

Penggunaan *Pre-trained Model* untuk Klasifikasi Kualitas Sekrup

Muhammad Rafif^{*1}, Diyah Utami Kusumaning Putri², Lukman Awaludin³

¹Program Studi Elektronika dan Instrumentasi, FMIPA UGM, Yogyakarta, Indonesia

^{2,3}Departemen Ilmu Komputer dan Elektronika, FMIPA UGM, Yogyakarta, Indonesia

e-mail: *muhammad.rafif@mail.ugm.ac.id, diyah.utami.k@mail.ugm.ac.id,

lukman.awaludin@ugm.ac.id

Abstrak

Inspeksi kualitas produk berbasis citra merupakan hal yang penting bagi industri manufaktur. Tugas tersebut sebagian besar masih dilakukan oleh manusia yang memiliki unit per hour rendah. Metode konvensional untuk inspeksi citra masih mengandalkan metode berbasis fitur, yang memiliki masalah sulitnya generalisasi dan ekstraksi fitur. Masalah tersebut diatasi dengan metode CNN, tetapi CNN membutuhkan data yang besar dan waktu training yang lama. Penggunaan pre-trained model dan augmentasi citra dapat menyelesaikan permasalahan pada metode-metode sebelumnya. Namun, belum ada penelitian yang secara lengkap meneliti dan membandingkan performa berbagai pre-trained model dan variasi augmentasi citra untuk tugas inspeksi citra kualitas produk manufaktur.

Proses penelitian menggunakan dataset sekrup berjenis multi class dan binary class pada 33 jenis pre-trained model dan 8 jenis augmentasi citra. Pengujian pre-trained model menggunakan dataset gabungan seluruh jenis augmentasi citra. Model dengan akurasi tertinggi adalah EfficientNetV2-L untuk dataset multi class (97.8%) dan VGG-19 untuk dataset binary class (96.5%). Augmentasi citra dengan signifikansi tertinggi terhadap performa model adalah blur, dengan akurasi 81.1% pada multi class dan 92% pada binary class. Keseluruhan proses pengujian pre-trained model dan augmentasi citra berjalan dengan baik.

Kata kunci—Inspeksi kualitas produk, Pre-trained model, Augmentasi citra

Abstract

Image-based product quality inspection is important for the manufacturing industry. Most of these tasks are still carried out by humans who have low units per hour. Conventional methods for image inspection still rely on feature-based methods, which have difficulties in generalizing and extracting features. This problem is solved by using CNN method, but CNN requires large data and long training time. Pre-trained model and image augmentation can solve the problems in the previous methods. However, there has been no research that has completely examined and compared the performance of various pre-trained models and variations of image augmentation for image inspection task of manufactured products quality.

The research process uses a multi-class and binary-class screw dataset on 33 pre-trained model types and 8 image augmentation types. Testing the pre-trained model using a combined dataset of all image augmentation types. Model with the highest accuracy is EfficientNetV2-L for multi class dataset (97.8%) and VGG-19 for binary class dataset (96.5%). Image augmentation with the highest significance to model performance is blur, with an accuracy of 81.1% in multi class and 92% in binary class. The whole process of testing pre-trained model and image augmentation went well.

Keywords—Product quality inspection, Pre-trained model, Image augmentation

1. PENDAHULUAN

Inspeksi kualitas produk berbasis citra merupakan hal yang sangat penting bagi industri manufaktur, terutama pada sektor 3C (*Computers, Communications, dan Consumer Electronics*) [1]. Mereka perlu memisahkan produk yang telah memenuhi standar dan menyingkirkan produk yang cacat, untuk memastikan kualitas produk tetap baik [2]. Tugas tersebut sebagian besar masih dilakukan secara manual oleh manusia yang hanya dapat bekerja untuk waktu yang terbatas [1]. Meningkatnya kompleksitas produk juga dapat mendorong kesalahan dalam proses inspeksi manual [3]. Penerapan inspeksi manual pada faktanya masih luas digunakan saat ini dan memiliki hasil yang akurat, tetapi hal tersebut masih tidak efisien dikarenakan *unit per hour rate* dari inspeksi manual masih rendah [4,5].

Pendekatan konvensional untuk inspeksi citra (*optical/visual inspection*) sebagian besar masih mengandalkan metode berbasis fitur (*feature-based approach*). Metode tersebut memerlukan manusia untuk menemukan pola cacat pada citra atau dengan kata lain proses ekstraksi fitur masih dilakukan secara manual, serta menghasilkan performa yang baik untuk tugas-tugas tertentu [1]. Permasalahan utama pada metode berbasis fitur adalah sulit untuk digeneralisasi pada tugas-tugas spesifik lainnya karena fitur citra harus diekstrak terlebih dahulu.

Pendekatan *convolutional neural networks* (CNN) dapat mengatasi permasalahan pada metode berbasis fitur. CNN adalah konsep dimana proses ekstraksi fitur dilakukan secara otomatis oleh *neural network* terhadap sebuah data yang berisi kumpulan citra berlabel dan dapat diterapkan untuk banyak tugas klasifikasi [3]. CNN akan memetakan gambar dan mendeteksi sejumlah representasi fitur abstrak, mulai dari yang sederhana hingga yang lebih kompleks [6]. Tantangan utama dalam menggunakan pendekatan tersebut adalah untuk memperoleh data yang besar dengan resolusi citra yang sesuai, serta proses *training* yang cukup lama. Permasalahan kekurangan data dan lamanya waktu *training* dapat diselesaikan dengan sebuah metode bernama *transfer learning* [1,7].

Metode *transfer learning* akan mentransfer bobot *pre-trained model* hasil ekstraksi fitur dari domain sumber ke domain target [1]. Umumnya proses ini dilakukan dengan cara *training* sebuah model pada *dataset* yang besar, seperti ImageNet. ImageNet melalui *ImageNet Large Scale Visual Recognition Challenge* (ILSVRC) telah menjadi tolok ukur berbagai model untuk melakukan tugas-tugas dalam deteksi objek dan klasifikasi citra [8]. Nakashima dkk. (2020) [9] meneliti perbandingan penggunaan model *transfer learning* berbasis ImageNet (AlexNet) dengan model arsitektur CNN biasa. Hasilnya didapatkan bahwa model *transfer learning* dapat menghasilkan akurasi sebesar 88% dibandingkan 71% pada model CNN biasa.

Penggunaan *transfer learning* dapat disertai dengan penerapan augmentasi citra (*image augmentation*) pada *dataset*. Kedua metode tersebut dapat meningkatkan efisiensi *training* pada model dengan *dataset* yang kecil [10]. Penelitian Fu dkk. (2019) [11] menganalisis penggunaan augmentasi citra dan pengaruhnya pada akurasi model. Akurasi model mencapai 86.7% dengan augmentasi citra berupa perubahan kecerahan, *flip* gambar, dan rotasi gambar, sedangkan 66.7% akurasi pada *dataset* yang tidak dilakukan augmentasi citra sama sekali.

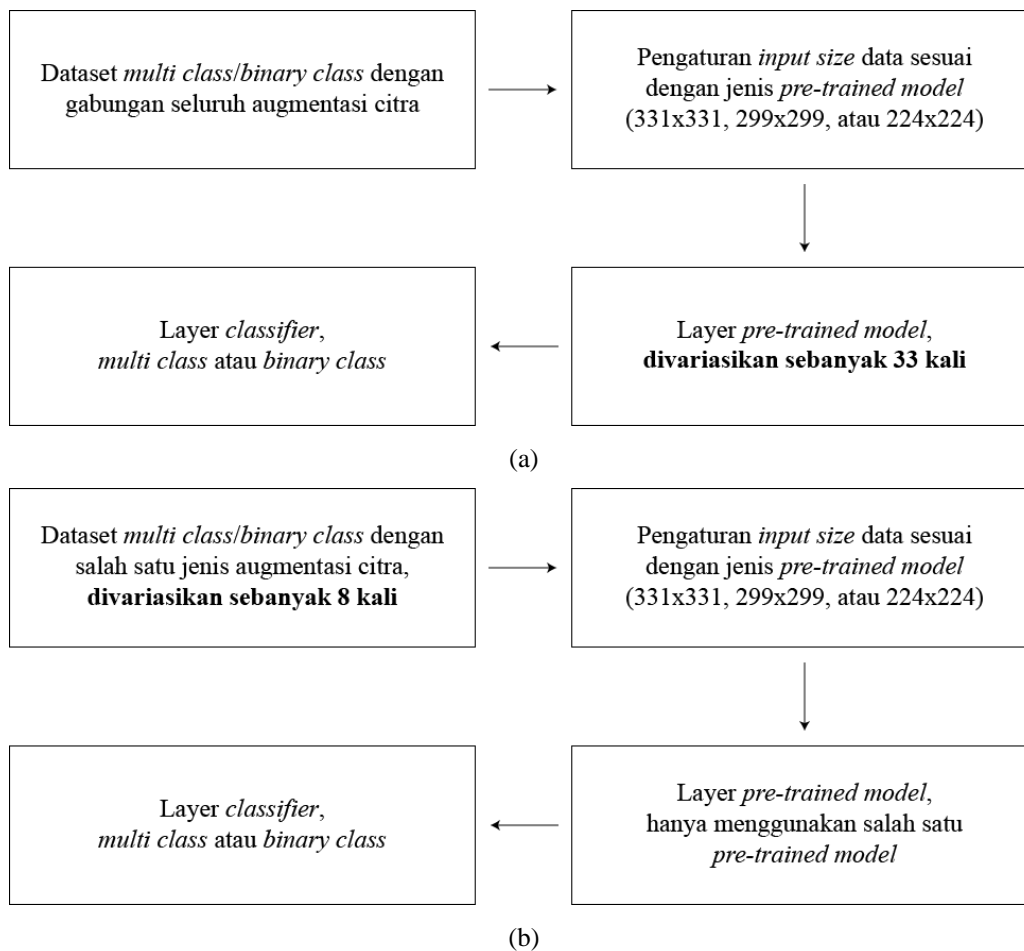
Metode *transfer learning* dan augmentasi citra telah terbukti sebagai solusi penyelesaian masalah-masalah pada tugas inspeksi citra. Sangat banyak jenis *pre-trained model* dan jenis augmentasi citra yang dapat digunakan, tetapi jika dilihat lebih dalam pada penelitian terdahulu terkait tugas klasifikasi menggunakan *transfer learning*, sebagian besar hanya menggunakan satu atau beberapa *pre-trained model* saja sebagai perbandingan performanya, serta beberapa penelitian juga tidak menerapkan augmentasi citra pada data mereka. Penelitian ini melakukan analisis dan perbandingan performa penggunaan berbagai *pre-trained model* dan augmentasi citra pada *dataset* untuk menyelesaikan tugas klasifikasi kualitas produk manufaktur.

2. METODE PENELITIAN

2.1 Rancangan Sistem

Eksperimen dalam penelitian ini dibagi menjadi 2 jenis, yaitu eksperimen jenis-jenis *pre-trained model* dan eksperimen jenis-jenis augmentasi citra menggunakan dataset *multi class* dan *binary class*. Eksperimen jenis-jenis *pre-trained model* dilakukan dengan cara menguji 33 jenis *pre-trained model* secara bergantian menggunakan gabungan seluruh jenis augmentasi citra untuk melihat model yang memiliki performa paling baik. *Pre-trained model* yang digunakan berjumlah 33, yaitu DenseNet (121, 169, 201), EfficientNetV1 (B0-B7), EfficientNetV2 (B0-B3, S, M, L), InceptionResNetV2, InceptionV3, MobileNetV1, MobileNetV2, NASNet (Mobile dan Large), ResNetV1 (50, 101, 152), ResNetV2 (50, 101, 152), VGG (16 dan 19), dan Xception.

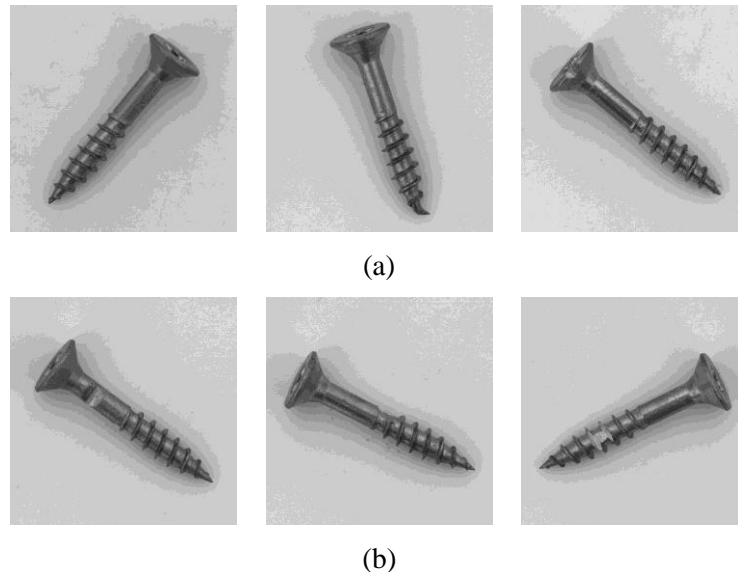
Eksperimen jenis-jenis augmentasi citra dilakukan dengan cara: (1) menguji 8 jenis augmentasi citra secara bergantian menggunakan 1 *pre-trained model* yang sama untuk melihat jenis augmentasi citra yang berdampak paling signifikan pada performa model; (2) 1 kali percobaan tanpa diterapkan augmentasi citra sama sekali, sehingga hanya menggunakan data asli, untuk melihat performa ketika tidak menggunakan augmentasi citra dan menjadi acuan perbandingan augmentasi citra. Jenis augmentasi citra yang digunakan adalah rotasi, *flip*, *shift*, *salt-pepper noise*, *gaussian noise*, perubahan kecerahan, perubahan kontras, dan blur. Alur kerja seluruh eksperimen dalam penelitian ini dapat dilihat pada gambar 1.



Gambar 1 Alur kerja eksperimen jenis-jenis *pre-trained model* (a) dan eksperimen jenis-jenis augmentasi citra (b)

2.2 Dataset

Data yang digunakan untuk pengujian performa adalah *dataset* MVTEC Anomaly Detection (MVTEC AD) dengan objeknya adalah sekrup. *Dataset* digunakan sebagai 2 jenis, yaitu *multi class* dan *binary class*. *Dataset multi class* terbagi atas enam kelas, yaitu *good*, *manipulated front*, *scratch head*, *scratch neck*, *thread side*, dan *thread top*, dengan masing-masing kelas berjumlah 22 gambar. *Dataset binary class* terbagi atas 2 kelas, yaitu OK (dari kelas *good*) dan NG (lima kelas selain *good*), dengan masing-masing kelas berjumlah 115 gambar. Gambar *dataset* hanya diambil dari sudut pandang sisi depan, sehingga tidak terdapat gambar dengan sudut pandang bagian atas dan bawah sekrup. Gambar masing-masing kelas dapat dilihat pada gambar 2.



Gambar 2 *Dataset* sekrup (a) *good*, *manipulated front*, *scratch head* (kiri ke kanan), (b) *scratch neck*, *thread side*, *thread top* (kiri ke kanan)

2.3 Proses Augmentasi Citra

Augmentasi citra bertujuan untuk menambah sampel baru pada data latih dengan cara memodifikasi data asli. *Dataset* sekrup sudah dalam rasio aspek 1:1, sehingga tidak diperlukan *cropping* terlebih dahulu. Selanjutnya, dilakukan berbagai jenis augmentasi citra pada *dataset* yang kemudian digunakan untuk mencari jenis augmentasi citra yang paling signifikan terhadap performa model.

Terdapat 8 jenis augmentasi citra yang digunakan, yaitu rotasi, *flip*, *shift*, *salt-pepper noise*, *gaussian noise*, perubahan kecerahan, perubahan kontras, dan blur. Seluruh operasi augmentasi citra tersebut akan menambah jumlah *dataset* dari 22 gambar per kelas menjadi 198 gambar per kelas untuk *multi class*, serta 115 gambar per kelas menjadi 1,035 gambar per kelas untuk *binary class*. Kemudian, *dataset* tersebut akan dibagi menjadi data latih, data validasi, dan data tes dengan rasio 80:10:10.

Parameter yang digunakan pada masing-masing jenis augmentasi citra telah disesuaikan sehingga seluruh data yang dihasilkan merupakan data baru, unik, dan tidak terdapat duplikasi data. Rincian parameter yang digunakan adalah sebagai berikut:

1. Rotasi
Sudut rotasi acak antara 90, 180, dan 270 derajat.
2. *Flip*
Jenis *flip* acak antara horizontal dan vertikal.
3. *Shift*

Jumlah pergeseran piksel acak antara -25, -20, -10, -5, 5, 10, 15, 20, dan 25 piksel untuk koordinat x maupun y.

4. *Salt-pepper Noise*
Noise sebanyak 1% dari total piksel secara acak.
5. *Gaussian Noise*
Standar deviasi (*spread*) dari distribusi sebesar 0.5 dengan *mean* 0.
6. Perubahan Kecerahan
Perubahan kecerahan dengan nilai acak antara -50, -40, -30, -20, -10, 10, 20, 30, 40, dan 50, dengan acuan nilai 0 adalah gambar normal/asli.
7. Perubahan Kontras
Perubahan kontras dengan nilai acak antara 0.5, 0.6, 0.7, 0.8, 0.9, 1.1, 1.2, 1.3, 1.4, dan 1.5, dengan acuan nilai 1 adalah gambar normal/asli.
8. Blur
Intensitas blur dengan besar kernel 3x3.

2.4 Pembuatan Model

Model yang digunakan dalam penelitian ini terdiri atas sebuah *pre-trained model* yang layer atasnya diganti dengan beberapa layer *classifier* baru untuk disesuaikan dengan *dataset* sekrup. Layer *classifier* tersebut adalah *global average pooling layer* untuk membuat data *flattened*, *dropout layer* untuk mencegah *overfitting*, dan *dense layer* sebagai *fully connected layer* sekaligus layer terakhir untuk klasifikasi. Sebelum dilakukan *training* pada *dataset* sekrup, layer *pre-trained model* perlu di-*freeze* agar tidak terjadi *training* ulang pada seluruh *pre-trained model* dengan cara mengubah atribut *trainable* pada layer *pre-trained model* menjadi *False*. Pengetahuan dan bobot yang dimiliki *pre-trained model* akan dimanfaatkan untuk mengekstrak fitur yang berarti pada *dataset* baru, sedangkan layer *classifier*-nya akan bertugas untuk tugas klasifikasi data.

Proses selanjutnya adalah menginput *dataset* yang telah dibuat sebelumnya pada model untuk dilakukan proses *training*. *Dataset* perlu disesuaikan bentuknya (*image shape*) dengan bentuk input *pre-trained model* yang digunakan (331x331, 299x299, atau 224x224). Beberapa *pre-trained model* juga membutuhkan jenis *preprocessing input* tertentu berupa skala piksel gambar dalam [0, 255] ataupun [-1, 1], maupun konversi warna gambar RGB ke BGR. Hal tersebut dapat dilakukan dengan menyertakan layer tambahan pada model sebelum layer utama/layer dasar berupa fungsi *preprocess input* yang terdapat pada masing-masing *pre-trained model*. Model yang telah dibuat akan di-*compile* dan dilakukan proses *training* pada *dataset* sekrup sebanyak 10 *epochs* dengan *learning rate* sebesar 0.0001.

2.5 Fine-tuning Model

Pada tahap sebelumnya, proses *training* hanya dilakukan pada layer *classifier*, sedangkan layer *pre-trained model* di-*freeze* dan bobotnya tidak berubah sepanjang proses *training*. Sebuah cara untuk meningkatkan performa model lebih baik adalah dengan melakukan fine-tuning pada model. *Fine-tuning* dilakukan dengan meng-*unfreeze* beberapa layer atas dari *pre-trained model*, sehingga proses *training* tidak hanya mengubah bobot dari layer *classifier*, tetapi juga beberapa layer atas *pre-trained model*. Hal tersebut akan membuat bobot lebih cocok dan spesifik pada *dataset* sekrup.

Proses fine-tuning dapat dilakukan dengan mengatur atribut *trainable* pada layer *pre-trained model* menjadi *True*. *Fine-tuning* dapat diterapkan secara efisien pada beberapa layer teratas saja dari sebuah *pre-trained model*, karena semakin atas layer, maka semakin terspesialisasi/spesifik layer tersebut, sehingga layer-layer tersebut yang perlu diadaptasi terhadap *dataset* baru. Penelitian ini melakukan *fine-tuning* pada 10% layer teratas dari sebuah *pre-trained model* yang digunakan dan di-*training* dengan 10 *epochs*. Parameter *learning rate* saat *fine-tuning* perlu diperhatikan untuk mencegah *overfitting* yang dapat terjadi karena model yang sangat besar di-*training* pada *dataset* yang jauh lebih kecil dibanding *dataset* aslinya,

sehingga digunakan *learning rate* 10 kali lebih kecil dibanding *learning rate* sebelum *fine-tuning*, yaitu sebesar 0.00001.

2.6 Pengujian Sistem

Pengujian sistem dilakukan terhadap seluruh eksperimen yang telah direncanakan untuk melihat perbandingan performa setiap *pre-trained model* dan signifikansi setiap jenis augmentasi citra terhadap performa model. Seluruh eksperimen tersebut diuji dengan beberapa metrik performansi pada tabel 1. Selanjutnya, dilakukan pencatatan terhadap seluruh eksperimen. Hasil eksperimen tersebut dianalisis untuk ditentukan model yang paling akurat (akurasi tertinggi), paling ringan (waktu *training* dan *testing* tercepat), dan paling efisien (kombinasi seluruh metrik paling baik), serta augmentasi citra yang paling signifikan meningkatkan performa model.

Tabel 1 Metrik Performansi

No	Metrik Performansi	Keterangan
1	Akurasi	Akurasi digunakan untuk mengevaluasi jumlah benar seluruh prediksi terhadap seluruh total data. Metrik ini digunakan karena persebaran data yang seimbang diantara 2 kelas tersebut. Semakin besar nilai akurasi, maka semakin baik performa model, dan sebaliknya.
2	<i>Loss</i>	<i>Loss</i> digunakan untuk melihat jumlah eror yang terjadi pada model. Metrik ini mengukur seberapa baik atau buruk sebuah model dalam melakukan tugasnya. Semakin kecil nilai <i>loss</i> , maka semakin baik performa model, dan sebaliknya.
3	Presisi	Presisi merepresentasikan jumlah prediksi positif yang benar (<i>true positive</i>) berbanding dengan total prediksi positif (<i>true positive</i> + <i>false positive</i>). Semakin besar nilai presisi, maka semakin baik performa model, dan sebaliknya.
4	<i>Recall</i>	<i>Recall</i> merepresentasikan jumlah prediksi positif yang benar (<i>true positive</i>) berbanding dengan total data positif sebenarnya (<i>true positive</i> + <i>false negative</i>). Semakin besar nilai <i>recall</i> , maka semakin baik performa model, dan sebaliknya.
5	<i>F1-score</i>	<i>F1-score</i> menggabungkan perhitungan antara presisi dan <i>recall</i> , yang memasukkan <i>false positive</i> maupun <i>false negative</i> dalam perhitungannya. <i>F1-score</i> didefinisikan sebagai $2 * \text{presisi} * \text{recall}$ berbanding dengan jumlah nilai presisi dan <i>recall</i> . Semakin besar nilai <i>f1-score</i> , maka semakin baik performa model, dan sebaliknya.
6	Waktu <i>Training</i>	Waktu <i>training</i> mengindikasikan kompleksitas dan kebutuhan tenaga komputasi untuk melakukan <i>training</i> model. Jika waktu <i>training</i> cepat, maka dapat diartikan bahwa model ringan dan tidak membutuhkan banyak tenaga komputasi, dan sebaliknya.
7	Waktu <i>Testing</i>	Waktu <i>testing</i> mengindikasikan kompleksitas dan tenaga komputasi yang dibutuhkan untuk menjalankan model. Semakin cepat waktu <i>testing</i> , maka semakin ringan model tersebut dalam proses pengujian.

3. HASIL DAN PEMBAHASAN

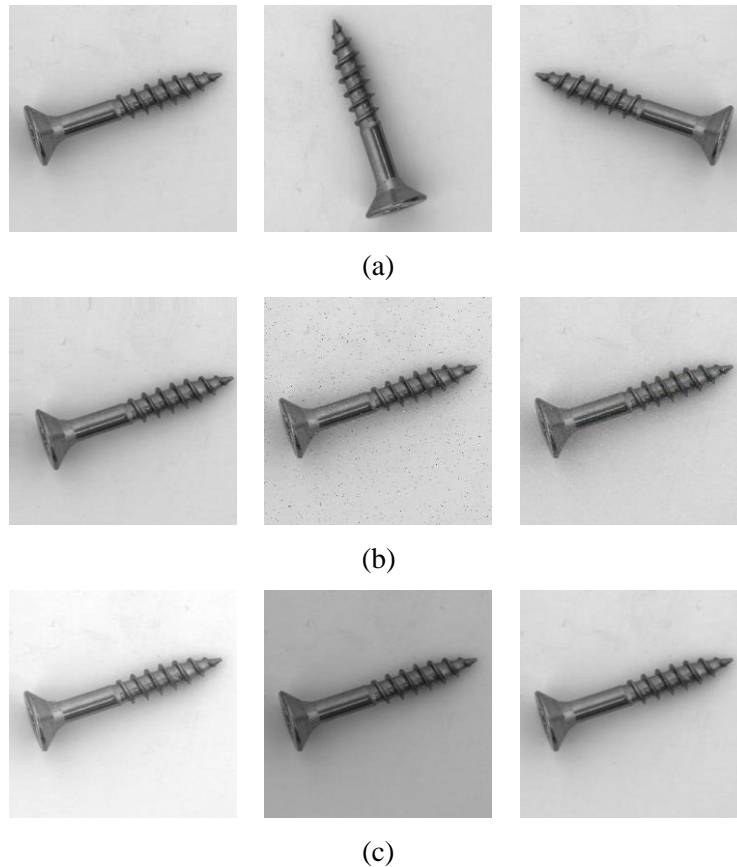
Penelitian ini secara umum mencakup 2 jenis eksperimen: (1) eksperimen jenis-jenis *pre-trained model* yang menggunakan 1 *dataset* yang berisi gabungan seluruh jenis augmentasi citra ditambah data asli; (2) eksperimen jenis-jenis augmentasi citra menggunakan 9 *dataset*

yang mencakup 1 *dataset* yang berisi hanya data asli dan 8 *dataset* yang masing-masing berisi satu jenis augmentasi citra ditambah data asli.

Eksperimen jenis-jenis *pre-trained model* menguji 33 jenis *pre-trained model* pada *library* TensorFlow Keras. Setelah eksperimen tersebut selesai, penelitian dilanjutkan untuk eksperimen jenis-jenis augmentasi citra menggunakan *pre-trained model* yang paling akurat (akurasi tertinggi) pada eksperimen sebelumnya. Eksperimen-eksperimen di atas berlaku untuk *dataset multi class* maupun *binary class*. Seluruh hasil eksperimen kemudian dicatat untuk dianalisis performanya menggunakan beberapa metrik performansi.

3.1 Hasil Proses Augmentasi Citra

Pembuatan *dataset-dataset* yang digunakan dalam penelitian ini dimulai dari proses augmentasi citra. Proses augmentasi citra melibatkan 8 jenis augmentasi citra yang diterapkan pada data asli *dataset* sekrup. Hasil dari proses augmentasi citra kemudian dijadikan *dataset* dengan cara dilakukan pembagian data menjadi data latih, data validasi, dan data tes. Contoh dari hasil proses augmentasi citra terhadap *dataset* sekrup dapat dilihat pada gambar 3.



Gambar 3 Hasil augmentasi citra (a) tanpa augmentasi, rotasi 90 derajat, *flip* horizontal (kiri ke kanan), (b) *shift* 25x25, *salt-pepper noise*, *gaussian noise* (kiri ke kanan), (c) perubahan kecerahan, perubahan kontras, blur (kiri ke kanan)

3.2 Hasil Eksperimen Jenis-Jenis Pre-trained Model

Eksperimen jenis-jenis *pre-trained model* dilakukan pada 33 *pre-trained model* secara berulang sebanyak 3 kali per model. Hal tersebut dilakukan untuk mencegah terjadinya hasil performa yang bias atau kebetulan. Parameter yang digunakan adalah 10 *epochs training* sebelum *fine-tuning* dengan *learning rate* 0.0001, 10 *epochs training* setelah *fine-tuning* dengan

learning rate 0.00001, ukuran *batch* 16, dan layer *pre-trained model* yang di-*fine-tuning* sebanyak 10% dari total jumlah layer.

Tabel 2 Hasil performa model pada *dataset multi class*

No	Model	Akurasi	Presisi	Recall	F1-Score	Loss	Waktu Training (Detik)	Waktu Testing (Detik)
1	DenseNet-121	56.1%	56.5%	56.1%	55.1%	1.38	87.959	0.914
2	DenseNet-169	65.6%	66.5%	65.6%	63.2%	1.285	117.776	1.163
3	DenseNet-201	78.6%	80.3%	78.6%	78.0%	1.037	147.167	1.412
4	EfficientNetV1-B0	76.7%	75.9%	76.7%	75.5%	0.942	68.451	0.703
5	EfficientNetV1-B1	83.9%	84.2%	83.9%	83.8%	0.576	92.8	0.889
6	EfficientNetV1-B2	85.8%	86.6%	85.8%	85.8%	0.566	96.257	0.907
7	EfficientNetV1-B3	87.5%	88.1%	87.5%	87.2%	0.467	118.2	1.096
8	EfficientNetV1-B4	88.6%	90.0%	88.6%	88.4%	0.377	152.224	1.402
9	EfficientNetV1-B5	90.3%	90.8%	90.3%	90.2%	0.377	204.206	1.813
10	EfficientNetV1-B6	91.1%	92.1%	91.1%	91.1%	0.316	260.124	2.282
11	EfficientNetV1-B7	93.0%	93.5%	93.0%	93.0%	0.266	349.712	3.024
12	EfficientNetV2-B0	66.9%	67.1%	66.9%	66.3%	1.202	61.913	0.64
13	EfficientNetV2-B1	67.5%	68.6%	67.5%	67.0%	1.15	74.503	0.731
14	EfficientNetV2-B2	74.2%	74.8%	74.2%	74.0%	0.931	81.14	0.793
15	EfficientNetV2-B3	75.0%	73.6%	75.0%	72.6%	0.897	99.616	0.948
16	EfficientNetV2-S	63.6%	63.0%	63.6%	61.7%	1.104	132.727	1.236
17	EfficientNetV2-M	95.3%	95.5%	95.3%	95.2%	0.151	234.687	1.966
18	EfficientNetV2-L	97.8%	97.8%	97.8%	97.8%	0.071	408.654	3.308
19	InceptionResNetV2	78.6%	84.5%	78.6%	78.7%	0.603	262.273	2.505
20	InceptionV3	81.4%	83.3%	81.4%	81.4%	0.827	109.339	1.121
21	MobileNetV1	78.6%	80.6%	78.6%	78.7%	0.853	40.656	0.483
22	MobileNetV2	72.8%	76.7%	72.8%	73.1%	0.93	47.239	0.535
23	NasNet Mobile	53.1%	54.2%	53.1%	52.6%	1.37	97.885	0.876
24	NasNet Large	66.1%	71.4%	66.1%	64.0%	0.912	703.381	6.615
25	ResNetV1-50	82.5%	84.5%	82.5%	82.1%	0.494	91.084	0.9
26	ResNetV1-101	87.2%	88.2%	87.2%	87.2%	0.436	153.168	1.377
27	ResNetV1-152	89.7%	90.4%	89.7%	89.5%	0.257	218.221	1.91
28	ResNetV2-50	83.6%	85.8%	83.6%	83.5%	0.466	87.31	0.863
29	ResNetV2-101	90.6%	91.1%	90.6%	90.6%	0.231	149.901	1.35
30	ResNetV2-152	92.2%	93.0%	92.2%	92.2%	0.259	214.274	1.915
31	VGG-16	82.2%	83.8%	82.2%	81.9%	0.664	108.192	1.293
32	VGG-19	86.4%	88.2%	86.4%	86.5%	0.445	131.096	1.463
33	Xception	71.9%	76.3%	71.9%	69.7%	0.965	180.482	1.753

Tabel 2 menunjukkan hasil performa jenis-jenis *pre-trained model* yang telah diuji dengan total 20 *epochs* sebanyak 3 kali perulangan pada *dataset multi class*. *Pre-trained model* yang paling akurat adalah EfficientNetV2-L dengan nilai akurasi sebesar 97.8%, *f1-score* sebesar 97.8%, dan *loss* 0.071; paling ringan adalah MobileNetV1 dengan waktu *training* 40.656 detik dan waktu *testing* 0.483 detik; serta paling efisien adalah EfficientNetV2-M dengan nilai akurasi sebesar 95.3%, *f1-score* sebesar 95.2%, *loss* 0.151, waktu *training* 234.687 detik, dan waktu *testing* 1.966 detik.

Tabel 3 Hasil performa model pada *dataset binary class*

No	Model	Akurasi	Presisi	Recall	F1-Score	Loss	Waktu Training (Detik)	Waktu Testing (Detik)
1	DenseNet-121	82.4%	80.5%	87.2%	83.2%	0.463	147.358	1.482
2	DenseNet-169	83.8%	84.6%	84.0%	83.7%	0.417	201.223	1.997
3	DenseNet-201	86.5%	85.1%	89.4%	86.8%	0.369	246.343	2.34
4	EfficientNetV1-B0	82.5%	82.4%	80.0%	82.7%	0.399	112.709	1.126
5	EfficientNetV1-B1	91.8%	90.8%	93.3%	92.0%	0.23	152.359	1.433
6	EfficientNetV1-B2	89.9%	91.1%	88.8%	89.8%	0.248	160.269	1.5
7	EfficientNetV1-B3	90.4%	90.2%	90.7%	90.4%	0.225	196.461	1.833
8	EfficientNetV1-B4	90.7%	94.5%	86.5%	90.3%	0.206	255.141	2.314
9	EfficientNetV1-B5	91.2%	92.1%	90.4%	91.1%	0.185	343.44	3.048
10	EfficientNetV1-B6	93.6%	92.0%	95.5%	93.7%	0.159	438.466	3.851
11	EfficientNetV1-B7	95.7%	95.4%	96.1%	95.7%	0.096	591.824	5.066
12	EfficientNetV2-B0	84.8%	85.7%	84.3%	84.7%	0.447	101.652	1.017
13	EfficientNetV2-B1	84.0%	82.4%	86.5%	84.4%	0.438	122.988	1.181
14	EfficientNetV2-B2	84.9%	85.6%	84.0%	84.8%	0.393	131.739	1.27
15	EfficientNetV2-B3	87.3%	86.0%	89.4%	87.6%	0.31	165.097	1.567
16	EfficientNetV2-S	87.7%	88.3%	87.2%	87.6%	0.337	218.427	2.048
17	EfficientNetV2-M	93.3%	91.1%	96.2%	93.5%	0.225	390.018	3.241
18	EfficientNetV2-L	95.7%	94.1%	97.4%	95.8%	0.142	690.967	5.531
19	InceptionResNetV2	91.5%	92.2%	91.0%	91.5%	0.259	444.67	4.165
20	InceptionV3	89.3%	90.3%	88.1%	89.2%	0.267	199.659	1.378
21	MobileNetV1	89.4%	94.3%	84.3%	88.7%	0.307	68.351	0.75
22	MobileNetV2	85.0%	93.8%	75.0%	83.3%	0.358	78.734	0.816
23	NasNet Mobile	80.7%	82.5%	78.2%	80.2%	0.506	156.841	1.357
24	NasNet Large	93.1%	95.6%	90.4%	92.9%	0.226	1211.772	11.33
25	ResNetV1-50	95.5%	95.8%	95.5%	95.5%	0.145	154.622	1.455
26	ResNetV1-101	94.4%	94.3%	94.6%	94.4%	0.145	260.513	2.251
27	ResNetV1-152	93.9%	95.7%	92.0%	93.8%	0.133	373.624	3.245
28	ResNetV2-50	92.8%	91.9%	93.9%	92.9%	0.195	148.588	1.404
29	ResNetV2-101	96.1%	97.2%	95.2%	96.1%	0.111	253.843	2.215
30	ResNetV2-152	94.1%	96.3%	91.7%	93.9%	0.236	361.051	3.192
31	VGG-16	92.6%	91.1%	94.8%	92.8%	0.238	186.508	2.068
32	VGG-19	96.5%	97.1%	95.8%	96.5%	0.134	226.419	2.432
33	Xception	87.4%	86.3%	89.7%	87.8%	0.334	310.376	2.962

Tabel 3 menunjukkan hasil performa jenis-jenis *pre-trained model* yang telah diuji dengan total 20 *epochs* sebanyak 3 kali perulangan pada *dataset binary class*. *Pre-trained model* yang paling akurat adalah VGG-19 yang memiliki nilai akurasi sebesar 96.5%, *f1-score* sebesar 96.5%, dan *loss* 0.134; paling ringan adalah MobileNetV1 dengan waktu *training* 68.351 detik dan waktu *testing* 0.75 detik; serta paling efisien adalah ResNetV1-50 dengan nilai akurasi sebesar 95.5%, *f1-score* sebesar 95.5%, *loss* 0.145, waktu *training* 154.62 detik, dan waktu *testing* 1.455 detik.

Secara umum, pengujian model pada *dataset binary class* memiliki hasil yang lebih tinggi dibandingkan pengujian model pada *dataset multi class*, tetapi dengan waktu *training* dan *testing* yang lebih lama. Hal tersebut disebabkan karena klasifikasi *multi class* memiliki kelas yang lebih banyak dan lebih rumit, tetapi dengan total data di dalamnya lebih sedikit dibandingkan *dataset binary class* yang hanya memiliki 2 kelas, sehingga perbedaan performa,

waktu *training*, dan waktu *testing* model pada kedua jenis *dataset* ini menjadi cukup terlihat secara keseluruhan pengujian.

3.3 Hasil Eksperimen Jenis-Jenis Augmentasi Citra

Eksperimen jenis-jenis augmentasi citra dilakukan pada 8 jenis augmentasi citra secara berulang sebanyak 3 kali masing-masing. Seluruh parameter, metrik performansi, dan proses pengujian dilakukan sama seperti eksperimen jenis-jenis *pre-trained model* sebelumnya. Perbedaannya hanya terletak pada *dataset* yang digunakan, yaitu menggunakan 9 *dataset* mencakup 1 *dataset* yang berisi hanya data asli 8 *dataset* yang masing-masing berisi satu jenis augmentasi citra ditambah data asli. *Dataset-dataset* tersebut digunakan untuk pengujian menggunakan *pre-trained model* yang memiliki akurasi paling tinggi pada eksperimen sebelumnya, yaitu EfficientNetV2-L untuk *dataset multi class*, dan VGG-19 untuk *dataset binary class*. Eksperimen jenis-jenis augmentasi citra menemukan jenis augmentasi citra yang paling signifikan dalam meningkatkan performa model.

Tabel 4 Hasil performa jenis-jenis augmentasi citra pada *dataset multi class*

No	Jenis Augmentasi	Akurasi	Presisi	Recall	F1-Score	Loss	Waktu Training (Detik)	Waktu Testing (Detik)
1	Tanpa Augmentasi	18.5%	26.9%	37.0%	29.1%	1.816	64.679	1.639
2	Rotasi	53.7%	62.4%	61.1%	56.9%	1.048	108.873	2.344
3	Flip	57.8%	68.2%	57.8%	58.2%	1.115	109.078	2.351
4	Shift	77.8%	80.8%	77.8%	76.0%	0.58	108.699	2.343
5	Salt-pepper Noise	64.1%	72.3%	67.8%	65.9%	0.781	108.582	2.334
6	Gaussian Noise	63.3%	70.9%	63.3%	62.1%	0.945	109.077	2.329
7	Perubahan Kecerahan	73.3%	82.7%	73.3%	73.1%	0.698	109.03	2.36
8	Perubahan Kontras	58.7%	76.8%	62.2%	60.8%	1.365	108.749	2.332
9	Blur	81.1%	87.0%	81.1%	80.3%	0.57	108.421	2.322

Tabel 4 menunjukkan hasil performa penerapan augmentasi citra pada *dataset multi class* menggunakan model EfficientNetV2-L. Secara umum, dapat dilihat bahwa seluruh jenis augmentasi citra memiliki pengaruh positif terhadap performa model jika dibandingkan dengan tanpa penerapan augmentasi sama sekali yang memiliki nilai akurasi sebesar 27.9%, *f1-score* sebesar 29.1%, dan *loss* 1.816. Waktu *training* dan *testing* juga secara umum sama untuk seluruh jenis augmentasi citra, sehingga letak perbedaan signifikansi jenis augmentasi citra terhadap performa ada pada nilai akurasi, *f1-score*, dan *loss*.

Blur adalah jenis augmentasi citra yang paling signifikan untuk meningkatkan performa model diantara jenis-jenis augmentasi citra yang lainnya, dengan nilai akurasi sebesar 81.1%, *f1-score* sebesar 80.3%, dan *loss* 0.57. Sedangkan jenis augmentasi citra yang memiliki signifikansi performa paling rendah adalah rotasi dengan nilai akurasi sebesar 53.7%, *f1-score* sebesar 56.9%, dan *loss* 1.048. Meskipun blur dapat meningkatkan performa model secara signifikan pada *dataset multi class*, tetapi masih lebih rendah dibandingkan dengan performa ketika seluruh jenis augmentasi citra digabungkan.

Tabel 5 Hasil performa jenis-jenis augmentasi citra pada *dataset binary class*

No	Jenis Augmentasi	Akurasi	Presisi	Recall	F1-Score	Loss	Waktu Training (Detik)	Waktu Testing (Detik)
1	Tanpa Augmentasi	77.8%	83.2%	72.2%	75.9%	0.441	27.277	0.339
2	Rotasi	79.7%	82.1%	76.8%	78.8%	0.52	51.722	0.577
3	Flip	84.8%	85.0%	85.5%	84.7%	0.37	51.78	0.584

4	<i>Shift</i>	89.8%	92.9%	87.0%	89.5%	0.251	51.802	0.572
5	<i>Salt-pepper Noise</i>	80.4%	77.0%	87.0%	81.6%	0.419	51.856	0.58
6	<i>Gaussian Noise</i>	84.8%	84.5%	85.5%	84.9%	0.323	51.978	0.581
7	Perubahan Kecerahan	91.3%	94.4%	88.4%	91.0%	0.265	51.672	0.573
8	Perubahan Kontras	87.0%	84.9%	89.9%	87.3%	0.363	51.457	0.566
9	Blur	92.0%	92.9%	91.3%	92.0%	0.212	51.292	0.567

Tabel 5 menunjukkan hasil performa penerapan augmentasi citra pada *dataset binary class* menggunakan model VGG-19. Sama seperti pada *dataset multi class*, secara umum penerapan augmentasi citra berpengaruh positif pada peningkatan performa model dengan waktu *training* dan *testing* yang relatif sama untuk seluruh jenis augmentasi citra. Tanpa penerapan augmentasi citra, model mencatatkan nilai akurasi sebesar 77.8%, *f1-score* sebesar 75.9%, dan *loss* 0.441.

Jenis augmentasi citra yang paling signifikan dan paling tidak signifikan terhadap performa model juga sama seperti pada *dataset multi class*. Blur adalah jenis augmentasi yang paling signifikan dalam peningkatan performa model dengan nilai akurasi sebesar 92.0%, *f1-score* sebesar 92.0%, dan *loss* 0.212. Sedangkan jenis augmentasi citra yang paling rendah signifikansinya adalah rotasi dengan nilai akurasi sebesar 79.7%, *f1-score* sebesar 78.8%, dan *loss* 0.52. Meskipun blur dapat meningkatkan performa model secara signifikan pada *dataset binary class*, tetapi masih lebih rendah dibandingkan dengan performa ketika seluruh jenis augmentasi citra digabungkan. Hasil eksperimen secara lengkap yang terdapat dalam penelitian ini dapat diakses di [spreadsheets hasil performa](#).

4. KESIMPULAN

Eksperimen jenis-jenis *pre-trained model* menggunakan 1 *dataset* yang berisi gabungan seluruh jenis augmentasi citra ditambah data asli. Eksperimen ini dilakukan pada 33 jenis *pre-trained model* yang terdapat pada *library* TensorFlow Keras. Model dengan performa paling baik (rata-rata akurasi, presisi, *recall*, dan *f1-score* tertinggi) adalah EfficientNetV2-L untuk *dataset multi class* dan VGG-19 untuk *dataset binary class*: paling ringan (waktu *training* dan *testing* tercepat) adalah MobileNetV1 untuk kedua *dataset*; serta paling efisien (kombinasi seluruh metrik paling baik) adalah EfficientNetV2-M untuk *dataset multi class* dan ResNetV1-50 untuk *dataset binary class*. Eksperimen ini juga melakukan perbandingan penggunaan *fine-tuning* pada model yang dibandingkan dengan tidak menggunakan *fine-tuning*. *Pre-trained model* dengan diterapkan *fine-tuning* hanya pada 10% layer teratasnya menghasilkan peningkatan performa yang sangat signifikan, seperti pada gambar 6.2 dan 6.3.

Eksperimen kedua adalah eksperimen jenis-jenis augmentasi citra menggunakan 9 *dataset* yang mencakup 1 *dataset* berisi gabungan seluruh jenis augmentasi citra ditambah data asli dan 8 *dataset* yang masing-masing berisi satu jenis augmentasi citra ditambah data asli. Eksperimen ini dijalankan menggunakan *pre-trained model* yang memiliki performa paling baik pada penelitian sebelumnya, yaitu EfficientNetV2-L untuk *dataset multi class* dan VGG-19 untuk *dataset binary class*. Secara umum, penggunaan augmentasi citra memberikan pengaruh yang positif terhadap performa model pada *dataset multi class* maupun *binary class*. Hasil pengujian juga menunjukkan bahwa jenis augmentasi citra yang paling signifikan dalam meningkatkan performa model pada *dataset multi class* maupun *binary class* adalah blur, sedangkan yang paling rendah signifikansinya adalah rotasi. Metrik performansi lainnya, yaitu waktu *training* dan waktu *testing*, secara umum memiliki waktu yang sama untuk seluruh jenis augmentasi citra, baik pada *dataset multi class* maupun *dataset binary class*.

5. SARAN

Penelitian ini dapat dikembangkan lebih lanjut dan dapat menggunakan saran-saran sebagai berikut:

1. Eksperimen serupa dapat diterapkan pada *dataset* dengan objek lain untuk melihat perbedaan yang dapat terjadi jika objek berbeda.
2. *Pre-trained model* yang digunakan dalam eksperimen dapat menggunakan model dari sumber selain *library* TensorFlow Keras, misalnya berasal dari TensorFlow Hub.
3. Parameter yang digunakan, seperti *optimizer*, *learning rate*, dan persentase layer yang dilakukan *fine-tuning* dapat dioptimalkan agar model mencapai performa maksimalnya.
4. Eksperimen jenis-jenis augmentasi citra dapat dilakukan dengan cara yang berbeda, misalnya mengkombinasikan blur dan *shift* pada satu *dataset*, atau kombinasi rotasi, *flip*, dan perubahan kecerahan pada satu *dataset*.

DAFTAR PUSTAKA

- [1] S. Kim, W. Kim, Y.-K. Noh, and F. C. Park, "Transfer learning for automated optical inspection," *2017 International Joint Conference on Neural Networks (IJCNN)*, 2017.
- [2] X. Chen, D. Wang, J. Shao, and J. Fan, "Plastic gasket defect detection based on Transfer learning," *Scientific Programming*, vol. 2021, pp. 1–11, 2021.
- [3] H. Würschinger, M. Mühlbauer, M. Winter, M. Engelbrecht, and N. Hanenkamp, "Implementation and potentials of a machine vision system in a series production using Deep Learning and low-cost hardware," *Procedia CIRP*, vol. 90, pp. 611–616, 2020.
- [4] M. Gao, P. Song, F. Wang, J. Liu, A. Mandelis, and D. W. Qi, "A novel deep convolutional neural network based on resnet-18 and transfer learning for detection of wood knot defects," *Journal of Sensors*, vol. 2021, pp. 1–16, 2021.
- [5] M. Abu, A. Amir, Y. H. Lean, N. A. Zahri, and S. A. Azemi, "The performance analysis of Transfer Learning for steel defect detection by using Deep Learning," *Journal of Physics: Conference Series*, vol. 1755, no. 1, p. 012041, 2021.
- [6] S. Khan, H. Rahmani, S. S. A. Ali, and M. Bennamoun, *A guide to Convolutional Neural Networks for Computer Vision*. San Rafael, CA: Morgan & Claypool Publishers, 2018.
- [7] N. O'Mahony, S. Campbell, A. Carvalho, S. Harapanahalli, G. V. Hernandez, L. Krpalkova, D. Riordan, and J. Walsh, "Deep learning vs. Traditional Computer Vision," *Advances in Intelligent Systems and Computing*, pp. 128–144, 2019.
- [8] W. Nash, T. Drummond, and N. Birbilis, "A review of deep learning in the study of materials degradation," *npj Materials Degradation*, vol. 2, no. 1, 2018.
- [9] K. Nakashima, F. Nagata, H. Ochi, A. Otsuka, T. Ikeda, K. Watanabe, and M. K. Habib, "Detection of minute defects using transfer learning-based CNN Models," *Artificial Life and Robotics*, vol. 26, no. 1, pp. 35–41, 2020.
- [10] R. Yamashita, M. Nishio, R. K. Do, and K. Togashi, "Convolutional Neural Networks: An overview and application in Radiology," *Insights into Imaging*, vol. 9, no. 4, pp. 611–629, 2018.
- [11] J. Fu, X. Zhu, and Y. Li, "Recognition of surface defects on steel sheet using transfer learning," *arXiv preprint arXiv:1909.03258*, 2019.