Convolutional	128	3 x 3/2	64 x 64
2 x	Convolutional Convolutional Residual	64 128	1 x 1 3 x 3	64 x 64
	Convolutional	256	3 x 3/2	32 x 32
8 x	Convolutional Convolutional Residual	128 256	1 x 1 3 x 3	32 x 32
	Convolutional	512	3 x 3/2	16 x 16
8 x	Convolutional Convolutional Residual	256 512	1 x 1 3 x 3	16 x 16
	Convolutional	1024	3 x 3/2	8 x 8
4 x	Convolutional Convolutional Residual	512 1024	1 x 1 3 x 3	8 x 8
	Avgpool Connected Softmax		Global 1000	

Table 1 explains the architecture of the darknet-53 yolo model including its parameters and the size of the layers. This new network is much more powerful than Darknet19 but still more efficient [9] than ResNet-101 or ResNet-152.

Steps in Building the Model

- The first step is to label the images using labelling and map it with defective classes and the image bounding box parameters are stored in a text file.
- The second step is to configure the training parameters which are used to train the dataset. These configuration parameters specify the input image size and the number of channels. It also contains a few parameters like momentum and decay that control how the weight is updated.
- The third step is to set up the Learning Rate, Steps, Scales, Burn-In (warm-up). The Burn-In is used to control the learning rate because the learning rate should be high during the training initialization, and it should be reduced as the model gets trained. The training occurs faster when the learning rate is low.
- The fourth step is to augment the data to produce a bigger training set for imbalanced classes and effectively train the model.
- The fifth step is to set the batch size and epoch for training. The use of epoch is that the entire dataset cannot be used for training, because it will throw a memory error and the samples are split and sent for training in terms of an epoch. The batch size determines how many instances should be present. The model is tested using loss and map (moving average precision) functions.

EXPERIMENTAL RESULTS AND DISCUSSIONS

When the training is started the average loss is very high during early iterations and tends to slow down after a certain time. The training is continued until the average loss becomes low (=0.6). Usually, 2000 iterations are sufficient for each class (object) but not less than the number of training images and not less than 6000 iterations in total [8]. But for a more precise definition to stop training, notice the varying indicators of error, and it should stop when no longer decreases 0.50 avg:

```
1: 420.369049, 420.369049 avg, 0.000000 rate, 2.076527 seconds, 1 images
Loaded: 0.000031 seconds
Region 16 Avg IOU: 0.486668, Class: 0.499934, Obj: 0.499776, No Obj: 0.499833, .5R: 0.000000,
.75R: 0.000000,  count: 1
Region 23 Avg IOU: nan, Class: nan, Obj: nan, No Obj: 0.498697, .5R: nan, .75R: nan,  count: 0
2: 421.127350, 420.444885 avg, 0.000000 rate, 2.082417 seconds, 2 images
Loaded: 0.000027 seconds
Region 16 Avg IOU: 0.774884, Class: 0.501205, Obj: 0.499345, No Obj: 0.499833, .5R: 1.000000,
.75R: 1.000000,  count: 1
Region 23 Avg IOU: nan, Class: nan, Obj: nan, No Obj: 0.498695, .5R: nan, .75R: nan,  count: 0
```

Figure 2: Start of Training

Figure 2 shows the initialization of training. The training loss is high during the first iteration (420.36) and the training should continue until a minimal loss is attained.

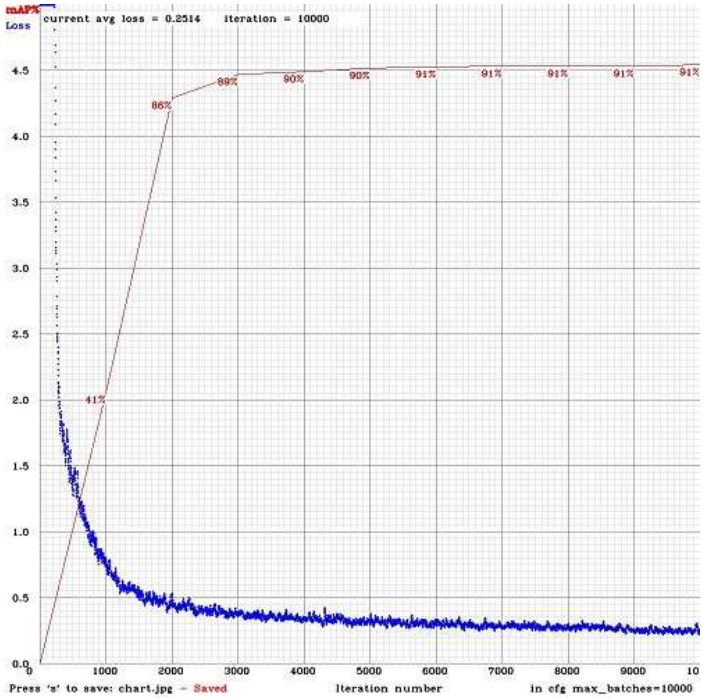

Figure 3: Iterations vs. Mean Average Loss.

The figure 3 shoes mAP (Mean Average Precision) chart (red-line) in the Loss-chart Window. The X-axis is the number of epochs and the Y-axis shows the training error percentage. The mAp will be calculated for every 4 Epochs using valid=valid.txt file that is specified in obj. data file (1 Epoch = images in train txt/batch iterations) (to change the max x-axis value - change max_batches= parameter to 2000*classes, f.e. max batches=8000 for 4 classes)

```
4717: 0.478695, 0.633565 avg, 0.001000 rate, 1.094805 seconds, 4717 images
Loaded: 0.000035 seconds
Region 16 Avg IOU: nan, Class: nan, Obj: nan, No Obj: 0.001769, .5R: nan, .75R: nan,  count: 0
Region 23 Avg IOU: 0.757523, Class: 0.979580, Obj: 0.177507, No Obj: 0.000248, .5R: 1.000000,
.75R: 1.000000,  count: 1
4718: 0.590653, 0.629274 avg, 0.001000 rate, 1.088090 seconds, 4718 images
Loaded: 0.000037 seconds
Region 16 Avg IOU: 0.742906, Class: 0.996573, Obj: 0.281718, No Obj: 0.002035, .5R: 1.000000,
.75R: 0.000000,  count: 1
```

Figure 4 End of Training.

Figure 4 shows the end of the training. When the loss is minimized (=0.6) after certain iterations the training can be stopped. If continued after this the change in loss is very low and may even over fit the model.

Accuracy

After the minimum loss is achieved the training is stopped, the weights file with a minimum loss, and maximum map (mean average precision) is selected and tested on the validation dataset. The accuracy of the model can be calculated using precision and recall [10]. The parameters used to determine accuracy are

- IoU (intersection over union) - average intersection over union of objects and detections for a certain threshold = 0.24
- mAP (mean average precision) - mean value of average precisions for each class, where average precision is average value of 11 points on PR-curve for each possible threshold (each probability of detection) for the same class (Precision-Recall in terms of PascalVOC, where Precision=TP/(TP+FP) and Recall=TP/(TP+FN))

Table 2: Backbones of Convolutional Architectures

Backbone	Top-1	Top-5	Bn Ops	BFLOP/s	FPS
Darknet-19	74.1	91.8	7.29	1246	171
ResNet-101	77.1	93.7	19.7	1039	53
ResNet-152	77.6	93.8	29.4	1090	37
Darknet-53	77.2	93.8	18.7	1457	78

Table 2 explains the accuracy, billions of operations, billion floating-point operations per second, and frames per second (FPS) for various networks. Each network is trained with identical settings and tested at 1600×256, single crop accuracy. Thus Darknet-53 performs on par with state-of-the-art classifiers but with fewer floating-point operations and more speed. Darknet-53 is better than ResNet-101 and 1.5 times faster. Darknet-53 has a similar performance to ResNet-152 and is 2 times faster. Darknet-53 also achieves the highest measured floating-point operations per second. This means the network structure better utilizes the GPU, making it more efficient to evaluate and thus faster. That's mostly because ResNets have just way too many layers and aren't very efficient.

Advantages & Limitations

The increase in the amount of data will significantly increase the model capacity to accurately detect the defects in the steel surfaces. Due to fast-moving production lines, this yolo v3 can do an online inspection at a faster rate. The functionality of neural networks is that with an increase in data points the depth of the neural network can be increased for better learning. The Convolutional Neural Network (CNN) works well on huge data and models complex patterns better than machine learning algorithms and adds advantage over machine learning algorithms with very little overfit to the training data. The limitation of deep learning algorithms is data. If sufficient data is available, the algorithm works well and it will learn data points very well.

CONCLUSIONS

In the steel manufacturing industry, defects are a major problem and it might lead to bad quality products of the industry. Identifying defects during the production process will help to eliminate the defective products in the market. In this project, a model is built to detect the defects on the steel surfaces at a faster rate in a single shot. According to the map (mean average precision), it is claimed that YOLO is more accurate than SSD [11] and faster than Resnet while SSD is faster than YOLO. Similarly, the result of the implementation matches the studied results regardless of whether one image or real-time testing. Besides, in terms of real-time implementation, the output result shows no difference between image and video, within one second, the SSD detector processes 10 times more frames than YOLO. However, with the YOLOv3 model, its velocity, which is able to process about 9 frames per second, is way better than YOLOv2 although it is slightly slower than SSD's 15 frames per second.

REFERENCES

1. *Yuji Iwahori, Yohei Takada, Tokiko Shiina, Yoshinori Adachi "Defect Classification of Electronic Board Using Dense SIFT and CNN" 22nd International Conference on Knowledge-Based and Intelligent Information & Engineering Systems.*
2. *Wuqin Tang, Qiang Yang, Kuixiang Xiong, Wenjun Yan "Deep learning-based automatic defect identification of photovoltaic module using electroluminescence images" Solar Energy 201 (2020) 453–460*
3. *Benjamin Staara,*, Michael Lütjena, Michael Freitag "Anomaly detection with convolutional neural networks for industrial surface inspection", 12th CIRP Conference on Intelligent Computation in Manufacturing Engineering, 18-20 July 2018.*
4. *Joseph Redmon, Ali Farhadi, "YOLOv3: An Incremental Improvement" in arxiv:1804.0276, Issue 8 April 2018.*
5. *J. Redmon and A. Farhadi. Yolo9000: Better, faster, stronger. In Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on, pages 6517–6525. IEEE, 2017.*
6. *Public available at the Kaggle platform, Severstal-steel-defect-detection, Oct. 2019. Available: https://www.kaggle.com/c/severstal-steel-defect-detection*
7. *Joseph Redmon, Ali Farhadi, "YOLOv3: An Incremental Improvement" in arxiv: 1804.0276, Issue 8 April 2018.*
8. *J. Redmon. Darknet: Open-source neural networks inc. http://pjreddie.com/darknet/, 2013–2016*
9. *J. Redmon and A. Farhadi. Yolo9000: Better, faster, stronger. In Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on, pages 6517–6525. IEEE, 2017.treatment", article, IEEE, 2012.*
10. *A blog on Github, "Yolo v4, v3 and v2 for Windows and Linux" February 2020, accessed on October 8, 2020. Available online https://github.com/AlexeyAB/darknet*
11. *Shaojie Yang, Xiang Li, Xiaodong Jia , Yinglu Wang , Haodong Zhao , Jay Lee " deep learning based intelligent defect detection of cutting wheels" in 48th SME North American Manufacturing Research Conference, NAMRC 48*