An Improved GANs Model for Steel Plate Defect Detection

Renjie Tang1,2• and Keming Mao3

1School of Computer science and engineering, Southeast University, Nanjing, Jiangsu Province, 211189, China
2China Mobile Group Zhejiang Co., Ltd., Hangzhou, Zhejiang Province, 310016, China
3College of Software, Northeastern University, Shenyang, Liaoning Province, 110004, China

*Corresponding author

Abstract. Automatic steel plate defect detection is very important for it can monitor the product quality. This paper makes a study on steel plate defect detection based on machine learning. The main difficult is that there is not enough data to make powerful detection models. We propose a Generative Adversarial Networks based method to generate synthetic training image. A novel structure is designed with type related variable incorporated in Generator and a classification branch added to Discriminator. With expanded dataset, two detection algorithm, Faster R-CNN and YOLO are adopted. Various model structures, optimization methods, batch sizes and model execution time are evaluated and the influence of parameters are also analyzed. The experimental results show that the proposed novel data generation method can effectively improve the model performance.

1. Introduction

With the rapid growing of construction industry, steel plate is widely used in fields of bridge, automobile and aerospace, etc. The surface quality not only make affect on the subsequent production, but also ensure the service life and the competitiveness. Therefore, detect surface defects is a main task of steel industry. Automated steel plate defect detection has been developed since 1970s. Compared with manually inspection, it is efficient and accurate. It can be classified as traditional technology, laser scanning based and machine vision based. Traditional methods include eddy current testing [1,2], infrared testing and ultrasonic testing [3]. They are with high energy consumption and vulnerable to environmental. Laser scanning method [4] improves the sensitivity and versatility, but it is not very good for tiny defects. Using computer vision to solve steel defect detection has attracted more attentions in recent years. In [5], local normalization was used to enhanced the image and projection profile was applied for defects location. Tian et al. [6] combined genetic algorithm and extreme learning for defects classification. Suvdaa et al. [7] detected steel defects based on scale-invariant feature transform(SIFT) feature representation and support vector machine (SVM) classifier. Paulraj et al. [8] employed discrete fourier transform and neural network. Ghorai et al. [9] employed wavelet sets for steel defect detection. Yazdchi et al. [10] presented an algorithm based on multifractal. Landstrom et al. [11] proposed a method based on 3D contour and statistical logistic regression. These algorithms achieved good results. However, there still exists some problems to be solved, for example, complex structure, sensitive to noise and poor versatility.
Form the year 2012, deep learning models especially CNNs models have greatly improved the model accuracy. [12] trained a CNNs model for metal surface defects on photometric stereo image. A Faster R-CNN based method was presented for surface damage detection in [13]. Multi-group convolutional neural network (MG-CNN) was proposed in [14]. However, insufficient training data are rarely mentioned, which is common case in industrial domain. It is a fact that construct dataset with detailed and accurate annotations are time consuming. It needs much works of professional experts. To address this problem, this work proposes a novel Generative Adversarial Networks (GANs) based model for data augment, which aims to improve model performance using data augment. Based on naive GANs model, we combine class type into GANs model, and multiple types data generation can be gained. Training dataset is expanded with these generated synthetic image data and the model performance can be improved. Extensive experiments are tested to evaluate the performance on various parameters. The rest of this paper is organized as follows: Section 2 introduces the framework of defect detection on steel plate. Section 3 presents the basic CNNs model. Section 4 gives the proposed GANs based synthetic image generation. Objection detection algorithms are explained in Section 5. Section 6 gives the experimental evaluation. Finally, Section 7 concludes this paper.

2. Framework of Steel Plate Defect Detection

Fig. 1 gives the framework of steel plate defect detection. In training, batch of images are fed into CNNs model. Regions and their types are computed on feature maps. The parameters are tuned by back-propagation and SGD based loss function. The trained model is used in process of testing.

3. Basic CNNs Model

CNNs model is the backbone for feature extraction and representation of deep learning. It mainly includes the following operations (layers).

3.1. Convolution Layer

Convolution layer contains trainable filters[15]. Generally, the filter slides along the image with certain stride. The image local part is convolved with the filter. As shown in equation 1, \( \text{conv}(x, y) \) and \( c(i, j) \) represent convolution result and convolution kernel. \( I(x, y) \) represents the input image. \( k \) is the kernel size.

\[
\text{conv}(x, y) = \sum_{i} \sum_{j} c(i, j) * I(x - j, y - j)
\]

3.2. Activation Layers

Activation function can increase the model nonlinearity and representation ability. The popular activation functions include \( \text{sigmoid}, \text{tanh} \) and \( \text{ReLU} \), etc. Equation 2 gives the \( \text{ReLU} \) function[16], where \( f(x) \) represents the activation value. \( \text{ReLU} \) can solve gradient vanishing problem, and it is more effective.

\[
f(x) = \max(0, x)
\]
3.3. Pooling Layers
Pool operation is an effectively way to reduce the size of feature map. It can speed up the computation and prevent over-fitting. Max pooling and mean pooling are two widely used methods [17].

3.4. Fully Connected Layer
In this layer, all nodes of current layer and previous layer are fully connected. It is often used as the output layer, and the number of node denotes the types that model outputs.

3.5. Loss Function and Model Training
For model training, equation 3 defines the loss function for calculating the deviations between model predicted label \( h_\theta(x_i) \) and ground truth label \( y_i \). Equation 4 and 5 give the deviation based on SGD.

\[
J(\theta) = -\frac{1}{2m} \sum_{i=1}^{m} (h_\theta(x_i) - y_i)^2 + \frac{r}{2} R(\theta)
\]

\[
\nabla_{\theta_j} J(\theta) = -\frac{1}{m} \sum_{i=1}^{m} (h_\theta(x_i) - y_i)x_i + r\theta_j
\]

\[
\theta_j = \theta_j + \alpha \nabla_{\theta_j} J(\theta)
\]

4. Type Related GANs Model for Steel Plate Defect Synthetic Image Generation
GANs was proposed in [18]. Fig.2 gives the basic framework of GANs, which includes one Generator \( G \) and one Discriminator \( D \). \( G \) models the distribution over real data \( p_{data} \) and generates data \( G(z) \) through a random Gaussian noise \( z \). \( D \) outputs probability that \( G(z) \) derives from real data rather than generated one.

\[
\begin{align*}
\min_G \max_D V(D,G) &= E_{x \sim p_{data}(x)}[\log(D(x))] + E_{z \sim p_z(z)}[\log(1 - D(G(z)))] \\
\max_D V(D,G) &= E_{x \sim p_{data}(x)}[\log(D(x))] + E_{z \sim p_z(z)}[\log(1 - D(G(z)))] \\
\min_G V(D,G) &= E_{z \sim p_z(z)}[\log(1 - D(G(z)))]
\end{align*}
\]

Figure 2. Basic structure of GANs model.

The objective function of GANs is defined as equation 6. It is a min-max problem, where \( p_{data}(x) \) and \( p_z(z) \) give the distribution of real data \( x \) and noise \( z \) respectively. For optimization, \( D \) is optimized as equation 7 with \( G \) fixed. Then \( G \) is optimized as equation 8 with \( D \) fixed.

In our research, multiple types defects are needed. We integrate multiple types data generation in one GANs model. As shown in fig.3. A class specific variable \( c \) is incorporated in \( G \), and the generated synthetic data is represented with \( G(z, c) \). The output of \( D \) has two branches, \( D(x) \) is used to distinguish data as real or fake, and \( D(c|x) \) is used for classify the data type.
Figure 3. Structure of the proposed type related GANs model. The objective function and optimization can be redefined as equation 9-11.

\[
\max_D V(D, G) = E_{x \sim P_{d}(x)}[\log(D(x))] + E_{z \sim P_{z}(z)}[\log(D(c \mid x))] \\
+ E_{z \sim P_{z}(z)}[\log(1 - D(G(z, c)))] + E_{z \sim P_{z}(z)}[\log(1 - D(c \mid G(z, c)))] \\
(9)
\]

\[
\max_D V(D, G) = E_{x \sim P_{d}(x)}[\log(D(x))] + E_{z \sim P_{z}(z)}[\log(D(c \mid x))] \\
+ E_{z \sim P_{z}(z)}[\log(1 - D(G(z, c)))] + E_{z \sim P_{z}(z)}[\log(1 - D(c \mid G(z, c)))] \\
(10)
\]

\[
\min_G V(D, G) = E_{z \sim P_{z}(z)}[\log(1 - D(G(z, c)))] + E_{z \sim P_{z}(z)}[\log(1 - D(c \mid G(z, c)))] \\
(11)
\]

Figure 4. Framework of steel plate defect training image generation. Based on the above GANs framework, more training data for steel plate defects detection can be generated. Fig. 4 gives the procedure for steel defect detection training image generation.

5. Object Detection Algorithms

Two advanced object detection models, Faster R-CNN [19] and YOLO [20] are explained.

5.1. Faster R-CNN

Faster R-CNN contains RPN and Fast R-CNN. RPN uses a small sliding window over feature map to generate candidate regions. All regions are set with objecteness scores. Each region is processed by ROI pooling. The features of objects are transformed into a fixed-size vector, which is finally fed into Fast R-CNN. Multiclass cross entropy is used as loss function. NMS is used to remove the redundant objects.

The output of RPN is \((x, y, w, h, S_f)\), where \((x, y)\) represents the position of region proposal, and \((w, h)\) means width and height of region proposal. \(S_f\) indicates foreground probability of region proposal. The output of Fast R-CNN is \((x, y, w, h, S_c)\), where \((x, y)\) and \((w, h)\) represent the parameters of predicted bounding box, and \(S_c\) represents the score of the specific class in the bounding box.

5.2. YOLO

YOLO is an one-stage algorithm, which directly predicts the type and location of objects from input image. As for the latest version of YOLO, YOLOv3 first fed the input images into the CNNs model for
feature map extraction. YOLOv3 uses Darknet-53 without fully connected layer and three different scale feature maps. K-means clustering is used to obtain the prior size of bounding boxes. The output of each box has five basic parameters \((x, y, w, h, \text{confidence})\), where \((x, y)\) and \((w, h)\) represent the center point coordinates and the width and height of bounding box. The confidence indicates the probability that bounding box contains object and the accuracy of bounding box.

6. Experiment Result

6.1. Raw Dataset

There are totally 250 steel plate image samples (containing 1600 defect regions, 1000 pitting and 600 scratches). The regions and types of defects are annotated manually. Original images are collected from the production line of a steel company by digital camera. The size of original images are 3700 pixels * 3000 pixels. Each image has an extra *.xml file containing the position of the bounding box and the type of steel surface defects. The format of *.xml file is the same as VOC2007 dataset. Figure 5 shows two types of steel surface defects. 80% of dataset are used for training model, and the rest are used for testing.

![Figure 5](image)

(a) demonstrates samples of scratch; (b) demonstrates samples of pitting. The green boxes denote the locations of the defect.

6.2. Experiment Setting

All codes are implemented with Python. TensorFlow/Keras are selected as deep learning framework. Experiments are evaluated on Ubuntu 16.04 OS, Intel i5 CPU, 32G RAM, Nvidia GTX 1080Ti 11G GPU.

6.2.1. Setting for Faster R-CNN. VGG16 and Resnet50 are used as backbone network. Scales with box areas of 32, 64, 128, 256, 512 and different aspect ratios of 0.5, 1, 2, 3, 5 are set. The mini-batch is 32. 2000 and 750 are set for RPN proposals in training and generation. The initial learning rate is 0.001. The momentum is 0.9 and learning rate decay factor is 0.1. The maximum number of iteration step is 100000.

6.2.2. Setting for YOLOv3. The backbone network are Darknet53 and Darknet59. Three kinds of a prior boxes are set, and 9 sizes of a prior boxes are clustered. The parameters of NMS algorithm, batchsize, initial learning, momentum and maximum number of iteration step are the same as Faster R-CNN.

6.3. Evaluation of Accuracy Performance

\(mAP\) (mean average precision), a widely used metric is adopted, which is defined as equation 12.

\[
mAP = \frac{1}{|C|} \sum_{q \in C} AP(q)
\]

\[\text{precision} = \frac{TP}{TP + FP}
\]

\[\text{recall} = \frac{TP}{TP + FN}
\]
**AP** (average precision) indicates the area under **PR** (precision-recall) curve. **Precision** reflects the proportion of correctly detected defects to regions that all detected. **Recall** reflects the proportion of correctly detected defects to all ground truth defects. **Precision** and **recall** are calculated as equations 13 and 14. A defect is correctly detected only if the **IOU** is greater than 0.5. **mAP** measures the average performance of a model on all categories.

The accuracy evaluation is given in table 1. Models are listed in 1st column. The 2nd column represents the data augment. ‘Y(traditional)’ means traditional data augment, including random cropping, flipping and perturbation. ‘Y(GANs)’ means the model is trained with our proposed GANs based data augment.

**Table 1.** Accuracy evaluation of steel plate defect detection.

| Model               | Data augment | Training mAP | Testing mAP |
|---------------------|--------------|--------------|-------------|
| Faster R-CNN(VGG16) | Y(traditional) | 0.957        | 0.853       |
| Faster R-CNN(VGG16) | Y(GANs)      | 0.945        | 0.887       |
| Faster R-CNN(VGG16) | N            | 0.987        | 0.714       |
| Faster R-CNN(Resnet50) | Y(traditional) | 0.945        | 0.841       |
| Faster R-CNN(Resnet50) | Y(GANs)      | 0.963        | 0.875       |
| Faster R-CNN(Resnet50) | N            | 0.987        | 0.725       |
| YOLOv3(Darknet53) | Y(traditional) | 0.92         | 0.829       |
| YOLOv3(Darknet53) | Y(GANs)      | 0.961        | 0.866       |
| YOLOv3(Darknet53) | N            | 0.991        | 0.727       |
| YOLOv3(Darknet59) | Y(traditional) | 0.93         | 0.838       |
| YOLOv3(Darknet59) | Y(GANs)      | 0.943        | 0.882       |
| YOLOv3(Darknet59) | N            | 0.993        | 0.741       |

We can see that there is about 10% gap between models. Model training with data augment is equivalent to increase the size of training data. Meanwhile, there are about 3%-5% promotion with our proposed GANs based method than traditional data augment method. This demonstrates that our proposed method can generate more realistic synthetic data, and the detection model can be further improved. The result shows that Faster R-CNN with VGG16 obtain the best performance, with mAP value of 0.887.

### 6.4. Evaluation of Optimization algorithm
**Adam**, **Adagrad** and **SGD** are evaluated with other parameters keep the same. All optimization algorithms can get converge after about 30 epoch. Compared with **Adam** and **Adagrad**, models trained with **SGD** are more stable. This result shows that **SGD** is suitable for most tasks although its simplicity.

### 6.5. Evaluation of Batch Size
In this subsection, influence of batch size is evaluated. Batch sizes with 16, 32, 48 and 128 are tested while other parameters keep the same. Models with batch size 16 convergences slow. Models with batch size 32 get superior accuracy. Compared with models trained with batchsize 16, 32, 64 and 128, models with batchsize 32 are promoted by 0.61%, 0.83%, 0.78% and 1.05% for mAP respectively.

### 6.6. Evaluation of Running Efficiency
**Table 2.** Running efficiency evaluation of steel plate defect detection(seconds).

| Model               | Batch size for training time | Testing time |
|---------------------|-----------------------------|--------------|
|                    | 16  | 32  | 64  | 128 |          |
| Faster R-CNN(VGG16) | 2.85 | 4.03 | 5.37 | 8.69 | 0.13     |
| Faster R-CNN(Resnet50) | 1.94 | 3.53 | 4.68 | 7.16 | 0.117    |
| YOLOv3(Darknet53) | 1.17 | 2.12 | 3.97 | 6.29 | 0.089    |
| YOLOv3(Darknet59) | 1.28 | 2.31 | 4.01 | 6.54 | 0.091    |
Table 2 gives the results of model running efficiency evaluation. Faster R-CNN with VGG16 consumes maximal training time, with 2.85s, 4.03s, 5.37s and 8.69s. For testing time, YOLOv3 runs faster.

7. Conclusion and Future Works
This study makes a research on steel plate defect detection. The benchmark dataset is first collected from a production line, and there are two types of defect, scratch and pitting. GANs based synthetic image data augment method is proposed. Using generated image data, training dataset is expanded, and the model performance is improved. Through this case study, we can conclude that it is possible to detect defects on steel plate surface using CNNs model with preferable accuracy and efficiency. Our future works will concentrate on these aspects: (1) More data should be generated to enlarge the benchmark dataset; (2) Customed CNNs model structures should be studied so performance can be further improved.

References
[1] Bin Gao, Libing Bai, Wai Lok Woo, Guiyun Tian, Yuhua Cheng: Automatic Defect Identification of Eddy Current Pulsed Thermography Using Single Channel Blind Source Separation. IEEE Trans. Instrumentation and Measurement 63(4): 913-922 (2014)
[2] M. Bentoumi, P. Aknin, G. Bloch: On-line rail defect diagnosis with differential eddy current probes and specific detection processing. Eur. Phys. J.-Appl. Phys., vol. 23, no. 3, pp. 227-233, 2003.
[3] F. C. Cruz, E. F. S. Filho, M. C. Albuquerque, I. C. Silva, C. T. Farias, L. L. Gouvêa: Efficient feature selection for neural network based detection of flaws in steel welded joints using ultrasound testing. Ultrasonics, vol.73, pp. 1-8, Jan. 2017.
[4] David Martin, Domingo Miguel Guinea, Maria C. García-Alegre, E. Villanueva, Domingo Guinea: Multi-modal defect detection of residual oxide scale on a cold stainless steel strip. Mach. Vis. Appl.21(5): 653-666 (2010)
[5] Qingyong Li, Shengwei Ren: A Real-Time Visual Inspection System for Discrete Surface Defects of Rail Heads. IEEE Trans. Instrumentation and Measurement 61(8): 2189-2199 (2012)
[6] S. Tian, K. Xu: An algorithm for surface defect identification of steel plates based on genetic algorithm and extreme learning machine. Metals, 7 (8) (2017), p. 311
[7] B. Suvdaa, J. Ahn, J. Ko: Steel surface defects detection and classification using SIFT and voting strategy. International Journal of Software Engineering and its Application, 6 (2) (2012), pp. 161-165
[8] Paulraj, M. P., Shukry, A. M., Yaacob, S., Adom, A. H., Krishnan, R. P: Structural steel plate damage detection using DFT spectral energy and artificial neural network. In Signal Processing and Its Applications (CSPA), 2010 6th International Colloquium on (pp. 1–6). IEEE.
[9] S. Ghorai, Anirban Mukherjee, M. Gangadaran, Pranab K. Dutta: Automatic Defect Detection on Hot-Rolled Flat Steel Products. IEEE Trans. Instrumentation and Measurement 62(3): 612-621 (2013)
[10] Mohammadreza Yazdchi, Mehran Yazdi, Arash Golibagh Mahyari: Steel Surface Defect Detection Using Texture Segmentation Based on Multifractal Dimension. ICDCIP 2009: 346-350
[11] A. Landstrøm, M.J. Thurley: Morphology-based crack detection for steel slabs. IEEE Journal of Selected Topics in Signal Processing, 6 (7) (2012), pp. 866-875
[12] Daniel Soukup, Reinhold Huber-Mörk: Convolutional Neural Networks for Steel Surface Defect Detection from Photometric Stereo Images. ISVC (1) 2014: 668-677
[13] Cha, Y.J.; Choi, W.; Suh, G.; Mahmoudkhani, S.; Büyükoztürk, O: Autonomous Structural Visual Inspection Using Region-Based Deep Learning for Detecting Multiple Damage Types. Comput.-Aided Civ. Infrastruct. Eng. 2017, doi:10.1111/mice.12334.
[14] Di He, Ke Xu, Peng Zhou: Defect detection of hot rolled steels with a new object detection framework called classification priority network. Computers Industrial Engineering 128: 290-297 (2019)
[15] Keming Mao, Duo Lu, Dazhi E, Zhenhua Tan: A Case Study on Attribute Recognition of Heated
Metal Mark Image Using Deep Convolutional Neural Networks. Sensors 18(6): 1871 (2018)

[16] Glorot, X., Bordes, A., and Bengio, Y: Deep sparse rectifier neural networks. In JMLR WCP: Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics (AISTATS 2011). 130, 297

[17] Scherer, D., Muller, A. Behnke, S: Evaluation of pooling operations in convolutional architectures for object recognition, in K. Diamantaras, W. Duch, and L. S. Iliadis (eds.), Artificial Neural Networks–ICANN 2010, Springer Verlag, Berlin Heidelberg, 92–101.

[18] I. Goodfellow et al., “Generative adversarial nets,” in Proc. NIPS, 2014, pp. 2672–2680.

[19] Shaoqing Ren, Kaiming He, Ross B. Girshick, Jian Sun: Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. CoRR abs/1506.01497 (2015)

[20] Joseph Redmon, Ali Farhadi: YOLOv3: An Incremental Improvement. CoRR abs/1804.02767 (2018)