



## SISTEM INSPEKSI PERMUKAAN BAJA BERBASIS *DEEP LEARNING* MENGGUNAKAN METODE *ANCHOR-FREE*

Singgih Yuliyanto<sup>a</sup>, Nurinda Fadhilah Amani<sup>b</sup>, Fityanul Akhyar<sup>c</sup>, Koredianto Usman<sup>d</sup>

<sup>a</sup> [singgihyl@student.telkomuniversity.ac.id](mailto:singgihyl@student.telkomuniversity.ac.id) , Universitas Telkom, Bandung

<sup>b</sup> [nurinda@student.telkomuniversity.ac.id](mailto:nurinda@student.telkomuniversity.ac.id), Universitas Telkom, Bandung

<sup>c</sup> [fityanulakhyar@student.telkomuniversity.ac.id](mailto:fityanulakhyar@student.telkomuniversity.ac.id), Universitas Telkom, Bandung

<sup>d</sup> [korediantousman@telkomuniversity.ac.id](mailto:korediantousman@telkomuniversity.ac.id) , Universitas Telkom, Bandung

### ABSTRACT

*Steel is one of the important materials in the industry. Steel may have defects in the production process that can affect the steel products. Therefore, the detection of steel surface defects is an important process to control the quality of steel products. An efficient steel surface detection process is carried out by automating steel images taken using a camera. We use an anchor-free model FoveaBox. FoveaBox is an accurate and flexible model for detecting objects and has a simple architecture. This study uses the NEU-DET dataset consists of six types of steel surface defects, namely crazing, inclusion, patches, pitted surface, rolled-in scale, and scratches, each with a total of 300 data.. The test results on the system show that the method used has a good detection performance with a mean average precision of 0.834 or 83.4% at a learning rate of 0.001, Optimizer SGD, sigma 0.6, and the number of epochs 24. This detection method can detect steel surface defects. This detection method can effectively detect steel surface defects with similar foreground and background characteristics. With an accuracy threshold of 80%, the method used in this study has an adequate precision value.*

*Keywords: anchor-free, FoveaBox, object detection, single-stage detection, surface defect detection*

### ABSTRAK

Baja merupakan salah satu material penting dalam industri. Dalam proses produksi baja dimungkinkan terjadi kecacatan, cacat tersebut dapat mempengaruhi kualitas baja. Oleh karena itu, deteksi cacat permukaan baja merupakan proses yang penting untuk mengontrol kualitas produk baja. Proses deteksi permukaan baja secara efisien dilakukan dengan otomatisasi menggunakan citra baja yang diambil menggunakan kamera. Pada penelitian ini digunakan model deteksi *anchor-free* yaitu FoveaBox untuk melakukan deteksi cacat. FoveaBox merupakan sebuah model *deep learning* yang akurat dan fleksibel dalam mendeteksi objek serta memiliki arsitektur yang sederhana. Penelitian ini menggunakan dataset NEU-DET yang terdiri enam jenis cacat permukaan baja yaitu *crazing, inclusion, patches, pitted surface, rolled-in scale* dan *scratches* masing-masing dengan jumlah data 300. Hasil pengujian terhadap sistem menunjukkan bahwa metode yang digunakan memiliki kinerja deteksi yang baik dengan nilai *mean average precision* 0,834 atau 83,4% pada *learning rate* sebesar 0,001, *Optimizer* SGD, *sigma* sebesar 0,6 dan jumlah *epoch* sebanyak 24. Metode deteksi ini mampu mendeteksi cacat permukaan baja dengan karakteristik *foreground* dan *background* yang mirip secara efektif. Dengan nilai ambang akurasi sebesar 80% maka metode yang diteliti ini memiliki nilai presisi yang memadai.

Kata Kunci : deteksi cacat permukaan, FoveaBox, *single-stage detection, anchor-free*, deteksi objek

## 1. PENDAHULUAN

Baja merupakan salah satu bahan baku yang penting dalam industri. Eksploitasi baja menduduki peringkat pertama dan produksinya terdiri atas hampir 95% dari produk barang berbahan logam. Pada proses produksi baja dimungkinkan terjadi berbagai macam kecacatan. Tidak hanya berpengaruh terhadap kualitas tampilan produk, cacat tersebut juga mempengaruhi kinerja dan keamanan produk [1]. Dari permasalahan tersebut dibutuhkan adanya pengawasan terhadap kualitas baja. Saat ini sebagian besar proses pengawasan masih dilakukan dengan cara manual. Metode tersebut memiliki banyak kelemahan seperti biaya yang tinggi, akurasi rendah, efisiensi rendah, dan tidak bisa memenuhi kebutuhan produksi otomatis industri modern [2].

Pemanfaatan metode *deep learning* untuk deteksi cacat memiliki banyak kelebihan yaitu: memiliki tingkat akurasi yang tinggi, efisiensi tinggi, dan dapat bekerja dengan cepat. Transformasi dari pengawasan dengan mata manusia menjadi pengawasan dengan mesin sudah menjadi tren dalam deteksi cacat permukaan. Beberapa penelitian tentang deteksi cacat permukaan baja telah dilakukan. Penelitian sebelumnya menggunakan dataset NEU-DET dan melakukan pengembangan dari model *baseline* yaitu Faster R-CNN menjadi model baru yang disebut dengan *Defect Inspection Network* (DIN) [3]. Pada penelitian berikutnya dilakukan dengan mengembangkan model *Single Shot Multibox Detector* (SSD) [4]. Dilakukan juga penelitian menggunakan metode *Multilevel-Feature Fusion Network* (MFN) yang mendapatkan nilai *mean average precision* sebesar 82,3% untuk deteksi cacat [5].

Pada penelitian ini, sistem dirancang untuk melakukan deteksi terhadap cacat permukaan baja berbasis *deep learning* dengan menggunakan metode *anchor-free*. Sedangkan manfaat dalam penelitian dapat merekomendasikan sistem dengan performansi yang terbaik dan diharapkan lebih tinggi presisinya dibandingkan dengan penelitian sebelumnya.

## 2. TINJAUAN PUSTAKA

### 2.1 Deteksi Objek

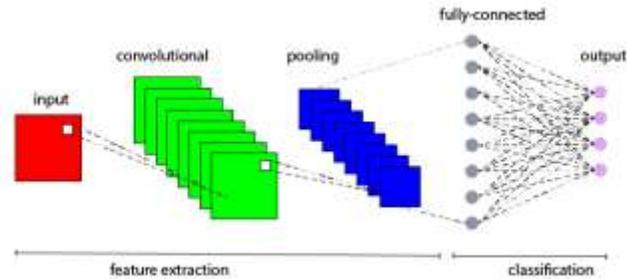
Deteksi objek merupakan bagian dari *computer vision* yang berhubungan dengan proses deteksi objek visual kelas tertentu dalam bentuk gambar seperti manusia, hewan dan mobil [6]. Deteksi objek menjadi perhatian dalam beberapa tahun terakhir, perkembangannya selama dua dekade terakhir dapat dilambangkan sebagai sejarah dari *computer vision*. Tujuan dari deteksi objek yaitu untuk mengembangkan model dan teknik komputasi dari suatu informasi yang dibutuhkan oleh komputer. Deteksi objek dapat dilihat sebagai kombinasi dari dua pekerjaan, yaitu lokalisasi objek atau dimana objek tersebut berada dan pengenalan visual atau bagaimana bentuk dari objek tersebut.

### 2.2 Anchor Free

Metode *anchor-free* telah banyak digunakan dalam dalam deteksi objek karena memiliki arsitektur sederhana dan kinerja yang baik [7]. Berbeda dengan metode *anchor-base* yang memperbaiki kandidat *anchor boxes* yang sudah di tentukan sebelumnya, metode *anchor-free* secara langsung memperkirakan posisi objek yang dideteksi. FCOS [8] melakukan klasifikasi setiap piksel di dalam *bounding box* serta meregresi jarak antar lokasi piksel dengan *bounding box*, model ini menghasilkan prediksi yang intensif untuk target. CenterNet [9] merupakan pengembangan dari CornerNet [10] yang menambahkan sebuah cabang untuk deteksi pusat dan sebagian besar meningkatkan kinerja dengan validasi titik pusat. CentripetalNet [11] memprediksi posisi dan pergeseran sentripetal dari titik sudut dan mencocokkan sudut yang memiliki hasil pergeseran sejajar. Pendekatan dengan pencocokan titik sudut dianggap lebih akurat daripada pendekatan lainnya. Pada penelitian ini Penulis menggunakan metode *anchor-free* yaitu FoveaBox [12]. Model ini mengusulkan sebuah objek fovea untuk penetapan target dan digunakan sebagai acuan untuk memprediksi batas objek (kiri, kanan, atas, bawah) pada setiap objek yang dideteksi.

### 2.3 Convolutional Neural Network

*Convolutional Neural Network* (CNN) merupakan salah satu metode dari *deep learning* yang paling populer untuk diuji oleh para peneliti [13]. CNN dikembangkan dari *Multilayer Perceptron* (MLP) yang fokus dirancang untuk mengolah data dua dimensi dan termasuk dalam jenis *deep neural network*. Secara umum, CNN diaplikasikan dalam bidang *computer vision*, misalnya: deteksi objek, klasifikasi objek dan lain-lain. CNN menggunakan operasi matematika yang biasa disebut dengan konvolusi. Memiliki arsitektur yang cara kerjanya menyerupai neuron atau saraf otak manusia [14]. Arsitektur CNN terdiri dari tiga *layer* utama, yaitu *convolutional layer*, *pooling layer*, dan *fully-connected layer* yang ditunjukkan pada Gambar 1. Contoh arsitektur CNN yang sering digunakan adalah VGGNet, MobileNet [15], GoogleNet dan ResNet.



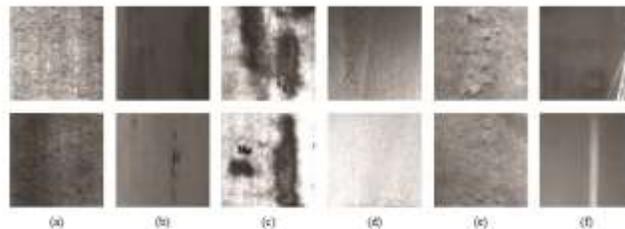
Gambar 1. Arsitektur Convolutional Neural Network

### 3. METODOLOGI PENELITIAN

Penelitian ini menggunakan PyTorch, yaitu sebuah kerangka kerja *deep learning* yang digunakan untuk membangun model. Seluruh percobaan penelitian dilakukan dengan *mmdetection toolbox* [16] dan diimplementasikan menggunakan Python 3.8.5. Penelitian ini menggunakan metode *anchor-free* yaitu FoveaBox.

#### 3.1 Dataset

Dataset yang diuji pada penelitian ini diambil dari dataset NEU-DET [17]. Dataset tersebut merupakan kumpulan data cacat permukaan baja yang dirilis oleh Northeastern University, di dalamnya terdapat enam cacat permukaan baja, yaitu *crazing* (Cr), *inclusion* (In), *patches* (Pa), *pitted surface* (Ps), *rolled-in scale* (Rs), dan *scratches* (Sc). Terdapat 300 gambar sampel untuk setiap cacat dan totalnya terdapat 1.800 gambar sampel *grayscale* dengan resolusi  $200 \times 200$  piksel. Untuk penggunaan deteksi objek, dataset menyediakan file anotasi *bounding box* yang disimpan dalam dokumen XML, dokumen tersebut menunjukkan kategori dan lokasi cacat pada setiap gambar. Contoh dari berbagai jenis cacat yang terdapat pada dataset NEU-DET ditunjukkan pada Gambar 2.



Gambar 2. Dataset Neu-Det: (A) Crazing, (B) Inclusion, (C) Patches, (D) Pitted Surface, (E) Rolled-In Scale, Dan (F) Scratches

#### a. Preprocessing

Sebelum proses pelatihan model, Penelitian ini membagi dataset NEU-DET menjadi data latih dan data uji. Rasionya adalah 8:2, masing-masing berisi 1.440 dan 360 gambar. Selain itu, juga digunakan operasi augmentasi data (*resize*, *random flip*, *normalize*) pada proses *training* untuk meningkatkan keberagaman data.

#### b. Arsitektur Model

Model deteksi yang digunakan pada penelitian deteksi cacat ini adalah FoveaBox dengan ResNet101 sebagai *backbone* dan megadopsi *Feature Pyramid Network* (FPN). FoveaBox merupakan sebuah model *Convolutional Neural Network* (CNN) yang menghasilkan *bounding box* untuk setiap objek yang ada pada gambar yang diberikan.

##### 1. ResNet 101

*Residual Network* (ResNet) merupakan sebuah jaringan *bottom-up* yang digunakan untuk mengekstraksi fitur dari gambar masukan [18]. Penulis menggunakan jaringan ResNet dengan 101 *layer*. Jaringan *bottom-up* tersebut memiliki satu tingkat piramida pada setiap tahap yang digunakan untuk mengekstrak peta fitur dari gambar masukan [19]. Peta fitur tersebut mengalami proses konvolusi  $1 \times 1$  untuk pengurangan dimensi saluran. Keluaran dari jalur *bottom-up* bertindak sebagai peta fitur referensi untuk jalur *top-down* dengan koneksi lateral.

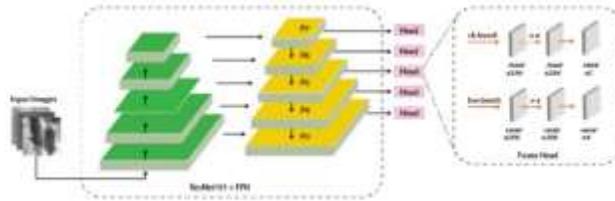
*Transfer learning* juga digunakan dalam penelitian ini. *Transfer learning* merupakan pendekatan yang dilakukan di mana model yang dilatih untuk suatu tugas tertentu dapat digunakan untuk menginisialisasi parameter model yang akan dilatih untuk tugas lain. Bobot ResNet 101 yang digunakan diinisialisasi ke bobot yang telah dilatih sebelumnya yang tersedia pada *torchvision*. Hal tersebut dapat meningkatkan akurasi dan menghemat waktu pelatihan model [19].

##### 2. Feature Pyramid Network

Secara umum *Feature Pyramid Network* (FPN) menggunakan arsitektur *top-down*. Setiap tingkat piramida FPN digunakan untuk mendeteksi objek dengan skala tertentu. Pada penelitian ini setiap tingkat piramida memiliki skala 32 - 512 yang dimulai dari P3, P4, P5, P6, dan P7.

### 3. FoveaBox

Dalam penelitian ini digunakan model jenis *anchor-free* untuk melakukan deteksi cacat, model tersebut adalah FoveaBox. Foveabox merupakan model *single-stage detection* yang akurat dan fleksibel dalam mendeteksi objek serta memiliki arsitektur sederhana [12]. Cara kerja dari model ini yaitu secara langsung memprediksi kemungkinan adanya suatu objek dan *boundary box* yang sesuai untuk setiap daerah positif pada setiap masukan. Keluaran dari *backbone* akan masuk ke dalam FoveaHead. FoveaHead memiliki dua subcabang, pertama digunakan untuk mengklasifikasikan daerah yang sesuai, dan yang kedua digunakan untuk memprediksi  $(x1, y1, x2, y2)$  dari *boundary box*. Arsitektur FoveaBox ditunjukkan pada Gambar 3.



Gambar 3. Arsitektur Foveabox

#### A. Performansi Sistem

Performansi sistem dapat diketahui dengan parameter *mean average precision*. *Mean average precision* (mAP) merupakan parameter yang digunakan untuk mengevaluasi kinerja model secara keseluruhan, yang merupakan nilai rata-rata dari *Average Precision* (AP) keseluruhan kelas [2], [5]. Secara persamaan matematis perhitungan mAP dapat dituliskan menjadi:

$$Recall = \frac{TP}{TP+FN} \quad (1)$$

$$Precision = \frac{TP}{TP+FP} \quad (2)$$

$$AP = \frac{Precision+Recall}{2} \quad (3)$$

$$mAP = \frac{\sum_{i=1}^K AP_i}{K} \quad (4)$$

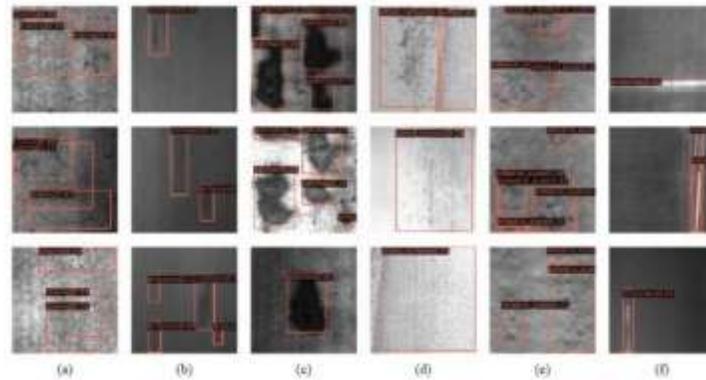
Dimana TP adalah *True Positive*, merupakan jumlah data positif yang terdeteksi benar oleh sistem. FN adalah *False Negative*, merupakan jumlah data negatif namun terdeteksi salah oleh sistem. FP adalah *False Positive*, merupakan jumlah data positif namun terdeteksi salah oleh sistem. Parameter K merupakan jumlah kategori objek.

## 4. HASIL DAN PEMBAHASAN

### 4.1 Pengujian Sistem

Dilakukan pengujian sistem dengan melihat pengaruh parameter *learning rate*, *optimizer*, sigma, dan jumlah *epoch* terhadap nilai *mean average precision* (mAP). Semakin tinggi nilai mAP maka dapat diartikan bahwa sistem bekerja dengan baik untuk mendeteksi objek. Hasil deteksi cacat permukaan baja ditunjukkan pada Gambar 4.

Selanjutnya dilakukan pengujian *learning rate* terhadap mAP. Pada skenario pertama, pengujian dilakukan pada parameter *learning rate* yang digunakan pada tahap *training*. Pengujian ini menggunakan tiga nilai *learning rate* berbeda yaitu 0,01, 0,001, dan 0,0001 yang dilakukan untuk mengetahui pengaruh *learning rate* terhadap performansi sistem yang ditunjukkan dengan nilai *mean average precision*. Hasil dari pengujian *learning rate* ditunjukkan pada Tabel 1.



Gambar 4. Hasil Deteksi Cacat: (A) Crazing, (B) Inclusion, (C) Patches, (D) Pitted Surface, (E) Rolled-In Scale, Dan (F) Scratches

Dari hasil pengujian pada Tabel 1 ditunjukkan bahwa learning rate 0,001 memiliki nilai *mean average precision* yang paling tinggi yaitu sebesar 75,1%. Dapat disimpulkan bahwa penggunaan learning rate yang optimal pada nilai 0,001.

**Tabel 1. Pengujian Learning Rate**

Learning Rate	Mean Average Precision
0,01	4,7%
0,001	<b>75,1%</b>
0,0001	69,2%

Selanjutnya dilakukan pengujian *optimizer* terhadap mAP. Setelah melakukan pengujian *learning rate*, skenario berikutnya yaitu melakukan pengujian pada parameter *optimizer*. Pada pengujian ini digunakan tiga jenis *optimizer* yaitu Adam, SGD, dan RMSprop. Berdasarkan hasil pengujian yang sudah dilakukan sebelumnya, maka pada tahap pengujian ini digunakan nilai *learning rate* 0,001. Hasil pengujian pengaruh *optimizer* terhadap nilai *mean average precision* ditunjukkan pada Tabel 2.

**Tabel 2. Pengujian Optimizer**

Optimizer	Mean Average Precision
Adam	37,5%
SGD	<b>75,0%</b>
RMSprop	- %

Dari Tabel 2, diketahui bahwa penggunaan *optimizer* SGD memiliki nilai *mean average precision* yang paling tinggi yaitu 75,0%. SGD merupakan salah satu *optimizer* yang paling sering digunakan dalam membangun suatu model karena memiliki kinerja yang baik, begitu pun dengan penerapannya pada penelitian ini. Berbanding terbalik dengan penggunaan *optimizer* RMSprop yang tidak cocok untuk model yang digunakan.

Langkah ketiga dilakuakn pengujian sigma terhadap mAP. Pada skenario ketiga, pengujian dilakukan untuk mengetahui pengaruh parameter sigma terhadap nilai *mean average precision*. Sigma merupakan sebuah parameter yang terdapat dalam model Foveabox. Parameter ini merupakan *scaling factor* yang mengontrol penentuan area positif dan area negatif dari sebuah citra dalam proses *training*. Pada tahap pengujian ini menggunakan *learning rate* 0,001 dan *optimizer* SGD. Digunakan lima nilai sigma yang berbeda yaitu 0,4, 0,5, 0,6, 0,7, dan 0,8. Hasil pengujian pengaruh parameter sigma terhadap *mean average precision* ditunjukkan pada Tabel 3.

**Tabel 3. Pengujian Sigma**

Sigma	Mean Average Precision
0,4	78,0%
0,5	79,3%
0,6	<b>80,1%</b>
0,7	79,1%
0,8	77,6%

Dari Tabel 3 diketahui bahwa penggunaan nilai sigma dari 0,4 sampai 0,8 Pada nilai sigma 0,6 memiliki nilai *mean average precision* yang paling tinggi yaitu 80,1%, sedangkan pada sig ma 0,8 memiliki nilai *mean average precision* yang paling rendah yaitu 77,6%

Langkah terakhir adalah pengujian jumlah *epoch* terhadap mAP. Pada skenario terakhir, pengujian dilakukan pada jumlah *epoch* yang digunakan pada tahap *training* model. Pengujian ini dilakukan untuk mengetahui pengaruh jumlah *epoch* terhadap performansi sistem yang ditunjukkan dengan nilai *mean average precision*. Parameter yang digunakan dalam pengujian ini yaitu *learning rate* 0,001, *optimizer* SGD, dan *sigma* 0,6. Tahap ini dilakukan dengan empat jumlah *epoch* yang berbeda yaitu 8, 16, 24, dan 32. Hasil perngujian jumlah *epoch* ditunjukkan pada Tabel 4.

Tabel 4. Pengujian Jumlah Epoch

Jumlah Epoch	Mean Average Precision
8	74,6%
16	80,0%
24	<b>83,4%</b>
32	83,3%

Dari Tabel 4, diketahui bahwa titik optimal *epoch* yang digunakan berjumlah 24 karena memiliki nilai *mean average precision* yang paling tinggi yaitu 83,4%. Penambahan jumlah *epoch* yang digunakan dalam proses *training* berpengaruh terhadap performansi suatu model dalam mendeteksi objek. Hal tersebut dapat ditunjukkan pada pengujian ini bahwa nilai *mean average precision* mengalami kenaikan dan mencapai titik optimal pada saat *epoch* berjumlah 24.

#### 4.2 Perbandingan Hasil Deteksi dengan Model Lain

Untuk memverifikasi kinerja model dalam mendeteksi cacat permukaan baja, model yang digunakan pada penelitian ini yaitu FoveaBox akan dibandingkan dengan model lainnya. Model lain yang menjadi pembanding adalah FCOS, CenterNet, dan CentripetalNet. Hasil perbandingan dengan model lain ditunjukkan pada Tabel 5.

Tabel 5. Perbandingan Hasil Dengan Model Lain

Model	AP (%)						mAP (%)
	Cr	In	Pa	Ps	Rs	Sc	
FCOS	53,1	88,5	88,3	<b>88,9</b>	70,4	86,6	79,3
CenterNet	40,4	77,7	88,2	86,6	64,4	78,9	72,7
CentripetalNet	31,0	83,1	83,7	68,3	48,7	72,0	64,5
FoveaBox	<b>63,6</b>	<b>90,0</b>	<b>92,4</b>	88,5	<b>76,6</b>	<b>89,4</b>	<b>83,4</b>

Dari Tabel 5 diketahui bahwa model yang digunakan pada penelitian ini lebih unggul dibandingkan dengan model *anchor-free* lainnya yaitu FCOS, CenterNet, dan CentripetalNet. FoveaBox memiliki nilai *average precision* (AP) yang lebih tinggi untuk setiap kelas, kecuali pada kelas *pitted surface* (Ps) yang terpaut 0,4% lebih rendah dibandingkan dengan model FCOS. Untuk nilai *mean average precision* (mAP), FoveaBox memiliki nilai yang paling tinggi mencapai 83,4%. Dengan demikian dapat disimpulkan bahwa model yang digunakan untuk mendeteksi cacat permukaan baja pada penelitian ini dapat bekerja dengan baik.

## 5. KESIMPULAN DAN SARAN

Berdasarkan hasil dan analisis terhadap pengujian pada penelitian yang dilakukan, perancangan sistem deteksi cacat permukaan baja dengan menggunakan metode *anchor-free* yaitu FoveaBox dapat berjalan dengan baik. Sistem mampu mendeteksi enam jenis cacat yaitu *crazing*, *inclusion*, *patches*, *pitted surface*, *rolled-in scale*, dan *scratches* dengan nilai *mean average precision* mencapai 83,4%. Hasil tersebut diperoleh dengan parameter *learning rate* 0,001, *optimizer* SGD, *sigma* 0,6 dan jumlah *epoch* 24. Nilai *mean average precision* sebesar 83,4% ini sedikit lebih tinggi dibandingkan dengan penelitian sebelumnya yaitu 82,3% [5].

Saran untuk penelitian selanjutnya, perlu dilakukan penelitian untuk meningkatkan presisi lebih lanjut dengan menggunakan pengembangan *optimizer* yang ada. Disamping itu juga perlu diteliti tentang waktu komputasi yang dibutuhkan oleh model dalam mendeteksi suatu objek.

## DAFTAR PUSTAKA

- [1] J. Xing dan M. Jia, "A convolutional neural network-based method for workpiece surface defect detection," *Measurement*, vol. 176, hal. 109185, 2021.
- [2] X. Chen, J. Lv, Y. Fang, dan S. Du, "Online Detection of Surface Defects Based on Improved YOLOV3," *Sensors*, vol. 22, no. 3, hal. 817, 2022.
- [3] R. Hao, B. Lu, Y. Cheng, X. Li, dan B. Huang, "A steel surface defect inspection approach towards smart industrial monitoring," *J. Intell. Manuf.*, 2020, doi: 10.1007/s10845-020-01670-2.
- [4] C. Y. Lin, C. H. Chen, C. Y. Yang, F. Akhyar, C. Y. Hsu, dan H. F. Ng, "Cascading Convolutional Neural Network for

- Steel Surface Defect Detection,” *Adv. Intell. Syst. Comput.*, vol. 965, hal. 202–212, 2020, doi: 10.1007/978-3-030-20454-9\_20.
- [5] Y. He, K. Song, Q. Meng, dan Y. Yan, “An End-to-End Steel Surface Defect Detection Approach via Fusing Multiple Hierarchical Features,” *IEEE Trans. Instrum. Meas.*, vol. 69, no. 4, hal. 1493–1504, 2020, doi: 10.1109/TIM.2019.2915404.
- [6] Z. Zou, Z. Shi, Y. Guo, dan J. Ye, “Object detection in 20 years: A survey,” *arXiv Prepr. arXiv1905.05055*, 2019.
- [7] H. Ma, S. T. Acton, dan Z. Lin, “CAT: Centerness-Aware Anchor-Free Tracker,” *Sensors (Basel)*, vol. 22, no. 1, hal. 354, Jan 2022, doi: 10.3390/s22010354.
- [8] Z. Tian, C. Shen, H. Chen, dan T. He, “Fcos: A simple and strong anchor-free object detector,” *IEEE Trans. Pattern Anal. Mach. Intell.*, 2020.
- [9] X. Zhou, D. Wang, dan P. Krähenbühl, “Objects as points,” *arXiv Prepr. arXiv1904.07850*, 2019.
- [10] H. Law dan J. Deng, “Cornernet: Detecting objects as paired keypoints,” in *Proceedings of the European conference on computer vision (ECCV)*, 2018, hal. 734–750.
- [11] Z. Dong, G. Li, Y. Liao, F. Wang, P. Ren, dan C. Qian, “CentripetalNet: Pursuing High-quality Keypoint Pairs for Object Detection,” Mar 2020, Diakses: Apr 08, 2022. [Daring]. Tersedia pada: <http://arxiv.org/abs/2003.09119>.
- [12] T. Kong, F. Sun, H. Liu, Y. Jiang, L. Li, dan J. Shi, “FoveaBox : Beyond Anchor-Based Object Detection,” vol. 29, hal. 7389–7398, 2020.
- [13] J. Gu dkk., “Recent advances in convolutional neural networks,” *Pattern Recognit.*, vol. 77, hal. 354–377, 2018.
- [14] Z. Q. Zhao, P. Zheng, S. T. Xu, dan X. Wu, “Object Detection with Deep Learning: A Review,” *IEEE Trans. Neural Networks Learn. Syst.*, vol. 30, no. 11, hal. 3212–3232, 2019, doi: 10.1109/TNNLS.2018.2876865.
- [15] H. Andrew dkk., “Searching for mobilenetv3,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, hal. 1314–1324.
- [16] K. Chen dkk., “MMDetection: Open mmlab detection toolbox and benchmark,” *arXiv Prepr. arXiv1906.07155*, 2019.
- [17] K. Song dan Y. Yan, “A noise robust method based on completed local binary patterns for hot-rolled steel strip surface defects,” *Appl. Surf. Sci.*, vol. 285, hal. 858–864, 2013, doi: <https://doi.org/10.1016/j.apsusc.2013.09.002>.
- [18] S. Gonzalez, C. Arellano, dan J. E. Tapia, “Deepblueberry: Quantification of blueberries in the wild using instance segmentation,” *Ieee Access*, vol. 7, hal. 105776–105788, 2019.
- [19] P. Malhotra, S. Gupta, D. Koundal, A. Zaguia, M. Kaur, dan H.-N. Lee, “Deep Learning-Based Computer-Aided Pneumothorax Detection Using Chest X-ray Images,” *Sensors (Basel)*, vol. 22, no. 6, hal. 2278, Mar 2022, doi: 10.3390/s22062278.