

IMPLEMENTASI ALGORITMA *FASTER R-CNN* DALAM DETEKSI TEMPAT PARKIR MOBIL DI AREA KAMPUS

LAPORAN TUGAS AKHIR

Proposal ini disusun untuk memenuhi salah satu syarat memperoleh gelar
Sarjana Strata 1 (S1) pada Program Studi Teknik Informatika
Fakultas Teknologi Industri Universitas Islam Sultan Agung Semarang



DISUSUN OLEH:

ALIF HAKIM AL FARUQ

NIM 32602100021

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM SULTAN AGUNG
SEMARANG**

2025

FINAL PROJECT

***IMPLEMENTATION OF FASTER R-CNN ALGORITHM IN CAR PARKING
LOT DETECTION IN CAMPUS AREA***

*Proposed to complete the requirement to obtain a bachelor's degree (S-1)
at Informatics Engineering Departement of Industrial Technology Faculty
Sultan Agung Islamic University*



ARRANGED BY:

UNISSULA

ALIF HAKIM AL FARUQ

NIM 32602100021

**MAJORING OF INFORMATICS ENGINEERING
INDUSTRIAL TECHNOLOGY FACULTY
SULTAN AGUNG ISLAMIC UNIVERSITY
SEMARANG**

2025

LEMBAR PENGESAHAN
TUGAS AKHIR

IMPLEMENTASI ALGORITMA *FASTER R-CNN* DALAM DETEKSI
TEMPAT PARKIR MOBIL DI AREA KAMPUS

ALIF HAKIM AL FARUQ
NIM 32602100021

Telah dipertahankan di depan tim penguji ujian sarjana tugas akhir
Program Studi Teknik Informatika
Universitas Islam Sultan Agung
Pada tanggal : 02 Juni 2025....

TIM PENGUJI UJIAN SARJANA :

Moch. Taufik, S.T., M.IT

NIK. 210604034

(Ketua Penguji)

03/06/2025

Arief Marwanto, ST., M.Eng.

Ph.D

NIK. 210600015

(Anggota Penguji)

03/06/2025

Andi Riansyah, S.T., M.Kom

NIK. 210616053

(Pembimbing)

03/06/2025

Semarang, 3 Juni 2025

Mengetahui,

Kaprodi Teknik Informatika
Universitas Islam Sultan Agung

Moch. Taufik, S.T., M.IT

NIK. 210604034

SURAT PERNYATAAN KEASLIAN TUGAS AKHIR

Yang bertanda tangan di bawah ini:

Nama : Alif Hakim Al Faruq

NIM : 32602100021

Judul Tugas Akhir : Implementasi Algoritma *Faster R-CNN* Dalam Deteksi Tempat Parkir Mobil di Area Kampus

Bahwa dengan ini saya menyatakan bahwa judul dan isi Tugas Akhir yang saya buat dalam rangka menyelesaikan Pendidikan Strata Satu (S1) Teknik Informatika tersebut adalah asli dan belum pernah diangkat, ditulis ataupun dipublikasikan oleh siapapun baik keseluruhan maupun sebagian, kecuali yang secara tertulis diacu dalam naskah ini dan disebutkan dalam daftar pustaka, dan apabila di kemudian hari ternyata terbukti bahwa judul Tugas Akhir tersebut pernah diangkat, ditulis ataupun dipublikasikan, maka saya bersedia dikenakan sanksi akademis. Demikian surat pernyataan ini saya buat dengan sadar dan penuh tanggung jawab.

Semarang, 4 Juni 2025

Yang menyatakan,



Alif Hakim Al Faruq

PERNYATAAN PERSETUJUAN PUBLIKASI KARYA ILMIAH

Saya yang bertanda tangan di bawah ini:

Nama : Alif Hakim Al Faruq

NIM : 32602100021

Program Studi : Teknik Informatika

Fakultas : Teknologi Industri

Alamat Asal : Sintang

Dengan ini menyatakan karya ilmiah berupa tugas akhir dengan judul : Implementasi Algoritma *Faster R-CNN* Dalam Deteksi Tempat Parkir Mobil di Area Kampus.

Menyetujui menjadi hak milik Universitas Islam Sultan Agung serta memberikan hak bebas royalti Non-Eksklusif untuk disimpan, dialihmediakan, dikelola dan pangkalan data dan dipublikasikan di internet dan media lain untuk kepentingan akademis selama tetap menyantumkan nama penulis sebagai pemilik hak cipta. Pernyataan ini saya buat dengan sungguh-sungguh. Apabila di kemudian hari terbukti ada pelanggaran Hak Cipta/Plagiarisme dalam karya ilmiah ini, maka segala bentuk tuntutan hukum yang timbul akan saya tanggung secara pribadi tanpa melibatkan Universitas Islam Sultan Agung.

Semarang, 4 Juni 2025

Yang menyatakan,



Alif Hakim Al Faruq

KATA PENGANTAR

Dengan mengucapkan syukur alhamdulillah atas kehadiran ALLAH SWT yang telah memberikan rahmat dan karunia-Nya kepada penulis, sehingga dapat menyelesaikan Tugas Akhir dengan judul “Implementasi Algoritma *Faster R-CNN* Dalam Deteksi Tempat Parkir Mobil di Area Kampus” dengan baik. Tugas akhir ini disusun dan dibuat dengan adanya bantuan dari berbagai pihak, berupa materi maupun teknis, oleh karena itu saya selaku penulis mengucapkan terima kasih kepada:

1. Rektor Universitas Islam Sultan Agung Bapak Prof. Dr. H. Gunarto, S.H., M.H yang mengizinkan penulis menimba ilmu di kampus ini.
2. Dekan Fakultas Teknologi Industri Ibu Dr. Ir. Hj. Novi Marlyana, S.T., M.T., IPU., ASEAN Eng.
3. Dosen pembimbing penulis Bapak Andi Riansyah, S.T., M.Kom yang telah meluangkan waktu dan memberi ilmu dalam penyusunan tugas akhir.
4. Orang Tua penulis yang telah memberikan do'a dan kepercayaan kepada penulis sehingga dapat menyelesaikan tugas akhir.
5. Saudara/i penulis yang telah mendukung dan membantu dalam penyelesaian tugas akhir.
6. Teman-teman yang telah membantu, memberikan motivasi dalam proses penyelesaian tugas akhir.
7. Kepada seseorang yang telah memotivasi saya untuk menyelesaikan tugas akhir.
8. Terima kasih banyak kepada pihak yg tidak bisa disebutkan satu persatu.

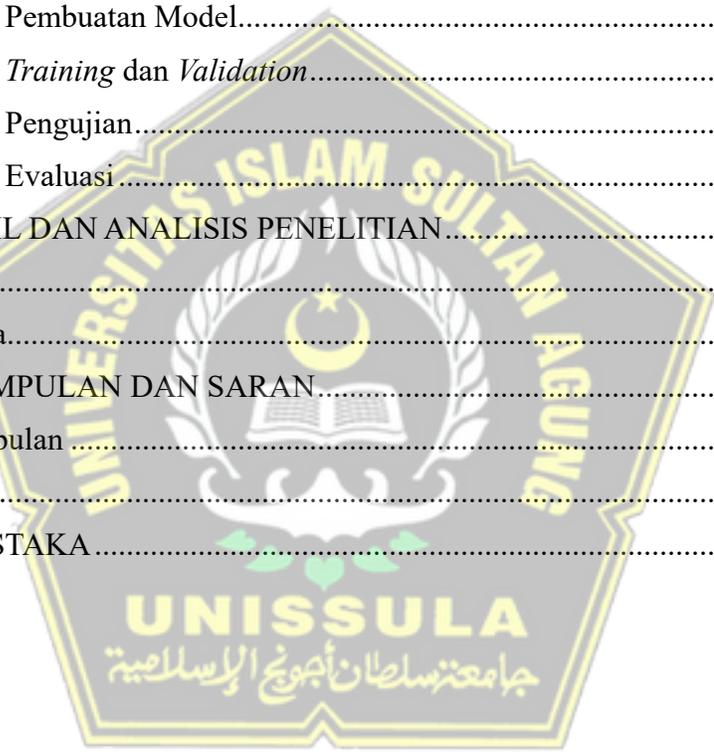
Semarang, Juni 2025

Penulis

DAFTAR ISI

HALAMAN JUDUL.....	i
LEMBAR PENGESAHAN TUGAS AKHIR.....	ii
SURAT PERNYATAAN KEASLIAN TUGAS AKHIR.....	ii
PERNYATAAN PERSETUJUAN PUBLIKASI KARYA ILMIAH.....	iii
KATA PENGANTAR.....	v
DAFTAR ISI.....	vi
DAFTAR TABEL.....	viii
DAFTAR GAMBAR.....	ix
ABSTRAK.....	x
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	2
1.3 Batasan Masalah.....	3
1.4 Tujuan Penelitian.....	3
1.5 Manfaat Penelitian.....	3
1.6 Sistematika Penulisan.....	4
BAB II TINJAUAN PUSTAKA DAN DASAR TEORI.....	5
2.1 Tinjauan Pustaka.....	5
2.2 Dasar Teori.....	10
2.2.1 Tempat Parkir.....	10
2.2.2 <i>Computer Vision</i>	11
2.2.3 <i>OpenCV</i>	12
2.2.4 <i>Machine Learning</i>	13
2.2.5 <i>Deep Learning</i>	14
2.2.6 <i>Convolutional Neural Network (CNN)</i>	14
2.2.7 <i>ResNet50</i>	15
2.2.8 <i>Faster Region - Convolutional Neural Network (Faster R-CNN)</i>	16

BAB III METODOLOGI PENELITIAN.....	20
3.1 Studi Literatur	20
3.2 Rancang Alur Model	20
3.3 Tahapan Perancangan Model	22
3.3.1 Pengumpulan Data	22
3.3.2 <i>Preprocessing Data</i>	24
3.3.3 <i>Augmentation Data</i>	24
3.3.4 Pembuatan Model.....	27
3.3.5 <i>Training dan Validation</i>	27
3.3.6 Pengujian.....	28
3.3.7 Evaluasi.....	29
BAB IV HASIL DAN ANALISIS PENELITIAN.....	31
4.1 Hasil.....	31
4.2 Analisa.....	36
BAB V KESIMPULAN DAN SARAN.....	48
5.1 Kesimpulan.....	48
5.2 Saran.....	48
DAFTAR PUSTAKA.....



DAFTAR TABEL

Tabel 1. 1 Sistematika penulisan.....	4
Tabel 3. 1 <i>Confusion Matrix</i>	30
Tabel 4. 1 <i>Hyperparameter</i>	36
Tabel 4. 2 Hasil penelitian.....	37



DAFTAR GAMBAR

Gambar 2. 1 Tempat parkir	10
Gambar 2. 2 <i>Computer vision</i>	11
Gambar 2. 3 <i>OpenCV</i>	12
Gambar 2. 4 Cara kerja <i>Faster R-CNN</i>	16
Gambar 3. 1 <i>Flowchart</i> alur model.....	21
Gambar 3. 2 Dataset.....	23
Gambar 3. 3 Alur pengujian.....	28
Gambar 4. 1 <i>folder</i> dataset	31
Gambar 4. 2 <i>File</i> folder anotasi dan gambar.....	31
Gambar 4. 3 <i>Folder</i> dataset baru.....	32
Gambar 4. 4 Definisi mapping label.....	32
Gambar 4. 5 Hasil augmentasi	33
Gambar 4. 6 <i>Loss progression</i>	34
Gambar 4. 7 Hasil deteksi tempat parkir kosong	35
Gambar 4. 8 Hasil mAP.....	35



ABSTRAK

Pertumbuhan jumlah kendaraan bermotor, khususnya mobil, terus meningkat seiring dengan pertumbuhan populasi dan kebutuhan mobilitas, tidak hanya di kawasan perkotaan tetapi juga di lingkungan institusi pendidikan seperti kampus. Hal ini menimbulkan permasalahan krusial terkait keterbatasan ruang parkir. Kendati kampus telah menyediakan area parkir khusus, kurangnya informasi mengenai ketersediaan slot parkir menyebabkan pengguna kendaraan harus berkeliling mencari tempat parkir, yang berdampak pada waktu, bahan bakar, dan kelancaran lalu lintas. Teknologi kecerdasan buatan dan pengolahan citra digital menawarkan solusi dengan memanfaatkan kamera pengawas dan algoritma pembelajaran mesin seperti Faster R-CNN. Hasil pengujian model pada berbagai nilai threshold menunjukkan bahwa penurunan threshold secara konsisten meningkatkan performa deteksi objek. Pada threshold 0.5, meskipun precision sedikit turun menjadi 97,3%, recall mencapai 97,3% dan akurasi tertinggi sebesar 95%, menghasilkan f1-score sebesar 97,3%. Dengan demikian, penyesuaian threshold terbukti berpengaruh signifikan terhadap performa deteksi, dan threshold 0.5 direkomendasikan sebagai nilai optimal untuk mendeteksi objek pada sistem monitoring parkir berbasis pengolahan citra. Implementasi teknologi ini diharapkan dapat membantu pengelolaan parkir yang lebih efektif dan efisien di area kampus.

Kata Kunci: Deteksi objek, *OpenCV*, Tempat Parkir, *Faster R-CNN*, *Resnet50*

ABSTRACT

The growth in the number of motorized vehicles, especially cars, continues to increase along with population growth and mobility needs, not only in urban areas but also in educational institutions such as campuses. This raises crucial issues related to limited parking space. Although the campus has provided a dedicated parking area, the lack of information regarding the availability of parking slots causes vehicle users to drive around looking for a parking space, which has an impact on time, fuel, and traffic flow. Artificial intelligence and digital image processing technologies offer a solution by utilizing surveillance cameras and machine learning algorithms such as Faster R-CNN. The results of testing the model at various threshold values show that decreasing the threshold consistently improves the object detection performance. At a threshold of 0.5, although the precision dropped slightly to 97.3%, the recall reached 97.3% and the highest accuracy was 95%, resulting in an f1-score of 97.3%. Thus, threshold adjustment is proven to have a significant effect on detection performance, and a threshold of 0.5 is recommended as the optimal value for detecting objects in image processing-based parking lot monitoring systems. The implementation of this technology is expected to help more effective and efficient parking management in the campus area.

Keywords: Object detection, OpenCV, Parking area, Faster R-CNN, Resnet50

BAB I

PENDAHULUAN

1.1 Latar Belakang

Perkembangan jumlah kendaraan bermotor, khususnya mobil, terus meningkat seiring dengan pertumbuhan populasi dan kebutuhan mobilitas masyarakat (Sidik & Ansawarman, 2022). Fenomena ini tidak hanya terjadi di kawasan perkotaan, tetapi juga di lingkungan institusi pendidikan seperti kampus. Mobil menjadi moda transportasi utama bagi dosen, staf, dan sebagian mahasiswa karena memberikan kenyamanan dan fleksibilitas waktu. Namun, peningkatan jumlah mobil ini juga membawa dampak terhadap ketersediaan dan pengelolaan ruang parkir di area kampus.

Permasalahan ketersediaan tempat parkir di area kampus merupakan masalah yang krusial seiring bertambahnya volume kendaraan (Radiuk dkk., 2022). Meskipun kampus telah menyediakan area parkir khusus, keterbatasan informasi mengenai ketersediaan slot parkir menyebabkan banyak pengguna kendaraan harus berkeliling mencari tempat parkir yang tersedia. Hal ini tidak hanya membuang waktu dan bahan bakar, tetapi juga dapat menyebabkan kepadatan lalu lintas di area kampus dan ketidakteraturan dalam sistem parkir.

Di tengah tantangan tersebut, perkembangan teknologi kecerdasan buatan dan pengolahan citra digital menawarkan potensi solusi yang menjanjikan. Salah satu pendekatan yang banyak digunakan dalam deteksi objek secara otomatis adalah pemanfaatan kamera pengawas yang digabungkan dengan algoritma berbasis pembelajaran mesin, seperti *Convolutional Neural Network (CNN)*. Dalam konteks deteksi objek, algoritma *Faster R-CNN* telah dikenal sebagai metode yang unggul karena kemampuannya yang dapat beradaptasi dengan gambar yang berubah-ubah dan menyesuaikan seluruh jaringan untuk meningkatkan akurasi pengenalan jaringan (Vijiyakumar dkk., 2024).

Faster R-CNN adalah jaringan pembelajaran mendalam yang banyak digunakan untuk meningkatkan kecepatan komputasi dengan menggunakan teknik *Region Proposal Network* (Wahjuni dkk., 2023) dan *Convolutional Neural Network (CNN)* sebagai jaringan ekstraksi fitur. Metode ini menggabungkan dua tahap utama, yaitu region proposal dan klasifikasi objek, menjadi satu arsitektur terpadu yang mempercepat proses deteksi tanpa mengorbankan akurasi. Dengan memanfaatkan algoritma ini, sistem dapat secara otomatis mendeteksi slot parkir yang kosong maupun yang terisi berdasarkan citra yang diambil dari kamera yang terpasang di area parkir kampus.

Berdasarkan latar belakang tersebut, penelitian ini bertujuan untuk mengimplementasikan algoritma *Faster R-CNN* dalam mendeteksi keberadaan slot parkir kosong pada area parkir yang disediakan di lingkungan kampus. Diharapkan, hasil dari penelitian ini dapat memberikan kontribusi dalam menciptakan sistem pemantauan parkir yang cerdas dan efisien, yang pada akhirnya dapat meningkatkan kenyamanan dan keteraturan dalam pengelolaan parkir kendaraan di kampus.

1.2 Rumusan Masalah

Berdasarkan latar belakang permasalahan di atas, maka identifikasi permasalahan akan berfokus pada :

1. Bagaimana cara mengimplementasikan algoritma *Faster R-CNN* untuk mendeteksi tempat parkir yang kosong?
2. Seberapa akurat algoritma *Faster R-CNN* dalam mendeteksi lahan parkir yang kosong?

1.3 Batasan Masalah

Adapun batasan masalah dari penulisan proposal ini adalah :

1. Dataset yang digunakan berasal dari internet dan dokumentasi pribadi yang di ambil di lokasi tertentu.
2. Kualitas gambar dan Resolusi kamera algoritma akan diuji menggunakan rekaman video dari kamera laptop yang memiliki kualitas resolusi standar (480p hingga 1080p).
3. Penelitian ini akan dilakukan di area kampus dengan fokus pada lahan parkir yang memiliki marka pembatas.
4. Penelitian ini berfokus pada deteksi lahan parkir yang memiliki marka pembatas.

1.4 Tujuan Penelitian

1. Mengimplementasikan metode *Faster R-CNN* dalam deteksi area parkir.
2. Menganalisis tingkat akurasi metode *Faster R-CNN* dalam deteksi area parkir.

1.5 Manfaat Penelitian

1. Penelitian ini diharapkan dapat menjadi referensi dalam literatur tentang model deteksi, khususnya metode *Faster R-CNN*
2. Menjadi inovasi yang bisa digunakan oleh pihak kampus dalam menegakkan aturan perparkiran area.

1.6 Sistematika Penulisan

Sistematika penulisan yang akan digunakan oleh penulis dalam sebuah pembuatan laporan tugas akhir adalah sebagai berikut:

Tabel 1. 1 Sistematika penulisan

BAB I	PENDAHULUAN
	Bab ini berisi latar belakang pemilihan judul, rumusan masalah, batasan masalah, tujuan penelitian, serta sistematika penulisan.
BAB II	TINJAUAN PUSTAKA DAN DASAR TEORI
	Bab ini memuat penelitian-penelitian sebelumnya dan dasar teori untuk membantu penulis memahami bagaimana teori yang berhubungan dengan <i>FASTER R-CNN</i> dan pengimplementasian menggunakan <i>webcam</i> .
BAB III	METODE PENELITIAN
	Bab ini mengungkapkan proses tahapan-tahapan penelitian dimulai dari mendapatkan data hingga proses pengolahan data yang ada.
BAB IV	HASIL DAN ANALISIS PENELITIAN
	Bab ini penulis mengungkapkan hasil penelitian seperti pengumpulan data, pengolahan data, dan simulasi sistem deteksi tempat parkir mobil.
BAB V	KESIMPULAN DAN SARAN
	Bab ini penulis memaparkan kesimpulan proses penelitian dari awal hingga akhir serta saran untuk penelitian selanjutnya.

BAB II

TINJAUAN PUSTAKA DAN DASAR TEORI

2.1 Tinjauan Pustaka

Penelitian sebelumnya terkait sistem pendeteksi pelanggaran di pinggir jalan tol menggunakan algoritma *Deep Learning Faster Region-based Convolutional Neural Network (Faster R-CNN)* dilakukan oleh (Hamzah, 2022). Sistem ini bekerja dengan memproses video rekaman dari aplikasi HK TOLL APPS yang berasal dari CCTV, kemudian mendeteksi kendaraan yang melakukan pelanggaran. Jika terbukti melakukan pelanggaran, sistem akan memberikan notifikasi kepada petugas jalan tol melalui Telegram Bot. Hasil penelitian menunjukkan bahwa sistem mampu mendeteksi kendaraan pelanggar dengan baik. Model terbaik diperoleh dari pengujian menggunakan skema pembagian data latih dan uji sebesar 90%:10%, dengan konfigurasi variabel *Learning Rate* sebesar 0,0004, *Batch Size* 1, dan *Epoch* 100. Nilai evaluasi yang dihasilkan adalah akurasi sebesar 77,9%, *precision* 71,1%, *recall* 73,6%, dan *Mean Average Precision (mAP)* sebesar 77%.

Penelitian sebelumnya terkait sistem pendeteksi objek menggunakan metode *Faster R-CNN* dengan arsitektur Inception V2 dilakukan oleh (Putri & Fahrurrozi, 2022) menghasilkan rata-rata akurasi sebesar 89%. Penelitian ini menggunakan 2000 citra hewan yang diperoleh secara acak melalui ekstensi Chrome dan diberi label secara manual. Akuisisi citra mencakup variasi sudut pengambilan, jarak, serta jumlah objek dalam satu citra. Sistem dibangun melalui tahap pelatihan model, uji coba, serta pengukuran performa sistem pendeteksian objek. Berdasarkan hasil implementasi, sistem mampu mengklasifikasikan jenis hewan karnivora dan herbivora dengan baik. Nilai total loss dari hasil pelatihan model berada pada angka 0,06. Selain itu, nilai *recall* mencapai 100% pada kelompok citra hewan seperti Cheetah, Eagle, Komodo, Shark, Tiger, Bull,

Guineapig, dan Zebra, yang menunjukkan bahwa sistem klasifikasi yang dikembangkan memiliki performa yang sangat baik.

Penelitian sebelumnya terkait pendeteksian senjata genggam kelas pisau menggunakan metode *Faster R-CNN* Inception V2 dilakukan oleh (Pradana, 2022) dan menghasilkan akurasi sebesar 87%. Model dalam penelitian ini dilatih menggunakan dataset yang terdiri dari gambar-gambar senjata genggam, khususnya pisau, yang diperoleh dari rekaman bela diri serta berbagai gambar pisau dengan bentuk dan warna yang beragam. Model yang dikembangkan mampu menandai objek yang diduga sebagai senjata dengan kotak penanda pada citra digital. Proses pelatihan dan pengujian model dilakukan menggunakan platform Google Colab dengan pengujian terhadap 475 gambar digital. Hasil penelitian menunjukkan bahwa model mampu mendeteksi senjata genggam kelas pisau dengan tingkat akurasi yang cukup tinggi, sehingga dapat digunakan sebagai sistem pendeteksi awal untuk objek berbahaya pada gambar atau video.

Penelitian sebelumnya terkait pendeteksian objek sampah menggunakan metode *Faster R-CNN* dilakukan oleh (Rahman & Bambang, 2021) dengan tujuan untuk mendeteksi dan mengklasifikasikan objek sampah secara langsung melalui video real-time. Penelitian ini bertujuan mengembangkan sistem pendeteksi sampah yang dapat diterapkan pada perangkat tertentu sebagai solusi atas permasalahan sampah yang terus berlanjut. Hasil pengujian menunjukkan bahwa metode *Faster R-CNN* yang digunakan mampu mendeteksi objek sampah dari 100 gambar dengan tingkat akurasi sebesar 74%. Selain itu, sistem juga mampu melakukan pendeteksian objek secara real-time dengan kecepatan frame video sebesar 1 frame per second (fps). Hal ini menunjukkan bahwa metode tersebut cukup efektif untuk digunakan dalam sistem pemantauan sampah secara langsung.

Penelitian sebelumnya terkait pengenalan jenis burung lovebird menggunakan metode *Faster Region-based Convolutional Neural Network* (*Faster R-CNN*) dilakukan oleh (Charli, 2020) dengan tujuan untuk

mengidentifikasi spesies burung lovebird berdasarkan motif atau warna aslinya. Dalam penelitian ini digunakan dataset yang terdiri dari 808 gambar burung lovebird yang terbagi dalam 8 kelas, yaitu *Agapornis Personata*, *Agapornis Fischeri*, *Agapornis Lilianae*, *Agapornis Nigrigenis*, *Agapornis Roseicollis*, *Agapornis Taranta*, *Agapornis Pullarius*, dan *Agapornis Canus*. Hasil pelatihan model menunjukkan bahwa metode *Faster R-CNN* mampu mengenali spesies burung lovebird dengan tingkat akurasi sebesar 96% dan nilai loss sebesar 0,2%. Hal ini membuktikan bahwa metode ini sangat efektif untuk diterapkan dalam sistem pengenalan objek, khususnya untuk klasifikasi jenis burung berdasarkan ciri visual.

Penelitian sebelumnya terkait deteksi objek buah apel dilakukan oleh (Kong dkk., 2024) dengan mengembangkan metode baru bernama *Faster RFormer*, yang merupakan penggabungan antara arsitektur *Faster R-CNN* dengan backbone berupa *moving window transformer* serta penambahan fungsi *loss smoothing* yang ditingkatkan. Penelitian ini bertujuan untuk mengatasi keterbatasan model *Convolutional Neural Network* dalam mendeteksi objek pada lingkungan yang kompleks. Untuk menguji efektivitas metode yang diusulkan, peneliti membuat dataset deteksi apel khusus bernama AD-2023. Hasil penelitian menunjukkan bahwa model *Faster RFormer* berhasil mencapai performa yang unggul dengan nilai *mAP* sebesar 0,692, *AP@0.75* sebesar 0,796, dan *AP@0.5* sebesar 0,941, yang melampaui algoritma deteksi objek sebelumnya. Studi ini juga menetapkan tolok ukur baru dalam metodologi deteksi apel, serta menunjukkan potensi penerapan yang luas dalam bidang otomasi pertanian dan robotika.

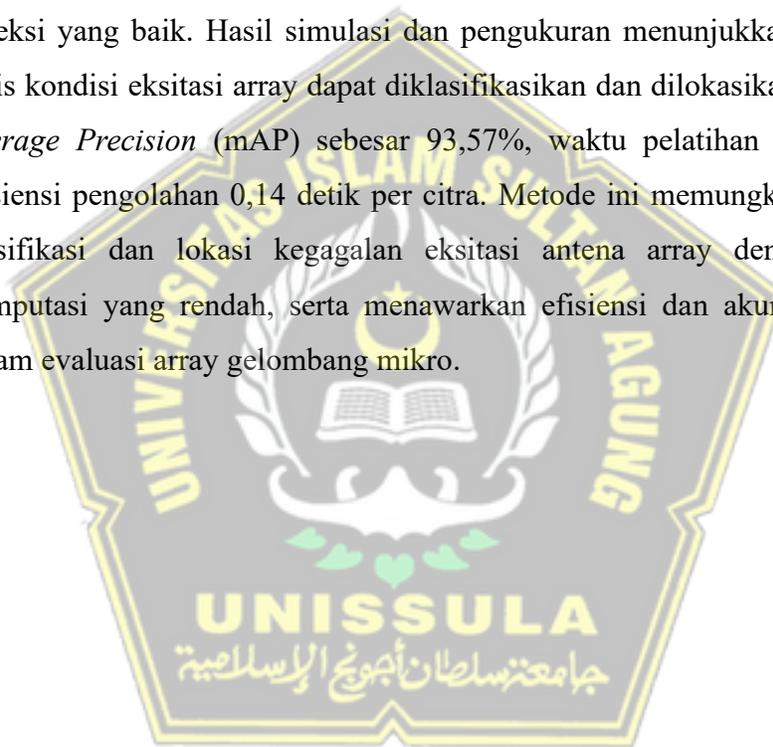
Penelitian sebelumnya terkait penerapan algoritma *Faster R-CNN* dilakukan oleh (Zhang dkk., 2021), yang mengusulkan sebuah skema implementasi teknis lengkap untuk tugas pengenalan peralatan dan deteksi status di ruang listrik. Dalam penelitian ini, peneliti mengumpulkan sebanyak 5600 gambar dari 100 kategori peralatan listrik melalui proses survei lapangan dan

pengambilan gambar secara langsung. Seluruh gambar diberi anotasi menggunakan perangkat lunak open-source LabelImg, kemudian dibagi menjadi 70% data latih, 10% data validasi, dan 20% data uji. Hasil pelatihan model menunjukkan nilai loss kurang dari 0.01%, serta menghasilkan *Mean Average Precision* (mAP) sebesar 91.3% pada pengujian gambar. Selain itu, pengujian pada video juga menunjukkan performa yang sangat baik, sehingga membuktikan efektivitas serta implementabilitas dari solusi teknis yang diajukan.

Penelitian sebelumnya terkait pemangkasan daun pada tanaman melon emas menggunakan metode *Faster Region-Convolutional Neural Network* (*Faster R-CNN*) dengan kamera RGB-Depth dilakukan oleh (Wahjuni dkk., 2023). Penelitian ini bertujuan untuk membedakan antara daun dan batang tanaman sebelum dilakukan proses pemangkasan secara otomatis oleh robot. Gambar yang digunakan dalam penelitian ini diambil dengan berbagai variasi intensitas cahaya, jarak, ketinggian, dan sudut pengambilan gambar menggunakan kamera RGB-Depth. Model *Faster R-CNN* dilatih selama 10 *Epoch* dengan dua skenario pengujian yang menggunakan nilai ambang Intersection over Union (IoU) sebesar 30% (Skenario 1) dan 70% (Skenario 2). Hasil pengujian menunjukkan bahwa Skenario 1 (IoU 30%) memberikan skor yang lebih baik pada data pengujian agregat, yaitu sebesar 70,02% dibandingkan dengan Skenario 2. Selain itu, ditemukan korelasi negatif antara intensitas cahaya dan skor *Mean Average precision* (mAP), yaitu sebesar $-0,292$. Tidak ditemukan pengaruh signifikan terhadap performa model berdasarkan variasi jarak dan sudut pengambilan gambar. Penelitian ini menyimpulkan bahwa nilai ambang IoU pada proposal region berpengaruh terhadap hasil deteksi yang diperoleh.

Penelitian sebelumnya yang relevan dengan penerapan algoritma *Faster R-CNN* dilakukan oleh (Yang dkk., 2025), yang memperkenalkan sebuah metode rekonstruksi sumber untuk mendeteksi kegagalan elemen antena array menggunakan data medan dekat. Dalam penelitian ini, peneliti memanfaatkan keunggulan konvergensi dan optimisasi pembelajaran mendalam untuk

mengatasi permasalahan ill-posed dari persamaan integral medan listrik (EFIE). Data citra pola medan dekat dari antena array dikumpulkan dan dibagi menjadi dataset pelatihan dan pengujian, lalu dimasukkan ke dalam model *Faster R-CNN* untuk mempelajari operator invers dari persamaan EFIE tersebut. Empat kondisi eksitasi yang berbeda dilatih dalam jaringan yang sama untuk mengevaluasi pengaruh kesalahan eksitasi terhadap ekstraksi fitur dan akurasi klasifikasi. Prototipe antena array dirancang dan diuji secara eksperimental dengan hasil deteksi yang baik. Hasil simulasi dan pengukuran menunjukkan bahwa empat jenis kondisi eksitasi array dapat diklasifikasikan dan dilokasikan dengan *Mean Average Precision* (mAP) sebesar 93,57%, waktu pelatihan 24 menit, serta efisiensi pengolahan 0,14 detik per citra. Metode ini memungkinkan diagnosis klasifikasi dan lokasi kegagalan eksitasi antena array dengan kebutuhan komputasi yang rendah, serta menawarkan efisiensi dan akurasi yang tinggi dalam evaluasi array gelombang mikro.



2.2 Dasar Teori

2.2.1 Tempat Parkir



Gambar 2. 1 tempat parkir

Tempat parkir adalah area yang disediakan bagi pengendara untuk menghentikan dan meninggalkan kendaraannya untuk sementara waktu (Dewi dkk., 2023; Dwitama dkk., 2025). Keberadaan tempat parkir memiliki peran yang penting dalam mendukung kelancaran transportasi, baik di area perumahan, perkantoran, pusat perbelanjaan, maupun fasilitas umum lainnya. Berdasarkan data dari Badan Pusat Statistik (BPS) pada periode 2021 hingga 2022, jumlah kendaraan bermotor di Indonesia mengalami lonjakan signifikan, dari 283.985.146 unit menjadi 296.523.634 unit. Lonjakan ini mencerminkan adanya peningkatan sekitar 12,5 juta unit kendaraan hanya dalam waktu satu tahun, yang tentu saja membawa implikasi besar terhadap kebutuhan lahan parkir.

Seiring dengan terus bertambahnya jumlah kendaraan, penyediaan prasarana parkir yang memadai menjadi semakin mendesak. Tanpa infrastruktur parkir yang memadai, pengendara akan kesulitan menemukan lokasi parkir yang

aman dan nyaman, yang pada akhirnya dapat menimbulkan permasalahan lain seperti parkir liar, kemacetan, serta gangguan terhadap kelancaran arus lalu lintas. Oleh karena itu, pengembangan dan perbaikan fasilitas parkir tidak hanya berfungsi sebagai solusi praktis bagi pengendara, tetapi juga sebagai bagian penting dalam upaya pengelolaan transportasi yang lebih efisien dan tertib di perkotaan.

2.2.2 *Computer Vision*



Gambar 2. 2 computer vision

Computer vision adalah cabang ilmu komputer yang membekali mesin dengan kemampuan "melihat" dan "memahami" gambar serta video. Dengan memanfaatkan algoritma canggih, komputer dapat mengenali wajah, objek, dan bahkan menafsirkan adegan kompleks seperti manusia. Penerapan *computer vision* sangat luas, mulai dari keamanan, otomotif, lingkungan, hingga medis. Singkatnya, *computer vision* memungkinkan mesin untuk memperoleh informasi berharga dari dunia visual dan mengambil tindakan berdasarkan pemahaman tersebut (Sutisna, 2023).

Computer vision menawarkan kemampuan untuk memproses data visual dengan cepat dan akurat, yang tidak hanya mengurangi kesalahan manusia, tetapi juga memungkinkan pengambilan keputusan secara otomatis. Teknologi ini mampu memproses ribuan gambar atau video dalam waktu singkat, membuatnya sangat efisien untuk berbagai aplikasi praktis. Selain itu, keunggulan lainnya terletak pada fleksibilitasnya, karena dapat diintegrasikan dengan teknologi lain

seperti kecerdasan buatan, *internet of things* (IoT), dan sistem berbasis cloud untuk memperluas kemampuannya dalam memahami konteks visual.me

Kemungkinan perkembangan *computer vision* di masa mendatang sangat menjanjikan, seiring dengan kemajuan teknologi komputasi dan pembelajaran mesin. Diharapkan bahwa algoritma pengolahan citra akan menjadi semakin canggih, sehingga memungkinkan komputer untuk memahami adegan visual dengan konteks yang lebih mendalam, seperti membaca emosi manusia atau memahami situasi yang kompleks. Selain itu, penggabungan teknologi seperti edge computing dan pemrosesan paralel diprediksi akan meningkatkan kecepatan dan efisiensi computer vision, menjadikannya teknologi yang semakin esensial dalam kehidupan sehari-hari dan berbagai sektor industri di masa depan.

2.2.3 *OpenCV*



Gambar 2. 3 *OpenCV*

Open Source Computer Vision Library atau yang lebih dikenal dengan OpenCV adalah perpustakaan sumber terbuka yang dibuat pertama kali oleh Intel pada tahun 1999 dan dikenal sangat populer untuk pemrosesan gambar dan penglihatan komputer. Kemudahan dan ketersediaan banyak fitur dalam *OpenCV* menjadikannya salah satu alat penting yang diperlukan dalam pengembangan aplikasi penglihatan komputer. Berikut adalah beberapa fitur utama *OpenCV*:

- a. Pemrosesan gambar: Memungkinkan memanipulasi gambar seperti pemotongan, pemosisian, dan transformasi.
- b. Pengenalan Pola dan Deteksi Objek: Mendukung pengenalan wajah, deteksi objek, dan pengenalan pola.

- c. Pengolahan Citra: Menyediakan algoritma untuk filterisasi, segmentasi, dan ekstraksi fitur citra.
- d. Kalibrasi Kamera: Memungkinkan kalibrasi kamera dan geometri stereo.
- e. Pemrosesan Video: Dapat memancarkan gambar dengan lebih baik.

OpenCV sudah tersedia dalam berbagai bahasa pemrograman, seperti C++, *Python*, *Java*, dan *MATLAB*, yang membuatnya lebih mudah untuk diintegrasikan ke berbagai platform dan digunakan dalam berbagai aplikasi(Maulana dkk., 2024).

2.2.4 Machine Learning

Machine learning adalah bagian dari kecerdasan buatan yang mengajarkan komputer untuk mempelajari informasi dari kumpulan data dan membuat prediksi atau tindakan berdasarkan interpretasi data tersebut(Chyan dkk., n.d.). Dengan kata lain, *machine learning* adalah pemrograman komputer yang menggunakan data masa lalu untuk mempelajari model, sehingga komputer dapat belajar dan memiliki performa terbaiknya(Sidik & Ansawarman, 2022). Istilah *machine learning* pertama kali dikemukakan oleh beberapa ilmuwan matematika seperti Adrien Marie Legendre, Thomas Bayes dan Andrey Markov pada tahun 1920 dengan mengemukakan dasar-dasar *machine learning* dan konsepnya. Sejak saat itu *machine learning* banyak yang mengembangkan. Salah satu contoh dari penerapan *machine learning* yang cukup terkenal adalah *Deep Blue* yang dibuat oleh IBM pada tahun 1996.

Potensi machine learning di masa mendatang sangat besar dan terus berkembang seiring dengan kemajuan teknologi dan meningkatnya ketersediaan data. Dengan kemampuan untuk menganalisis data dalam jumlah besar secara efisien, machine learning berpeluang merevolusi berbagai sektor, mulai dari kesehatan, transportasi, pendidikan, hingga industri manufaktur. Di masa depan, machine learning dapat digunakan untuk menciptakan sistem prediktif yang lebih akurat, otomatisasi proses bisnis yang kompleks, serta pengambilan keputusan yang lebih cerdas dan adaptif. Selain itu, integrasi machine learning dengan teknologi

lain seperti *Internet of things* (IoT), komputasi awan, dan edge computing akan membuka jalan bagi penerapan yang lebih luas dan inovatif, seperti kendaraan otonom, sistem deteksi dini penyakit, serta manajemen energi yang lebih efisien. Dengan terus meningkatnya kualitas algoritma dan daya komputasi, machine learning akan menjadi pilar utama dalam transformasi digital di berbagai bidang kehidupan.

2.2.5 Deep Learning

Deep learning atau pembelajaran mendalam adalah cabang dari *machine learning* yang mana memiliki metode pembelajaran dengan menggunakan jaringan syaraf tiruan *multi-layer* (Ilahiyah & Nilogiri, 2018). Jaringan syaraf tiruan ini dirancang menyerupai otak manusia, dengan masing-masing *neuron* dihubungkan bersama membentuk jaringan syaraf yang sangat kompleks (Claudia Rossi dkk., 2024). *Deep learning* tidak hanya meniru bagaimana otak manusia bekerja, namun juga berusaha menciptakan jaringan syaraf tiruan itu sendiri (Zamachsari & Puspitasari, 2021).

Potensi deep learning di masa mendatang sangat besar, terutama dalam mendorong otomatisasi dan kecerdasan buatan ke tingkat yang lebih tinggi. Dengan kemampuannya meniru cara kerja otak manusia melalui jaringan syaraf tiruan yang kompleks, deep learning berpeluang menjadi fondasi utama dalam pengembangan teknologi seperti kendaraan otonom, sistem diagnosis medis berbasis citra, pengenalan suara dan wajah, serta pengambilan keputusan berbasis data besar. Seiring bertambahnya data dan meningkatnya daya komputasi, model-model deep learning akan menjadi semakin akurat dan adaptif, memungkinkan penerapannya secara luas di berbagai sektor industri, pendidikan, dan layanan publik. Hal ini menunjukkan bahwa deep learning tidak hanya sebagai alat bantu teknologi, melainkan sebagai katalisator revolusi digital di masa depan.

2.2.6 Convolutional Neural Network (CNN)

CNN adalah versi resmi dari multilayer perceptrons. Multilayer perceptrons biasanya merujuk ke jaringan yang sepenuhnya terhubung, yaitu,

setiap Neuron dalam satu lapisan terhubung ke semua Neuron di lapisan berikutnya. "Koneksi penuh" dari jaringan-jaringan ini membuat mereka rentan terhadap *overfitting* data. Secara proses regularisasi termasuk menambahkan beberapa bentuk pengukuran bobot untuk fungsi kerugian. Namun, *CNN* mengambil pendekatan yang berbeda terhadap regularisasi mereka mengambil keuntungan dari pola hierarkis dalam data dan mengumpulkan pola yang lebih kompleks menggunakan pola yang lebih kecil dan lebih sederhana (Putra Salawazo, 2019).

2.2.7 ResNet50

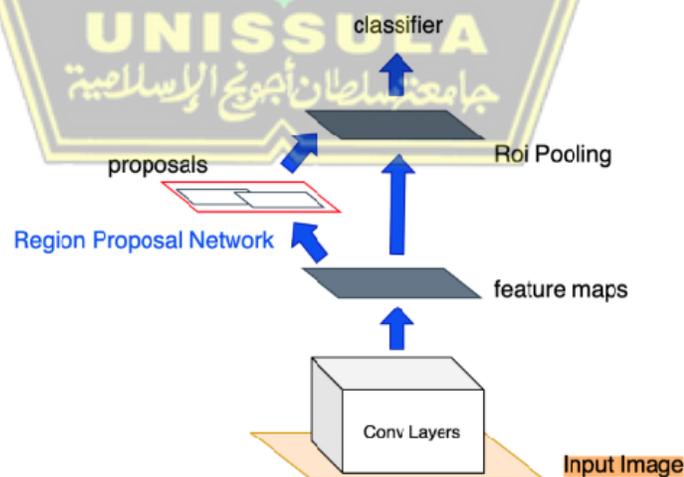
Arsitektur *ResNet* (Residual Networks) dikembangkan oleh He dan rekan-rekannya pada tahun 2016 untuk mengatasi masalah *vanishing gradient* pada jaringan yang sangat dalam (Bohlol dkk., 2025). Keunggulan utama dari arsitektur ini terletak pada penggunaan *skip connections* atau koneksi lompatan, yang memungkinkan aliran informasi melewati beberapa lapisan dan langsung mencapai lapisan yang lebih tinggi. *ResNet* tersedia dalam berbagai varian, seperti *ResNet50* dan *ResNet101*, yang masing-masing terdiri dari 50 dan 101 lapisan. Arsitektur ini meraih peringkat pertama dalam kompetisi ILSVRC 2015 dengan tingkat kesalahan hanya 3,57%. Karena keandalannya dalam mendeteksi fitur pada jaringan yang sangat dalam, *ResNet* telah menjadi pilihan populer dalam berbagai aplikasi visi komputer.

Model Hybrid *ResNet50+FPN* merupakan hasil pengembangan dari penggabungan dua arsitektur utama, yaitu *ResNet50* dan *Feature Pyramid Network (FPN)*, yang masing-masing menawarkan keunggulan dalam ekstraksi fitur pada citra (Vishal dkk., 2025). *ResNet50* dipilih sebagai komponen utama karena kemampuannya dalam menangani citra dengan pola kompleks, berkat koneksi residual yang mempermudah aliran informasi antar lapisan jaringan. Di sisi lain, *FPN* dimanfaatkan untuk memperkuat kemampuan model dalam mengekstraksi fitur pada berbagai skala, khususnya dalam mendeteksi objek berukuran kecil. Dengan menggabungkan kedua arsitektur ini, model hybrid

mampu meningkatkan ketepatan deteksi objek dalam citra, meskipun harus diakui bahwa kombinasi ini juga menambah kompleksitas komputasi, yang pada gilirannya dapat mempengaruhi efisiensi proses inferensi. Penerapan hybrid ini menegaskan pentingnya integrasi berbagai metode untuk mengoptimalkan performa deteksi visual, yang kemudian akan diuraikan lebih mendalam melalui pembahasan mengenai arsitektur *ResNet* yang menjadi fondasi utama dari model hybrid tersebut.

2.2.8 *Faster Region - Convolutional Neural Network (Faster R-CNN)*

Faster R-CNN merupakan algoritma deteksi objek berbasis wilayah terbaru yang menunjukkan hasil luar biasa pada berbagai deteksi objek. *R-CNN* atau *Region Convolutional Neural Network* merupakan metode pendeteksian objek yang masuk ke dalam ranah computer vision yang berbasis jaringan konvolusi atau *CNN*.(Devi, 2023). Pada awalnya, *R-CNN* pada tahun 2015 dibuat untuk melakukan pendeteksian objek yang mengkombinasikan algoritme *Region Proposal Network (RPN)* dan *Convolutional Neural Network (CNN)*. *Convolutional Neural Network (CNN)* termasuk dalam satu turunan jaringan saraf tiruan yang paling berkembang(Wardani, 2021).



Gambar 2. 4 cara kerja *Faster R-CNN*

Pada tahapan model *Faster R-CNN*, pertama-tama akan diproses oleh *CNN* (*Convolutional neural network*). *CNN* akan mencari pola-pola atau fitur seperti tepi, sudut, tekstur, atau bentuk di dalam dataset. *CNN* akan memberikan sebuah “peta fitur” (*feature map*), peta fitur ini bisa dianggap sebagai gambaran kasar dari gambar. Adapun rumus yang digunakan sebagaimana rumus nomor 1.

$$F = f_{CNN}(I) \quad (1)$$

Keterangan:

- F = feature map
- f_{CNN} = *konvolusional* yang digunakan untuk mengekstrak fitur visual dari gambar
- I = gambar input mentah, biasanya dalam format RGB, berukuran $H \times W \times 3$ (tinggi, lebar, 3 channel warna)

Kemudian di proses dengan *Region Proposal Network (RPN)*. Fungsi *RPN* yaitu untuk mencari kemungkinan lokasi objek pada gambar. *RPN* bertugas untuk menghasilkan area-area yang berpotensi mengandung objek kendaraan dari citra input. *RPN* akan membuat *bounding box* (kotak pembatas) yang menunjukkan lokasi di mana kendaraan mungkin berada dengan rumus nomor 2.

$$t_x = \frac{(x - x_a)}{w_a}$$

$$t_y = \frac{(y - y_a)}{h_a}$$

(2)

$$t_w = \log\left(\frac{w}{w_a}\right)$$

$$t_h = \log\left(\frac{h}{h_a}\right)$$

Keterangan:

- (x_a, y_a, w_a, h_a) : anchor box
- (x, y, w, h) : ground truth box

Langkah selanjutnya adalah *ROI Pooling*, peran *ROI Pooling* adalah untuk mengumpulkan area yang sesuai ke dalam peta fitur dengan ukuran tetap sesuai dengan koordinat posisi dari frame yang telah dipilih sebelumnya, sehingga dapat memfasilitasi klasifikasi (Fang dkk., 2025) dengan operasi sesuai rumus nomor 3.

$$f_{ROI} = ROI Pool(F, ROI) \quad (3)$$

Keterangan:

- F = feature map
- ROI = Region of interest
- $ROI Pool$ = Region of interest pooling
- f_{ROI} = hasil dari pooling

ROI Pooling mengubah ukuran area-area menjadi ukuran tetap sehingga dapat diproses oleh lapisan berikutnya. *Faster R-CNN* adalah langkah terakhir, yang mana memiliki dua langkah utama, yaitu *klasifikasi* dan *regresi*.

- Langkah *klasifikasi* bertujuan untuk menentukan apakah setiap area gambar yang diambil oleh *RPN* berisi objek yang relevan atau tidak. Langkah ini dilakukan sesuai rumus nomor 4.

$$P(c|ROI) = \frac{e^{z_c}}{\sum_{j=1}^C e^{z_j}} \quad (4)$$

Keterangan:

- $P(c|ROI)$ = probabilitas ROI mengandung objek dari kelas ke- c
- z_c = Skor logit mentah dari kelas- c
- C = jumlah total kelas
- e = basis logaritma natural (~ 2.718)

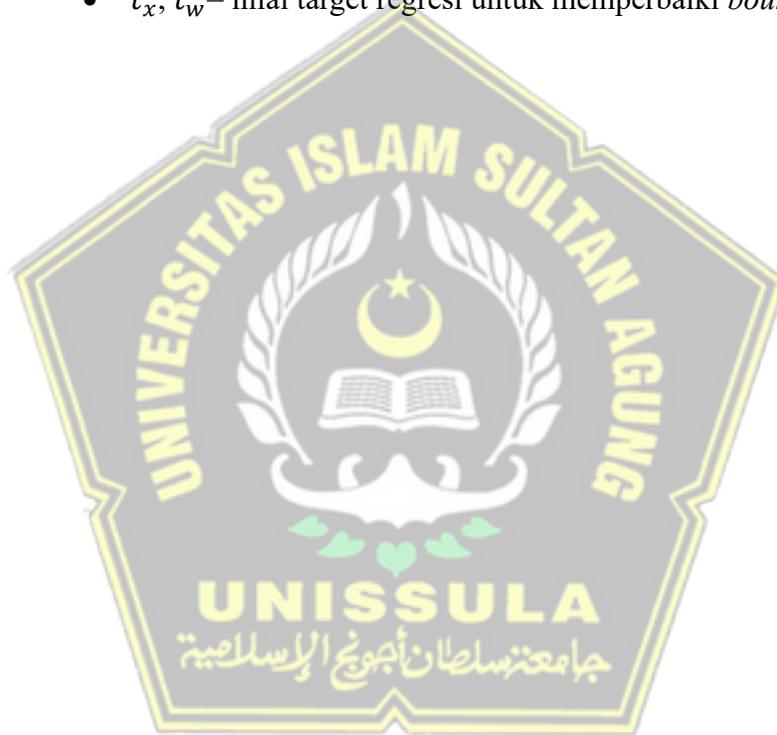
Hasil dari langkah ini adalah probabilitas atau *confidence score* untuk setiap kelas yang menunjukkan seberapa yakin area tersebut mengandung objek yang relevan.

- Langkah *regresi* bertujuan untuk memperbaiki lokasi dan ukuran *bounding box* dari objek yang terdeteksi dengan operasi rumus nomor 5.

$$t_x^* = \frac{(x^* - x_{roi})}{w_{roi}}, t_w^* = \log\left(\frac{w^*}{w_{roi}}\right) \quad (5)$$

Keterangan:

- x^*, w^* = koordinat tengah dan lebar ground truth box
- x_{roi}, w_{roi} = koordinat tengah dan lebar dari RoI
- t_x^*, t_w^* = nilai target regresi untuk memperbaiki *bounding box*.



BAB III

METODOLOGI PENELITIAN

3.1 Studi Literatur

Dalam proses penelitian ini, peneliti secara sistematis mengumpulkan berbagai referensi yang mencakup *e-book*, artikel ilmiah, jurnal, skripsi, dan tesis yang telah disusun oleh peneliti sebelumnya sebagai bagian dari upaya untuk memperkaya pemahaman teori yang mendasari. Selain itu, peneliti juga memanfaatkan sejumlah laman atau situs web di internet yang menyediakan informasi terkait topik penelitian, yang meliputi konsep dasar mengenai bahu jalan, pemahaman mendalam tentang teknologi *Deep Learning*, arsitektur *Convolutional Neural Network* (CNN), serta algoritma *Faster R-CNN*. Langkah ini dilakukan dengan tujuan untuk memastikan bahwa peneliti memiliki pemahaman yang memadai dan komprehensif mengenai literatur yang relevan, sehingga dapat mendukung penyusunan penelitian secara menyeluruh dan mendalam serta menghasilkan analisis yang valid dan sesuai dengan tujuan penelitian yang ditetapkan.

3.2 Rancang Alur Model

Dalam pembuatan sistem deteksi tempat parkir kendaraan menggunakan webcam yang memanfaatkan metode *Faster R-CNN*, terdapat tahap perancangan alur model untuk mempermudah peneliti dalam melakukan pekerjaannya.



Gambar 3. 1 *Flowchart* alur model

Pada gambar 3.1 menampilkan diagram alur (*flowchart*) yang menggambarkan proses utama dalam pengembangan model computer vision. Proses dimulai dari tahap pengumpulan dataset, di mana data gambar dikumpulkan untuk digunakan dalam pelatihan model. Setelah data terkumpul,

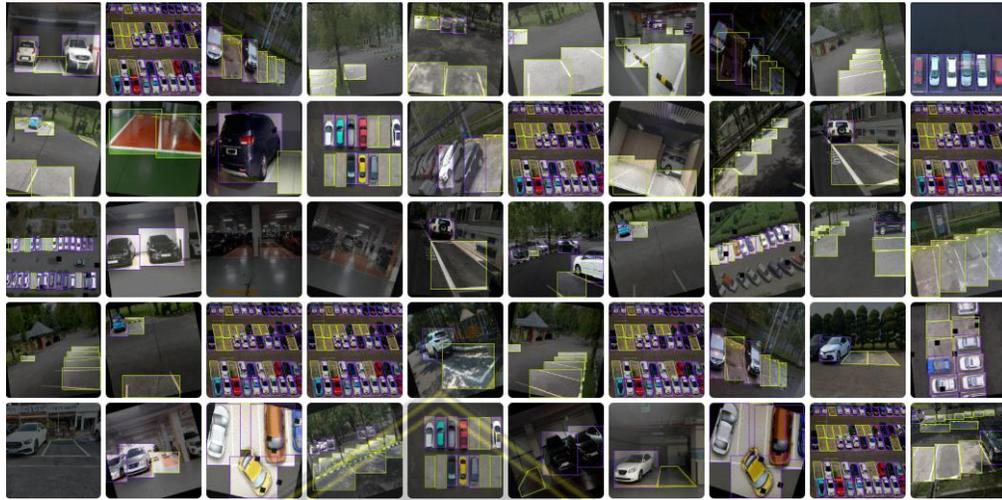
dilakukan pemberian anotasi label pada gambar, yang berfungsi untuk memberikan tanda atau label pada objek-objek dalam gambar agar komputer dapat mempelajarinya. Langkah berikutnya adalah memulai *training*, yaitu pelatihan model menggunakan dataset yang telah dianotasi. Proses ini bertujuan untuk membuat model mampu mengenali dan memahami pola-pola visual pada gambar.

Selanjutnya, diagram menunjukkan bahwa setelah model dilatih, dilakukan pengujian model untuk mengevaluasi performa dan akurasi model tersebut. Jika hasil pengujian memuaskan, model kemudian diimplementasikan dalam sistem nyata untuk digunakan sesuai kebutuhan. Akhir dari alur ini ditandai dengan tahap selesai, yang menandakan bahwa proses pengembangan model computer vision telah rampung. Diagram ini memberikan gambaran yang jelas dan terstruktur mengenai tahapan yang harus dilalui untuk membangun sistem computer vision yang efektif dan efisien.

3.3 Tahapan Perancangan Model

3.3.1 Pengumpulan Data

Dataset yang digunakan dalam penelitian ini berasal dari situs web *Roboflow*, *Roboflow* sendiri merupakan platform yang dirancang khusus untuk membantu *AI engineer* untuk memperoleh dan memproses dataset untuk pengaplikasian proyek *computer vision*, lalu kita ambil data dengan format data Pascal VOC (*Visual Object Classes*) sebanyak 3.000 data. Setiap gambar dalam dataset ini dilengkapi dengan informasi berupa anotasi, label, dan ukuran, yang disimpan dalam file berformat *.xml*. Dataset ini kemudian diproses dan dibagi untuk keperluan pelatihan (*training*) dan validasi (*validation*).



Gambar 3. 2 Dataset

Pada gambar 3.2 merupakan contoh dataset yang akan digunakan dalam proses pelatihan (*training*) dan validasi (*validation*) model deteksi objek, khususnya untuk mendeteksi kendaraan pada area parkir. Setiap gambar dalam kumpulan data ini telah dianotasi dengan *bounding box* berwarna kuning yang menandai posisi kendaraan seperti mobil dan motor. Anotasi ini berfungsi sebagai label yang digunakan oleh model untuk mempelajari dan mengenali objek yang ingin dideteksi. Dataset ini mencakup berbagai kondisi pencahayaan, sudut pengambilan gambar, serta kepadatan kendaraan yang berbeda-beda, sehingga model yang dilatih dapat lebih andal dalam menghadapi situasi di dunia nyata. Data ini kemudian dibagi menjadi dua bagian, yaitu data pelatihan yang digunakan untuk mengajarkan model mengenali pola dan karakteristik kendaraan, serta data validasi yang digunakan untuk mengevaluasi performa model secara berkala selama proses pelatihan berlangsung. Dengan penggunaan data yang beragam dan sudah dianotasi dengan baik seperti pada gambar di atas, model yang dikembangkan diharapkan dapat memiliki tingkat akurasi dan generalisasi yang tinggi dalam mendeteksi kendaraan di berbagai kondisi.

3.3.2 Preprocessing Data

Preprocessing dataset merupakan serangkaian tindakan yang dilakukan untuk mempersiapkan *dataset* supaya sesuai dengan yang dibutuhkan model dalam pembelajarannya. Tahapan *preprocessing* ini meliputi pemeriksaan *bounding box* dan label yang sudah ada untuk disesuaikan dengan kebutuhan dalam pembuatan model. Gambar yang dirasa tidak sesuai akan disaring untuk dimasukkan ke dalam daftar abai. Kemudian, *dataset* yang sudah dipisahkan akan dilakukan augmentasi untuk memperkaya keragaman gambar melalui transformasi yang diberikan. Hal ini dilakukan untuk meningkatkan kualitas data serta memastikan konsistensi data sehingga model dapat belajar dengan baik. Selain itu, dengan melakukan *preprocessing* bisa mengurangi potensi *overfitting* dan membuat model lebih *robust* terhadap variasi gambar berupa rotasi, pencahayaan, maupun skala yang berbeda-beda.

3.3.3 Augmentation Data

Augmentasi gambar merupakan cara yang digunakan untuk mengubah dan memodifikasi gambar sesuai dengan ketentuan yang kita inginkan. Tujuan dari penerapan teknik ini adalah untuk meningkatkan kemampuan model dalam mengenali pola dan mengurangi potensi terjadinya *overfitting*. *Overfitting* dalam kecerdasan buatan adalah masalah umum yang terjadi karena model belajar terlalu detail terhadap data latih, sehingga memiliki kinerja yang baik pada saat data latih tetapi buruk ketika mengenali data baru.

Augmentasi gambar bertujuan untuk menambah keanekaragaman data pelatihan dengan menciptakan gambar-gambar baru yang divariasikan dari gambar aslinya. Teknik ini bekerja dengan membuat salinan gambar asli lalu diterapkan berbagai modifikasi hingga terbentuk variasi baru yang tetap mewakili informasi yang sama. Variasi ini dihasilkan dari berbagai transformasi yang diterapkan. Transformasi ini adalah operasi yang memanipulasi gambar, baik secara posisi, warna, maupun pixel, untuk menciptakan tampilan baru.

Pada umumnya transformasi terbagi ke dalam macam-macam kategori, seperti transformasi geometris, warna, *noise*, *blurring*, dan pixel. Dimana masing-masing kategori transformasi memiliki jenis yang berbeda-beda seperti berikut:

- *Flipping* adalah salah satu bentuk transformasi geometris yang bekerja dengan cara membalik citra secara horizontal atau vertical. Transformasi ini menjaga struktur objek dalam gambar untuk mengubah orientasi sehingga model belajar untuk mengenali objek dalam posisi cermin
- *Rotation* merupakan transformasi yang mensimulasikan kondisi nyata Dimana objek gambar dapat dilihat dari sudut pandang yang berbeda. *Rotation* membantu model belajar mendeteksi objek meskipun tampilannya diputar.
- *Scaling* adalah transformasi geometris yang mengubah ukuran gambar dengan cara memperbesar atau memperkecil. Transformasi ini berfungsi untuk melatih model dalam mendeteksi objek dengan berbagai ukuran.
- *Brightness* merupakan transformasi yang mengubah tingkat kecerahan gambar, membuat citra menjadi lebih terang atau gelap. *Brightness* sangat berguna untuk mensimulasikan kondisi pencahayaan seperti siang, malam, atau bayangan sehingga model *robust* terhadap perubahan pencahayaan.
- *Contrast* berfungsi untuk memodifikasi perbedaan antara area terang dan gelap pada gambar. Transformasi ini membantu model belajar dari gambar dengan kualitas visual yang berbeda, baik dengan detail tajam maupun buram.
- *Gaussian noise* adalah transformasi yang menambahkan bintik-bintik acak pada gambar, Dimana distribusi noise mengikuti pola *gaussian* (distribusi normal). Transformasi ini berguna untuk mensimulasikan gangguan seperti kualitas kamera rendah atau pengaruh lingkungan.
- *Motion blur* merupakan transformasi *blurring* yang menambahkan efek kabur menyerupai citra yang dihasilkan oleh kamera yang bergerak atau objek yang bergerak cepat. Efek ini mensimulasikan kondisi dunia nyata saat gambar

buram akibat Gerakan sehingga model dapat belajar untuk mendeteksi pola meskipun citra tidak tajam.

- *Normalization* adalah transformasi pixel yang mengubah rentang nilai pixel yang biasanya dari rentang asli 0-255 menjadi 0-1. Transformasi ini membuat data lebih seragam dan membantu dalam perhitungan numerik, karena model deep learning lebih mudah bekerja dengan nilai kecil yang teratur. *Normalization* membantu mencegah masalah seperti *vanishing gradients* dan mempercepat proses pelatihan.
- *Standardization* termasuk transformasi pixel yang mengubah nilai pixel menjadi memiliki rata-rata 0 dan standar deviasi 1. Membuat data menjadi lebih stabil dan terdistribusi normal sehingga memudahkan model untuk mempelajari pola dalam data.

Augmentasi gambar bekerja dengan cara menghasilkan variasi gambar yang berbeda dari gambar asli. transformasi di atas adalah terbentuknya variasi data citra yang berbeda-beda sehingga model dapat dilatih untuk mendeteksi objek dalam berbagai kondisi. Transformasi geometris seperti flipping, rotation, dan scaling mengajarkan model mengenali objek dari sudut pandang dan ukuran yang beragam. Transformasi brightness dan contrast membantu model tetap akurat meskipun gambar berbeda kondisi pencahayaannya. Penambahan noise dan blur, seperti gaussian noise dan motion blur, mensimulasikan gangguan pada citra sehingga model mampu mendeteksi objek meskipun gambar tidak bersih atau kabur. Normalization dan standardization membuat data pixel lebih seragam dan stabil, sehingga memudahkan model dalam proses pembelajaran. Dengan variasi data ini, model menjadi lebih robust dan mampu mendeteksi objek dengan baik dalam kondisi nyata yang bervariasi.

3.3.4 Pembuatan Model

Dalam pembuatan model yang digunakan untuk deteksi tempat parkir, peneliti menggunakan *Google Colaboratory* sebagai media untuk menuliskan sintaks dari bahasa pemrograman *python*. Model yang dibuat ini akan menggunakan metode *Faster R-CNN* dalam membantu sistem mendeteksi objek. *Faster R-CNN* dipilih karena menggunakan dua tahap deteksi (*region proposal network* dan klasifikasi *regresi bounding box*), sehingga lebih konsisten dalam mengenali objek, bahkan dalam kondisi pencahayaan atau sudut pandang yang bervariasi.

3.3.5 Training dan Validation

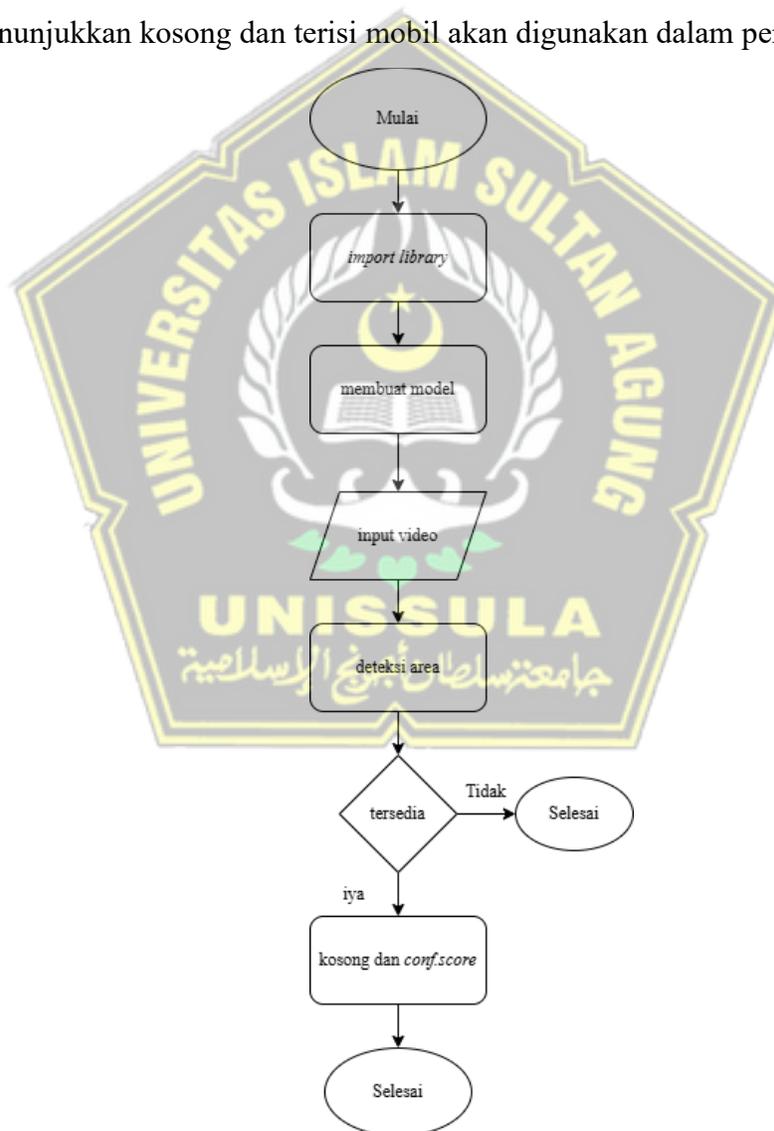
Tahapan *training* adalah tahapan ketika model yang sudah dibuat pada *google colaboratory* akan dilatih menggunakan dataset yang sudah disiapkan sebelumnya oleh peneliti dengan jumlah 3000 data. Data *training* mendapatkan lebih banyak data karena kebutuhan sistem untuk melatih model. Pada data yang digunakan ini sudah terdapat *bounding box* pada setiap gambarnya yang tentunya memudahkan peneliti untuk melatih model mengenali objek yang akan dijadikan kelas. Tahapan *training* ini meliputi penyesuaian jumlah *Epoch* yang nantinya akan mempengaruhi performa model yang dihasilkan. Apabila *Epoch* terlalu banyak, model akan cenderung mempelajari data terlalu detail, akibatnya akan kehilangan kemampuan generalisasi. Sebaliknya, apabila *Epoch* terlalu sedikit mengakibatkan model belum cukup untuk mempelajari dataset yang digunakan.

Setelah dilakukan tahapan *training*, maka dilanjutkan tahapan *validation*. Tujuan dari tahapan ini adalah untuk menilai performa dari model yang sudah dilatih sebelumnya. Apabila dalam tahapan validasi diketahui model belum mencapai performa terbaik, maka akan dilakukan tahapan *training* kembali berupa penambahan atau pengurangan *Epoch*. Jumlah *Epoch* yang ideal tergantung pada kompleksitas data, model, dan tujuan pelatihan. Model diuji menggunakan *validation* dataset yang berbeda dari data *training* untuk

mengukur kinerja dan mencegah *overfitting* lalu menilai *overfitting* jika *loss* validasi naik sementara *loss training* turun.

3.3.6 Pengujian

Peneliti akan melakukan pengujian terhadap model yang telah dibuat pada area parkir yang memiliki marka pembatas. Hasil deteksi dari model yang dibuat akan menjadi acuan peneliti untuk menyimpulkan performa maupun akurasi dari model. Model yang telah dilatih dengan gambar-gambar tempat parkir yang menunjukkan kosong dan terisi mobil akan digunakan dalam pengujian.



Gambar 3. 3 Alur pengujian

Flowchart yang ditampilkan pada gambar 3.3 merupakan alur dari tahapan pengujian dalam penelitian yang berkaitan dengan deteksi area parkir menggunakan video. Pengujian dimulai dengan mengimpor pustaka (*library*) yang diperlukan untuk menjalankan model dan mengelola data. Selanjutnya, model deteksi yang telah dibuat atau dilatih sebelumnya diaktifkan. Setelah model siap, video sebagai data uji dimasukkan untuk dianalisis. Proses berlanjut dengan mendeteksi area parkir dalam video tersebut. Kemudian dilakukan pengecekan terhadap ketersediaan area parkir; jika tidak ada area yang tersedia, maka pengujian dihentikan. Namun jika tersedia, sistem akan mengevaluasi status kekosongan area serta nilai *confidence score* dari prediksi model untuk mengukur tingkat keyakinan hasil deteksi. Tahapan ini diakhiri dengan menyimpan atau mencatat hasil pengujian, yang nantinya akan digunakan untuk analisis lebih lanjut dalam penelitian.

3.3.7 Evaluasi

Pada tahapan evaluasi model, peneliti akan menggunakan beberapa matrik evaluasi seperti *accuracy*, *precision*, *recall*, dan *f1-score*. Untuk rumus dari masing-masing metrik adalah sebagai berikut:

- a. *Accuracy* merupakan proses untuk mengukur seberapa sering model membuat prediksi yang benar dibandingkan dengan seluruh prediksi yang dilakukan.

$$\frac{TP+TN}{Total} \quad (6)$$

- b. *Precision* adalah persentase deteksi positif yang benar (benar terdeteksi sebagai tempat parkir) dibandingkan dengan semua deteksi positif.

$$\frac{TP}{TP+FP} \quad (7)$$

- c. *Recall* merupakan persentase tempat parkir yang berhasil terdeteksi dibandingkan dengan semua tempat parkir yang ada dalam *dataset*.

$$\frac{TP}{TP+FN} \quad (8)$$

- d. *F1-score* merupakan metrik yang mengukur keseimbangan antara *precision* dan *recall* dalam satu nilai. Semakin tinggi nilainya, maka semakin baik performa model secara keseluruhan dalam mendeteksi dan tidak salah deteksi.

$$\frac{2}{\frac{1}{Precision} + \frac{1}{Recall}} \quad (9)$$

Matrik evaluasi ini diperoleh dari hasil *confusion matrix*, adalah sebuah tabel yang menunjukkan jumlah klasifikasi dari hasil data uji benar dan salah.

Tabel 3. 1 *Confusion Matrix*

	Prediksi Positif	Prediksi Negatif
Positif	TP	FN
Negatif	FP	TN

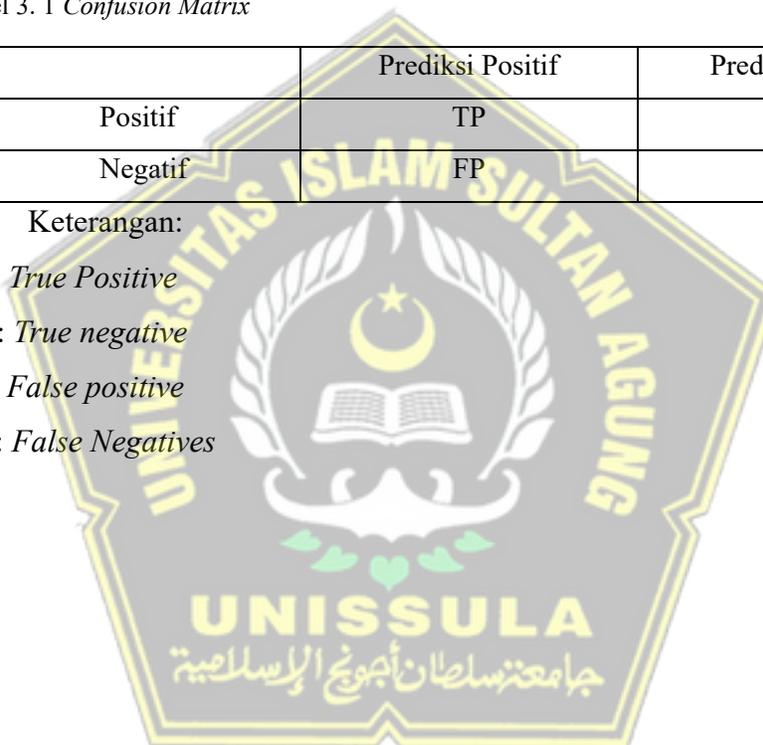
Keterangan:

TP: *True Positive*

TN: *True negative*

FP: *False positive*

FN: *False Negatives*

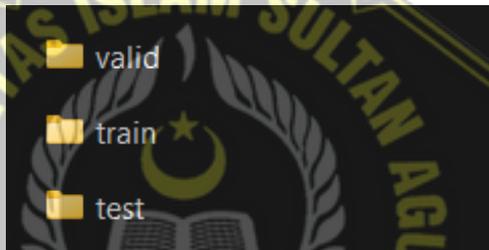


BAB IV

HASIL DAN ANALISIS PENELITIAN

4.1 Hasil

Berikut adalah hasil dari pengerjaan model yang telah di buat, Data yang telah terkumpul pada penelitian ini sebanyak 3.000 gambar yang bersumber dari *website roboflow* dimana data terbagi dalam tahapan *train*, *valid*, dan *test*. Data yang diperoleh sudah memiliki *bouding box* yang tersimpan dalam *file* tipe *.xml* dengan kelas “*vacant*” dan “*occupied*”. Pada gambar 4.1 dan 4.2 menunjukkan kondisi awal dari *dataset* yang diperoleh dari *roboflow*.



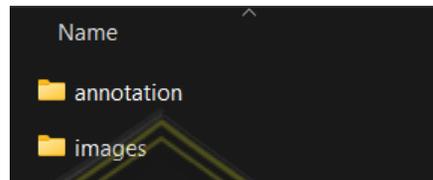
Gambar 4. 1 folder dataset

Pada gambar 4.1 adalah folder mentahan untuk data yang akan kita olah, dapat dilihat struktur *folder dataset valid, train, dan test* yang akan kita bagi untuk dilakukan pelatihan dan evaluasi model. Dalam pembuatan model kita akan menggunakan dataset *train dan valid*.

Name	Type
-_jpg.rf.301502166d0618623f0...	JPG File
-_jpg.rf.301502166d0618623f0...	Microsoft Edge HT...
001_20220315_174309-mp4_20...	JPG File
001_20220315_174309-mp4_20...	Microsoft Edge HT...
001_20220315_174309-mp4_20...	JPG File
001_20220315_174309-mp4_20...	Microsoft Edge HT...
001_20220315_174309-mp4_20...	JPG File
001_20220315_174309-mp4_20...	Microsoft Edge HT...
1-_jpg.rf.28cfb77848667ee068...	JPG File

Gambar 4. 2 File folder anotasi dan gambar

Pada gambar 4.2 adalah isi dari *folder train* dan *valid* yang akan digunakan, data yang diperoleh akan disatukan semua. Data tersebut disatukan dalam satu *folder* dengan membuat *subfolder annotation* yang berisi *file annotation* dan *images* yang menyimpan seluruh *file* gambar seperti pada gambar 4.3 untuk digunakan dalam tahapan pembuatan model.



Gambar 4. 3 Folder dataset baru

pada gambar 4.3 adalah *folder annotation* dan *images* yang telah kita pisahkan. folder ini dibuat untuk memastikan bahwa *pipeline* pemrosesan data, pelatihan, dan evaluasi model dapat berjalan dengan baik tanpa terjadinya *error*. Setelah menyatukan data ke dalam struktur *folder*, proses augmentasi, pembacaan *file*, serta pemetaan gambar dengan anotasi akan lebih efisien dan minim terjadinya kesalahan saat dieksekusi dalam kode. Sebelum dilakukan split data, data akan dilakukan label mapping yang bertujuan untuk menyederhanakan dan menyatukan berbagai variasi label yang terdapat pada dataset agar menjadi mudah dan seragam diproses oleh model. Pada pembuatan model deteksi tempat parkir terdapat label seperti "*vacant*" dan "*occupied*". Pada label "*occupied*" akan menunjukkan kategori yaitu "terisi" dan pada label "*vacant*" akan menunjukkan kategori "kosong". Dengan label mapping, variasi "terisi" akan dipetakan menjadi angka representative seperti angka 0 dan variasi "kosong" menjadi angka 1 seperti pada gambar 4.4.

```
# Definisikan mapping label
label_mapping = {
    "occupied": 0,
    "vacant": 1
}
```

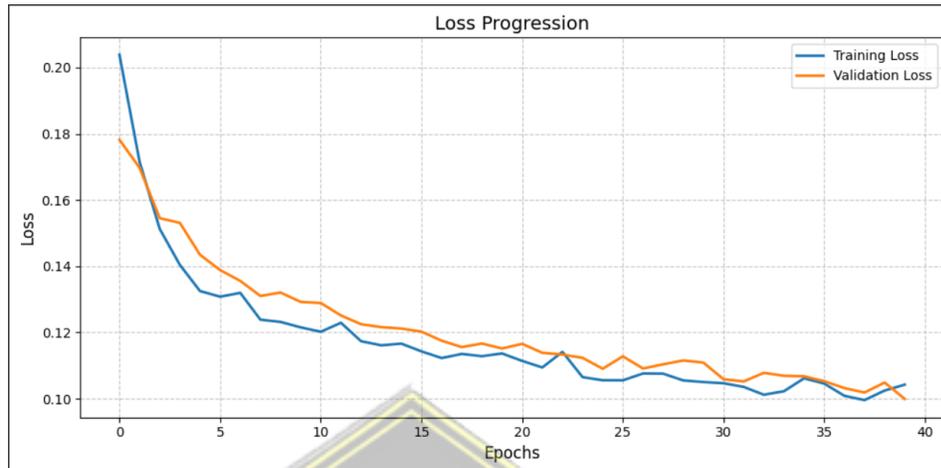
Gambar 4. 4 Definisi mapping label

Data tersebut kemudian dilakukan augmentasi untuk memperkaya keragaman data menggunakan berbagai transformasi dengan hasil seperti gambar 4.5. gambar tersebut menunjukkan hasil deteksi tempat parkir dengan label “background” dan “kosong”, yang dianotasikan menggunakan *bounding box* berwarna hijau.



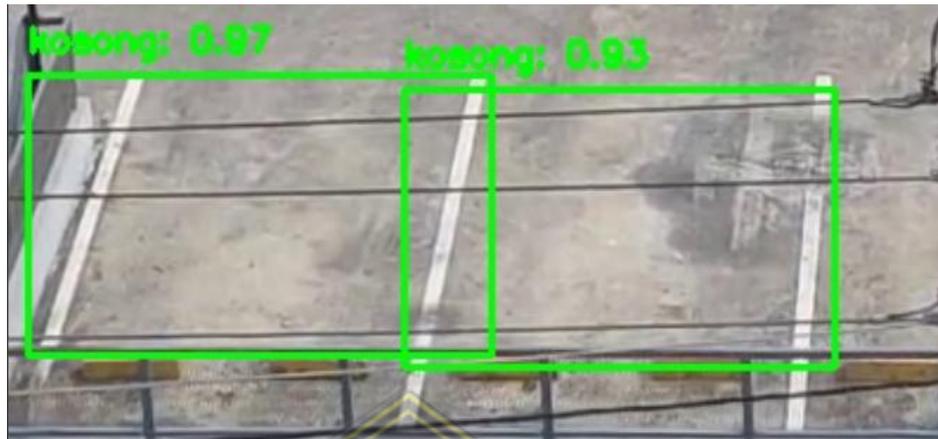
Gambar 4. 5 Hasil augmentasi

Hasil pembuatan model yang dilakukan dengan tahapan *training* dan *validation* menggunakan dataset yang ada akan menghasilkan grafik *loss*. Grafik *loss* pada gambar 4.6 merupakan hasil dari pelatihan model selama 40 *Epoch*, garis yang berwarna biru menunjukkan *training loss* dan garis berwarna orange menunjukkan *validation loss*.



Gambar 4. 6 *loss progression*

Grafik *loss progression* pada *training loss* (warna biru) menunjukkan nilai loss dengan kesalahan prediksi model terhadap data latih yang cenderung turun secara bertahap, meski terdapat sedikit fluktuasi. Pada *validation loss* (warna orange) menunjukkan kesalahan prediksi model terhadap data validasi untuk menilai kemampuan generalisasi model ke data baru. Kedua *loss (training & validation)* menurun seiring bertambahnya *Epoch* menunjukkan bahwa model belajar secara efektif dari data. Pada pola ini menunjukkan kedua *loss* memiliki indikasi *positif* bahwa model tidak hanya menghafal data latih, tapi juga bisa memprediksi data baru dengan baik sehingga tidak ada tanda-tanda *overfitting* dan model cocok jika digunakan pada data nyata. Dengan demikian, hasil penelitian menunjukkan bahwa proses pelatihan model berjalan secara efektif dan menghasilkan performa yang baik.



Gambar 4. 7 hasil deteksi tempat parkir kosong

Gambar 4.7 di atas menampilkan hasil deteksi objek dari model yang digunakan untuk mendeteksi status parkir mobil. Dalam gambar tersebut, model mendeteksi dua area parkir yang ditandai dengan kotak hijau dan label bertuliskan "kosong" yang berarti area tersebut tidak ditempati kendaraan. Model juga menampilkan skor kepercayaan (*confidence score*) dengan nilai 0.97 dan 0.93, yang menunjukkan bahwa model cukup yakin terhadap deteksinya bahwa area tersebut kosong. Kotak hijau mengelilingi area parkir yang kosong dengan jelas dan menandakan lokasi deteksi tersebut. Secara keseluruhan, hasil deteksi ini menunjukkan bahwa model dapat mendeteksi ruang parkir kosong dengan cukup baik dan memberikan informasi visual yang jelas melalui *bounding box* dan label *confidence*. Model lalu diunduh dalam *file .pth* untuk dilakukan pengujian menggunakan *google colaboratory*. Pada gambar 4.7 menunjukkan hasil dari pengujian sistem deteksi tempat parkir yang telah dikembangkan. Di mana sistem mampu mengenali tempat parkir yang kosong dan terisi dengan cukup akurat lalu memberikan *bounding box* berwarna hijau secara otomatis.

```
Mean Average Precision[0.5:0.95:0.05] : 0.2523
Mean Average Precision @ 0.5          : 0.3729
Mean Average Precision @ 0.75         : 0.2929
```

Gambar 4. 8 hasil mAP

Mean Average Precision (mAP) [0.5:0.95:0.05] sebesar 0.2523 menunjukkan rata-rata performa deteksi objek pada berbagai tingkat ketelitian, dari IoU 0.5 hingga 0.95. Hasil mAP @ 0.5 sebesar 0.3729 menunjukkan performa model dalam mendeteksi objek dengan ketelitian minimum 50%, yang berarti cukup toleran. Sedangkan hasil mAP @ 0.75 sebesar 0.2929 menggambarkan kinerja model pada tingkat ketelitian yang lebih tinggi (75%), yang membutuhkan *bounding box* prediksi yang lebih presisi. Semakin tinggi nilai mAP, semakin baik model dalam mendeteksi objek dengan benar; di sini terlihat model memiliki performa lebih baik pada IoU 0.5 dibandingkan dengan IoU 0.75. Secara keseluruhan, nilai mAP yang didapatkan cukup baik, tetapi masih memiliki ruang untuk peningkatan terutama pada tingkat ketelitian yang lebih tinggi.

4.2 Analisa

Pada tabel 4.1 menunjukkan *hyperparameter* yang digunakan dalam pembuatan model. Parameter-parameter ini sangat memengaruhi performa dan hasil akhir dari model, baik dari segi kecepatan konvergensi, akurasi deteksi, maupun kestabilan selama pelatihan.

Tabel 4.1 *Hyperparameter*

Parameter	Nilai
<i>Batch Size</i>	3
<i>Epoch</i>	40
<i>Learning Rate</i>	0.0003
Optimizer	Adam
<i>Threshold</i>	0.25
<i>IoU Threshold</i>	0.5

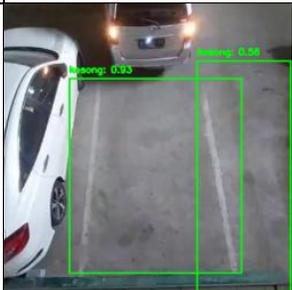
Tabel 4.1 menjelaskan beberapa parameter penting yang digunakan dalam proses pelatihan model deteksi objek. *Batch Size* diatur sebesar 4, yang berarti

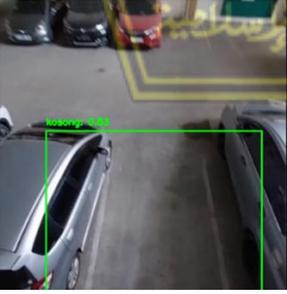
setiap iterasi pelatihan akan memproses 4 gambar sekaligus sebelum memperbarui bobot model. *Epoch* yang digunakan adalah 40, menunjukkan bahwa dataset dilatih selama 40 putaran penuh, sehingga model diharapkan dapat mempelajari pola data dengan baik. *Learning Rate* diatur ke 0.0003, yang mengontrol seberapa besar langkah pembaruan bobot model dilakukan pada setiap iterasi. Optimizer yang digunakan adalah Adam (*Adaptive Moment Estimation*), sebuah algoritma pengoptimalan yang menggabungkan kecepatan konvergensi dari metode momentum dan *adaptive Learning Rate*, sehingga membuat proses pelatihan lebih cepat dan stabil.

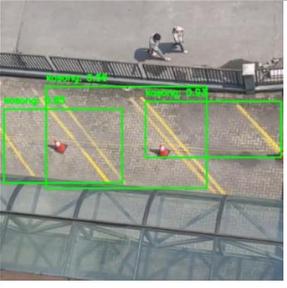
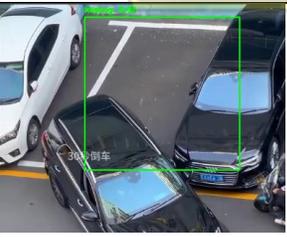
Selain itu, *Threshold* ditetapkan pada 0.25, yang artinya model akan menganggap deteksi valid hanya jika confidence score (tingkat keyakinan) dari *bounding box* prediksi lebih besar dari 25%. *IoU Threshold* diset pada 0.5, yang menentukan bahwa prediksi *bounding box* dianggap benar (*True Positive*) hanya jika tingkat tumpang tindih antara prediksi dan ground truth lebih dari 50%. Pemilihan parameter-parameter ini mempengaruhi hasil pelatihan dan performa model, termasuk akurasi, kecepatan deteksi, dan kemampuan membedakan antara objek yang sebenarnya dan yang salah.

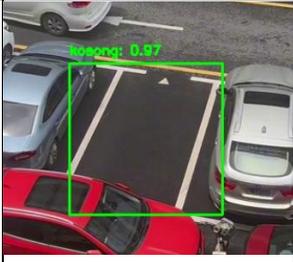
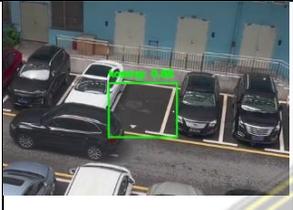
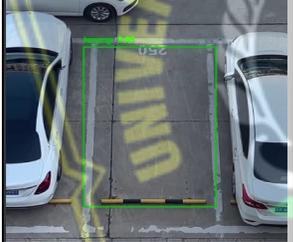
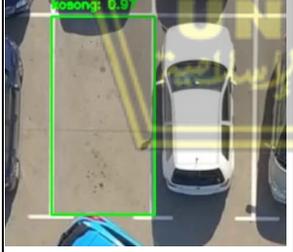
Hasil menggunakan parameter-parameter dengan nilai tersebut, memperoleh hasil pengujian seperti dalam tabel 4.2 berikut:

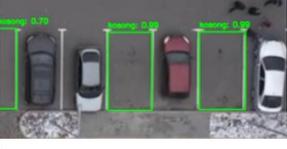
Tabel 4. 2 Hasil penelitian

No	Gambar	Pred	Act	Conf. Score	Threshold				
					0.9	0.8	0.7	0.6	0.5
1.		Kosong	Kosong	0.93	TP	TP	TP	TP	TP

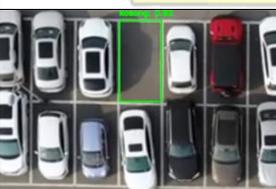
2.		kosong	kosong	0.59	FN	FN	FN	FN	TP
3.		kosong	kosong	0.90	TP	TP	TP	TP	TP
4.		kosong	kosong	0.98	TP	TP	TP	TP	TP
5.		kosong	kosong	0.95	TP	TP	TP	TP	TP
6.		kosong	kosong	0.83	FN	TP	TP	TP	TP

7.		kosong	Kosong	0.93	TP	TP	TP	TP	TP
8.		kosong	Kosong	0.98	TP	TP	TP	TP	TP
9.		kosong	kosong	0.97	TP	TP	TP	TP	TP
10.		kosong	kosong	0.81	FN	TP	TP	TP	TP
11.		kosong	kosong	0.99	TP	TP	TP	TP	TP
12.		kosong	kosong	0.68	FN	FN	FN	TP	TP

13.		kosong	kosong	0.97	TP	TP	TP	TP	TP
14.		kosong	kosong	0.88	FN	TP	TP	TP	TP
15.		kosong	kosong	0.78	FN	FN	TP	TP	TP
16.		Kosong	kosong	0.68	FN	FN	FN	TP	TP
17.		Kosong	Kosong	0.97	TP	TP	TP	TP	TP
18.		Kosong	Kosong	0.98	TP	TP	TP	TP	TP

19.		Kosong	Kosong	0.99	TP	TP	TP	TP	TP
20.		Kosong	Kosong	0.99	TP	TP	TP	TP	TP
21.		Kosong	Kosong	0.83	FN	TP	TP	TP	TP
22.		Kosong	Kosong	0.99	TP	TP	TP	TP	TP
23.		Kosong	Kosong	0.99	TP	TP	TP	TP	TP
24.		Kosong	Kosong	0.99	TP	TP	TP	TP	TP

25.		Kosong	Kosong	0.99	TP	TP	TP	TP	TP
26.		Kosong	Kosong	0.96	TP	TP	TP	TP	TP
27.		Kosong	Kosong	0.99	TP	TP	TP	TP	TP
28.		Kosong	Kosong	0.99	TP	TP	TP	TP	TP
29.		Kosong	Kosong	0.79	FN	FN	TP	TP	TP
30.		Kosong	Kosong	0.52	FN	FN	FN	FN	TP

31.		Kosong	Kosong	0.95	TP	TP	TP	TP	TP
32.		Kosong	Kosong	0.91	TP	TP	TP	TP	TP
33.		Kosong	Kosong	0.73	FN	FN	TP	TP	TP
34.		Kosong	Kosong	0.94	TP	TP	TP	TP	TP
35.		Kosong	Kosong	0.95	TP	TP	TP	TP	TP
36.		Kosong	Kosong	0.92	TP	TP	TP	TP	TP
37.		Kosong	Kosong	0.93	TP	TP	TP	TP	TP

38.		Terisi	Terisi	0.0	TN	TN	TN	TN	TN
39.		Kosong	Terisi	0.5	TN	TN	TN	TN	FP
40.		Terisi	Kosong	0.0	FN	FN	FN	FN	FN

Pada tabel 4.2 menunjukkan prediksi (*pred*) model dibandingkan dengan label aktual (*Act*) apakah slot parkir tersebut kosong atau tidak. Kolom *Conf.Score* menunjukkan tingkat keyakinan model terhadap prediksi yang dibuat, sedangkan di setiap *Threshold*, sistem mengevaluasi apakah keyakinan prediksi melebihi ambang batas. Jika *Confident Score* melebihi *Threshold* tertentu dan label prediksi cocok dengan label aktual, maka akan dicatat sebagai *True Positive* (TP), sebaliknya jika *Confident Score* melebihi *Threshold* tetapi prediksi tidak sesuai label aktual maka akan tercatat sebagai *False Negative* (FN). Dengan adanya berbagai *Threshold* ini, tabel memberikan informasi mengenai bagaimana

sensitivitas model berubah seiring dengan perubahan tingkat ambang batas. Semakin rendah *Threshold*, model akan lebih sering mencatat prediksi sebagai benar (TP), namun potensi kesalahan juga meningkat.

Evaluasi pada model ini menggunakan beberapa metrik penting, yaitu *accuracy*, *precision*, *recall*, dan *f1-score*. Metrik *accuracy* dihitung menggunakan rumus 6 untuk memberikan ukuran seberapa baik model bekerja secara keseluruhan, dengan menghitung persentase prediksi yang benar dibandingkan dengan semua data yang diuji. Akurasi yang tinggi menunjukkan bahwa model membuat prediksi yang benar untuk sebagian besar data. *Precision* menggunakan rumus nomor 7, kegunaan rumus ini untuk mengukur seberapa akurat model dalam mendeteksi objek kelas yang positif, yaitu seberapa besar dari semua yang diprediksi positif oleh model. Semakin tinggi *precision*, semakin sedikit kesalahan prediksi positif yang dibuat model. Nilai *recall* dihitung menggunakan rumus nomor 8 untuk mengukur seberapa baik model dalam menemukan semua objek yang sebenarnya positif. Lalu *recall* memberi tahu seberapa banyak dari semua objek yang seharusnya terdeteksi, benar-benar terdeteksi oleh model. Semakin tinggi *recall*, semakin sedikit objek positif yang terlewat. Selain itu, *f1-score* dengan menggunakan rumus nomor 9 untuk mengukur keseimbangan antara *precision* dan *recall* dari suatu model. *Precision* sendiri mengukur seberapa akurat prediksi positif model, sementara *recall* mengukur seberapa baik model menemukan semua data positif yang benar. *F1-score* sangat berguna ketika kita ingin mempertimbangkan baik akurasi maupun kelengkapan deteksi, terutama jika data yang digunakan tidak seimbang, misalnya ketika jumlah data positif jauh lebih sedikit daripada data negatif. Rumus ini menyatukan kedua metrik tersebut dalam satu angka, sehingga semakin tinggi nilai *f1-score*, semakin baik performa model dalam hal prediksi positif yang benar dan kemampuan menemukan semua data positif yang ada. Dengan demikian, *f1-score* memberikan gambaran yang lebih menyeluruh mengenai performa model daripada hanya melihat presisi atau *recall* secara terpisah.

Berdasarkan hasil pengujian model yang telah dilakukan, diperoleh nilai dari masing-masing metrik. Hasil ini dapat menjadi tumpuan untuk menilai model sudah layak digunakan pada aplikasi nyata atau memerlukan pengembangan lebih lanjut. Adapun hasil pengujian model yang dilakukan dengan hasil berikut:

1. Pada nilai *Threshold* 0.9 menghasilkan 26 *True Positive*, 0 *false positive*, 2 *true negative*, dan 12 *False Negative*. Sehingga diperoleh nilai *accuracy* sebesar 70%, *precision* sebesar 100%, *recall* sebesar 68,4%, dan *f1-score* sebesar 81,2%.
2. Pada nilai *Threshold* 0.8 menghasilkan 30 *True Positive*, 0 *false positive*, 2 *true negative*, dan 8 *False Negative*. Sehingga diperoleh nilai *accuracy* sebesar 80%, *precision* sebesar 100%, *recall* sebesar 78,9%, dan *f1-score* sebesar 88,2%.
3. Pada nilai *Threshold* 0.7 menghasilkan 33 *True Positive*, 0 *false positive*, 2 *true negative*, dan 5 *False Negative*. Sehingga diperoleh nilai *accuracy* sebesar 87,5%, *precision* sebesar 100%, *recall* sebesar 86,8%, dan *f1-score* sebesar 92,9%.
4. Pada nilai *Threshold* 0.6 menghasilkan 35 *True Positive*, 0 *false positive*, 2 *true negative*, dan 3 *False Negative*. Sehingga diperoleh nilai *accuracy* sebesar 92,5%, *precision* sebesar 100%, *recall* sebesar 92,1%, dan *f1-score* sebesar 95,9%.
5. Pada nilai *Threshold* 0.5 menghasilkan 37 *True Positive*, 1 *false positive*, 1 *true negative*, dan 1 *False Negative*. Sehingga diperoleh nilai *accuracy* sebesar 95%, *precision* sebesar 97,3%, *recall* sebesar 97,3%, dan *f1-score* sebesar 97,3%.

Berdasarkan hasil pengujian model pada berbagai nilai *Threshold*, dapat disimpulkan bahwa penurunan *Threshold* secara konsisten meningkatkan performa model dalam mendeteksi objek. Pada *Threshold* 0.9, meskipun *precision* mencapai 100%, *recall* masih cukup rendah di angka 68,4% sehingga

f1-score hanya 81,2%, sedangkan akurasi sebesar 70%. Seiring penurunan *Threshold* ke 0.8 dan 0.7, model menunjukkan peningkatan *recall* menjadi 78,9% dan 86,8%, serta *f1-score* meningkat menjadi 88,2% dan 92,9%, dengan akurasi yang juga naik menjadi 80% dan 87,5%. Pada *Threshold* 0.6, performa mendekati sempurna dengan *recall* sebesar 92,1%, *f1-score* sebesar 95,9%, dan akurasi mencapai 92,5%. Akhirnya, pada *Threshold* 0.5, meskipun *precision* sedikit turun menjadi 97,3% karena adanya satu *false positive*, *recall* juga mencapai 97,3% dan akurasi tertinggi sebesar 95%, menunjukkan bahwa model mampu mendeteksi objek secara hampir sempurna dengan tingkat kesalahan yang sangat rendah. Keseluruhan hasil ini menunjukkan bahwa penyesuaian *Threshold* sangat berpengaruh terhadap performa deteksi, dan *Threshold* 0.5 direkomendasikan untuk hasil optimal.

Besar dan kecil *threshold* berpengaruh terhadap hasil prediksi yang dihasilkan. Semakin besar *threshold* pada model, maka hasil prediksinya semakin bernilai tinggi karena selama pelatihan model terlatih dengan standar yang ketat, artinya hasil pelatihan model yang menunjukkan nilai *threshold* di bawah ketentuan akan dianggap *miss*. Begitu pula dengan sebaliknya, apabila *threshold* yang diterapkan kecil maka Tingkat toleransi yang diberikan pada pelatihan model akan lebih besar dan ini mengakibatkan prediksi dengan *threshold* kecil akan dianggap benar.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil penelitian menunjukkan bahwa model deteksi tempat parkir menggunakan metode *Faster R-CNN* mampu memberikan performa yang baik dalam mendeteksi objek. Pengujian model dengan berbagai nilai *Threshold* memperlihatkan bahwa performa model cenderung meningkat seiring dengan penurunan *Threshold*. Hal ini ditandai dengan peningkatan *accuracy*, *recall*, dan *f1-score* yang mencapai nilai mendekati sempurna pada *Threshold* 0,5 dengan *accuracy* sebesar 95%, *precision* sebesar 97,3%, *recall* sebesar 97,3%, dan *f1-score* sebesar 97,3%. Meskipun *precision* konsisten tinggi pada semua *Threshold*, *recall* yang semula rendah pada *Threshold* tinggi meningkat signifikan seiring penurunan *Threshold*, hal ini menunjukkan penurunan jumlah *False Negative* yang dihasilkan model. Demikian pula dapat dilihat pada *Threshold* 0,9 dengan *accuracy* sebesar 70%, *precision* sebesar 100%, *recall* sebesar 68,4%, dan *f1-score* sebesar 81,2%. Hasil ini membuktikan bahwa penyesuaian *Threshold* memiliki pengaruh besar terhadap performa deteksi model, dan *Threshold* 0,5 direkomendasikan sebagai ambang keputusan optimal untuk menghasilkan kinerja model terbaik.

5.2 Saran

Berikut adalah saran yang dapat dipertimbangkan untuk pengembangan dan perbaikan model deteksi tempat parkir menggunakan metode *Faster R-CNN* di masa mendatang. Meskipun model ini telah menunjukkan performa yang baik dengan akurasi mencapai 95%, *precision* sebesar 97,3%, *recall* sebesar 97,3%, dan *f1-score* sebesar 97,3% pada *Threshold* 0,5, masih terdapat beberapa aspek yang dapat ditingkatkan untuk mencapai akurasi dan generalisasi yang lebih baik pada berbagai kondisi nyata. Selain itu, pemilihan *Threshold* yang optimal juga

perlu disesuaikan dengan kebutuhan aplikasi, apakah lebih mementingkan sensitivitas atau ketepatan.

Beberapa saran pengembangan ke depan adalah sebagai berikut:

1. Mengeksplorasi penggunaan backbone lain seperti *ResNet*, *EfficientNet*, atau *MobileNet* untuk meningkatkan akurasi sekaligus efisiensi komputasi.
2. Menambah variasi data latih, terutama dari berbagai kondisi pencahayaan, sudut pandang, dan jenis tempat parkir yang berbeda, agar model lebih robust terhadap variasi nyata di lapangan.
3. Menggunakan teknik augmentasi data yang lebih beragam untuk memperkuat kemampuan generalisasi model, seperti rotasi, flipping, scaling, dan penyesuaian warna.
4. Melakukan evaluasi menggunakan dataset eksternal atau video real-time untuk menguji performa model dalam situasi sesungguhnya.
5. Menerapkan teknik hyperparameter tuning secara sistematis untuk mendapatkan kombinasi parameter yang optimal dalam pelatihan model.
6. Mengembangkan aplikasi berbasis web atau mobile yang dapat memanfaatkan model ini secara langsung untuk kebutuhan pemantauan atau deteksi otomatis di area parkir.

Dengan mengimplementasikan saran-saran tersebut, diharapkan sistem deteksi tempat parkir berbasis *Faster R-CNN* ini dapat terus ditingkatkan performanya menjadi lebih baik dan dapat memberikan kebermanfaatan secara luas bagi masyarakat dalam pengelolaan area parkir yang lebih efisien dan tertib.

DAFTAR PUSTAKA

- Badan Pusat Statistik Indonesia. (29 Februari 2024). *Perkembangan Jumlah Kendaraan Bermotor Menurut Jenis, 2021-2022*. Diakses pada 6 Mei 2025, dari <https://www.bps.go.id/id/statistics-table/2/NTcjMg==/perkembangan-jumlah-kendaraan-bermotor-menurut-jenis--unit-.html>
- Bohlol, P., Hosseinpour, S., & Soltani Firouz, M. (2025). Improved food recognition using a refined ResNet50 architecture with improved fully connected layers. *Current Research in Food Science*, 10(December 2024). <https://doi.org/10.1016/j.crfs.2025.101005>
- Charli, F., Syaputra, H., Akbar, M., Sauda, S., & Panjaitan, F. (2020). Implementasi Metode Faster Region Convolutional Neural Network (Faster R-CNN) Untuk Pengenalan Jenis Burung Lovebird. *Journal of Information Technology Ampera*, 1(3), 185–197. <https://doi.org/10.51519/journalita.volume1.issue3.year2020.page185-197>
- Chyan, P., Arni, S., & Thayf, M. S. S. (n.d.). *Pengantar Machine Learning PT . MIFANDI MANDIRI DIGITAL*.
- Claudia Rossi, I., Armanto, & Hasanah Bimastari Aviani, T. (2024). Penerapan Metode Deep Learning Pada Sistem Pendeteksi Gambar Notasi Balok. *Kohesi: Jurnal Multidisiplin Saintek*, 2(12), 86–96. <https://ejournal.warunayama.org/kohesi>
- Devi, D. T., Santos, V. A., & Rosyani, P. (2023). Analisa Penggunaan Metode Faster R-CNN dalam Pengenalan Wajah: Systematic Literature Review. *Buletin Ilmiah Ilmu Komputer dan Multimedia (BIKMA)*, 1(1), 258–262. <https://jurnalmahasiswa.com/index.php/biikma>
- Dewi, K., Krisdiyanto, A., Heriyono, F. R., Kriswandatu, A., & Kriswandatu, A. S. (2023). Analysis of Visitor Density Influence on Parking Lot in Kedungmundu Market, Semarang City. *International Journal of Social Service and Research*, 3(2), 543–549. <https://doi.org/10.46799/ijssr.v3i2.266>
- Dwitama, A., Wijayanti, M., Meysawati, M., Fauziah, Fitriyani, Y., & Agustien, D.

- (2025). *Prototype sistem ketersediaan dan kapasitas tempat parkir mobil berbasis arduino uno*. 4(1), 31–39.
- Fang, Q., Jiang, A., Liu, M., & Zhao, S. (2025). Faster R-CNN model for target recognition and diagnosis of scapular fractures. *Journal of Bone Oncology*, 51. <https://doi.org/10.1016/j.jbo.2025.100664>
- Hamzah, R. A., Setianingsih, C., Nugrahaeni, R. A., Hanafia, S. R., & Fuadi, F. (2022). Parking Violation Detection on The Roadside of Toll Roads with Intelligent Transportation System Using Faster R-CNN Algorithm. *Proceedings - International Conference on Informatics and Computational Sciences, 2022-Septe*(3), 169–174. <https://doi.org/10.1109/ICICoS56336.2022.9930590>
- Ilahiyah, S., & Nilogiri, A. (2018). Implementasi Deep Learning Pada Identifikasi Jenis Tumbuhan Berdasarkan Citra Daun Menggunakan Convolutional Neural Network _ Ilahiyah _ JUSTINDO (Jurnal Sistem dan Teknologi Informasi Indonesia). *JUSTINDO(Jurnal Sistem & Teknologi Informasi Indonesia)*, 3(2), 49–56.
- Kong, X., Li, X., Zhu, X., Guo, Z., & Zeng, L. (2024). Detection model based on improved faster-RCNN in apple orchard environment. *Intelligent Systems with Applications*, 21(October 2023), 200325. <https://doi.org/10.1016/j.iswa.2024.200325>
- Maulana, A., Auliatunnajah, F., Rosidin, N., Ramadien Rizki Darmawan, M., & Rosyani, P. (2024). Implementasi OpenCV dengan Metode Image Thresholding pada Gambar. *Jurnal Artificial Inteligent dan Sistem Penunjang Keputusan*, 2(1), 27–32. <https://jurnalmahasiswa.com/index.php/aidanspk>
- Pradana, I. C., Mulyanto, E., & Rachmadi, R. F. (2022). Deteksi Senjata Genggam Menggunakan Faster R-CNN Inception V2. *Jurnal Teknik ITS*, 11(2). <https://doi.org/10.12962/j23373539.v11i2.86587>
- Putra Salawazo, V. M., Jaya Gea, D. P., Gea, R. F., & Azmi, F. (2019). Implementasi Metode Convolutional Neural Network (CNN) Pada Pengenalan Objek Video CCTV. *Indonesia Jalan Sekip Sikaming*, 3(1), 74–79.

- Putri, S. T. E., & Fahrurozi, A. (2022). Pendeteksian Objek Pada Citra Hewan Karnivora Dan Herbivora Menggunakan Faster R-Cnn. *Jurnal Ilmiah Informatika Komputer*, 27(1), 32–42. <https://doi.org/10.35760/ik.2022.v27i1.5858>
- Radiuk, P., Pavlova, O., El Bouhissi, H., Avsiyevych, V., & Kovalenko, V. (2022). Convolutional Neural Network for Parking Slots Detection. *CEUR Workshop Proceedings*, 3156(June), 284–293.
- Rahman, M. F., & Bambang, B. (2021). Deteksi Sampah pada Real-time Video Menggunakan Metode Faster R-CNN. *Applied Technology and Computing Science Journal*, 3(2), 117–125. <https://doi.org/10.33086/atcsj.v3i2.1846>
- Sidik, A. D., & Ansawarman, A. (2022). Prediksi Jumlah Kendaraan Bermotor Menggunakan Machine Learning. *Formosa Journal of Multidisciplinary Research*, 1(3), 559–568. <https://doi.org/10.55927/fjmr.v1i3.745>
- Sutisna, R. E. (2023). Penerapan Algoritma K-Means Clustering Dalam Proses Pengelompokan Kasus Meninggal Dunia Covid-19 Di Indonesia. *JATISI (Jurnal Teknik Informatika dan Sistem Informasi ...)*, 10(1), 30–37. <https://jurnal.mdp.ac.id/index.php/jatisi/article/view/2420> <https://jurnal.mdp.ac.id/index.php/jatisi/article/download/2420/1119>
- Vijiyakumar, K., Govindasamy, V., & Akila, V. (2024). An effective object detection and tracking using automated image annotation with inception based faster R-CNN model. *International Journal of Cognitive Computing in Engineering*, 5(July), 343–356. <https://doi.org/10.1016/j.ijece.2024.07.006>
- Vishal, K. S., Srinidhi, S. K., Shashank, U. S., & Singh, R. P. (2025). ScienceDirect Enhanced Small Object Detection in Remotely Sensed Images using Enhanced Hybrid Small Object Detection in Remotely Sensed Images using and Feature Pyramid Network Hybrid ResNet50 and Feature Pyramid Network. *Procedia Computer Science*, 258(2024), 1813–1822. <https://doi.org/10.1016/j.procs.2025.04.433>
- Wahjuni, S., Wulandari, & Nurarifah, H. (2023). Faster RCNN based leaf segmentation using stereo images. *Journal of Agriculture and Food Research*, 11(January),

100514. <https://doi.org/10.1016/j.jafr.2023.100514>

Wardani, A. S., Setianingsih, C., & Dirgantara, F. M. (2021). Sistem Deteksi Pelanggaran Social Distancing Di Ruang Terbuka Menggunakan Algoritma Faster R-CNN. *e-Proceeding of Engineering*, 8(5), 6670–6677.

Yang, B., Wei, Y., Shi, J., Hong, T., Li, L., & Xu, K. Da. (2025). Diagnosis of array antennas based on near-field data using Faster R-CNN. *Materials and Design*, 254(April). <https://doi.org/10.1016/j.matdes.2025.114033>

Zamachsari, F., & Puspitasari, N. (2021). Penerapan Deep Learning dalam Deteksi Penipuan Transaksi Keuangan Secara Elektronik. *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, 5(2), 203–212. <https://doi.org/10.29207/resti.v5i2.2952>

Zhang, Q., Chang, X., Meng, Z., & Li, Y. (2021). Equipment detection and recognition in electric power room based on faster R-CNN. *Procedia Computer Science*, 183, 324–330. <https://doi.org/10.1016/j.procs.2021.02.066>

