

PERFORMANCE COMPARISON OF FASTER R-CONVOLUTIONAL NEURAL NETWORK (CNN) AND EFFICIENTNET FOR TRAIN DETECTION UNDER DIVERSE LIGHTING AND IMAGE QUALITY CONDITIONS

Suastika Yulia Riska^{*1}, Achmad Noercholis²

^{1,2}Informatic Departement, Technology and Desain Faculty, Institut Teknologi dan Bisnis Asia, Malang, Indonesia

Email: ¹riska.suastika@asia.ac.id, ²anoercholis@asia.ac.id

(Article received: September 5, 2024; Revision: October 4, 2024; published: December 29, 2024)

Abstract

Object detection using computer vision has seen rapid advancements, especially with the advent of deep learning architectures such as Faster R-CNN and EfficientNet. This study compares the performance of the two models in detecting trains in various lighting conditions and noise disturbances. The dataset used consisted of 4500 train images which were divided into 70% for training, 20% for validation, and 10% for testing, reflecting real-world conditions. The evaluation was carried out using the Intersection over Union (IoU), Average Precision (AP), and Average Recall (AR) metrics. The results show that Faster R-CNN consistently excels in terms of detection accuracy, especially in less than ideal lighting conditions and under rain noise interference. In sufficient lighting conditions, Faster R-CNN showed a slightly superior AP score with a score of 0.844. As the lighting decreased, the difference between the two models became more pronounced, with Faster R-CNN recording an AP value of 0.810. In conditions with rain noise interference, the object detection performance of both models decreased more significantly, but the Faster R-CNN still excelled with an AP value of 0.798. Although EfficientNet is more efficient in terms of training speed, with a time of 5 hours and 37 minutes, and a smaller model size, Faster R-CNN shows higher reliability in complex environmental situations. This research provides important insights for the development of reliable and efficient train detection systems, taking into account the trade-off between resource efficiency and detection accuracy.

Keywords: Average Precision (AP), Average Recall (AR), EfficientNet, Faster R-CNN, Intersection over Union (IoU), Train Detection

PERBANDINGAN PERFORMA FASTER R-CONVOLUTIONAL NEURAL NETWORK (CNN) DAN EFFICIENTNET UNTUK DETEKSI KERETA API PADA BERAGAM PENCAHAYAAN DAN KONDISI KUALITAS GAMBAR

Abstrak

Deteksi objek menggunakan visi komputer telah mengalami kemajuan pesat, terutama dengan hadirnya arsitektur pembelajaran mendalam seperti Faster R-Convolutional Neural Network (CNN) dan EfficientNet. Penelitian ini membandingkan kinerja kedua model dalam mendeteksi kereta api di berbagai kondisi pencahayaan dan gangguan *noise*. Dataset yang digunakan terdiri dari 4500 gambar kereta api yang dibagi menjadi 70% untuk *training*, 20% untuk validasi, dan 10% untuk *testing*, mencerminkan kondisi dunia nyata. Evaluasi dilakukan menggunakan metrik *Intersection over Union (IoU)*, *Average Precision (AP)*, dan *Average Recall (AR)*. Hasil penelitian menunjukkan bahwa Faster R-CNN secara konsisten unggul dalam hal akurasi deteksi, terutama pada kondisi pencahayaan yang kurang ideal dan di bawah gangguan *noise* hujan. Pada kondisi pencahayaan yang cukup, Faster R-CNN menunjukkan nilai AP sedikit lebih unggul dengan skor 0.844. Ketika pencahayaan berkurang, perbedaan antara kedua model menjadi lebih nyata, dengan Faster R-CNN mencatat nilai AP 0.810. Dalam kondisi dengan gangguan *noise* hujan, performa deteksi objek dari kedua model mengalami penurunan yang lebih signifikan, namun Faster R-CNN tetap unggul dengan nilai AP 0.798. Meskipun EfficientNet lebih efisien dalam hal kecepatan training, dengan waktu 5 jam 37 menit, dan ukuran model yang lebih kecil, Faster R-CNN menunjukkan keandalan yang lebih tinggi dalam situasi lingkungan yang kompleks. Penelitian ini memberikan wawasan penting bagi pengembangan sistem deteksi kereta api yang andal dan efisien, dengan mempertimbangkan *trade-off* antara efisiensi sumber daya dan akurasi deteksi.

Kata kunci: *Average Precision (AP)*, *Average Recall (AR)*, Deteksi Kereta Api, EfficientNet, Faster R-CNN, *Intersection over Union (IoU)*,

1. PENDAHULUAN

Deteksi objek menggunakan visi komputer telah berkembang pesat dalam beberapa tahun terakhir, terutama dengan munculnya arsitektur pembelajaran mendalam yang semakin canggih. Penerapan penting dari teknologi ini adalah sistem keamanan lalu lintas, khususnya untuk deteksi kereta api [1]. Sistem ini berperan penting dalam mendeteksi keberadaan kereta api dengan cepat dan akurat dalam berbagai kondisi pencahayaan dan kualitas gambar.

Peran penting dalam sistem deteksi ditunjukkan dengan adanya keandalan sistem deteksi untuk mengurangi risiko kecelakaan, terutama di lingkungan dengan fluktuasi cahaya yang tinggi, seperti terowongan, ruang terbuka, dan persimpangan yang tidak dijaga [2]. Dalam konteks ini, Faster R-CNN, yang merupakan salah satu arsitektur CNN telah menjadi salah satu metode pendeteksian objek paling populer sejak diluncurkan pada tahun 2015 [3][4]. Metode ini banyak digunakan dalam berbagai aplikasi seperti pendeteksian kereta api.

Secara umum, Faster R-CNN berkinerja sangat baik dalam kondisi pencahayaan tertentu [5]. Namun, arsitektur ini cenderung rusak bila terkena kondisi pencahayaan ekstrem, sehingga menghasilkan kinerja yang buruk [3][6]. Dalam situasi ini, pengembangan lebih lanjut diperlukan untuk memastikan keandalan sistem deteksi dalam kondisi pencahayaan yang berbeda. Seiring kemajuan teknologi, EfficientNet berkembang menjadi arsitektur yang memberikan keseimbangan antara akurasi dan efisiensi komputasi [7]. Model ini menunjukkan kinerja yang stabil dalam berbagai kondisi pencahayaan, menjadikannya alternatif yang menarik untuk tugas deteksi objek di lingkungan transportasi [8][9][10]. Namun demikian, masih terdapat tantangan, terutama yang berkaitan dengan penurunan kinerja pada gambar beresolusi rendah, yang menunjukkan bahwa masih ada ruang untuk optimalisasi lebih lanjut dalam situasi ini. Perbandingan antara Faster R-CNN dan EfficientNet menunjukkan bahwa masing-masing arsitektur memiliki kelebihan dan kekurangan.

Perbandingan tersebut ditunjukkan dengan, Faster R-CNN lebih unggul dalam situasi yang memerlukan deteksi cepat dan akurat, terutama pada gambar dengan *noise* visual tinggi [11]. Namun, efisiensi penyimpanan dan daya komputasi EfficientNet menjadikannya pilihan yang menarik dalam kondisi tertentu [12]. Hal ini menggambarkan pentingnya memilih arsitektur yang sesuai berdasarkan tujuan deteksi.

Untuk mengatasi keterbatasan masing-masing arsitektur, berbagai pendekatan telah diusulkan,

termasuk kombinasi Faster R-CNN dengan teknik augmentasi data untuk meningkatkan kinerja deteksi pada kondisi pencahayaan yang buruk [13][14]. Pendekatan ini menunjukkan bahwa dengan augmentasi yang tepat, tantangan dalam kondisi pencahayaan rendah atau kontras tinggi dapat diatasi, sehingga meningkatkan keandalan sistem deteksi [6][15][16].

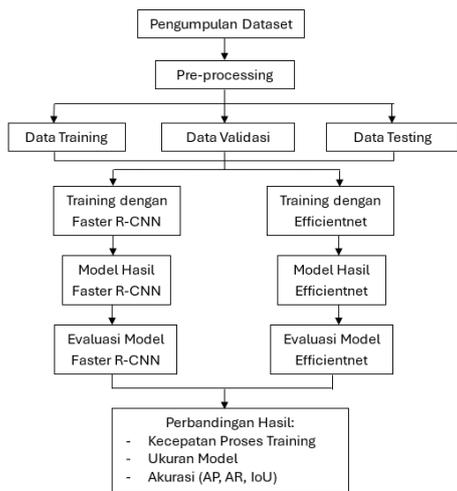
Sementara itu, pengembangan lebih lanjut pada EfficientNet menunjukkan potensinya dalam menghadapi variasi kualitas gambar yang disebabkan oleh perubahan cuaca. Meskipun kecepatan pemrosesan tetap menjadi tantangan, kemampuan untuk mempertahankan akurasi deteksi yang tinggi pada kondisi yang sulit menjadikan EfficientNet sebagai kandidat kuat untuk aplikasi dalam kondisi lingkungan yang kompleks [7][17][18].

Sebagai upaya untuk menggabungkan keunggulan dari kedua arsitektur ini, beberapa penelitian terbaru telah mengembangkan model hybrid yang mengintegrasikan fitur-fitur terbaik dari Faster R-CNN dan EfficientNet [19][20]. Model ini menunjukkan hasil yang menjanjikan dalam meningkatkan akurasi deteksi dan mengurangi kesalahan deteksi, terutama pada gambar dengan variasi pencahayaan yang ekstrem [9][21][22].

Dengan dasar tersebut, penelitian ini bertujuan untuk melakukan komparasi menyeluruh antara Faster R-CNN dan EfficientNet dalam konteks deteksi kereta api di berbagai kondisi pencahayaan dan kualitas gambar. Penelitian ini diharapkan dapat memberikan wawasan lebih lanjut mengenai kelebihan dan kekurangan masing-masing arsitektur, serta memberikan kontribusi signifikan dalam pengembangan sistem deteksi kereta api yang lebih andal dan efisien di Indonesia, yang memiliki tantangan lingkungan yang kompleks dan beragam.

2. METODE PENELITIAN

Pada penelitian ini menggunakan 4500 gambar kereta api, yaitu gambar kereta dari berbagai kondisi dan pencahayaan. Blok diagram sistem pada penelitian ini ditunjukkan pada gambar 1.



Gambar 1. Blok Diagram Perbandingan Performa Faster R-CNN Dan EfficientNet

Berdasarkan Gambar 1, langkah yang dilakukan pada penelitian ini dimulai dengan pengumpulan dataset. Setelah dataset dibagi dalam data *training*, data validasi, dan data *testing*, dilakukan perbandingan Arsitektur CNN. Pada penelitian ini, membandingkan 2 Arsitektur CNN, yaitu Faster R-CNN dan EfficientNet. Adapun aspek hasil perbandingan dari kedua arsitektur tersebut adalah kecepatan proses training, ukuran model, dan akurasi.

2.1. Dataset

Dataset yang digunakan dalam penelitian ini berasal dari berbagai sumber untuk memastikan keragaman dan representativitas gambar kereta api dalam berbagai kondisi. Sumber data meliputi foto kereta api yang diambil menggunakan kamera, tangkapan gambar dari video kereta api di YouTube, gambar kereta api yang diambil dari hasil *browsing*, dan dataset yang tersedia di platform Roboflow. Berdasarkan sumber dataset tersebut, dataset yang terkumpul sebanyak 1500 gambar kereta api.

Kondisi dataset gambar kereta api pada penelitian ini dikelompokkan menjadi 3, yaitu gambar dengan kondisi pencahayaan yang cukup, gambar dengan kondisi pencahayaan yang rendah, dan gambar dengan tingkat *noise* yang tinggi. Contoh dataset yang digunakan pada penelitian ini ditunjukkan pada Tabel 1.

Tabel 1. Contoh Variasi Dataset Kereta Api

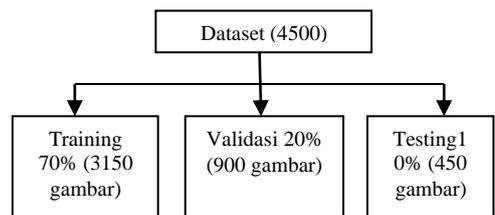
Gambar Kereta Api	Kondisi Dataset
	Pencahayaan yang baik

Gambar Kereta Api	Kondisi Dataset
	Pencahayaan yang kurang baik
	Terdapat Banyak Noise

Tahap preprocessing data gambar kereta api dilakukan dengan cara *me-resize* gambar menjadi ukuran 416 x 416 piksel dan dilakukan *auto orientation* untuk memperbaiki orientasi gambar yang mungkin tidak konsisten. Tahap *preprocessing* perlu dilakukan untuk memastikan konsistensi dalam pengulahan data pada tahap selanjutnya.

Setelah itu dilakukan proses augmentasi data untuk menambah variasi data dan meningkatkan kemampuan generalisasi model. Proses *Augmentasi* melibatkan beberapa Teknik, antara lain penambahan efek blur, rotasi, penyesuaian kecerahan, serta penambahan *noise* pada gambar. Proses *augmentasi* yang dilakukan pada penelitian ini menghasilkan peningkatan jumlah dataset, yaitu 4500 gambar kereta api. Sehingga memungkinkan model untuk dilatih dengan data yang lebih bervariasi dan mencerminkan kondisi dunia nyata yang lebih beragam.

Pada penelitian dilakukan pembagian antara data training, data validasi, dan data testing, dengan perbandingan adalah 70% data training, 20% data validasi, dan 10% data testing. Pembagian dataset ditunjukkan pada Gambar 2.



Gambar 2. Pembagian Dataset

2.2. Proses Training

Pada penelitian ini dilakukan perbandingan performa dalam proses deteksi obyek antara arsitektur EfficientNet dan Faster R-CNN. Namun, perlu dilakukan penyesuaian khusus pada arsitektur EfficientNet. Secara default, arsitektur EfficientNet dirancang sebagai model klasifikasi, bukan deteksi. Oleh karena itu, agar arsitektur EfficientNet dapat berfungsi sebagai model deteksi, dilakukan modifikasi menggunakan TensorFlow's Object Detection API. Melalui pendekatan ini, arsitektur EfficientNet diubah menjadi model deteksi yang

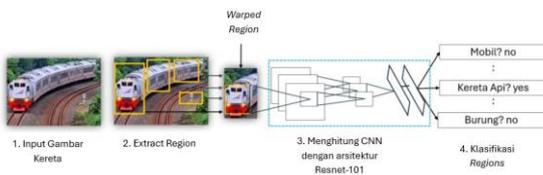
mampu mengidentifikasi dan melokalisasi objek kereta api dalam gambar.

Arsitektur Faster R-CNN secara default merupakan arsitektur deteksi objek, digunakan sebagai pembanding tanpa memerlukan modifikasi tambahan. Kedua arsitektur tersebut kemudian dilatih menggunakan dataset yang sama, yang telah mengalami proses augmentasi untuk memastikan konsistensi dalam proses evaluasi performa. Selain itu, untuk menjaga keseragaman dan menghindari bias dalam perbandingan, pengaturan *hyperparameter* yang sama diterapkan pada kedua arsitektur selama proses training. Pengaturan *hyperparameter* yang dilakukan antara lain *learning rate*, *batch size*, dan jumlah *epoch* disesuaikan agar sesuai dengan arsitektur. Namun, tetap konsisten di antara kedua arsitektur tersebut. Pengaturan ini memastikan bahwa perbandingan performa didasarkan pada kemampuan intrinsik model dalam mendeteksi objek, bukan pada perbedaan dalam konfigurasi training.

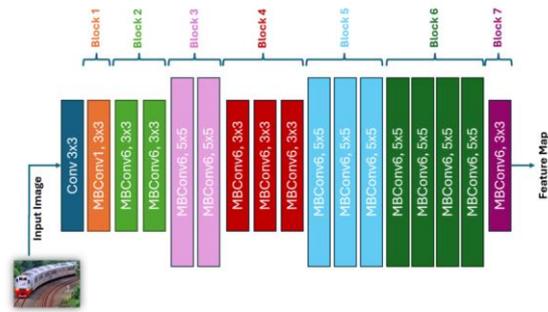
Dengan pendekatan ini, dapat dilakukan evaluasi kemampuan kedua arsitektur dalam mendeteksi objek kereta api pada berbagai kondisi pencahayaan dan *noise*, serta membandingkan efisiensi dan akurasi keduanya dalam mendeteksi objek.

2.2. Arsitektur CNN

Pada penelitian menggunakan 2 arsitektur CNN yang digunakan untuk mendeteksi obyek kereta api dengan berbagai kondisi dan pencahayaan. Arsitektur yang digunakan dalam penelitian ini adalah Faster R-CNN dengan arsitektur Resnet-101 dan EfficientNet. Arsitektur yang digunakan dalam penelitian ini meliputi Faster R-CNN dan EfficientNet. Setelah dilakukan pembagian dataset menjadi training, validasi, dan testing, dilakukan proses pelatihan (training) untuk masing-masing arsitektur. Hasil dari proses pelatihan ini adalah model yang dihasilkan oleh arsitektur Faster R-CNN dan EfficientNet. Selanjutnya, dilakukan evaluasi terhadap kedua model tersebut. Berdasarkan hasil evaluasi, dilakukan perbandingan terhadap kecepatan proses pelatihan, ukuran model, dan akurasi masing-masing arsitektur. Adapun arsitektur Faster R-CNN ditunjukkan pada Gambar 3, Arsitektur EfficientNet ditunjukkan pada Gambar 4.



Gambar 3. Arsitektur Faster R-CNN



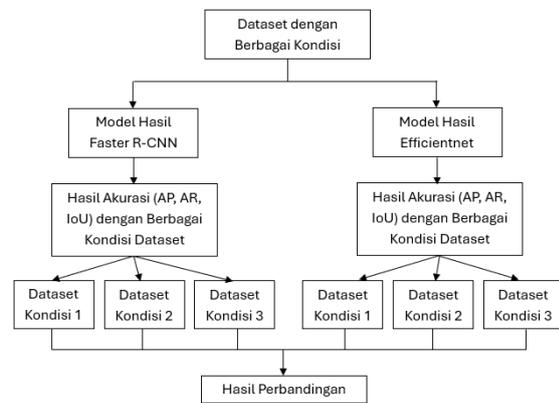
Gambar 4. Arsitektur EfficientNet

2.3. Berbandingan Akurasi Variasi Dataset

Terdapat tiga variasi dataset yang digunakan pada penelitian ini, yaitu:

1. Dataset Kondisi 1: Dataset gambar kereta dengan pencahayaan yang baik
2. Dataset Kondisi 2: Dataset gambar kereta dengan pencahayaan yang kurang baik
3. Dataset Kondisi 3: Dataset gambar kereta dengan kondisi banyak *noise*. Skenario yang digunakan pada penelitian ini adalah menggambarkan kondisi saat hujan.

Ketiga kondisi tersebut masing-masing diujikan pada Model hasil Faster R-CNN dan EfficientNet. Kemudian hasil akurasi dibandingkan pada setiap kondisi dataset. Sehingga, diperoleh hasil setiap akurasi pada masing-masing arsitektur Faster R-CNN dan EfficientNet. Adapun blok diagram variasi dataset yang diterapkan pada penelitian ini ditunjukkan pada Gambar 5.



Gambar 5. Blok Diagram Perbandingan Akurasi Variasi Dataset

2.4. Evaluasi

Evaluasi kinerja model dilakukan menggunakan metrik *Intersection over Union (IoU)*, *Average Precision (AP)*, dan *Average Recall (AR)*. Metrik ini dipilih untuk menilai sejauh mana prediksi model mendekati *ground truth* serta seberapa baik model mampu mendeteksi objek secara konsisten dalam berbagai kondisi lingkungan.

IoU adalah rasio antara area tumpang tindih (intersection) antara kotak prediksi (predicted bounding box) dan kotak sebenarnya (ground truth bounding box) terhadap area gabungan (union) dari

kedua kotak tersebut [23]. Secara matematis, IoU dihitung sebagai:

$$IoU = \frac{\text{Area of Intersection}}{\text{Area of Union}} \quad (1)$$

Dimana:

1. *Area of Intersection*: Merupakan area di mana kotak prediksi dan kotak sebenarnya saling tumpang tindih.
2. *Area of Union*: Merupakan gabungan dari area kotak prediksi dan kotak sebenarnya.

Average Recall (AR) adalah ukuran yang mengukur seberapa baik model dapat mendeteksi semua objek yang ada dalam gambar, terlepas dari threshold yang digunakan. AR dihitung sebagai rata-rata dari recall pada berbagai nilai *Intersection over Union (IoU)* thresholds, yang mengukur seberapa baik area prediksi tumpang tindih dengan area sebenarnya dari objek.

Average Precision (AP) adalah ukuran yang menggabungkan *presisi* dan *recall* pada berbagai threshold untuk menentukan seberapa baik model dapat memprediksi objek yang benar dalam gambar [24]. AP diukur dengan menghitung area di bawah kurva *Precision-Recall (PR curve)*, yang merupakan grafik antara presisi (jumlah prediksi benar dari total prediksi) dan *recall* (jumlah prediksi benar dari total objek sebenarnya).

Hasil penelitian ini bertujuan untuk mengevaluasi ketepatan dan keandalan kedua model dalam berbagai kondisi lingkungan yang mungkin ditemui dalam aplikasi nyata. Pada penelitian ini analisis dilakukan terhadap kondisi pencahayaan dan adanya *noise*, berupa hujan, yang dapat mempengaruhi kinerja deteksi dari masing-masing model. Selain itu, perbandingan ini diharapkan dapat memberikan pengetahuan mengenai kekuatan dan kelemahan dari model yang diuji dalam konteks deteksi objek dengan tingkat kesulitan yang tinggi.

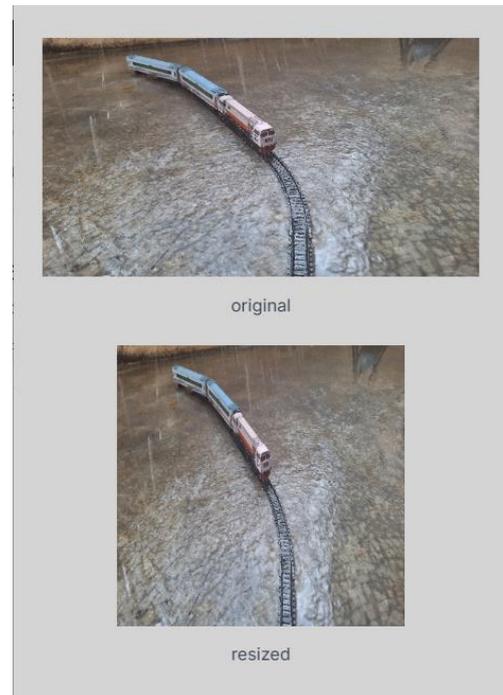
3. HASIL DAN PEMBAHASAN

Dalam penelitian ini, dilakukan perbandingan performa antara model EfficientNet dan Faster R-CNN dalam mendeteksi satu kelas objek, yaitu kereta api, pada berbagai kondisi pencahayaan dan tingkat *noise*. Percobaan ini dilakukan menggunakan dataset yang terdiri dari 4500 gambar, yang dibagi menjadi 70% untuk *training*, 20% untuk validasi, dan 10% untuk *testing*. Kondisi pencahayaan yang diuji mencakup pencahayaan baik, pencahayaan kurang kurang baik, serta adanya *noise* dalam bentuk hujan.

3.1. Hasil Preprocessing

Tahap *preprocessing* data gambar kereta api dilakukan dengan cara *re-size* gambar menjadi ukuran 416 x 416 piksel dan dilakukan auto orientation untuk memperbaiki orientasi gambar

yang mungkin tidak konsisten. Hasil *resize* ditunjukkan pada Gambar 6.



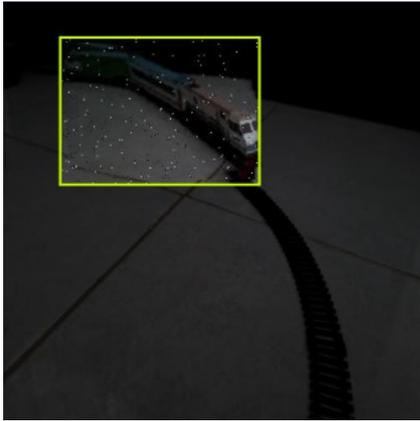
Gambar 6. Hasil *Resize Image*

Kemudian dilakukan proses augmentasi data untuk menambah variasi data penelitian. Adapun teknik yang dilakukan untuk proses augmentasi antara lain *blur* sampai 2.5px, rotasi 90° serah dan berlawanan arah jarum jam, *cropping*, rotasi antara -15° dan +15°, *brightness*, dan *noise* 1.32% pixels.

Proses Augmentasi melibatkan beberapa Teknik, antara lain penambahan efek blur, rotasi, penyesuaian kecerahan, serta penambahan *noise* pada gambar. Beberapa hasil *preprocessing* ditunjukkan pada Gambar 7 dan Gambar 8.



Gambar 7. Hasil Rotasi Citra



Gambar 8. Hasil Penambahan Noise

Adanya tahap *preprocessing* menunjukkan hasil pengujian model bisa mengatasi berbagai keadaan didunia nyata.

3.2. Hasil Training

Berdasarkan hasil percobaan yang telah dilakukan, terdapat perbedaan yang signifikan antara arsitektur EfficientNet dan Faster R-CNN dalam hal kecepatan training dan ukuran model. EfficientNet menunjukkan waktu *training* yang lebih singkat, yaitu 5 jam 37 menit, dibandingkan dengan Faster R-CNN yang memerlukan 9 jam 5 menit. Selain itu, ukuran model EfficientNet lebih kecil, yaitu 321.2 MB, jika dibandingkan dengan Faster R-CNN yang mencapai 475.7 MB. Ukuran model yang lebih kecil pada EfficientNet menunjukkan kemampuannya dalam menyediakan efisiensi ruang penyimpanan tanpa mengorbankan performa, sementara Faster R-CNN, dengan arsitektur yang lebih kompleks, memerlukan ruang penyimpanan yang lebih besar.

Tabel 2. Perbandingan Hasil Kecepatan Training dan Ukuran Model

Indikator performance	EfficientNet	Faster R-CNN
Kecepatan Training	5 jam 37 menit	9 jam 5 menit
Ukuran Model	321.2 MB	475.7 MB

EfficientNet memerlukan waktu yang lebih singkat untuk training dibandingkan dengan Faster R-CNN. Hal ini menunjukkan bahwa EfficientNet lebih efisien dalam hal penggunaan sumber daya komputasi atau lebih cepat dalam konvergensi dibandingkan dengan Faster R-CNN.

Berdasarkan arsitekturnya, EfficientNet dirancang untuk efisiensi komputasi dengan mengoptimalkan keseimbangan antara akurasi dan kecepatan. Sedangkan Faster R-CNN, dengan arsitektur yang lebih kompleks dan menggunakan komponen seperti *Region Proposal Networks* (RPN), memerlukan lebih banyak waktu untuk proses *training*. Berdasarkan Parameter dan Kinerjanya, EfficientNet memiliki lebih sedikit parameter dibandingkan Faster R-CNN, yang dapat

menyebabkan konvergensi lebih cepat dalam *training*.

EfficientNet menghasilkan model dengan ukuran file yang lebih kecil dibandingkan dengan Faster R-CNN. Beberapa faktor yang mempengaruhi perbedaan ukuran ini adalah

1. Arsitektur Model

EfficientNet menggunakan teknik kompresi dan optimisasi yang memungkinkan model menjadi lebih kecil tanpa mengorbankan performa secara signifikan. Faster R-CNN, terutama jika menggunakan *backbone* seperti ResNet 101, memiliki ukuran model yang lebih besar karena komponen-komponen tambahan seperti RPN dan *classifier*.

2. Parameter

Meskipun Faster R-CNN memiliki jumlah parameter yang lebih besar, hal ini menyebabkan ukuran model yang lebih besar. Efisiensi dalam desain EfficientNet juga berkontribusi pada ukuran file yang lebih kecil.

EfficientNet lebih cocok untuk situasi yang memerlukan waktu training cepat dan ukuran model kecil, seperti pada perangkat dengan sumber daya terbatas atau aplikasi mobile. Sebaliknya, Faster R-CNN lebih sesuai untuk aplikasi yang membutuhkan deteksi objek yang lebih akurat dengan fitur-fitur kompleks, meskipun memerlukan waktu training lebih lama dan ukuran model yang lebih besar. Sehingga, EfficientNet memberikan waktu training yang lebih cepat, bermanfaat untuk iterasi yang lebih efisien atau ketika sumber daya komputasi terbatas, serta menghasilkan model yang lebih kecil, sehingga memungkinkan penggunaan pada perangkat dengan memori terbatas.

3.3. Hasil Average Precision (AP) pada Berbagai Level IoU

Perbandingan kinerja deteksi objek pada EfficientNet dan Faster R-CNN berdasarkan hasil pengujian menggunakan *Average Precision* (AP) pada berbagai Level IoU ditunjukkan pada Tabel 3.

Tabel 3. Perbandingan Hasil AP pada Berbagai IoU

Average Precision	EfficientNet	Faster R-CNN
(AP) @[IoU=0.50:0.95 area= all maxDets=100]	0.80	0.825
(AP) @[IoU=0.50 area= all maxDets=100]	0.970	0.983
(AP) @[IoU=0.75 area= all maxDets=100]	0.930	0.962
(AP) @[IoU=0.50:0.95 area= small maxDets=100]	0.780	0.806
(AP) @[IoU=0.50:0.95 area=medium maxDets=100]	0.790	0.820
(AP) @[IoU=0.50:0.95 area= large maxDets=100]	0.800	0.828

Berdasarkan Tabel 3, beberapa metrik Average Precision (AP), Faster R-CNN secara konsisten menunjukkan keunggulan dibandingkan EfficientNet. Pada metrik AP @[IoU=0.50:0.95 |

area=all | maxDets=100], Faster R-CNN mencapai skor 0.825, sedikit lebih tinggi daripada EfficientNet yang mencetak 0.80. Hal ini mengindikasikan bahwa Faster R-CNN lebih andal dalam mendeteksi objek dengan berbagai tingkat IoU, mencerminkan kemampuannya dalam menangani variasi konteks yang lebih luas.

Pada tingkat IoU yang lebih rendah (AP @[IoU=0.50 | area=all | maxDets=100]), kedua model menunjukkan performa yang sangat baik, tetapi Faster R-CNN tetap unggul dengan skor 0.983 dibandingkan 0.970 untuk EfficientNet. Ini menunjukkan bahwa Faster R-CNN memiliki ketepatan deteksi yang lebih baik meskipun dalam situasi dengan kesepakatan yang lebih rendah antara prediksi dan *ground truth*.

Ketika metrik dipersempit pada tingkat IoU yang lebih ketat (AP @[IoU=0.75 | area=all | maxDets=100]), performa Faster R-CNN tetap unggul dengan skor 0.962 dibandingkan 0.930 untuk EfficientNet. Keunggulan ini menegaskan kemampuan Faster R-CNN dalam mendeteksi objek dengan akurasi tinggi, terutama ketika diperlukan kesesuaian yang lebih presisi antara prediksi dan objek sebenarnya.

Dalam analisis berdasarkan ukuran objek, Faster R-CNN menunjukkan keunggulan dalam mendeteksi objek kecil (AP @[IoU=0.50:0.95 | area=small | maxDets=100]) dengan skor 0.806 dibandingkan dengan 0.780 untuk EfficientNet. Demikian pula, pada objek berukuran medium (AP @[IoU=0.50:0.95 | area=medium | maxDets=100]) dan besar (AP @[IoU=0.50:0.95 | area=large | maxDets=100]), Faster R-CNN terus menunjukkan keunggulannya dengan masing-masing skor 0.820 dan 0.828, dibandingkan dengan 0.790 dan 0.800 untuk EfficientNet.

Secara keseluruhan, hasil ini menunjukkan bahwa Faster R-CNN memiliki keunggulan dalam mendeteksi objek pada berbagai ukuran dan tingkat kesepakatan *IoU*, menegaskan keandalannya dalam berbagai situasi deteksi yang lebih kompleks.

3.4. Hasil Average Recall (AR) pada Berbagai Kondisi Deteksi

Dalam evaluasi performa model deteksi objek, penelitian ini membandingkan hasil *Average Recall (AR)* antara model EfficientNet dan Faster R-CNN dalam berbagai kondisi deteksi. Pengujian dilakukan dengan mempertimbangkan batas maksimum deteksi (*maxDets*) yang berbeda serta area objek yang bervariasi dari kecil hingga besar.

Tabel 4. Perbandingan Hasil Average Recall (AR) pada Berbagai Kondisi Deteksi

Average Recall	EfficientNet	Faster R-CNN
(AR) @[IoU=0.50:0.95 area=all maxDets= 1]	0.825	0.855
(AR) @[IoU=0.50:0.95 area=all maxDets= 10]	0.840	0.855

(AR) @[IoU=0.50:0.95 area=all maxDets=100]	0.850	0.855
(AR) @[IoU=0.50:0.95 area=small maxDets=100]	0.810	0.827
(AR) @[IoU=0.50:0.95 area=medium maxDets=100]	0.825	0.831
(AR) @[IoU=0.50:0.95 area=large maxDets=100]	0.845	0.858

Tabel 4 menunjukkan bahwa Faster R-CNN secara konsisten menunjukkan performa yang unggul dibandingkan EfficientNet, meskipun perbedaannya kecil. Pada metrik AR @[IoU=0.50:0.95 | area=all | maxDets=1], Faster R-CNN mencatat nilai 0.855, sedikit lebih tinggi daripada EfficientNet yang memperoleh nilai 0.825. Ini mengindikasikan bahwa Faster R-CNN memiliki keunggulan dalam mendeteksi objek ketika hanya satu deteksi yang diperbolehkan, dengan *recall* yang lebih baik pada level *IoU* yang lebih rendah.

Ketika jumlah maksimum deteksi ditingkatkan menjadi 10 (AR @[IoU=0.50:0.95 | area=all | maxDets=10]), Faster R-CNN tetap menunjukkan performa yang lebih baik dengan nilai AR 0.855, dibandingkan dengan EfficientNet yang memperoleh 0.840. Hal ini menunjukkan bahwa Faster R-CNN lebih konsisten dalam mendeteksi beberapa objek secara bersamaan dengan *recall* yang lebih tinggi.

Pada metrik AR @[IoU=0.50:0.95 | area=all | maxDets=100], perbedaan antara kedua model semakin kecil, dengan Faster R-CNN memperoleh nilai AR 0.855 dan EfficientNet mencapai 0.850. Meskipun performa kedua model hampir sama dalam mendeteksi banyak objek, Faster R-CNN tetap sedikit lebih unggul.

Dalam hal deteksi objek kecil (AR @[IoU=0.50:0.95 | area=small | maxDets=100]), Faster R-CNN menunjukkan keunggulan dengan nilai AR 0.827 dibandingkan EfficientNet yang memperoleh nilai 0.810. Ini menunjukkan bahwa Faster R-CNN lebih efektif dalam mendeteksi objek kecil dengan *recall* yang lebih baik.

Pada kategori objek medium (AR @[IoU=0.50:0.95 | area=medium | maxDets=100]), Faster R-CNN mencatat nilai AR 0.831, sedikit lebih tinggi dari pada EfficientNet yang memperoleh nilai 0.825. Meskipun perbedaannya kecil, hal ini mengindikasikan bahwa Faster R-CNN memiliki keunggulan dalam mendeteksi objek berukuran sedang.

Untuk objek besar (AR @[IoU=0.50:0.95 | area=large | maxDets=100]), kedua model menunjukkan performa yang sangat baik, dengan EfficientNet memperoleh nilai AR 0.845 dan Faster R-CNN mencapai 0.858. Meskipun Faster R-CNN sedikit lebih unggul, kedua model dapat mendeteksi objek besar dengan tingkat *recall* yang tinggi. Secara keseluruhan, hasil evaluasi ini menunjukkan bahwa Faster R-CNN sedikit lebih unggul dibandingkan EfficientNet dalam berbagai skenario deteksi, terutama dalam hal *recall* pada berbagai ukuran objek dan jumlah deteksi.

3.5. Pengujian Performa Model Deteksi

Penelitian ini melakukan pengujian performa model dalam berbagai kondisi pencahayaan dan gangguan lingkungan. Evaluasi performa deteksi objek dari model EfficientNet dan Faster R-CNN pada tiga kondisi pencahayaan yang berbeda yaitu, pencahayaan yang baik, pencahayaan yang kurang baik, dan kondisi dengan *noise* hujan. Pengujian dilakukan menggunakan 100 gambar untuk setiap

kondisi pencahayaan tersebut. Evaluasi dilakukan menggunakan metrik *Average Precision (AP)* dan *Average Recall (AR)* untuk memberikan gambaran menyeluruh mengenai akurasi dan kemampuan deteksi kedua model dalam berbagai situasi. Metrik ini membantu menilai efektivitas model dalam mendeteksi objek dengan tepat dan konsisten dalam kondisi yang berbeda-beda. Hasil Evaluasi ditunjukkan pada Tabel 5.

Tabel 4. Evaluasi AP dan AR Pada Berbagai Kondisi

AP dan AR	Pencahayaannya Cukup		Pencahayaannya Kurang		Dengan Noise (Hujan)	
	EfficientNet	Faster R-CNN	EfficientNet	Faster R-CNN	EfficientNet	Faster R-CNN
AP @[IoU=0.50:0.95 area=all]	0.837	0.844	0.786	0.810	0.747	0.798
AR @[IoU=0.50:0.95 area=all maxDets=100]	0.881	0.892	0.843	0.869	0.799	0.851

Berdasarkan Tabel 4, kinerja deteksi objek di pada berbagai kondisi pencahayaan dan gangguan *noise*, baik EfficientNet maupun Faster R-CNN menunjukkan variasi performa dalam menangani situasi berbeda.

Pada kondisi pencahayaan yang cukup, nilai *Average Precision (AP)* menunjukkan bahwa Faster R-CNN sedikit lebih unggul dengan skor 0.844 dibandingkan EfficientNet yang memperoleh 0.837. Meskipun perbedaannya tipis, kedua model menunjukkan performa yang baik, mampu mendeteksi objek dengan akurat dalam kondisi pencahayaan optimal. *Average Recall (AR)* juga memperkuat keunggulan Faster R-CNN, dengan nilai 0.892 dibandingkan dengan 0.881 pada EfficientNet. Hal tersebut menunjukkan bahwa Faster R-CNN memiliki kemampuan yang lebih baik dalam mendeteksi semua objek dalam gambar dengan pencahayaan yang cukup.

Ketika pencahayaan berkurang, perbedaan antara kedua model menjadi lebih nyata. Faster R-CNN mencatat nilai AP 0.810, lebih tinggi dibandingkan dengan 0.786 yang diperoleh EfficientNet. Hal tersebut menunjukkan keunggulan Faster R-CNN dalam mempertahankan akurasi deteksi objek meskipun kondisi pencahayaan tidak ideal. Pada metrik AR, Faster R-CNN juga menunjukkan performa yang lebih baik dengan nilai 0.869, sementara EfficientNet memperoleh 0.843. Hal tersebut mengindikasikan bahwa Faster R-CNN lebih mampu mengatasi penurunan kualitas gambar yang disebabkan oleh kurangnya pencahayaan, dengan AR yang lebih tinggi.

Dalam kondisi dengan gangguan *noise* hujan, performa deteksi objek dari kedua model mengalami penurunan yang lebih signifikan. Faster R-CNN tetap unggul dengan nilai AP 0.798 dibandingkan dengan 0.747 pada EfficientNet, menunjukkan bahwa Faster R-CNN lebih tahan terhadap gangguan *noise* dan dapat mendeteksi objek dengan lebih akurat dalam kondisi yang tidak ideal. Nilai AR di bawah *noise* hujan juga menegaskan keunggulan Faster R-CNN, dengan skor 0.851 dibandingkan dengan 0.799 pada EfficientNet. Hasil ini

menunjukkan bahwa Faster R-CNN tidak hanya lebih efektif dalam mendeteksi objek, tetapi juga lebih handal dalam memastikan deteksi yang akurat meskipun ada gangguan *noise*. Contoh hasil deteksi ditunjukkan pada Gambar 9 dan 10.



Gambar 9. Hasil Deteksi Menggunakan Efficientnet



Gambar 10. Hasil Deteksi Menggunakan Faster R-CNN

Secara keseluruhan, analisis ini menegaskan bahwa meskipun kedua model memiliki kemampuan deteksi yang kuat, Faster R-CNN lebih konsisten dalam menunjukkan performa yang lebih baik, terutama dalam situasi pencahayaan yang kurang ideal dan dengan adanya gangguan *noise*, menjadikannya pilihan yang lebih unggul untuk aplikasi dalam kondisi lingkungan yang beragam.

4. DISKUSI

Pada penelitian ini dilakukan perbandingan Model EfficientNet dan Faster R-CNN dalam aspek kecepatan proses *training*, ukuran model, serta Akurasi menggunakan evaluasi AP, AR, dan IoU.

Adapun arsitektur yang digunakan pada Faster R-CNN adalah Rester-101. Penelitian sebelumnya telah dilakukan deteksi kereta api menggunakan Faster R-CNN dengan arsitektur VGG-16. Pengujian dataset dilakukan dengan menerapkan keberagaman kondisi cahaya, yaitu gambar kereta pada siang, sore, dan malam hari. Adapun evaluasi yang digunakan adalah *confusion matrix*, dengan rata-rata deteksi objek kereta api sebesar 87,83% [10]. Penelitian yang sebelumnya juga dilakukan dengan menerapkan Faster R-CNN untuk mendeteksi sepeda motor dengan akurasi 95% [9].

Dalam banyak penelitian model EfficientNet sering digunakan untuk proses klasifikasi. Namun, pada penelitian ini, Model EfficientNet dimodifikasi menjadi model deteksi menggunakan TensorFlow's Object Detection API. Penelitian sebelumnya tentang klasifikasi penyakit daun tanaman menggunakan EfficientNet, menunjukkan nilai akurasi yang cukup tinggi yaitu, 99,71% [25]. Model EfficientNet menghasilkan akurasi yang tinggi dalam kasus klasifikasi, karena secara *default* dikembangkan untuk klasifikasi.

Pada penelitian ini digunakan metode evaluasi *IoU*, *AP*, dan *AR*. *IoU*, *AP*, dan *AR* lebih efektif digunakan daripada *confusion matrix* untuk menilai kinerja model pada dataset yang beragam. Ketiga metrik tersebut memberikan penilaian yang lebih tepat terhadap kualitas deteksi objek yang beragam [16][26]. *IoU* mengukur seberapa baik prediksi model sesuai dengan lokasi objek yang sebenarnya, yang sangat penting dalam kondisi pencahayaan dan *noise* yang bervariasi [27]. Sementara itu, *AP* dan *AR* menawarkan evaluasi yang lebih komprehensif dengan mempertimbangkan kinerja model pada berbagai *threshold*, memungkinkan analisis yang lebih mendalam terhadap kemampuan model dalam mendeteksi objek di berbagai kondisi beragam [26]. Dibandingkan dengan *confusion matrix*, yang hanya memberikan gambaran umum tentang benar atau salahnya prediksi, *IoU*, *AP*, dan *AR* memberikan informasi yang lebih relevan dan detail, terutama dalam konteks deteksi objek di lingkungan yang kompleks.

Secara umum pada penelitian ini, Faster R-CNN menunjukkan performa yang lebih unggul dibandingkan dengan EfficientNet pada berbagai metrik *Average Precision*, terutama dalam hal akurasi deteksi objek pada berbagai ukuran dan tingkat kesepakatan. Meskipun EfficientNet menunjukkan performa yang baik, terutama pada level $IoU=0.50$, Faster R-CNN memiliki keunggulan dalam mendeteksi objek dengan detail dan akurasi yang lebih tinggi, serta pada berbagai ukuran objek.

Berdasarkan analisis, Faster R-CNN umumnya menunjukkan nilai *Average Recall* yang lebih tinggi dibandingkan EfficientNet, terutama dalam kondisi deteksi dengan batas maksimum yang lebih rendah dan objek kecil. EfficientNet mendemonstrasikan performa yang kompetitif, tetapi Faster R-CNN

memiliki keunggulan dalam berbagai kondisi deteksi, terutama dalam hal recall untuk objek kecil dan dengan batas maksimum deteksi yang lebih tinggi.

Faster R-CNN menunjukkan kinerja yang lebih konsisten dan lebih baik dibandingkan EfficientNet di berbagai kondisi pencahayaan dan dengan adanya *noise* hujan. Perbedaan ini menyoroti kelebihan Faster R-CNN dalam menangani berbagai kondisi lingkungan yang menantang, sementara EfficientNet, meskipun performanya baik dalam kondisi optimal, menunjukkan penurunan yang lebih besar dalam situasi dengan pencahayaan kurang atau *noise*.

5. KESIMPULAN

Model EfficientNet dan Faster R-CNN memiliki kelebihan dan kekurangan masing-masing dalam berbagai aspek. Dalam hal efisiensi, EfficientNet lebih unggul dengan waktu training yang lebih cepat (5 jam 37 menit) dan ukuran model yang lebih kecil (321.2 MB). Sehingga, menjadikan pilihan yang tepat untuk perangkat dengan keterbatasan sumber daya.

Namun, Faster R-CNN unggul dalam hal kinerja deteksi, dengan nilai *Average Precision (AP)* dan *Average Recall (AR)* yang lebih tinggi di berbagai kondisi pencahayaan dan gangguan *noise*. Model ini menjadi pilihan yang lebih baik untuk aplikasi yang memerlukan akurasi tinggi dan ketahanan terhadap variasi lingkungan.

Secara keseluruhan, meskipun EfficientNet menawarkan keuntungan dalam hal efisiensi ukuran model dan waktu *training*, Faster R-CNN menunjukkan keunggulan dalam kinerja deteksi objek dan daya tahan terhadap berbagai kondisi lingkungan. Pemilihan antara kedua model ini harus mempertimbangkan *trade-off* antara kecepatan dan ukuran model terhadap akurasi deteksi dan kemampuan beradaptasi dalam kondisi yang bervariasi.

UCAPAN TERIMA KASIH

Peneliti mengucapkan terima kasih kepada Direktorat Riset, Teknologi, dan Pengabdian kepada Masyarakat Kementerian Pendidikan, Kebudayaan, Riset, dan Teknologi atas dukungan dan pendanaan yang diberikan. Bantuan ini sangat berharga dalam mendukung pelaksanaan penelitian ini, serta memungkinkan peneliti untuk mencapai hasil yang signifikan dalam upaya berkontribusi pada pengembangan ilmu pengetahuan dan teknologi di Indonesia.

DAFTAR PUSTAKA

- [1] T. Bai, J. Yang, G. Xu, and D. Yao, "An optimized railway fastener detection method based on modified Faster R-CNN," *Measurement*, vol. 182, p. 109742, 2021.

- [2] G. Rachmad, Y. Yulianto, and M. Nurul, "PALANG PINTU PERLINTASAN KERETA API OTOMATIS YANG TERINTEGRASI BERBASIS DATA GPS," *Kohesi J. Sains dan Teknol.*, vol. 2, no. 9, pp. 1–10, 2024.
- [3] W. Li, "Analysis of object detection performance based on Faster R-CNN," in *Journal of Physics: Conference Series*, 2021, vol. 1827, no. 1, p. 12085.
- [4] R. F. Alya, M. Wibowo, and P. Paradise, "Classification of Batik Motif Using Transfer Learning on Convolutional Neural Network (CNN)," *J. Tek. Inform.*, vol. 4, no. 1, pp. 161–170, 2023, doi: 10.52436/1.jutif.2023.4.1.564.
- [5] A. Jiang, R. Noguchi, and T. Ahamed, "Tree trunk recognition in orchard autonomous operations under different light conditions using a thermal camera and faster R-CNN," *Sensors*, vol. 22, no. 5, p. 2065, 2022.
- [6] K. Wang and M. Z. Liu, "Object recognition at night scene based on DCGAN and faster R-CNN," *IEEE Access*, vol. 8, pp. 193168–193182, 2020.
- [7] S. Kalvankar, H. Pandit, and P. Parwate, "Galaxy morphology classification using efficientnet architectures," *arXiv Prepr. arXiv2008.13611*, 2020.
- [8] L. T. Duong, T. B. Tran, N. H. Le, V. M. Ngo, and P. T. Nguyen, "Automatic detection of weeds: synergy between EfficientNet and transfer learning to enhance the prediction accuracy," *Soft Comput.*, vol. 28, no. 6, pp. 5029–5044, 2024.
- [9] M. Dian, L. Yudha, W. Setiawan, and Y. Wihardi, "Deteksi Sepeda Motor di Jalan Raya Menggunakan Faster R-CNN Berbasis VGG16 Detecting Motorcycle at Traffic Roads with Faster VGG16 based R-CNN," *JATIKOM J. Apl. dan Teor. Ilmu Komput.*, vol. 4, no. 2, pp. 10–13, 2021, [Online]. Available: <https://ejournal.upi.edu/index.php/JATIKOM> M.
- [10] J. Pardede and H. Hardiansah, "Deteksi Objek Kereta Api menggunakan Metode Faster R-CNN dengan Arsitektur VGG 16," *MIND (Multimedia Artif. Intell. Netw. Database) J. MIND*, vol. 7, no. 1, pp. 21–36, 2022.
- [11] G. I. Andaru, "Pengembangan Model Deteksi Untuk On-shelf Availability Produk menggunakan Yolov8 Pada Aplikasi Bergerak." Universitas Islam Indonesia, 2024.
- [12] M. Harahap and A. M. Husein, "Peneraparan Efficient-Net Dalam Mengklasifikasi Kanker Kulit," *Publ. BUKU UNPRI Press ISBN*, vol. 1, no. 1, 2024.
- [13] C. B. LIMBOING, "IMPLEMENTASI DEEP LEARNING UNTUK MENGANALISA BERBAGAI MACAM DAUN." Universitas Mercu Buana Jakarta, 2022.
- [14] R. Firmansyah, "Implementasi deep learning menggunakan convolutional neural network untuk klasifikasi bunga." Fakultas Sains dan Teknologi UIN Syarif Hidayatullah Jakarta, 2021.
- [15] X. Xu *et al.*, "Crack detection and comparison study based on faster R-CNN and mask R-CNN," *Sensors*, vol. 22, no. 3, p. 1215, 2022.
- [16] R. Budi, R. A. Harianto, and E. Setyati, "Segmentasi Citra Area Tumpukan Sampah Dengan Memanfaatkan Mask R-CNN," *INSYST J. Intell. Syst. Comput.*, vol. 5, no. 1, pp. 58–64, 2023.
- [17] R. A. Firmansah, H. Santoso, and A. Anwar, "Transfer Learning Implementation on Image Recognition of Indonesian Traditional Houses," *J. Tek. Inform.*, vol. 4, no. 6, pp. 1469–1478, 2023.
- [18] I. R. Fadhillah, M. M. Al Haromainy, and H. Maulana, "Implementasi Model Transfer Learning EfficientNet untuk Pendeteksian Bahasa Isyarat Indonesia (BISINDO) pada Perangkat Android," *JATI (Jurnal Mhs. Tek. Inform.)*, vol. 8, no. 4, pp. 7816–7822, 2024.
- [19] F. Charli, H. Syaputra, M. Akbar, S. Sauda, and F. Panjaitan, "Implementasi Metode Faster Region Convolutional Neural Network (Faster R-CNN) Untuk Pengenalan Jenis Burung Lovebird," *J. Inf. Technol. Ampera*, vol. 1, no. 3, pp. 185–197, 2020, doi: 10.51519/journalita.volume1.issue3.year2020.page185-197.
- [20] Y. Mardianto, T. Dewi, and P. Risma, "Analisis Klasifikasi Kematangan Buah Tomat dengan Pendekatan Transfer Learning Model EfficientNet," *Techno Bahari*, vol. 11, no. 1, pp. 20–25, 2024, doi: 10.52234/tb.v11i1.306.
- [21] D. P. Sidik, F. Utamingrum, and L. Muflikhah, "Penggunaan Variasi Model pada Arsitektur EfficientNetV2 untuk Prediksi Sel Kanker Serviks," ... *Teknol. Inf. dan Ilmu ...*, vol. 7, no. 5, pp. 2116–2121, 2023, [Online]. Available: <https://j-ptiik.ub.ac.id/index.php/j-ptiik/article/view/12656>.
- [22] A. Ganang Pradana, D. Rosal, I. M. Setiadi, and A. R. Muslikh, "Fine tuning model Convolutional Neural Network EfficientNet-B4 dengan augmentasi data untuk klasifikasi penyakit kakao Fine tuning Convolutional Neural Network model EfficientNet-B4 with data augmentation for cocoa disease

- classification,” *J. Inf. Syst. Appl. Dev.*, vol. 2, no. 1, pp. 1–11, 2024, doi: 10.26905/jisad.v2i1.11899.
- [23] M. A. J. Asshiddiqie, B. Rahmat, and F. T. Anggraeny, “Deteksi tanaman tebu pada lahan pertanian menggunakan metode convolutional neural network,” *J. Inform. dan Sist. Inf.*, vol. 1, no. 1, pp. 229–237, 2020.
- [24] R. S. Passa, S. Nurmaini, and D. P. Rini, “DETEKSI TUMOR OTAK PADA MAGNETIC RESONANCE IMAGING MENGGUNAKAN YOLOv7,” *J. Ilm. Matrik*, vol. 25, no. 2, pp. 116–121, 2023.
- [25] Ü. Atila, M. Uçar, K. Akyol, and E. Uçar, “Plant leaf disease classification using EfficientNet deep learning model,” *Ecol. Inform.*, vol. 61, p. 101182, 2021, doi: <https://doi.org/10.1016/j.ecoinf.2020.101182>.
- [26] D. G. Manurung *et al.*, “Deteksi Dan Klasifikasi Hama Potato Beetle Pada Tanaman Kentang Menggunakan YOLOV8,” *J. Teknol. Inf. dan Ilmu Komput.*, vol. 11, no. 4, pp. 723–734, 2024.
- [27] G. R. H. Galib, “Klasifikasi Area Tutupan Lahan Vegetasi Menggunakan Convolutional Neural Network.” Universitas Islam Negeri Maulana Malik Ibrahim, 2024.