

Inspeksi Kualitas Pengelasan Besi Menggunakan Teknik Segmentasi Citra Berbasis *Convolutional Neural Network*

Wahyono¹, Andi Dharmawan², Lukman Awaludin^{*3}, Oskar Natan⁴, Baskara⁵

^{1,2,3,4} Departemen Ilmu Komputer dan Elektronika, Universitas Gadjah Mada Sleman, Yogyakarta 55281, Indonesia

^{4,5} Department of Computer Science and Engineering, Toyohashi University of Technology Toyohashi, Aichi 441-8580, Japan

e-mail: ¹wahyo@ugm.ac.id, ²andi_dharmawan@ugm.ac.id, ^{*3}lukman.awaludin@ugm.ac.id, ⁴oskarnatan@ugm.ac.id, ⁵baskara.zv@tut.jp

Abstrak

Inspeksi pengelasan merupakan kebutuhan mutlak bagi dunia industri terutama yang bergerak dibidang otomotif untuk memastikan kualitas las. Namun demikian, sebagian besar industri masih menggunakan pemeriksaan manual yang bersifat subjektif dan penuh dengan bias yang dapat berakibat pada inkonsistensi dalam penilaian standar kualitas. Oleh karena itu, diperlukan suatu sistem cerdas yang dapat memeriksa kualitas pengelasan dengan konsisten. Penelitian ini bertujuan untuk membuat model kecerdasan buatan berbasis deep learning dan computer vision untuk mendeteksi area-area pengelasan dan mengklasifikasikannya kedalam kategori baik dan buruk. Model CNN dengan arsitektur UNet diadopsi untuk melakukan segmentasi citra pada gambar pengelasan besi. Studi penggunaan beberapa teknik ekstraksi fitur juga dilakukan untuk mendapatkan performa model terbaik berdasarkan skor IoU dan kecepatan konvergensi model. Berdasarkan hasil eksperimen, teknik CNN UNet terbukti mampu meningkatkan performa model dengan skor IoU sebesar 78,1% dan dengan kecepatan konvergensi dalam 144 epoch.

Kata kunci— Kualitas pengelasan, Segmentasi citra, Deep learning, Ekstraksi fitur.

Abstract

Welding inspection is an absolute necessity for the industrial world, especially those engaged in the automotive sector to ensure weld quality. However, most industries still use manual inspection which is subjective and full of bias which can result in inconsistencies in the assessment of quality standards. Therefore, intelligent system that can check the quality of welding consistently is needed. This study aims to create an artificial intelligence model based on deep learning and computer vision to detect welding spots and classify them into good and bad categories. CNN model with UNet architecture is adopted to perform image segmentation on iron welding images. Studies using several feature extraction techniques are also conducted to obtain the best model performance based on IoU scores and model convergence speed. Based on the experimental results, the UNet technique is proven to be able to improve the performance of the model with an IoU score of 78.1% and with a convergence speed of 144 epochs.

Keywords— Welding quality, Image segmentation, Deep learning, Feature extraction.

1. PENDAHULUAN

Inspeksi kualitas pengelasan merupakan kebutuhan yang sangat penting bagi dunia industri terutama yang bergerak dibidang otomotif untuk memastikan kualitas las. Inpeksi dilakukan dengan cara mendeteksi sebuah area pengelasan untuk kemudian diklasifikasikan kualitasnya apakah masuk ke dalam kelas baik atau buruk. Namun demikian, sebagian besar industri masih menggunakan pemeriksaan manual yang bersifat subjektif dan penuh dengan bias yang dapat berakibat pada inkonsistensi dalam penilaian standar kualitas. Ini dikarenakan pengetahuan dari inspektor yang berbeda satu sama lain. Selain itu, meskipun inspektor memiliki pengetahuan yang baik dalam penilaian kualitas pengelasan, pada suatu waktu akan mengalami penurunan konsenstrasi dalam penilaian yang bisa menyebabkan penilaian jadi lebih subjektif. Oleh sebab itu dibutuhkan sebuah sistem yang bisa digunakan untuk melakukan inpeksi kualitas pengelasan besi secara otomatis dan terus menerus. Dari beberapa teknik yang mungkin dilakukan, inpeksi pengelasan besi berbasis camera membutuhkan biaya yang lebih murah dibandingkan strategi lainnya. Sehingga sistem yang dikembangkan berbasis kepada teknik pengolahan dan analisis citra yang dihasilkan dari kamera.

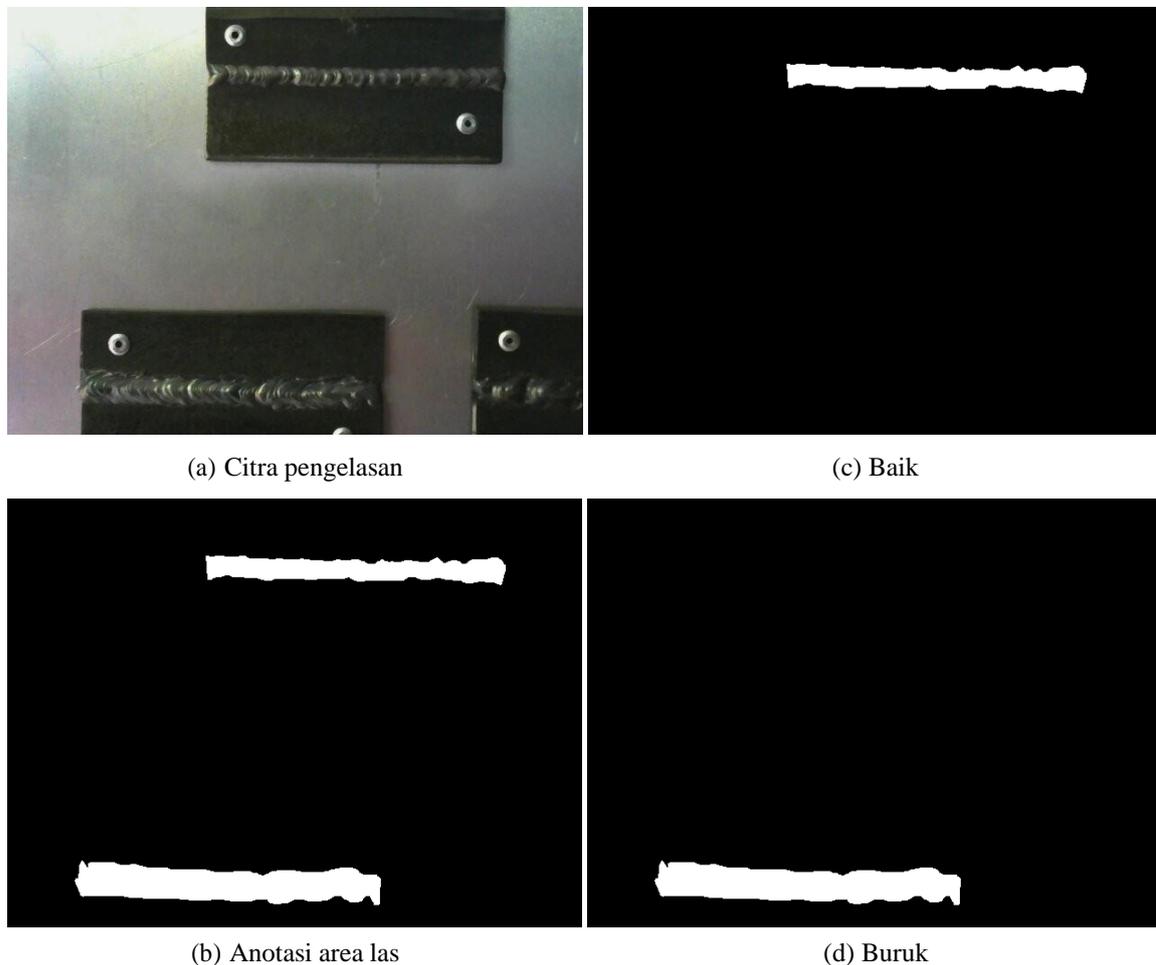
Dalam bidang pengolahan citra digital, segmentasi citra merupakan salah satu tahap penting yang biasanya dilakukan untuk menganalisis sebuah citra. Segmentasi citra bekerja dengan cara mengekstrasi area objek dari sebuah citra untuk kemudian dianalisis dengan teknik tertentu seperti klasifikasi, pengenalan, maupun identifikasi. Salah satu penerapan segmentasi citra adalah untuk melakukan inpeksi kualitas pada hasil pengelasan besi. Oleh sebab itu, dalam penelitian ini kami mencoba untuk mengusulkan sebuah metode segmentasi citra untuk melakukan inspeksi kualitas pengelasan besi berbasis pada data kamera. Setidaknya ada dua pendekatan yang mungkin dilakukan untuk menyelesaikan permasalahan ini yaitu segmentasi berbasis fitur warna dan segmentasi berbasis *deep learning*. Segmentasi berbasis fitur warna memiliki keunggulan yang lebih sederhana dalam prosesnya karena hanya memanfaatkan *threshold* warna untuk menentukan apakah sebuah piksel merupakan area pengelasan. Namun, metode ini sangat rentan terhadap perubahan pencahayaan saat pengambilan citra pengelasan sehingga akan menghasilkan akurasi yang buruk.

Pada penelitian ini, metode yang diusulkan berbasis *convolutional neural network* khususnya pada arsitektur UNet karena memiliki akurasi yang paling baik dalam kasus segmentasi citra pada aplikasi berbasis citra lainnya, misalnya pada pendeteksian objek, pengenalan objek, identifikasi objek dan lain-lain. Sehingga harapannya sistem yang dikembangkan menghasilkan akurasi yang baik dalam evaluasi kualitas pengelasan besi. Adapun kebaruan penelitian ini terangkum kedalam tiga poin berikut:

- Penelitian ini mengadopsi arsitektur UNet[1] dengan modifikasi[2] untuk melakukan segmentasi citra guna mendeteksi area-area pengelasan besi serta mengklasifikasikannya dalam kategori baik dan buruk.
- Penelitian ini melakukan studi komparasi metode data *preprocessing* yang meliputi RGB, LBP, *grayscale* dan citra gradient untuk menemukan teknik terbaik dalam mengekstraksi fitur citra guna meningkatkan performa segmentasi dari model UNet.
- Penelitian ini membuat dataset pengelasan sendiri dengan mengumpulkan citra-citra hasil pengelasan besi dan memberikan anotasi terhadap hasil pengelasan yang baik dan buruk. Kedepannya, dataset ini dapat digunakan untuk penelitian-penelitian dimasa mendatang.

Secara detail, konten manuskrip ini dibagi kedalam beberapa bab sebagai berikut. Pada

bab metodologi, dijelaskan metodologi penelitian yang meliputi informasi dataset, teknik *preprocessing*, arsitektur deep learning, serta konfigurasi eksperimen. Pada bab hasil dan pembahasan, dijelaskan hasil percobaan beserta pembahasan pada performa dari setiap teknik ekstraksi fitur. Terakhir, kesimpulan dan saran dari makalah ini disajikan pada bab kesimpulan dan saran sebagai pernyataan penutup atas hal yang ditemukan dalam penelitian.



Gambar 1: Sampel citra pengelasan beserta anotasinya.

2. METODE PENELITIAN

Seperti yang telah disebutkan pada Bab 1, penelitian ini bertujuan untuk mengembangkan model *deep learning* dengan arsitektur UNet untuk mendeteksi area-area pengelasan serta mengklasifikasikannya kedalam kategori baik dan buruk dengan melakukan segmentasi pada citra yang ditangkap oleh kamera. Dalam bab ini, teknik pengambilan data, data *preprocessing*, arsitektur model, dan konfigurasi training akan dijelaskan secara terperinci.

2.1 Dataset

Dataset diperlukan dalam paradigma *supervised learning*. Pada penelitian ini, beberapa citra hasil pengelasan besi yang diakuisisi dari bengkel pengelasan di daerah Jejeran, Wonokromo, Pleret, Bantul, D.I. Yogyakarta. Setiap citra diambil dengan resolusi tinggi dan lebar sebesar $H \times W = 640 \times 480$. Terdapat total citra pengelasan sebanyak 117 sampel. Setiap citra pengelasan kemudian diubah ukurannya menjadi $H \times W = 256 \times 256$ untuk menghemat biaya dan waktu komputasi.

Tahap selanjutnya dalam pembuatan dataset citra pengelasan besi adalah memberikan anotasi pada setiap area las pada setiap sampel citra. Anotasi ini nantinya akan digunakan sebagai *ground truth* pada proses *training*, sehingga model dapat mengetahui posisi area pengelasan beserta kategori kualitasnya. Dalam penelitian ini, aplikasi berbasis web LabelBox digunakan untuk memberikan anotasi poligon untuk menentukan luasan area las pada citra. Dari 117 sampel citra, terdapat total 136 area pengelasan dengan kategori 93 buruk dan 43 baik yang telah dianotasi beserta kategori klasifikasinya. Contoh beberapa citra beserta anotasi yang digunakan pada penelitian ini dapat dilihat pada Gambar 1.

Selanjutnya, seluruh anotasi ini diubah menjadi *ground truth mask* sehingga setiap sampel citra pengelasan memiliki *ground truth mask* $\mathbb{R}^{2 \times 256 \times 256}$, dimana 256×256 adalah dimensi $H \times W$ yang sama dengan data input citra, sedangkan angka 2 adalah jumlah *channel* (C) sesuai dengan jumlah kategori pengelasan yaitu baik dan buruk. Sementara itu, $\mathbb{R} \in \{0,1\}$ dimana suatu piksel bernilai 0 bila piksel tersebut tidak termasuk dalam area pengelasan dan bernilai 1 jika merupakan area pengelasan. Dengan mekanisme *ground truth* seperti ini, fungsi aktivasi sigmoid dapat diaplikasikan. Pembahasan lebih detail terkait dengan arsitektur model akan dibahas pada subbab 2.2. Adapun visualisasi *ground truth mask* baik dan buruk masing-masing dapat dilihat pada Gambar 1c dan Gambar 1d. Daerah berwarna hitam merupakan area yang tidak termasuk area pengelasan, sementara daerah putih merupakan area pengelasan untuk setiap kategori pengelasan (baik atau buruk) yang masing-masing menempati setiap *channel* dari *ground truth mask*.

2.1.1 Data Preprocessing

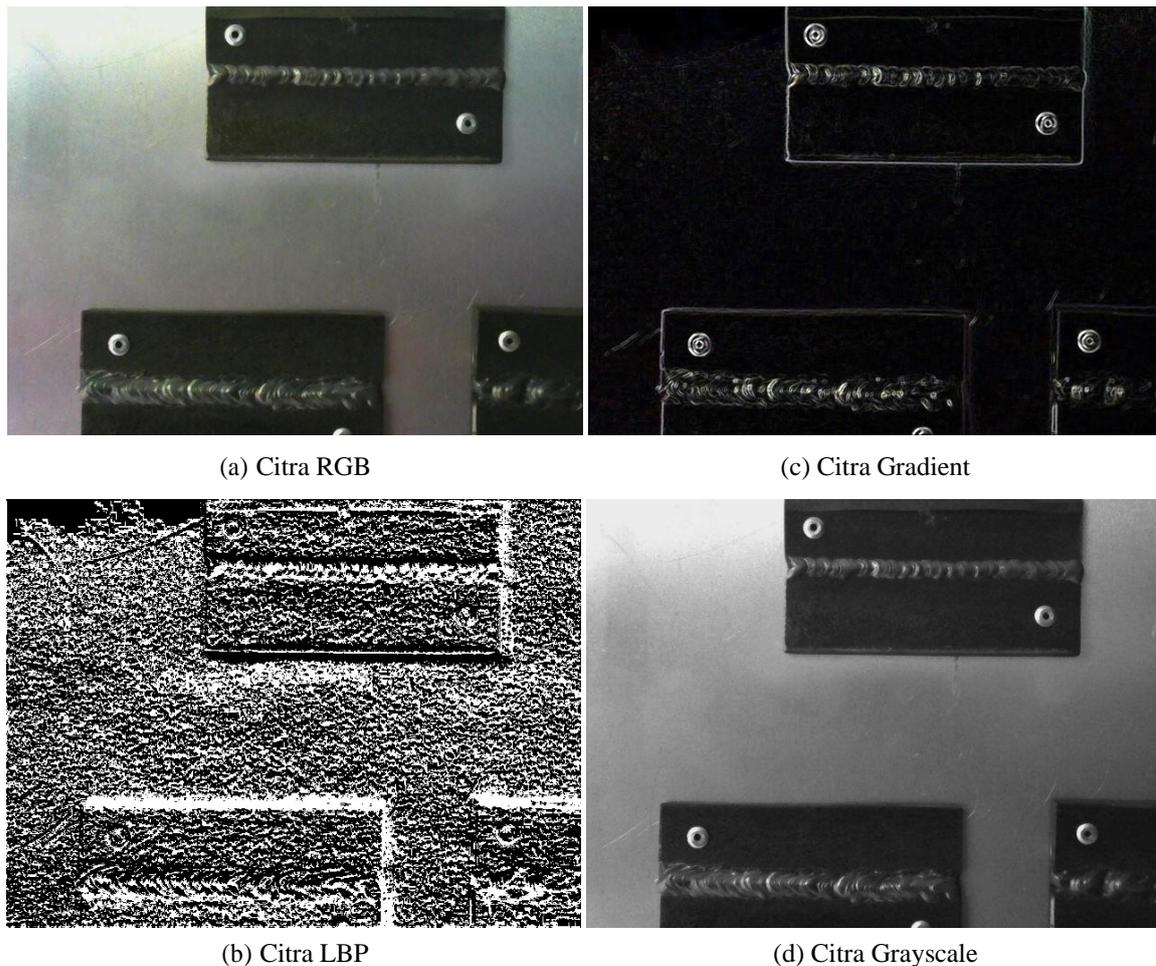
Pada penelitian ini, dilakukan studi komparasi dari beberapa teknik *preprocessing* untuk mendapatkan performa model terbaik. Terdapat empat jenis citra yang digunakan dalam penelitian ini yaitu:

1. Citra RGB: Pada citra RGB pada setiap piksel terdapat 3 channel merepresentasikan level intensitas warna *red*, *green*, dan *blue*. Contoh citra RGB dari dataset dapat dilihat pada Gambar 2a.
2. Citra *Grayscale*: Pada citra *grayscale* setiap piksel hanya terdapat 1 channel yang merepresentasikan level intensitas cahaya dari putih hingga hitam. Pada penelitian ini dilakukan konversi dari citra RGB untuk mendapatkan citra *grayscale* dengan menggunakan Persamaan (1)

$$Y = 0,299R + 0,587G + 0,114B \quad (1)$$

dimana Y merupakan nilai *grayscale*, R merupakan nilai intensitas *channel red*, G merupakan nilai intensitas *channel green*, dan B merupakan nilai intensitas *channel blue*. Contoh citra *grayscale* dari dataset dapat dilihat pada Gambar 2d.

3. Citra Gradient: Dilakukan ekstraksi fitur dari citra untuk mendapatkan nilai gradient dari citra pada arah x dan y . Citra gradient didapatkan dengan mengaplikasikan operator Sobel[3] terhadap citra *grayscale*. Contoh citra gradient dari dataset dapat dilihat pada Gambar 2c.
4. Citra LBP: Local Binary Pattern (LBP)[4] merupakan hasil ekstraksi fitur yang merepresentasikan pola tekstur dari citra. Tekstur didapatkan dengan membandingkan nilai piksel dengan piksel lain disekitarnya. Contoh citra LBP dari dataset dapat dilihat pada Gambar 2b.



Gambar 2: Sampel citra pengelasan setelah proses *preprocessing*.

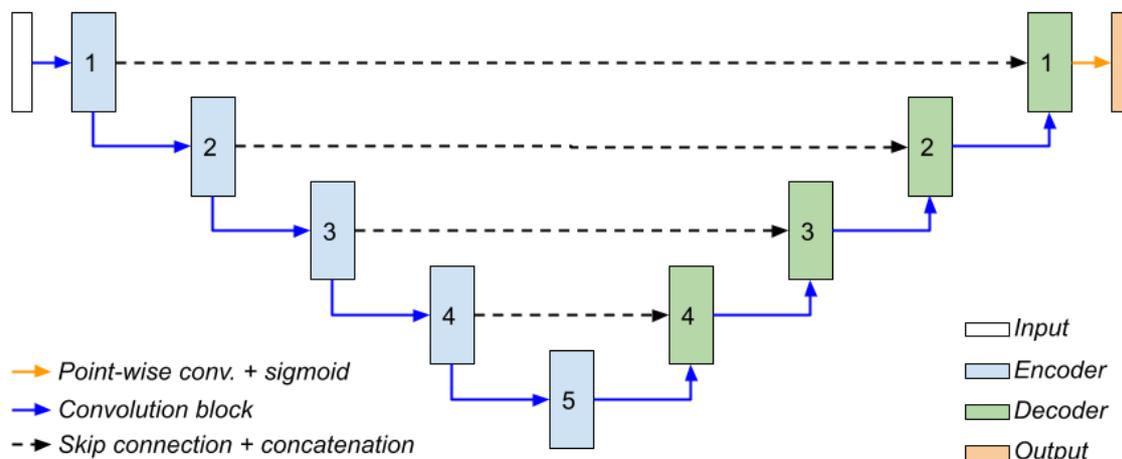
2.2 Arsitektur Deep Learning

Pada penelitian ini, model *deep learning* bernama UNet[1] diadopsi dengan modifikasi arsitektur dan tuning hyperparameter seperti yang dilakukan oleh Oskar dkk[2]. Secara garis besar, arsitektur UNet memiliki dua bagian utama yaitu *encoder* dan *decoder* yang dihubungkan dengan *skip connection* dan *concatenation* pada setiap *convolutional block*-nya. Bagian *encoder* berfungsi untuk untuk mengekstrak *discriminative feature*, sedangkan *decoder* digunakan untuk

Title of manuscript is short and clear, implies research results (First Author)

merekonstruksi citra pengelasan dengan hasil prediksi area pengelasan beserta klasifikasi kategorinya. Adapun ilustrasi dari arsitektur UNet ini dapat dilihat pada Gambar 3.

Setiap garis biru dengan tanda panah pada Gambar 3 adalah representasi dari *mini VGG convolution block*[5] yang terdiri dari dua layer konvolusi dengan *kernel* 3×3 , *batch normalization*[6], dan fungsi aktivasi ReLU[7]. Dalam arsitektur ini, *batch normalization* diterapkan pada setiap *convolution block* guna membuat proses training model menjadi lebih cepat dan stabil[6]. Kemudian, pada setiap *convolution block* di sisi *encoder* dilengkapi dengan *layer max pooling* dengan *kernel* 2×2 dan *stride* (2,2) untuk mengurangi dimensi spasial dari *feature maps* sebanyak setengahnya secara bertahap. Sedangkan pada setiap *convolution block* di sisi *decoder* dilengkapi dengan *layer bilinear upsampling* dengan *kernel* 2×2 dan *stride* (2,2) untuk menggandakan ukuran dimensi spasial *feature maps* sehingga ukuran dimensi spasial pada output sama dengan inputnya.



Gambar 3: Arsitektur UNet

Tabel 1: Detail ukuran setiap *feature maps* pada arsitektur UNet

No.	Dimensi ($C \times H \times W$)	Keterangan
Input	$(3 \times 256 \times 256)$ $(1 \times 256 \times 256)$ $(1 \times 256 \times 256)$ $(1 \times 256 \times 256)$	RGB Grayscale Citra Gradient LBP
1	$(16 \times 256 \times 256)$	
2	$(32 \times 128 \times 128)$	
3	$(64 \times 64 \times 64)$	
4	$(128 \times 32 \times 32)$	
5	$(256 \times 16 \times 16)$	
Output	$(2 \times 128 \times 128)$	Baik dan Buruk

Sementara itu, garis putus-putus berwarna hitam merupakan representasi dari proses *skip connection* dengan *concatenation* untuk menggabungkan *feature maps* dari *encoder* dan *decoder*. Selanjutnya, pada akhir arsitektur terdapat *point-wise convolution layer* dengan *kernel* 1×1 dan *stride* (1,1) serta diaktivasi oleh fungsi sigmoid yang bertugas untuk melakukan prediksi akhir.

Jumlah *channel* keluaran pada layer ini adalah sama dengan jumlah kelas pada dataset yaitu 2. Detail setiap ukuran *input* dan *output* pada setiap *convolution block* ditunjukkan pada Tabel 1.

2.3 Formulasi Fungsi Loss dan Metric

Dalam penelitian ini, gabungan *binary cross-entropy* (BCE)[8] dan *dice loss*[9] digunakan sebagai fungsi *loss* yang selanjutnya digunakan untuk mengkomputasi gradient dan update bobot model pada proses *backpropagation*. Sementara itu pada fungsi *metric*, *Jaccard Index* atau *Intersect over Union* (IoU) digunakan untuk menjustifikasi seberapa baik performa model pada setiap *epoch*-nya. Secara matematis, fungsi *loss* BCE dapat dituliskan seperti pada persamaan (2).

$$\mathcal{L}_{BCE} = \frac{1}{N} \sum_{i=1}^N y_i \times \log(\hat{y}_i) + (1 - y_i) \times \log(1 - \hat{y}_i) \quad (2)$$

dimana N adalah jumlah neuron pada layer output yaitu $2 \times 128 \times 128$, y_i dan \hat{y}_i adalah nilai *ground truth* dan nilai prediksi untuk piksel ke- i . Sementara itu, fungsi *dice loss* dapat dilihat pada persamaan (3).

$$\mathcal{L}_{Dice} = 1 - \frac{2|\hat{y} \cap y|}{|\hat{y}| + |y|} \quad (3)$$

Selanjutnya, total *loss* dapat dihitung dengan menjumlahkan kedua persamaan (2) dan persamaan (3) sebagai $\mathcal{L}_{BCEDice}$ dilihat pada persamaan (4).

$$\mathcal{L}_{BCEDice} = \alpha \mathcal{L}_{BCE} + \beta \mathcal{L}_{Dice} \quad (4)$$

dimana α dan β bernilai 1 sehingga setiap fungsi *loss* memiliki pengaruh yang sama. Dalam penelitian ini, *weight decay* w_d dengan nilai 0,0001 juga digunakan sebagai kompensasi untuk kompleksitas model dengan tujuan untuk menghindari *overfitting* dan meningkatkan generalisasi dari model itu sendiri[10]. Dengan demikian, fungsi *loss* akhir dapat dihitung dengan persamaan (5).

$$\mathcal{L} = \mathcal{L}_{BCEDice} + w_d \sum W \quad (5)$$

dimana $\sum W$ adalah jumlah total bobot model secara keseluruhan. Selanjutnya, fungsi *metric* yang digunakan dalam penelitian ini adalah IoU sebagaimana dapat dilihat pada persamaan (6).

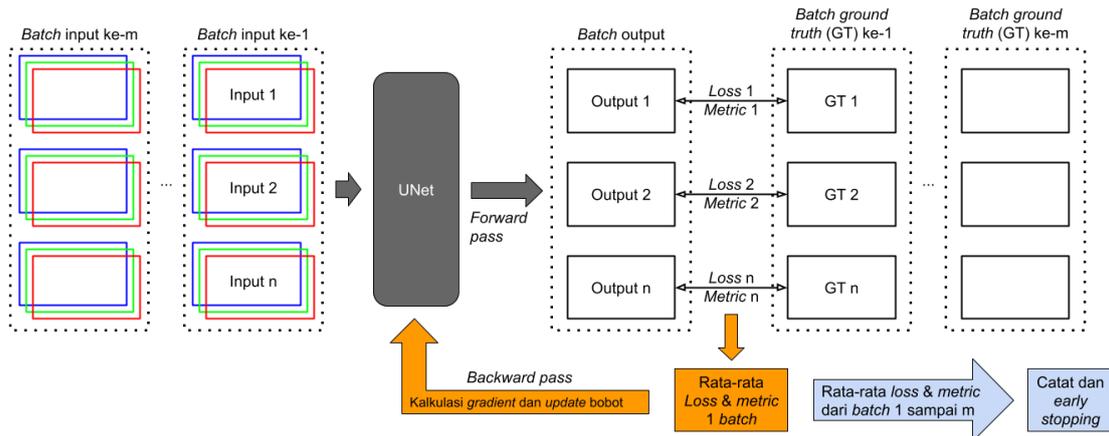
$$IoU = \frac{|\hat{y} \cap y|}{|\hat{y} \cup y|} \quad (6)$$

2.4 Konfigurasi Training

Proses *training* dilakukan dalam *virtual machine* yang disediakan oleh *Google Cloud Platform* dengan spesifikasi 1 buah GPU Tesla T4. Selanjutnya, dataset dibagi menjadi 2 bagian masing-masing untuk *training* dan validasi dengan perbandingan 80:20. Sehingga dari total 117 sampel citra akan diperoleh 94 sampel untuk data *training* dan 23 sampel untuk data validasi. Sedangkan ukuran *batch* diatur menjadi 8 sehingga dalam 1 *epoch* terdapat sebanyak 12 *training steps* dan 3 *validation steps*.

Optimizer stochastic gradient descent (SGD)[11] dengan momentum 0,9 digunakan untuk proses *training* model. Selama proses *training*, *learning rate* mula-mula diinisialisasi

dengan nilai 0,1 dan dikurangi setengahnya secara bertahap apabila tidak ada penurunan *validation loss* dalam kurun 15 *epoch* secara berturut-turut. Adapun *learning rate* minimum untuk proses *training* ini adalah 0,000025. Model akan terus *training* tanpa henti, akan tetapi proses tersebut akan dihentikan secara otomatis (*early stopping*) jika tidak ada kenaikan *validation IoU* dalam 50 *epoch* secara berturut-turut. Ilustrasi mekanisme *training* dengan skema *batch processing* dapat dilihat pada Gambar 4. Variabel n adalah jumlah sampel citra dalam 1 *batch* yang diset bernilai 8, sementara variabel m adalah jumlah *batch* yang diproses dalam 1 *epoch*. Variabel m dapat dihitung dengan membagi jumlah data pada *training set* atau *validation set* dengan *batch size*.



Gambar 4: Batch Processing

Pada fase *training*, satu *batch* yang terdiri dari 8 sampel citra diproses ke model untuk mendapatkan hasil prediksi untuk setiap sampel. Kemudian, hasil prediksi ini dibandingkan dengan setiap *ground truth* dan dihitung skor *metric* dan *loss*-nya. Kemudian, rata-rata nilai dalam satu *batch* juga dihitung. Rata-rata *loss* dalam 1 *batch* inilah yang digunakan untuk memperbarui bobot model selama proses *backpropagation*. Proses ini diulangi sampai semua *batch* telah diproses. Setelah menghitung rata-rata dari semua *batch loss* dan *batch IoU*, maka rata-rata dari *loss* dan *IoU* dalam 1 *epoch* dapat dihitung dan selanjutnya dicatat untuk memantau performa model selama fase *training*. Perlu dicatat bahwa proses tersebut juga berlaku untuk fase validasi, akan tetapi tidak ada proses update bobot dalam fase validasi. Proses dalam fase *training* dan validasi akan terus berlanjut dan hanya berhenti bila tidak ada kenaikan *validation IoU* dalam 50 *epoch* terakhir. Seperti yang disebutkan sebelumnya, rata-rata dari *loss* dan nilai *IoU* pada setiap *epoch* juga digunakan untuk menentukan kapan *learning rate* harus dikurangi.

3. HASIL DAN PEMBAHASAN

Dalam bab ini dibahas bagaimana pengaruh *preprocessing* terhadap performa model dalam melakukan segmentasi citra untuk mengenali area pengelasan dan mengklasifikasinya kedalam kategori baik dan buruk. Secara khusus, pembahasan dibagi menjadi dua bagian yaitu bagaimana masing-masing metode *preprocessing* berpengaruh terhadap performa model dan

kecepatan konvergensi model. Adapun tolak ukur yang dipakai untuk menjustifikasi performa model adalah nilai *validation IoU* seperti dirumuskan pada persamaan (6).

3.1 Pengaruh Terhadap Performa Segmentasi Model

Berikut hasil dari pengaruh berbagai macam *preprocessing* terhadap performa segmentasi model dengan memvariasikan bentuk citra sebagai inputan yaitu Citra RGB, Citra *Grayscale*, Citra Gradient, dan Citra LBP (perhatikan pada Gambar 2). Dengan bermacam bentuk citra masukan yang ada, penelitian ini mempunyai hipotesis bahwadengan citra LBP akan diperoleh hasil yang terbaik dengan inspeksi visual fokus pada objek *interest*-nya. Namun, setelah dilakukan percobaan, diperoleh hasil yang berbeda dari hipotesis seperti berikut ini.

3.1.1 Citra RGB Sebagai Input

Pada percobaan ini digunakan citra RGB sebagai input model tanpa adanya *preprocessing*. Beberapa variasi dari *batch size* dicoba dan didapatkan nilai *validation IoU* terbaik 0,793 pada *batch size* berukuran 32. Hasil dari percobaan ini dapat dilihat pada Tabel 2. Dari tabel dihasilkan bahwa semakin besar ukuran *batch size* maka akurasi model semakin baik dan sebaliknya.

Tabel 2: Hasil Training Dengan Input Citra RGB

<i>Batch Size</i>	<i>Epoch</i>	<i>Train Loss</i>	<i>Train IoU</i>	<i>Validation Loss</i>	<i>Validation IoU</i>
4	137	0,132	0,840	0,246	0,718
8	171	0,106	0,873	0,197	0,784
16	126	0,098	0,886	0,233	0,772
32	86	0,097	0,930	0,189	0,793

3.1.2 Citra Grayscale Sebagai Input

Pada percobaan ini digunakan citra *grayscale* hasil konversi dari citra RGB sebagai input model. Beberapa variasi dari *batch size* dicoba dan didapatkan skor *validation IoU* terbaik 0,756 pada *batch size* berukuran 32. Hasil dari percobaan ini dapat dilihat pada Tabel 3. Dari tabel dihasilkan bahwa semakin besar ukuran *batch size* maka akurasi model semakin baik.

Tabel 3: Hasil Training Dengan Input Citra *Grayscale*

<i>Batch Size</i>	<i>Epoch</i>	<i>Train Loss</i>	<i>Train IoU</i>	<i>Validation Loss</i>	<i>Validation IoU</i>
4	179	0,045	0,942	0,240	0,723
8	77	0,375	0,499	0,251	0,721
16	252	0,074	0,911	0,246	0,733
32	94	0,132	0,905	0,212	0,756

3.1.3 Citra Gradient Sebagai Input

Pada percobaan ini digunakan citra gradient sobel hasil konversi dari citra RGB sebagai input model. Beberapa variasi dari *batch size* dicoba dan didapatkan skor *validation IoU* terbaik 0,754

Title of manuscript is short and clear, implies research results (First Author)

pada *batch size* berukuran 32. Hasil dari percobaan ini dapat dilihat pada Tabel 4. Dari tabel dihasilkan bahwa semakin besar ukuran *batch size* maka akurasi model semakin baik.

Tabel 4: Hasil Training Dengan Input Citra Gradient

<i>Batch Size</i>	<i>Epoch</i>	<i>Train Loss</i>	<i>Train IoU</i>	<i>Validation Loss</i>	<i>Validation IoU</i>
4	174	0,046	0,945	0,296	0,646
8	173	0,123	0,916	0,269	0,695
16	189	0,151	0,826	0,270	0,668
32	180	0,107	0,871	0,231	0,754

3.1.1 Citra LBP Sebagai Input

Pada percobaan ini digunakan citra LBP Sobel hasil konversi dari citra RGB sebagai input model. Beberapa variasi dari *batch size* dicoba dan didapatkan skor *validation IoU* terbaik 0,730 pada *batch size* berukuran 16. Hasil dari percobaan ini dapat dilihat pada Tabel 5. Dari tabel dihasilkan bahwa semakin besar ukuran *batch size* maka akurasi model semakin baik.

Tabel 5: Hasil Training Dengan Input Citra LBP

<i>Batch Size</i>	<i>Epoch</i>	<i>Train Loss</i>	<i>Train IoU</i>	<i>Validation Loss</i>	<i>Validation IoU</i>
4	204	0,072	0,908	0,263	0,719
8	155	0,139	0,848	0,279	0,693
16	124	0,140	0,841	0,254	0,730
32	81	0,307	0,648	0,330	0,727

3.2 Pengaruh Preprocessing Terhadap Kecepatan Konvergensi

Berdasarkan hasil dari subbab 3.1 didapatkan perbandingan kecepatan konvergensi dari model dengan performa terbaik untuk tiap jenis *preprocessing*. Kecepatan konvergensi ditunjukkan pada Tabel 6. Seperti terlihat pada tabel didapatkan bahwa waktu *training* paling cepat ketika *grayscale* digunakan, tetapi apabila dilihat dari waktu konvergensinya, RGB memerlukan waktu *training* yang paling cepat dibandingkan Gradient dan LBP. Ini dikarenakan citra RGB memiliki jumlah *channel* yang lebih banyak dibandingkan citra lainnya.

Tabel 6: Jumlah *Epoch Early Stopping* Tiap Model

<i>Preprocessing</i>	<i>Early Stopping Epoch</i>	Waktu <i>Training</i>
RGB	136	10 menit, 3 detik
Grayscale	144	9 menit, 54 detik
Gradient	230	15 menit, 45 detik
LBP	174	11 menit, 36 detik

4. KESIMPULAN

Pada penelitian ini, model *deep learning* UNet digunakan untuk melakukan segmentasi data citra pengelasan kemudian mengklasifikasikannya kedalam dua kelompok yaitu pengelasan yang baik dan pengelesan yang buruk. Dataset yang digunakan untuk proses *training* dan validasi model didapatkan dengan mengakuisisi citra hasil pengelasan besi di suatu tempat *workshop* mekanik di Jejeran, Wonokromo, Pleret, Bantul, D.I. Yogyakarta. Studi teknik *preprocessing* (ekstraksi fitur) dilakukan guna mendapatkan model dengan performa terbaik dan waktu konvergensi tercepat. Dari penelitian ini dapat disimpulkan bahwa model UNet dengan teknik ekstraksi fitur RGB memiliki performa terbaik dengan nilai validasi IoU sebesar 78,1% dengan inialisasi *learning rate* 0,1 dan *batch size* 8. Dalam hal kecepatan konvergensi model, teknik *grayscale* dapat membantu model mencapai konvergensi setelah 144 epoch dengan waktu 9 menit 54 detik . Hal ini dikarenakan metode ekstraksi fitur *grayscale* hanya memiliki satu *channel* dan memiliki informasi piksel lebih baik dibandingkan LBP. Dimasa mendatang, penelitian terkait arsitektur *deep learning* untuk persoalan segmentasi dapat dilakukan untuk mendapatkan model dengan performa yang lebih baik. Selain itu, studi formulasi *loss function* juga dapat dilakukan untuk meningkatkan performa model.

UCAPAN TERIMA KASIH

Penelitian ini dilaksanakan dengan dana Penelitian Terapan Unggulan Perguruan Tinggi (PTUPT) dengan nomor hibah 6/E1/KP.PTNBH/2021 dan 1765/UN1/DITLIT/DITLIT/PT/2021. Penulis juga mengucapkan terima kasih kepada Pengelola bengkel pengelasan Balawan Yasa Construction di Jejeran, Wonokromo, Pleret, Bantul, D.I. Yogyakarta dan saudara Rizqi Pangestu yang telah membantu dalam proses pembuatan dataset dan anotasinya.

DAFTAR PUSTAKA

- [1] O. Ronneberger, P. Fischer, dan T. Brox. "U-Net: Convolutional Networks for Biomedical Image Segmentation". en. In: *Medical Image Computing and Computer Assisted Intervention – MICCAI 2015*. Ed. by N. Navab, J. Hornegger, W. M. Wells, dan A. F. Frangi. Lecture Notes in Computer Science. Cham: Springer International Publishing, 2015, pp. 234–241. ISBN: 9783319245744. DOI: 10.1007/978-3-319-24574-4_28.
- [2] N. Oskar, D. U. K. Putri, dan A. Dharmawan. "Deep Learning-based Weld Spot Segmentation using Modified UNet with Various Convolutional Blocks". In: *ICIC Express Letter B: Applications* 12.12 (2021).
- [3] I. Sobel. "An Isotropic 3x3 Image Gradient Operator". In: Presentation at Stanford A.I. Project 1968 (Feb. 2014).
- [4] T. Ojala, M. Pietikainen, dan T. Maenpaa. "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24.7 (July 2002), pp. 971–987. ISSN: 1939-3539. DOI: 10.1109/TPAMI.2002.1017623.
- [5] K. Simonyan dan A. Zisserman. "Very Deep Convolutional Networks for Large-Scale Image Recognition". In: *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*. Ed. by Y.

- Bengio dan Y. LeCun. 2015.
- [6] S. Ioffe dan C. Szegedy. “Batch normalization: accelerating deep network training by reducing internal covariate shift”. In: *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*. ICML’15. Lille, France: JM-LR.org, July 2015, pp. 448–456. (Visited on 09/16/2021).
- [7] V. Nair dan G. E. Hinton. “Rectified linear units improve restricted boltzmann machines”. In: *Proceedings of the 27th International Conference on International Conference on Machine Learning*. ICML’10. Haifa, Israel: Omnipress, June 2010, pp. 807–814. ISBN: 9781605589077.(Visited on 09/16/2021).
- [8] S. Mannor, D. Peleg, dan R. Rubinstein. “The cross entropy method for classification”. In: *Proceedings of the 22nd international conference on Machine learning*. ICML ’05. Bonn, Germany: Association for Computing Machinery, Aug. 2005, pp. 561–568. ISBN: 9781595931801. DOI: 10.1145/1102351.1102422. URL: <https://doi.org/10.1145/1102351.1102422>(visited on 09/16/2021).
- [9] C. H. Sudre, W. Li, T. Vercauteren, S. Ourselin, dan M. Jorge Cardoso. “Generalised DiceOverlap as a Deep Learning Loss Function for Highly Unbalanced Segmentations”. en. In: *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical DecisionSupport*. Ed. by M. J. Cardoso, T. Arbel, G. Carneiro, T. Syeda-Mahmood, J. M. R. Tavares, M. Moradi, A. Bradley, H. Greenspan, J. P. Papa, A. Madabhushi, J. C. Nascimento, J. S. Cardoso, V. Belagiannis, dan Z. Lu. Lecture Notes in Computer Science. Cham: Springer International Publishing, 2017, pp. 240–248. ISBN: 9783319675589. DOI: 10.1007/978- 3-319-67558-9_28.
- [10] A. Krogh dan J. A. Hertz. “A simple weight decay can improve generalization”. In: *Proceedings of the 4th International Conference on Neural Information Processing Systems*. NIPS’91. Denver, Colorado: Morgan Kaufmann Publishers Inc., Dec. 1991, pp. 950–957. ISBN: 9781558602229. (Visited on 09/16/2021).
- [11] I. Sutskever, J. Martens, G. Dahl, dan G. Hinton. “On the importance of initialization and momentum in deep learning”. In: *Proceedings of the 30th International Conference on International Conference on Machine Learning - Volume 28*. ICML’13. Atlanta, GA, USA: JMLR.org, June 2013, pp. III–1139–III–1147. (Visited on 09/16/2021).