

**PERHITUNGAN INDEKS TANAMAN SAWIT BERBASIS *DEEP LEARNING*
FASTER R-CNN DENGAN VARIASI *BACKBONE* RESNET**

(Skripsi)

Oleh

Rico Prediansyah

NPM 2115031082



**JURUSAN TEKNIK ELEKTRO
FAKULTAS TEKNIK
UNIVERSITAS LAMPUNG
BANDAR LAMPUNG
2025**

**PERHITUNGAN INDEKS TANAMAN SAWIT BERBASIS *DEEP LEARNING*
FASTER R-CNN DENGAN VARIASI *BACKBONE* RESNET**

Oleh

Rico Prediansyah

Skripsi

**Sebagai Salah Satu Syarat untuk Mendapatkan Gelar
SARJANA TEKNIK**

Pada

**Jurusan Teknik Elektro
Fakultas Teknik Universitas Lampung**



**JURUSAN TEKNIK ELEKTRO
FAKULTAS TEKNIK
UNIVERSITAS LAMPUNG
BANDAR LAMPUNG
2025**

ABSTRAK

PERHITUNGAN INDEKS TANAMAN SAWIT BERBASIS *DEEPLARNING* *FASTER R-CNN* DENGAN VARIASI *BACKBONE* RESNET

Oleh

Rico Prediansyah

Perhitungan jumlah pohon kelapa sawit yang akurat sangat penting dalam pengelolaan perkebunan, namun metode konvensional seringkali kurang efisien dan rentan terhadap kesalahan. Penelitian ini mengevaluasi model *Faster R-CNN* dengan berbagai backbone ResNet (ResNet18, ResNet34, ResNet50, ResNet101, dan ResNet152) dalam mendeteksi dan menghitung jumlah pohon sawit dari citra udara UAV, serta menentukan kualitas standar orthophoto yang optimal. Hasil penelitian menunjukkan bahwa model *Faster R-CNN* dengan backbone ResNet50 memiliki performa terbaik dengan akurasi 89.02% dan error paling rendah di semua format gambar. Format JPG menunjukkan performa terbaik pada ResNet152 (89.70%), sementara format PNG dan AVIF terbaik pada ResNet18 (90.8% dan 87.33%), serta format BMP terbaik pada ResNet50 (90.63%). Dalam pengujian terhadap augmentasi gambar, rata-rata akurasi tertinggi untuk format PNG diperoleh pada ResNet50 (85.76%), sedangkan pada format JPG, akurasi tertinggi juga diperoleh pada ResNet50 (85.97%). Pengujian standar orthophoto dengan augmentasi Brightness, Hue, dan Kontras meningkatkan akurasi deteksi, sedangkan augmentasi shear hingga 30 derajat tetap menjaga akurasi sekitar 80%. Pada pengujian cut-off, akurasi tetap stabil dalam kisaran 89-90%. Dengan demikian, model *Faster R-CNN* dengan backbone ResNet50 menjadi pilihan terbaik untuk deteksi dan perhitungan pohon sawit dengan akurasi tinggi pada berbagai kondisi gambar.

Kata Kunci: Sawit, *Faster R-CNN*, ResNet, *Deep Learning*, Deteksi, Hitung

ABSTRACT

PALM OIL INDEX CALCULATION BASED ON DEEP LEARNING FASTER R-CNN WITH RESNET BACKBONE VARIATIONS

By

Rico Prediansyah

Accurate palm oil tree counting is crucial in plantation management; however, conventional methods are often inefficient and prone to errors. This study evaluates the Faster R-CNN model with various ResNet backbones (ResNet18, ResNet34, ResNet50, ResNet101, and ResNet152) for detecting and counting palm trees from UAV aerial imagery and determining the optimal orthophoto quality standard. The results show that the Faster R-CNN model with the ResNet50 backbone performs best, achieving 89.02% accuracy with the lowest error across all image formats. The JPG format shows the best performance with ResNet152 (89.70%), while PNG and AVIF perform best with ResNet18 (90.8% and 87.33%), and BMP performs best with ResNet50 (90.63%). In image augmentation testing, the highest average accuracy for PNG format is obtained with ResNet50 (85.76%), while for JPG format, the highest accuracy is also achieved with ResNet50 (85.97%). Testing the orthophoto standard with Brightness, Hue, and Contrast augmentation enhances detection accuracy, while shear augmentation up to 30 degrees maintains accuracy at around 80%. In cut-off testing, the model's accuracy remains stable within the 89-90% range. Thus, the Faster R-CNN model with the ResNet50 backbone is the best choice for detecting and counting palm trees with high accuracy under various image conditions.

Keywords: *Palm, Faster R-CNN, ResNet, Deep Learning, Detection, Counting*

Judul

**; PERHITUNGAN INDEKS TANAMAN SAWIT
BERBASIS *DEEP LEARNING FASTER R-CNN*
DENGAN VARIASI *BACKBONE RESNET***

Nama Mahasiswa

: Rico Prediansyah

Nomor Pokok Mahasiswa

: 2115031082

Program Studi

: Teknik Elektro

Fakultas

: Teknik

MENYETUJUI

1. Komisi Pembimbing


Ing. Melvi, S.T., M.T., Ph.D.

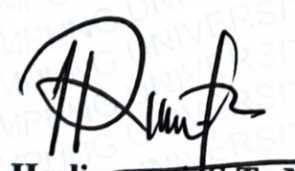
NIP. NIP. 197301182000032001


Aryanto, S.T., M.T.

NIP. 199006212019031011

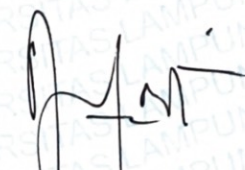
2. Mengetahui

Ketua Jurusan Teknik Elektro


Herlinawati, S.T., M.T.

NIP. 197103141999032001

Ketua Program Studi Teknik Elektro


Sumadi, S.T., M.T.

NIP. 197311042000031001

MENGESAHKAN

1. Tim Penguji

Ketua : Ing. Ardian Ulvan, S.T., M.Sc., Ph.D.


.....

Sekretaris : Aryanto, S.T., M.T.


.....

Penguji : Ing. Melvi, S.T., M.T., Ph.D.


.....



2. Dekan Fakultas Teknik

The official stamp of Universitas Lampung is a circular blue stamp. It contains the text "KEMENTERIAN PENDIDIKAN TINGGI, SAINS DAN TEKNOLOGI" around the top inner edge, "UNIVERSITAS LAMPUNG" around the bottom inner edge, and "FAKULTAS TEKNIK" at the bottom. In the center of the stamp is a blue emblem featuring a torch and two stylized figures. A signature is written over the stamp.

Dr. Eng. Ir. Helmy Fitriawan, S.T., M.Sc. /

NIP. 197509282001121002

Tanggal lulus Ujian Skripsi : 13 februari 2025

SURAT PERNYATAAN

Yang bertanda tangan di bawah ini:

Nama : Rico Prediansyah

NPM : 2115031082

Dengan ini saya menyatakan bahwa dalam skripsi ini tidak terdapat karya yang pernah dilakukan orang lain dan sepanjang sepengetahuan saya tidak terdapat atau diterbitkan oleh orang lain, kecuali secara tertulis diacu dalam naskah ini sebagaimana yang disebutkan dalam daftar Pustaka. Selain itu, saya menyatakan bahwa skripsi ini dibuat oleh saya sendiri.

Apabila pernyataan saya tidak benar, maka saya bersedia dikenai sanksi akademik sesuai dengan hukum yang berlaku.

Bandar Lampung, 19 Januari 2025



Rico Prediansyah

NPM. 2115031082

RIWAYAT HIDUP



Penulis lahir di Bandar Agung pada tanggal 25 Januari 2003 sebagai anak kedua dari dua bersaudara, putra dari Bapak Komar Rustendar dan Ibu Fitria Hartati. Pendidikan dasar diselesaikan di SD 2 Banding Agung pada tahun 2015, dilanjutkan dengan pendidikan menengah pertama di MTsN 2 OKU Selatan yang diselesaikan pada tahun 2018, dan menamatkan pendidikan menengah atas di SMA 2 OKU Selatan pada tahun 2021. Pada tahun yang sama, penulis diterima sebagai mahasiswa Jurusan Teknik Elektro, Fakultas Teknik, Universitas Lampung melalui jalur SBMPTN.

Selama menjadi mahasiswa, penulis aktif dalam berbagai organisasi dan kepanitiaan. Penulis menjabat sebagai Sekretaris Departemen Kominfo di Himpunan Mahasiswa Teknik Elektro (HIMATRO) pada tahun 2022, kemudian dipercaya sebagai Kepala Departemen Kominfo HIMATRO pada tahun 2023. Di tingkat universitas, penulis juga menjabat sebagai Direktur Jenderal Desain Grafis di BEM U KBM Unila pada tahun 2023 dan aktif sebagai Magang Humas Universitas Lampung di tahun yang sama. Pada tahun 2023, penulis berhasil lolos pendanaan dan melaksanakan kegiatan PPK ORMAWA HIMATRO serta PKKMM di Desa Gisting Permai, Tanggamus, Lampung. Pada tahun 2024, penulis menjadi mentor Magang Humas Universitas Lampung dan Asisten Laboratorium Telekomunikasi di Jurusan Teknik Elektro. Selain itu, penulis mengikuti Studi Independen Orbit Future Academy di PT. Orbit Ventura Indonesia pada 16 Februari hingga 30 Juni 2024, dengan proyek akhir berjudul Quick Detection System on Palm Trees (QUTIONS-PT) menggunakan Model YOLOv8, yang berhasil meraih juara 3 sebagai Best Project. Penulis juga mengikuti penelitian Unila Robotika dan Otomasi (URO) dan turut serta dalam Lomba Kontes Robot Terbang Indonesia (KRTI) 2024 yang diselenggarakan oleh Kementerian Pendidikan, Kebudayaan, Riset, dan Teknologi di Yogyakarta.

PERSEMBAHAN

Alhamdulillahirabbil'alamin, ku ucapkan Puji Syukur kepada Allah
Subhanahu Wa Ta'ala atas segala rahmat dan hidayah-Nya serta
sholawat kepada Nabi Muhammad Shallallahu Alaihi Wa Sallam yang
selalu menjadi teladan bagi kehidupanku.

Kupersembahkan dengan tulus Skripsi ini sebagai tanda terimakasih kepada:
Ayah dan Ibuku Tersayang

KOMAR RUSTENDAR dan FITRIA HARTATI

Serta Kakak ku Tersayang

Frengki Pahlevi

Tak lupa kupersembahkan juga kepada seluruh dosen,
keluarga angkatan 2021, dan
civitas Jurusan Teknik Elektro Universitas Lampung.

Seluruh organisasi atau lembaga ilmu yang mendidik,
mendewasakan, dan meningkatkan kemampuan saya

Dan Terimakasih untuk Diri Sendiri, **Rico Prediansyah**

MOTO

"Sebaik-baik manusia diantaramu adalah yang paling banyak manfaatnya bagi orang lain."
(H.R. Bukhari)

"Kalau bukan anak bangsa ini yang membangun bangsanya, siapa lagi? Jangan saudara mengharap orang lain yang datang membangun bangsa kita"
(BJ Habibie)

“Sedikit lebih beda, lebih baik daripada sedikit lebih baik”
(Pandji Pragiwaksono)

*“No one knows What awaits us Around the Corner,
so stop Planning life, And start living it”*
(Hanna Shebar)

SANWANCARA

Segala puji dan syukur penulis panjatkan ke hadirat Allah SWT atas limpahan rahmat, hidayah, dan inayah-Nya, sehingga penulis dapat menyelesaikan laporan skripsi yang berjudul **“PERHITUNGAN INDEKS TANAMAN SAWIT BERBASIS *DEEP LEARNING FASTER R-CNN* DENGAN VARIASI *BACKBONE RESNET*”** tepat pada waktunya. Laporan ini disusun sebagai salah satu syarat untuk meraih gelar Sarjana Teknik pada Jurusan Teknik Elektro, Fakultas Teknik, Universitas Lampung.

Shalawat dan salam senantiasa tercurah kepada Nabi Muhammad SAW, beserta keluarga, sahabat, dan para pengikutnya yang setia berjalan di atas jalan Islam hingga akhir zaman. Dalam proses pengerjaan skripsi ini, penulis banyak menerima bantuan berupa pemikiran dan dukungan moral dari berbagai pihak. Oleh karena itu, pada kesempatan ini, penulis ingin menyampaikan rasa terima kasih kepada:

1. Ibu Prof. Dr.Ir. Lusmeilia Afriani, D.E.A., I.P.M., selaku Rektor Universitas Lampung.
2. Bapak Dr. Eng. Ir. Helmy Fitriawan, S.T., M.Sc., selaku Dekan Fakultas Teknik Universitas Lampung, sekaligus Dosen Pembimbing Pendamping.
3. Ibu Herlinawati, S.T., M.T. selaku Ketua Jurusan Teknik Elektro Fakultas Teknik Universitas Lampung.
4. Bapak Sumadi, S.T., M.T. selaku Ketua Program Studi Teknik Elektro Fakultas Teknik Universitas Lampung
5. Bapak Ing. Ardian Ulvan, S.T., M.Sc. Ph.D. selaku Kepala Laboratorium Telekomunikasi Informasi, sekaligus penguji utama tugas akhir dan telah banyak membantu memberikan kritik dan saran kepada penulis.
6. Ibu Ing. Melvi, S.T., M.T. Ph.D. selaku pembimbing tugas akhir dan telah banyak membantu dan memberi manfaat terutama dalam hal menulis Tugas Akhir.

7. Bapak Aryanto, S.T., M.T. selaku dosen pembimbing kedua, yang telah banyak membimbing dan memberi dukungan kepada penulis.
8. Ibu Dr. Sri Purwiyanti, S.T., M.T., Ph.D. sebagai Dosen Pembimbing Akademik, yang telah banyak membimbing dan membantu penulis selama menjalani kuliah.
9. Seluruh Dosen dan karyawan Jurusan Teknik Elektro Universitas Lampung, berkatilmu yang telah diajarkan kepada penulis selama penulis menjalani masa studi di perkuliahan
10. Ayahandaku tersayang Komar Rustendar, Ibundaku tercinta Fitria Hartati, dan Kakak ku Frengki Pahlevi.
11. Rekan-rekan Teknik Elektro dan Teknik Informatika Universitas Lampung Angkatan 2021 yang telah memberikan dukungan moril kepada penulis.
12. Keluarga Besar HIMATRO yang telah memberikan ilmu dan pengalaman kepada penulis selama pendidikan baik secara langsung maupun tak langsung.
13. Keluarga Besar Magang Humas Unila yang telah memberikan ilmu dan pengalaman kepada penulis selama meningkatkan kemampuan kreatifitas dan inovasi.
14. Keluarga Besar BEM U KBM Unila 2023 yang telah memberikan pengalaman berharga dalam hal berfikir kritis dan kreatif
15. Keluarga Besar Unila Robotika dan Otomasi yang telah saling support dan memberikan pengalaman keteknikan yang luar biasa.
16. Semua pihak yang terlibat dalam menyelesaikan Skripsi ini yang tidak dapat penulis sebutkan satu persatu.

Penulis menyadari bahwa masih banyak kekurangan dalam penulisan skripsi ini.

Penulis mengharapkan adanya kritik dan saran yang membangun dari berbagai pihak demi kemajuan bersama.

Bandar Lampung, 19 Februari 2025

Penulis

Rico Prediansyah

DAFTAR ISI

Halaman

ABSTRAK	ii
ABSTRACT	iii
MENYETUJUI.....	iv
MENGESAHKAN	v
SURAT PERNYATAAN	vi
RIWAYAT HIDUP.....	vii
PERSEMBAHAN.....	viii
MOTO	ix
SANWANCARA	x
DAFTAR ISI	xii
DAFTAR TABEL	xiv
DAFTAR GAMBAR	xvi
I PENDAHULUAN	1
1.1. Latar Belakang	1
1.2. Rumusan Masalah	3
1.3. Tujuan Penelitian.....	3
1.4. Manfaat Penelitian.....	3
1.5. Batasan Masalah.....	4
1.6. Sistematika Penulisan.....	4
II TINJAUAN PUSTAKA	6
2.1. Penelitian Terdahulu	6
2.2. Citra Udara	8
2.3. <i>Deep Learning</i> (DL).....	11
2.4. <i>Computer Vision</i> (CV).....	12
2.5. <i>Convolutional Neural Network</i> (CNN)	13
2.7.1. Covolutional Layer.....	14
2.7.2. <i>Pooling Layer</i>	16
2.7.3. <i>Fully Connected Layer</i>	17

2.7.4. <i>Activation Function</i>	17
2.6. <i>Region-based Convolutional Neural Network (R-CNN)</i>	19
2.7. <i>Fast Region-based Convolutional Neural Network (Fast R-CNN)</i>	20
2.8. <i>Faster Region-based Convolutional Neural Network (Faster R-CNN)</i>	22
2.8.1. <i>Region Proposal Network (RPN)</i>	23
2.8.2. <i>Sharing Features for RPN and Fast R-CNN</i>	25
2.9. <i>Residual Network (ResNet)</i>	26
2.10. <i>Siklus Proyek AI</i>	29
III METODE PENELITIAN	34
3.1. <i>Waktu dan Tempat Penelitian</i>	34
3.2. <i>Alat dan Bahan Penelitian</i>	34
3.3. <i>Pembuatan Algoritma dan Testing</i>	35
3.4. <i>Diagram Alir Penelitian</i>	35
3.4.1. <i>Problem Scoping</i>	37
3.4.2. <i>Data Acquisition</i>	38
3.4.3. <i>Data Exploration</i>	39
3.4.4. <i>Modelling</i>	41
3.4.5. <i>Evaluation</i>	42
IV. HASIL DAN PEMBAHASAN	43
4.1. <i>Data Acquisition</i>	43
4.2. <i>Data Exploration</i>	44
4.2.1. <i>Datatest untuk Pengukuran Evaluasi</i>	44
4.2.2. <i>Datatest Perhitungan Pohon Sawit dengan Perbedaan Augmentasi</i>	47
4.3. <i>Proses Modeling</i>	51
4.3.1. <i>Persiapan Dataset</i>	52
4.3.2. <i>Training</i>	59
4.3.3. <i>Evaluasi dan Testing Gambar</i>	62
4.4. <i>Hasil Evaluasi Kinerja Model dalam berbagai Format Gambar</i>	70
4.5. <i>Perhitungan Pohon Sawit</i>	77
4.5.1. <i>Menentukan Skala Crop untuk Perhitungan</i>	77
4.5.2. <i>Perhitungan Prediksi Berdasarkan Augmentasi Gambar</i>	82
4.5.3. <i>Perhitungan Prediksi Gambar Lahan Sampoerna Agro</i>	90
BAB V PENUTUP	94
DAFTAR PUSTAKA	96
LAMPIRAN	100

DAFTAR TABEL

	Halaman
Tabel 3.1 <i>Problem Scoping</i>	37
Tabel 4.1 Augmentasi gambar uji	48
Tabel 4.2 Evaluasi kerja untuk format gambar JPG	70
Tabel 4.3 Evaluasi kerja untuk format gambar PNG	70
Tabel 4.4 Evaluasi kerja untuk format gambar BMP	71
Tabel 4.5 Evaluasi kerja untuk format gambar AVIF	72
Tabel 4.6 Normalisasi Parameter Evaluasi MSE dan RSME.....	73
Tabel 4.7 Pembobotan parameter evaluasi backbone terbaik	76
Tabel 4.8 Perhitungan <i>orthophoto</i> format PNG dengan ukuran <i>crop</i> skala 1:1	77
Tabel 4.9 Perhitungan <i>orthophoto</i> format PNG dengan ukuran <i>crop</i> skala 4:1	78
Tabel 4.10 Perhitungan <i>orthophoto</i> format JPG dengan ukuran <i>crop</i> skala 1:1	78
Tabel 4.11 Perhitungan <i>orthophoto</i> format JPG dengan ukuran <i>crop</i> skala 4:1	78
Tabel 4.12 Akurasi Perhitungan <i>orthophoto</i> format PNG dengan <i>crop</i> skala 1:1.....	79
Tabel 4.13 Akurasi Perhitungan <i>orthophoto</i> format PNG dengan <i>crop</i> skala 4:1.....	79
Tabel 4.14 Akurasi Perhitungan <i>orthophoto</i> format JPG dengan <i>crop</i> skala 1:1	81
Tabel 4.15 Akurasi Perhitungan <i>orthophoto</i> format JPG dengan <i>crop</i> skala 4:1	81
Tabel 4.16 Perhitungan untuk Augmetasi Level <i>Brightness</i> format PNG	82
Tabel 4.17 Perhitungan untuk Augmetasi Level <i>Hue</i> format PNG	83
Tabel 4.18 Perhitungan untuk Augmetasi Level Kontras format PNG	83
Tabel 4.19 Perhitungan untuk Augmetasi Level Saturasi format PNG.....	83
Tabel 4.20 Perhitungan untuk Augmetasi Level <i>Brightness</i> format JPG.....	84
Tabel 4.21 Perhitungan untuk Augmetasi Level <i>Hue</i> format JPG	84
Tabel 4.22 Perhitungan untuk Augmetasi Level Kontras format JPG	85
Tabel 4.23 Perhitungan untuk Augmetasi Level Saturasi format JPG	85
Tabel 4.24 Perhitungan untuk Augmetasi Sudut <i>Shear</i> format PNG	86

Tabel 4.25 Perhitungan untuk Augmetasi Sudut <i>Shear</i> format JPG	86
Tabel 4.26 Perhitungan untuk Augmetasi <i>Cut-off</i> format PNG	87
Tabel 4.27 Perhitungan untuk Augmetasi <i>Cut-off</i> format JPG.....	87
Tabel 4.28 Rata-rata akurasi augmentasi format gambar PNG.....	88
Tabel 4.29 Rata-rata akurasi augmentasi format gambar JPG	88
Tabel 4.30 Hasil prediksi kebun Sampoerna Agro dengan beberapa variasi <i>crop</i>	91

DAFTAR GAMBAR

	Halaman
Gambar 2.1 Fotogrametri Perkebunan Sawit	10
Gambar 2.2 <i>Mechine Learning</i> vs <i>Deep Learning</i>	11
Gambar 2.3 Alur Kerja CNN	14
Gambar 2.4 Ilustrasi Layer Input	15
Gambar 2.5 <i>Convolutional Layer</i>	16
Gambar 2.6 <i>Max Pooling</i>	17
Gambar 2.7 <i>Fully Connected Layer</i>	18
Gambar 2.8 Alur kerja R-CNN	19
Gambar 2.9 Alur kerja <i>Fast</i> R-CNN	21
Gambar 2.10 Perbandingan R-CNN, <i>Fast</i> R-CNN, dan <i>Faster</i> R-CNN	22
Gambar 2.11 Alur kerja <i>Faster</i> R-CNN	23
Gambar 2.12 RPN	24
Gambar 2.13 Pelatihan Alternatif 4-langkah.....	26
Gambar 2.14 VGG vs ResNet.....	27
Gambar 2.15 ResNet <i>block</i> diagram.	29
Gambar 2.16 Tahapan Siklus Proyek AI	30
Gambar 2.17 <i>Confution Matrix</i>	31
Gambar 3.1 Diagram <i>Flowchart</i> Penelitian	35
Gambar 3.2 Dataset Roboflow	39
Gambar 3.3 Fotogrametri Lahan Sawit Sampoerna Agro Mesuji.....	39
Gambar 3.4 Diagram <i>Flowchart</i> Data <i>Exploration</i>	40
Gambar 3.5 Diagram <i>Flowchart</i> <i>Modelling</i>	41
Gambar 4.1 Rincian dataset untuk data <i>training</i>	43
Gambar 4.2 Proses pemotongan gambar menggunakan <i>tools Split</i> di Photoshop	45
Gambar 4.3 Proses anotasi objek sawit setiap potongan gambar di Roboflow	46
Gambar 4.4 <i>Class</i> pada data di Roboflow.....	46

Gambar 4.5 Rincian data untuk pengujian Parameter Evaluasi.....	47
Gambar 4.6 Pengecekan <i>runtime</i> GPU T4.....	51
Gambar 4.7 Penjelasan GPU T4 yang di gunakan.....	52
Gambar 4.8 <i>Import</i> dataset dari Roboflow.....	53
Gambar 4.9 Menejemen <i>file</i> dan <i>folder</i> dataset	54
Gambar 4.10 Proses konversi format <i>file</i> gambar	55
Gambar 4.11 Melakukan <i>Import libraries</i> untuk <i>modelling</i>	56
Gambar 4.12 Visualisasi gambar <i>bounding box</i>	57
Gambar 4.13 Penyederhanaan dan konversi dataset	58
Gambar 4.14 Memuat data <i>training</i> dan <i>testing</i> model.....	59
Gambar 4.15 <i>Load</i> dan konfigurasi model.....	59
Gambar 4.16 Persiapan model pada device yang digunakan.....	60
Gambar 4.17 Meyiapkan <i>optimizer</i> dan memulai pelatihan model	61
Gambar 4.18 Menyimpan dan memuat bobot model.....	62
Gambar 4.19 Prediksi model deteksi objek <i>datatest</i>	63
Gambar 4.20 Prediksi satu sample gambar jumlah pohon sawit	64
Gambar 4.21 Proses pengumpulan prediksi pada data uji	65
Gambar 4.22 <i>Import</i> dan mengatur <i>library Utils</i>	66
Gambar 4.23 Parameter Evaluasi <i>Precision</i> , <i>recall</i> , <i>f1-score</i> , dan <i>mAP/AP</i>	66
Gambar 4.24 Parameter Evaluasi MSE dan RSME	67
Gambar 4.25 <i>Testing</i> dengan <i>Crop</i> lalu dideteksi dan dihitung	68
Gambar 4.26 Grafik Parameter Evaluasi <i>Backbone</i> dengan Format JPG.....	74
Gambar 4.27 Grafik Parameter Evaluasi <i>Backbone</i> dengan Format PNG.....	74
Gambar 4.28 Grafik Parameter Evaluasi <i>Backbone</i> dengan Format BMP	74
Gambar 4.29 Grafik Parameter Evaluasi <i>Backbone</i> dengan Format AVIF	75
Gambar 4.30 <i>Orthophoto</i> Area Kebun Sampoerna Agro, Mesuji, Lampung.	90
Gambar 4.31 Sampel crop 31 x 31 PNG hasil prediksi lahan Sampoerna Agro.....	92
Gambar 4.32 Sampel crop 31 x 31 JPG hasil prediksi lahan Sampoerna Agro	92

I PENDAHULUAN

1.1. Latar Belakang

Sektor perkebunan dan industri pengolahan kelapa sawit di Indonesia terus berkembang pesat dan memberikan kontribusi besar bagi perekonomian nasional [1, 2]. Namun, pengelolaan lahan sawit yang luas menghadirkan tantangan, khususnya dalam menghitung jumlah pohon secara akurat [3]. Selama ini, banyak perkebunan masih mengandalkan metode manual atau tradisional yang membutuhkan waktu lama dan cenderung kurang akurat. Kesalahan dalam perhitungan dapat menghambat pengambilan keputusan yang efektif, sehingga pengelola perkebunan memerlukan pendekatan baru untuk mengatasi masalah ini [4].

Digitalisasi melalui teknologi modern, seperti penggunaan UAV (*Unmanned Aerial Vehicles*) atau drone, menjadi solusi efektif. Drone dapat mengambil gambar lahan dari udara dengan kamera beresolusi tinggi yang terpasang padanya. Data yang diperoleh tidak hanya berupa gambar, tetapi juga hasil pemetaan seperti *orthophoto* dan fotogrametri, yang memberikan gambaran detail mengenai topografi dan distribusi pohon sawit [5]. Selain meningkatkan akurasi dan efisiensi, hasil pemetaan drone seperti *orthophoto* dan fotogrametri memberikan banyak manfaat tambahan. Data visual yang dihasilkan dapat diproses lebih lanjut menggunakan kecerdasan buatan berbasis *deep learning* untuk menghitung jumlah pohon sawit secara otomatis. Teknologi ini memungkinkan sistem untuk mendeteksi dan menghitung pohon sawit dengan akurasi tinggi, bahkan di area yang padat atau sulit dijangkau. Selain itu, sistem berbasis *deep learning* juga dapat menganalisis data untuk memberikan informasi tambahan, seperti identifikasi pola pertumbuhan dan kondisi kesehatan tanaman.

Dibandingkan metode konvensional yang mengandalkan pencatatan manual di lapangan, teknologi digital dalam perhitungan pohon (taksasi) menawarkan efisiensi yang jauh lebih tinggi. Sistem berbasis AI mampu menganalisis ribuan gambar udara dalam waktu singkat, menghasilkan laporan yang lebih cepat dan akurat dibandingkan dengan penghitungan manual yang memakan waktu berhari-hari. Teknologi ini juga mengurangi potensi kesalahan manusia serta biaya operasional yang dikeluarkan untuk survei lapangan. Dengan adanya solusi digital ini, perkebunan kelapa sawit dapat mengoptimalkan perencanaan produksi dan distribusi sumber daya dengan lebih baik, sehingga meningkatkan produktivitas serta keberlanjutan industri.

Penelitian ini akan menggunakan algoritma *Faster R-CNN (Region-based Convolutional Neural Network)* dengan menguji beberapa versi *backbone* ResNet dalam peningkatan akurasi deteksi dan perhitungan jumlah pohon kelapa sawit dari citra udara perkebunan PT Sampoerna Agro di Mesuji. Algoritma *Faster R-CNN* dikenal sebagai metode deteksi objek yang sangat baik di bandingkan metode *Convolutional* klasik sebelumnya dalam mengidentifikasi objek sekaligus dalam satu gambar [6]. Sedangkan ResNet merupakan model *deep learning* tingkat tinggi namun tetap optimal dan kompleksitas yang rendah sehingga menjadikan juara pertama deteksi *ImageNet* pada kompetisi ILSVRC & COCO 2015. Alur kerja ResNet, sebagai *backbone* dari *Faster R-CNN*, memungkinkan model untuk mengenali pola dan detail kecil yang mungkin terdapat dalam citra udara, seperti pohon yang berdekatan atau tumpang tindih. Kombinasi ini diharapkan dapat memberikan perhitungan yang lebih akurat untuk pengambilan keputusan di Perkebunan [7].

Model deteksi berbasis kecerdasan buatan memungkinkan pemantauan perkebunan kelapa sawit secara efisien dan hemat waktu, serta dapat mengurangi biaya dan meningkatkan efisiensi produksi dengan menyediakan data akurat untuk manajemen. Selain itu, data jumlah pohon dari citra udara sangat berguna untuk merencanakan jadwal panen dan memenuhi kebutuhan bahan baku industri pengolahan kelapa sawit. Dengan data yang lebih akurat dan terkini, pengelola perkebunan dapat membuat keputusan strategis yang lebih baik, seperti mengoptimalkan distribusi tenaga kerja, meningkatkan produktivitas lahan, serta merancang strategi replanting yang lebih efektif guna menjaga keberlanjutan industri kelapa sawit.

1.2. Rumusan Masalah

Adapun rumusan masalah dalam penelitian ini adalah sebagai berikut:

1. Bagaimana efektivitas dan evaluasi model *Faster* R-CNN dalam mendeteksi dan menghitung jumlah tanaman sawit dengan berbagai *backbone* ResNet, seperti ResNet18, ResNet34, ResNet50, ResNet101, dan ResNet152?
2. Jenis *backbone* ResNet manakah yang memberikan hasil terbaik dalam mendeteksi dan menghitung jumlah tanaman sawit menggunakan model *Faster* R-CNN?
3. Bagaimana kualitas standar *orthophoto* kebun sawit yang optimal agar dapat terdeteksi dan dihitung secara akurat menggunakan model *Faster* R-CNN dengan *backbone* ResNet?

1.3. Tujuan Penelitian

Adapun rumusan masalah dalam penelitian ini adalah sebagai berikut:

1. Melakukan pengujian parameter evaluasi model model *Faster* R-CNN dengan menguji berbagai variasi *backbone* ResNet untuk mendeteksi tanaman sawit.
2. Melakukan deteksi dan perhitungan tanaman sawit menggunakan model *Faster* R-CNN dengan menguji berbagai variasi versi *backbone* ResNet seperti ResNet18, ResNet34, ResNet50, ResNet101, dan ResNet152;
3. Menentukan jenis *backbone* ResNet yang paling baik pada model *Faster* R-CNN dalam mendeteksi dan menghitung jumlah tanaman sawit.
4. Menentukan kualitas standar *orthophoto* kebun sawit untuk mendeteksi dan menghitung menggunakan model *Faster* R-CNN dengan *backbone* ResNet;

1.4. Manfaat Penelitian

Adapun manfaat dalam penelitian ini adalah sebagai berikut:

1. Penelitian ini menggunakan teknologi modern sehingga menghasilkan perhitungan yang lebih cepat dan efisien dibandingkan dengan metode konvensional;

2. Penelitian ini membantu menentukan jenis *backbone* yang paling efektif dalam mendeteksi dan menghitung jumlah tanaman sawit, yang dapat diimplementasikan untuk optimasi pengelolaan perkebunan.
3. Penelitian ini akan memberikan standar kualitas *orthophoto* kebun sawit yang dapat diolah dan terdeteksi secara akurat oleh model *Faster R-CNN* dengan *backbone* ResNet sehingga meningkatkan efisiensi dalam pemetaan Perkebunan;

1.5. Batasan Masalah

Adapun manfaat dalam penelitian ini adalah sebagai berikut:

1. Deteksi dan perhitungan tanaman terbatas pada dataset *orthophoto* yang ada.
2. Penelitian ini hanya menggunakan model *Faster R-CNN* dengan *backbone* ResNet (ResNet18, ResNet34, ResNet50, ResNet101, dan ResNet152) untuk deteksi dan perhitungan jumlah tanaman sawit.
3. Citra pengujian *orthophoto* yang digunakan dalam penelitian ini hanya berasal dari kebun sawit PT Sampoerna Agro di Mesuji dan tidak mencakup variasi kondisi lingkungan lain.

1.6. Sistematika Penulisan

Adapun sistematika penulisan yang digunakan pada penelitian ini untuk memberikan suatu gambaran sederhana serta untuk memudahkan pemahaman pada penelitian ini yang dituliskan menjadi beberapa bab, sebagai berikut:

BAB I PENDAHULUAN

Bab ini menjelaskan tentang latar belakang, tujuan penelitian, rumusan masalah, batasan masalah, manfaat penelitian, hipotesis dan sistematika penulisan pada penelitian ini.

BAB II TINJAUAN PUSTAKA

Bab ini menjelaskan mengenai teori pendukung yang menjadi pengantar pemahaman dan berkaitan dengan materi penelitian yang diambil dari berbagai sumber ilmiah seperti buku dan jurnal.

BAB III METODOLOGI PENELITIAN

Bab ini memaparkan metodologi penelitian antara lain waktu dan tempat pengerjaan, alat dan bahan yang digunakan dalam pengerjaan penelitian, serta metode dan diagram penelitian yang akan digunakan dalam pengerjaan penelitian tugas akhir.

BAB IV HASIL DAN PEMBAHASAN

Bab ini berisikan tentang hasil dan analisa data yang didapatkan dari algoritma dan sistem yang dikembangkan pada penelitian ini.

BAB V PENUTUP

Bab ini berisikan kesimpulan yang didapat dari hasil analisa dan pembahasan juga berisikan saran yang membangun bagi semua pihak yang terlibat dalam pelaksanaan tugas akhir.

DAFTAR PUSTAKA

Bab ini berisikan referensi dari penulisan dan pelaksanaan proyek

II TINJAUAN PUSTAKA

2.1. Penelitian Terdahulu

Penelitian ini dilakukan oleh Juepeng Zheng, Weijia Li, Maocai Xia, Runmin Dong, Haohuan Fu, dan Shuai Yuan [6] yang berfokus pada deteksi pohon kelapa sawit menggunakan citra satelit resolusi tinggi [8] dan pesawat tanpa awak. Penelitian ini didasarkan pada deteksi pohon *deep learning* untuk deteksi tajuk pohon di Malaysia dengan metode CNN dengan presisi 96% dan selanjutnya Li dkk. [9] mengusulkan *one-stage* CNN dan *two-stage* CNN (TS-CNN) dengan evaluasi IOU F1-score 92.8%. Penelitian Juepeng Zheng dkk. [6] data diperoleh dari daerah Malaysia Selatan tahun 2006 dengan total 14 wilayah (6 wilayah *train*, 2 wilayah *validation*, dan 6 untuk pengujian). Penelitian ini mengadopsi *backbone* ZF, VGG16, dan VGG19. VGG16 memperoleh hasil paling baik diantara *backbone* lainnya. Model ini mengalami kesulitan mendeteksi kelapa sawit karena ukuran pohon yang kecil dalam citra dan banyaknya pohon serupa di sekitar kelapa sawit. Untuk mengatasi hal ini, mereka mengoptimalkan Jaringan Proposal Wilayah (*Region Proposal Network*) dan menambahkan strategi pasca-pemrosesan. Langkah ini berhasil mengurangi kesalahan deteksi dan menghasilkan rata-rata *precision* 95,05%, *recall* 94,68%, serta skor F1 sebesar 95%, yang lebih tinggi dibandingkan metode deteksi lainnya [6]. Penelitian ini membuktikan bahwa metode yang digunakan memperoleh evaluasi yang lebih baik untuk wilayah *data test* dan *data uji* yang sama. Namun metode ini tidak menguji Nilai *loss* pada model, tidak bagaimana bagaimana standar augmentasi *orthophoto* yang digunakan dan tidak berfokus untuk melakukan perhitungan dalam mendeteksi objek pohon sawit.

Penelitian oleh Adel Ammar, Anis Koubaa, dan Bilel Benjdira [10] mengembangkan sistem berbasis *deep learning* untuk menghitung dan menentukan lokasi pohon kelapa

sawit serta pohon umum secara otomatis dari citra udara menggunakan CNN. Penelitian didasarkan juga pada penelitian pertama mengenai deteksi pohon sawit yang dilakukan oleh Li dkk ditahun 2016 sampai 2029 [9] dan penelitian Juepeng Zheng dkk. [6] dengan wilayah yang sama. Penelitian adel dkk. [10] ini mempertimbangkan dataset citra udara yang lebih tinggi yaitu 2cm/px yang diperoleh dari perkebunan di Arab Saudi. Penelitian ini mengungkapkan geolokasi citra pohon sawit dengan pemberian tag geografis. Metode yang digunakan pada adel dkk. membandingkan empat model yang lebih canggih dalam mendeteksi objek, yaitu YOLOv3, YOLOv4, EfficientDet-D5, dan Faster R-CNN. Dataset diperoleh sebesar 279 gambar dan datatest 70 gambar dengan *class* “*palm trees*” dan “*other trees*”. Dengan variasi IoU nilai mAP (90%) terbaik adalah model EfficientDet-D5 pada IoU 0.5. dan *Faster R-CNN* dengan IoU yang sama hampir mencapai 70%. Namun *Faster RCNN* sangat baik untuk *class palm*, dan sangat buruk untuk *calss “other trees”*. Lebih jelasnya terbaik performa terbaik untuk *class “palm trees”* *Faster R-CNN* dengan IoU 0.6, penelitian ini mendapatkan *precision* sebesar 93%, *recall* 87%, *F1 score* 90% dan mAP 88% [10]. Penelitian ini hanya berfokus pada variasi IoU pohon sawit dengan pohon lain namun tidak untuk perhitungan pohon sawit, augmentasi, atau variasi bentuk gambar. Penelitian ini baik dalam mendeteksi lokasi objek pohon, namun berfokus pada metode *classification* pohon sawit terhadap pohon lain sehingga tidak menggaransi untuk medeteksi objek sawit dengan skala yang kecil.

Penelitian ini dilakukan oleh Pinaki Nath Chowdhury, Palaiahnakote Shivakumara, Lokesh Nandanwar, Faizal Samiron, Umapada Pal, dan Tong Lu [11] yang mengembangkan metode untuk menghitung jumlah pohon kelapa sawit dan kelapa pada citra drone dengan pendekatan berbasis deep learning. Penelitian ini sama-sama merujuk pada Juepeng Zheng dkk [6]. yang berfokus pada deteksi kualitas tinggi namun tidak untuk pohon-pohon yang lebat. Penelitian Pinaki dkk [11] berfokus pada perhitungan gambar dengan objek pohon yang lebat. Metode yang digunakan yaitu menggabungkan metode *Generalized Gradient Vector Flow* (GGVF) untuk mendeteksi titik dominan berdasarkan simetri arah cabang pohon dan YOLOv5 untuk menghilangkan area deteksi yang salah. Hasilnya menunjukkan metode ini mampu mengatasi tantangan latar belakang yang kompleks dan variasi sudut dalam citra udara, dengan *precision* rata-rata 69,83%, *recall* 98,06%, dan *F1 score* 81,57%, yang membuktikan efektivitas metode ini pada kondisi gambar dengan pohon yang padat atau campuran [11]. Penelitian ini juga berfokus pada metode terbaik untuk mengklasifikasikan antara pohon sawit dan pohon

kelapa dengan melihat arah cabang dengan menggaransi dalam hal kepadatan objek gambar. Namun tidak menggaransi model tersebut dapat mengklasifikasikan dan menghitung dengan baik pohon tersebut jika ukuran objek yang kecil.

Secara singkat, penelitian sebelumnya menunjukkan bahwa sebagian besar model belum benar-benar melakukan perhitungan pengujian gambar *orthophoto*, melainkan hanya berfokus pada deteksi objek. Selain itu, penelitian tersebut belum menguji berbagai variasi standar augmentasi warna dan pencahayaan yang dapat meningkatkan efisiensi dalam proses perhitungan. Tak ada juga penelitian yang menguji pengaruh variasi format gambar dalam mendeteksi dan menghitung objek pohon sawit, terutama gambar yang diambil secara miring (*shear*) atau gambar *orthophoto* yang memiliki banyak bolong (*cut-off*). Penelitian-penelitian tersebut juga belum menguji perhitungan pohon sawit di area yang sangat luas dengan kepadatan pohon yang tinggi.

Penelitian ini bertujuan untuk mengatasi kekurangan tersebut dengan menggunakan metode *cropping* cepat untuk menghitung area lahan yang sangat luas, bahkan hingga ratusan hektar. Model yang diteliti diharapkan dapat menentukan kualitas standar gambar uji dengan performa terbaik, dengan berbagai variasi *backbone* dari ResNet dan model *Faster R-CNN*. Selain berfokus pada standar gambar dan perhitungan di area luas, penelitian ini juga mengevaluasi gambar dari data uji dan dataset yang berbeda, sehingga lebih efisien untuk diterapkan pada studi kasus di lahan pohon sawit di wilayah lain. Evaluasi dilakukan dengan menambahkan pengujian parameter *error* seperti MSE dan RMSE, serta standar evaluasi lainnya seperti *precision*, *recall*, *F1 score*, dan *mean Average Precision* (mAP).

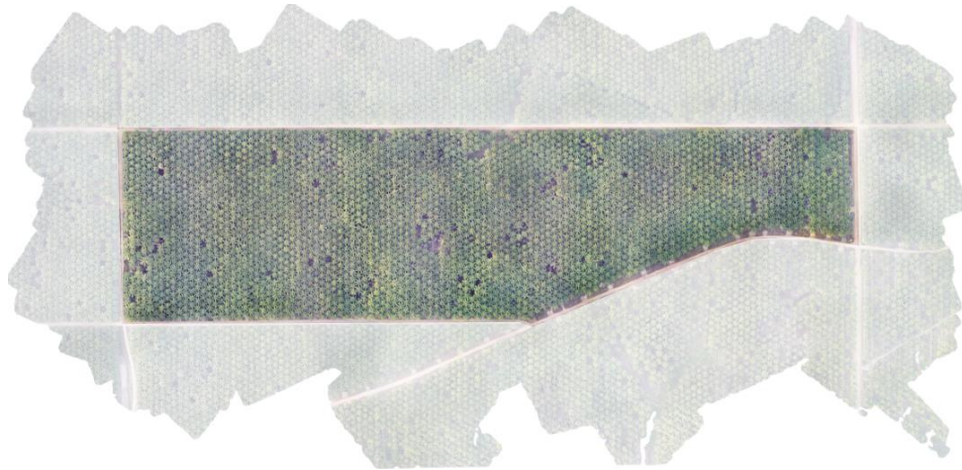
2.2. Citra Udara

Fotogrametri adalah seni, ilmu, dan teknologi untuk mendapatkan informasi yang akurat tentang objek serta lingkungan fisik melalui proses perekaman, pengukuran, dan interpretasi citra fotografi. Dalam proses ini, aspek-aspek geometris pada foto udara, seperti sudut, jarak, koordinat, dan lainnya, menjadi faktor utama. Pesatnya perkembangan teknologi di bidang survei dan pemetaan juga didukung oleh penggunaan Wahana Unmanned Aerial Vehicle (UAV) [12]. Pemetaan menggunakan foto udara biasanya mampu menghasilkan peta dengan tingkat akurasi yang cukup tinggi. Foto

udara ini memberikan sumber data berupa *orthophoto* yang telah melewati tahap pengecekan ketelitian. *Orthophoto* tersebut memuat berbagai objek yang terekam, seperti bangunan, vegetasi, maupun elemen non-bangunan lainnya. Pemetaan secara fotogrametri tidak dapat lepas dari referensi pengukuran secara terestris, mulai dari penetapan ground controls (titik dasar kontrol) hingga kepada pengukuran batas tanah. Batas-batas tanah yang diidentifikasi pada peta foto harus diukur di lapangan. Fotogrametri dapat mencakup dua bidang yaitu fotogrametri metrik dan fotogrametri interpretative [13].

1. Fotogrametri metrik adalah proses pengukuran menggunakan wahana pemotretan udara yang melibatkan berbagai perhitungan untuk menentukan dimensi dan bentuk suatu objek. Tujuan utama dari fotogrametri metrik adalah memperoleh data kuantitatif, seperti jarak, sudut, luas, dan posisi objek. Proses ini membutuhkan peralatan khusus serta pengetahuan dan keahlian tertentu untuk memahami hubungan matematis antara sistem foto udara dan sistem di permukaan tanah, sehingga ukuran pada foto dapat dikonversi ke sistem koordinat tanah atau sebaliknya.
2. Fotogrametri interpretatif adalah aktivitas yang bertujuan untuk mengenali dan mengidentifikasi objek tertentu. Metode ini lebih berfokus pada pengumpulan data kualitatif melalui pengamatan, identifikasi, dan interpretasi foto udara.

Orthophoto hasil pemetaan ini dilakukan menggunakan *software* Agisoft Metashape, yang dapat *render* citra dari foto udara menjadi model tiga dimensi yang lebih mudah dipahami. Pada tahap ini, lahan-lahan yang telah dipetakan dipisahkan berdasarkan hasil hitungan manual untuk dijadikan data uji model. Pemisahan ini sangat penting untuk memastikan bahwa data yang digunakan dalam analisis adalah representatif dan akurat, sehingga dapat memberikan gambaran yang jelas mengenai kondisi lahan. Terlihat gambar 2.1 adalah *orthophoto* atau Fotogrametri yang sudah diolah menggunakan *software* Agisoft.

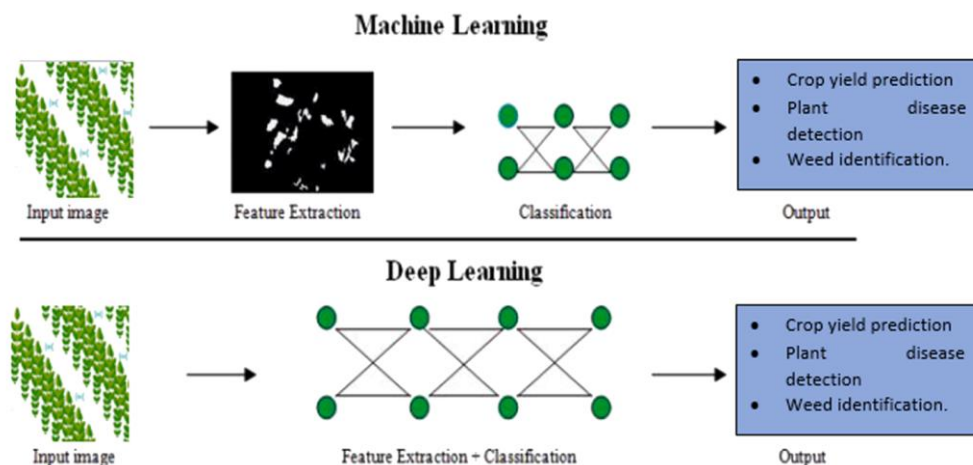


Gambar 2. 1 Fotogrametri Perkebunan Sawit [13]

Gambar 2.1 diatas merupakan citra udara menggunakan teknik fotogrametri kebun sawit PT Sampoerna Agro pada blok 44, Agisoft memungkinkan pengguna untuk mendapatkan informasi lebih detail mengenai kondisi lahan tersebut, termasuk pengenalan dan pengukuran objek seperti pohon sawit dengan akurasi tinggi. Data yang diperoleh dapat dimanfaatkan untuk berbagai analisis bentuk citra gambar, citra yang dihasilkan dapat disimpan dalam berbagai format seperti JPG/JPEG, PNG, AVIF, dan BMP, masing-masing dengan keunggulan dalam ukuran *file*, kualitas, dan kompresi. Format JPG/JPEG umumnya digunakan karena efisiensinya dalam kompresi, sedangkan PNG cocok untuk menjaga kualitas gambar tanpa kehilangan data. Format AVIF dan BMP, meskipun lebih jarang digunakan dalam pemetaan, memiliki kemampuan untuk menyimpan citra dalam kualitas tinggi, yang dapat diperlukan untuk analisis yang lebih mendetail [14]. Untuk meningkatkan akurasi model deteksi, augmentasi gambar seperti perubahan *brightness*, hue, saturasi, kontras, *shear*, dan *cut-off* diterapkan. *Brightness* (Kecerahan) mengacu pada tingkat terang atau gelapnya suatu gambar secara keseluruhan. *Hue* (Warna dasar) adalah atribut warna yang menentukan spektrum gelombang warna cahaya. *Saturasi* (Kejenuhan) menggambarkan intensitas atau kecerahan warna (tampak hidup) [15]. *Kontras* adalah perbedaan antara area terang dan gelap dalam gambar; kontras tinggi menciptakan perbedaan yang tajam. *Shear* adalah transformasi geometris yang memiringkan gambar dengan sudut tertentu, tanpa mengubah luasnya. Teknik augmentasi *cut-off* membantu memperbaiki gambar yang terpotong atau bolong setelah proses *render*, sehingga meningkatkan kemampuan model dalam mengenali objek pada citra dengan lebih baik.

2.3. Deep Learning (DL)

Deep learning merupakan cabang dari *Machine learning* bagian dari *Artificial Intelligence* yang memanfaatkan jaringan saraf tiruan atau *Artificial Neural Networks* (ANN) untuk menganalisis dan mempelajari pola dari data. Beberapa alur kerja *deep learning* yang telah diimplementasikan, seperti *Convolutional Neural Network* (CNN), *Deep Neural Network* (DNN), *Recurrent Neural Networks* (RNN), *Deep Belief Network* (DBN), dan *Fully Convolutional Network* (FCN), menunjukkan performa unggul dalam berbagai bidang, termasuk di sektor pertanian, pengenalan wajah, dan pemrosesan bahasa alami [16]. Pada sektor pertanian, kemampuan *deep learning* dalam klasifikasi dan deteksi objek telah menggantikan metode konvensional seperti *Machine learning* klasik yang memerlukan ekstraksi fitur data secara manual. Hal ini membuat *deep learning* lebih efisien untuk aplikasi berbasis citra, termasuk identifikasi tanaman, prediksi hasil panen, deteksi penyakit tanaman, dan identifikasi gulma, yang mendukung manajemen perkebunan dan peningkatan produktivitas [17]. Gambar 2.2 mengilustrasikan konsep *Machine learning* dan *deep learning* untuk pertanian. Setelah model dianggap akurat dan andal, model tersebut dapat diterapkan untuk melakukan prediksi atau mendeteksi anomali secara *real-time* [18].



Gambar 2. 2 *Machine Learning* vs *Deep Learning* [18]

Gambar 2.2 diatas membandingkan dua teknik kecerdasan buatan, yaitu *Machine Learning* (ML) dan *Deep Learning* (DL), yang keduanya digunakan untuk menganalisis gambar tanaman. ML memerlukan ekstraksi fitur secara manual sebelum klasifikasi, sedangkan DL dapat melakukannya secara otomatis melalui jaringan saraf tiruan yang dalam. DL umumnya lebih akurat, terutama untuk data yang kompleks, namun

membutuhkan lebih banyak data dan komputasi. Dalam perkembangan terkini, *deep learning* berperan penting dalam klasifikasi dan deteksi objek pada citra udara untuk berbagai aplikasi di bidang agrikultur, termasuk perhitungan jumlah vegetasi dan klasifikasi jenis tanaman [19]. Teknologi ini memiliki kemampuan untuk menangkap fitur visual secara otomatis tanpa intervensi manual selama tahap pelatihan, sehingga meningkatkan akurasi dalam pengenalan citra. Pada penelitian ini, algoritma *deep learning* yang digunakan adalah *Faster R-CNN* dengan ResNet sebagai *backbone*, sebuah alur kerja yang dirancang untuk mendeteksi dan menghitung objek dengan tingkat ketelitian yang tinggi. *Faster R-CNN*, yang memiliki kecepatan dan ketepatan tinggi dalam deteksi objek, sangat sesuai untuk penerapan deteksi jumlah pohon sawit dari citra udara.

Penelitian ini menggunakan citra udara perkebunan sawit milik PT Sampoerna Agro di Mesuji, Lampung. Setelah citra diperoleh, data tersebut digunakan untuk prediksi data hasil dan tambahan dataset pohon sawit yang sudah ada, kemudian dilakukan eksplorasi data seperti augmentasi di *roboflow* untuk meningkatkan variasi dan kualitas data latih. Proses *deep learning* dalam penelitian ini melibatkan metode *Faster R-CNN*, yang merupakan model deteksi objek dengan akurasi tinggi, terutama pada objek kompleks seperti pohon sawit dalam perkebunan padat. Model ini menggunakan *backbone* ResNet yang mampu mengekstraksi fitur-fitur penting dari citra secara otomatis, yang kemudian digunakan dalam proses klasifikasi dan deteksi objek secara tepat. Tahapan ini memungkinkan *deep learning* mengenali pohon sawit secara konsisten dalam berbagai kondisi citra udara, mendukung analisis lebih lanjut dan aplikasi manajemen perkebunan yang lebih efisien.

2.4. Computer Vision (CV)

Computer Vision (CV) merupakan cabang domain dari *Artificial Intelligence* yang dirancang untuk memungkinkan komputer melakukan analisis citra dan video guna memperoleh informasi serta pemahaman tentang objek dengan cara yang menyerupai atau bahkan melebihi kemampuan penglihatan manusia [20]. CV memungkinkan mesin untuk memahami rangsangan visual dalam berbagai tingkatan, mulai dari tugas sederhana seperti identifikasi tepi pada gambar hingga tugas yang lebih kompleks

seperti pemahaman adegan lengkap untuk deteksi objek dalam video real-time. Beberapa aspek penting dalam CV meliputi:

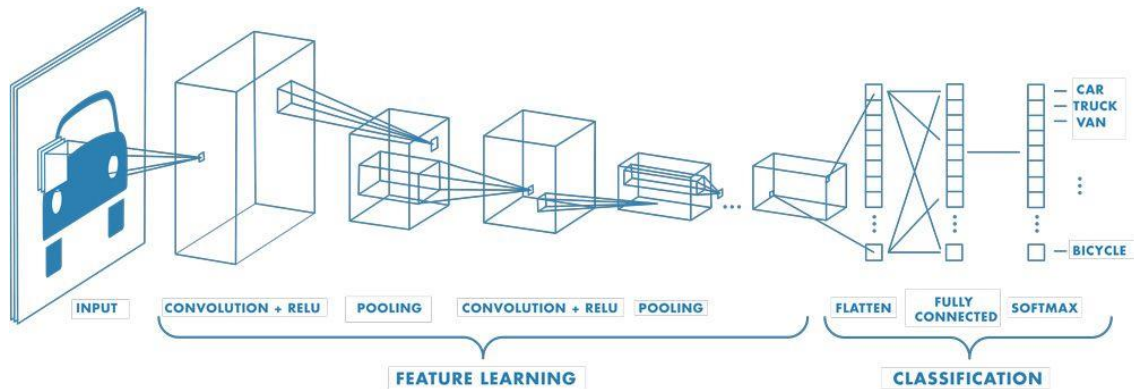
- **Pengenalan Pola:** Mengidentifikasi objek atau pola dalam gambar atau video, seperti mengenal pohon dan daun.
- **Segmentasi:** Memisahkan gambar menjadi bagian-bagian yang berbeda berdasarkan karakteristik tertentu, seperti warna, tekstur, atau intensitas gambar pohon dan daun sawit tersebut.
- **Deteksi Objek:** Menemukan dan mengidentifikasi objek pohon yang cacat, berjamur, kering, berubah, atau tidak berubah.
- **Pelacakan Objek:** Mengikuti pergerakan objek pohon sawit dari satu frame ke frame berikutnya dalam video atau kamera langsung.

Kemajuan dalam CV selama dekade terakhir dipengaruhi oleh tiga faktor utama: pertama, pematangan *deep learning* (DL), yang merupakan jenis pembelajaran mesin yang memungkinkan pembelajaran *end-to-end* dari fungsi yang kompleks menggunakan data mentah; kedua, kemajuan dalam komputasi lokal daya melalui pengembangan GPU yang semakin canggih; dan ketiga, ketersediaan sumber terbuka dari kumpulan data berlabel yang dapat digunakan untuk melatih algoritma-algoritma ini. Kombinasi dari ketiga elemen tersebut memberikan akses kepada peneliti, baik individu maupun kelompok, terhadap sumber daya yang diperlukan untuk memajukan bidang ini [21].

2.5. *Convolutional Neural Network* (CNN)

Convolutional Neural Network (CNN) adalah salah satu algoritma *deep learning* yang paling umum digunakan. Diperkenalkan pertama kali pada tahun 1960-an, CNN telah membuktikan kinerjanya yang menjanjikan di bidang *Computer Vision* dan diakui sebagai jaringan saraf yang paling mewakili *deep learning* [22]. CNN diterapkan untuk menangani tugas visual kompleks yang membutuhkan komputasi intensif dan digunakan secara luas dalam klasifikasi gambar, segmentasi, deteksi objek, pemrosesan video, pemrosesan bahasa alami, serta pengenalan suara [23,24]. Contoh implementasi CNN dapat ditemukan dalam analisis video yang dilakukan oleh Shri [25] dan analisis gambar oleh Roncancio [26]. Artikel ini berkontribusi dengan memberikan penjelasan ringkas namun menyeluruh tentang CNN, di mana setiap elemen pembentukannya disajikan

sebagai perspektif lain dalam metode AI. CNN memiliki empat lapisan utama: lapisan konvolusi, lapisan *Pooling*, lapisan *Fully Connected*, dan lapisan non-linearitas [27]. Ilustrasi dari keempat lapisan ini ditampilkan pada Gambar 2.3.

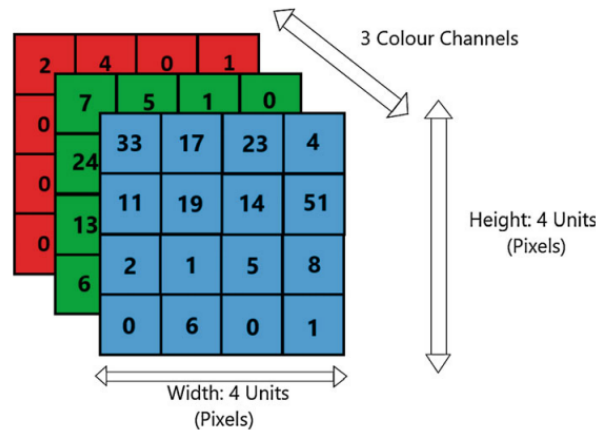


Gambar 2. 3 Alur Kerja CNN [28]

Gambar 2.3 di atas merupakan Alur Kerja CNN, prosesnya dimulai dari input gambar (misalnya, gambar mobil), kemudian melalui beberapa lapisan tersembunyi (hidden layers) yang terdiri dari konvolusi, *relu*, dan *pooling*. Lapisan-lapisan ini berfungsi untuk mengekstrak fitur-fitur penting dari gambar, seperti tepi, bentuk, dan tekstur. Setelah itu, fitur-fitur tersebut diratakan (*flatten*) dan dimasukkan ke dalam lapisan fully connected untuk melakukan klasifikasi akhir. Lapisan output (*softmax*) akan memberikan probabilitas untuk setiap kelas yang mungkin (misalnya, mobil, truk, van, sepeda). Dengan kata lain, CNN belajar untuk mengenali pola dalam gambar dan mengklasifikasikannya berdasarkan pola tersebut. Penjelasan lebih lanjut mengenai fungsi setiap lapisan akan dijelaskan dalam sub-bab berikut.

2.7.1. Covolutional Layer

Layer input pada *Convolutional Neural Network* (CNN) adalah lapisan pertama yang menerima data dan meneruskannya ke lapisan berikutnya. Lapisan ini menetapkan dimensi data, biasanya dalam bentuk *Tensor* 3D untuk gambar, seperti (224, 224, 3) yang mewakili tinggi 224 *pixel*, lebar 224 *pixel*, dan tiga saluran warna RGB (*Red*, *Green*, *Blue*). Fungsi utamanya adalah menyiapkan data tanpa komputasi kompleks untuk pemrosesan di lapisan-lapisan selanjutnya. Ilustrasi *Layer input* dapat dilihat pada gambar 2.4.

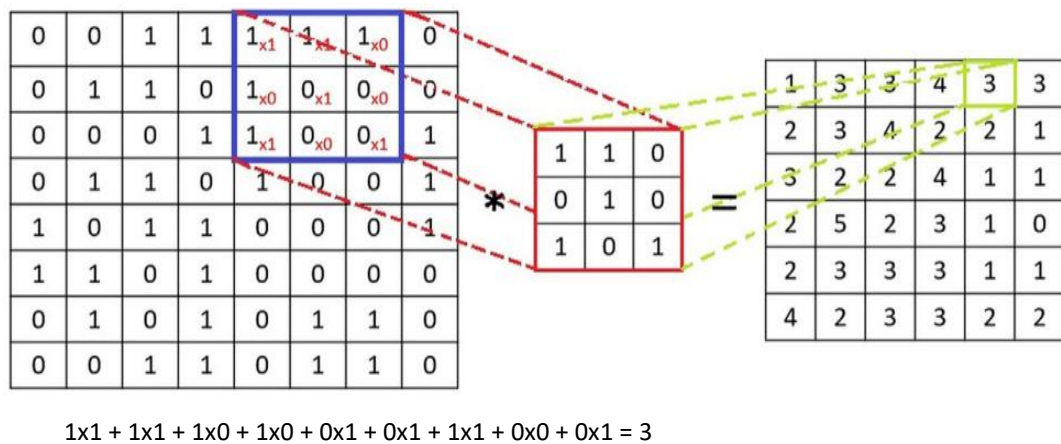


Gambar 2. 4 Ilustrasi *Layer Input* [29]

Gambar di atas menunjukkan ilustrasi untuk 3 channel dengan tinggi 4 pixel dan lebar 4 pixel, tentunya untuk lebar dan tinggi menyesuaikan gambar yang dimasukkan. Selanjutnya lapisan konvolusi pada CNN menggunakan filter *kernel* untuk melakukan konvolusi pada gambar *input*, mengekstraksi fitur dasar gambar. Filter ini memiliki dimensi sama namun parameter konstan yang lebih kecil dari gambar *input* [20]. Sebagai contoh, filter *kernel* 2D dengan ukuran $3 \times 3 \times 3$ memiliki panjang yang sesuai sebesar $f \times f \times 2$, di mana f bisa bernilai 3, 5, atau 7, dan seterusnya. Filter ini bergerak melintasi gambar *input*, menghitung hasil kali antara bobot filter dan nilai piksel, yang menghasilkan peta aktivasi 2D. CNN kemudian mempelajari fitur visual dari gambar tersebut.

Pada gambar 2.4 di atas, matriks 3×3 (filter atau kernel) digeserkan pada matriks input (gambar) secara bertahap. Pada setiap posisi, elemen-elemen yang sesuai pada filter dan matriks input dikalikan dan kemudian dijumlahkan untuk menghasilkan satu nilai pada matriks output. Proses ini berulang untuk setiap posisi yang mungkin dari filter pada matriks input, menghasilkan sebuah matriks fitur yang menangkap informasi lokal dari gambar input. Lapisan konvolusi ditentukan oleh tiga parameter utama: ukuran *kernel*, panjang langkah (*stride*), dan *padding* [30]. Ukuran *kernel* merujuk pada dimensi filter *kernel* yang digunakan untuk pemrosesan [31]. Panjang langkah menunjukkan seberapa jauh *kernel* bergeser sebelum menghitung titik perkalian untuk menghasilkan piksel *output* [32]. Sementara itu, *padding* mengacu

pada penambahan bingkai nol di sekitar *map input* [33]. Proses komputasi ini digambarkan pada Gambar 2.5

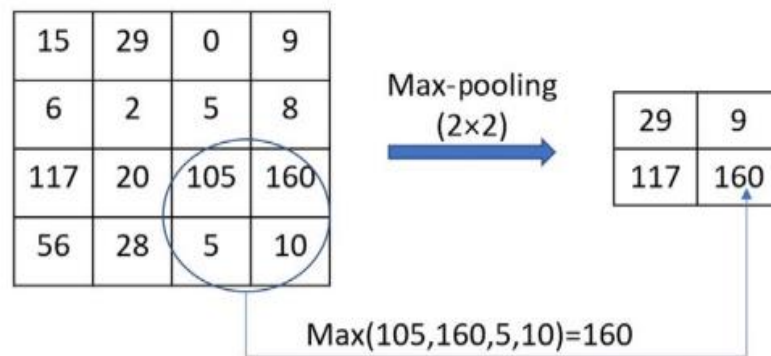


Gambar 2. 5 *Convolutional Layer* [34]

Pada gambar 2.5 diatas, gambar input tidak ditambahkan *padding*. Matriks 3x3 (*filter* atau *kernel*) digeserkan (*stride*) pada matriks input secara bertahap sebanyak 1 pixel. Pada setiap posisi, elemen-elemen yang sesuai pada *filter* dan matriks input dikalikan dan kemudian dijumlahkan untuk menghasilkan satu nilai pada matriks output, sebagai contoh di peroleh nilai 3. Proses ini berulang untuk setiap posisi yang mungkin dari filter pada matriks input, menghasilkan sebuah matriks fitur yang menangkap informasi lokal dari gambar input.

2.7.2. *Pooling Layer*

Lapisan *Pooling* menghubungkan dua lapisan konvolusi berturut-turut dan mengurangi jumlah parameter serta beban komputasi dengan membuat representasi down-sampling. Fungsi dalam lapisan *Pooling* dapat menghasilkan nilai maksimum atau rata-rata, dengan kombinasi maksimum yang sering digunakan untuk hasil optimal [35]. *Pooling* juga membantu mengurangi *overfitting* atau bobot komputasi. Gambar 2.6 menunjukkan operasi sederhana untuk mengurangi dimensi dari peta aktivasi menggunakan fungsi max-*Pooling*.



Gambar 2. 6 Max Pooling [34]

Pada contoh gambar di atas, *feature map* awal yang memiliki 16 piksel diubah menjadi *feature map* baru yang hanya memiliki 4 piksel melalui proses max-pooling. Meskipun dimensi berkurang, informasi penting seperti lokasi fitur dengan nilai tertinggi tetap dipertahankan. Region yang dipilih adalah 2x2 dan nilai maksimum dari region tersebut (yaitu 160) akan menjadi representasi dari region tersebut pada *feature map* yang baru.

2.7.3. Fully Connected Layer

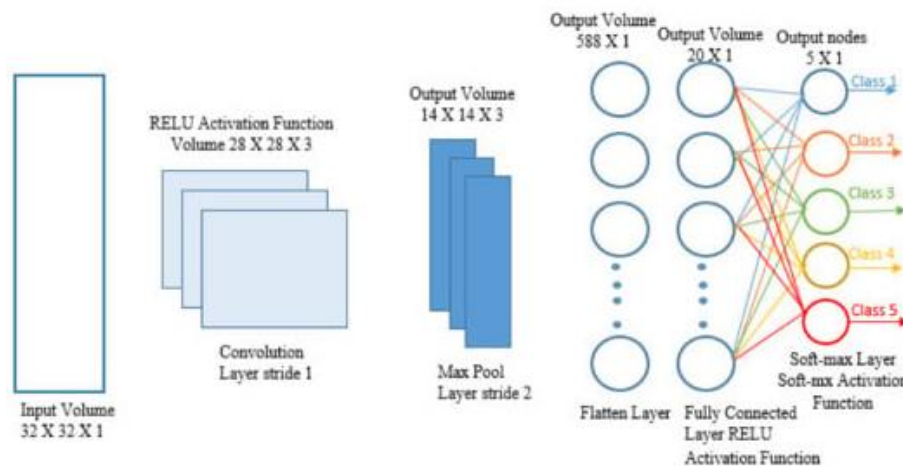
Lapisan ketiga adalah *Fully Connected Layer*, juga dikenal sebagai lapisan *output* konvolusi [26]. Lapisan ini mirip dengan jaringan saraf *feedforward* dan biasanya terletak di bagian bawah jaringan. *Layer Fully Connected* yang diterapkan mencakup *Layer Flatten* dan *Layer Dense*. *Layer Flatten* bertugas mengonversi *output* dari lapisan sebelumnya, yakni peta fitur dari lapisan konvolusi atau *Pooling*, menjadi vektor dengan satu dimensi. *Layer Dense* merupakan lapisan *Fully Connected* yang terdiri dari sejumlah neuron yang sepenuhnya terhubung dengan setiap neuron pada lapisan sebelumnya.

2.7.4. Activation Function

Activation adalah fungsi aktivasi yang diterapkan, *Output* yang telah difilter memberikan fungsi matematis lain yang disebut aktivasi [29]. Pada *Layer output*, fungsi aktivasi yang digunakan adalah *softmax*. *Layer output* menghasilkan hasil

akhir dari model, seperti prediksi kelas dalam tugas klasifikasi, sebagaimana terlihat pada Gambar 2.7 yang menggambarkan *Layer Fully Connected* dan *Layer Output* berwarna-warni. *Layer output* biasanya berupa *Layer* dense dengan fungsi aktivasi yang disesuaikan dengan jenis *output* yang diharapkan, seperti *softmax* untuk klasifikasi multikelas.

Setelah model selesai dibuat, tahap berikutnya adalah *compile* model dengan mengatur optimizer, *loss function*, dan *metric*. *Optimizer* adalah algoritma yang mengubah bobot model selama pelatihan untuk meminimalkan nilai *loss*. Dalam penelitian ini, digunakan optimizer jenis SGD (*Stochastic Gradient Descent*), yang memperbarui bobot model berdasarkan sebagian kecil data pada setiap iterasi, membantu konvergensi yang lebih stabil dan mengurangi kemungkinan terjebak di titik minimum lokal. Gambar 2.7 berikut merupakan mengilustrasikan *input layer* hingga detail *Fully Connected Layer*.



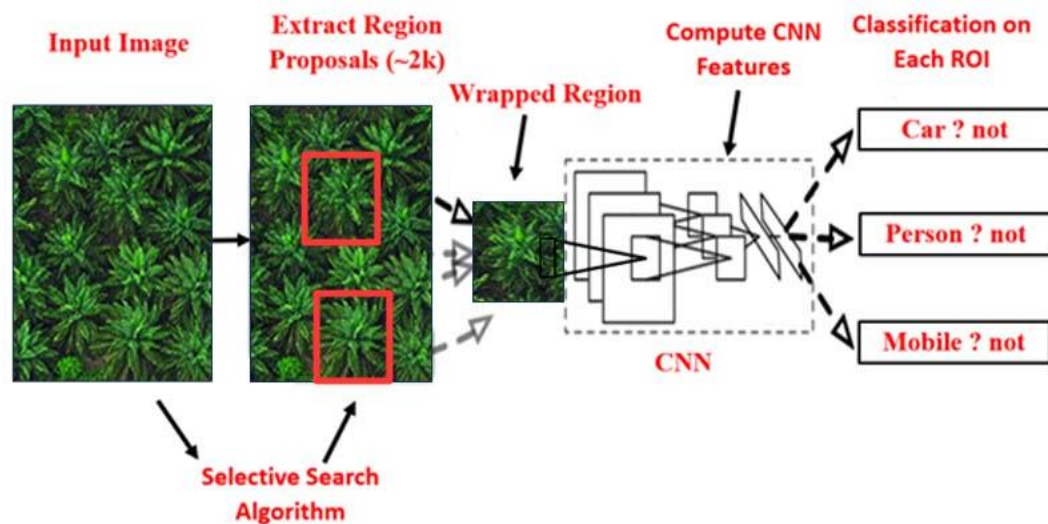
Gambar 2. 7 *Fully Connected Layer* [36]

Gambar diatas menunjukkan alur kerja *Fully Connected Layer* CNN. Input berupa gambar *grayscale* berukuran $32 \times 32 \times 1$ melewati *Convolution Layer* dengan *stride* 1, menghasilkan fitur berukuran $28 \times 28 \times 3$ setelah aktivasi ReLU. Kemudian, *Max Pooling Layer* dengan *stride* 2 mereduksi dimensi menjadi $14 \times 14 \times 3$. Data ini diratakan di *Flatten Layer* menjadi vektor 1D (588×1), yang diproses oleh *Fully Connected Layer* dengan aktivasi ReLU untuk mendeteksi pola lebih kompleks. Akhirnya, *Softmax Layer* menghasilkan probabilitas untuk 5 kelas, memungkinkan klasifikasi berdasarkan nilai probabilitas tertinggi.

2.6. Region-based Convolutional Neural Network (R-CNN)

R-CNN merupakan algoritma deteksi yang sederhana dan skalabel yang memiliki pendekatan untuk menggabungkan dua metode:

1. Penggunaan jaringan saraf konvolusional (CNN) berkapasitas tinggi pada *Region Proposal* yang dihasilkan secara bottom-up untuk melokalisasi dan melakukan segmentasi objek.
2. Ketika data pelatihan berlabel terbatas, pelatihan awal yang diawasi untuk tugas tambahan, diikuti dengan *fine-tuning* untuk domain tertentu, memberikan peningkatan kinerja yang signifikan.



Gambar 2. 8 Alur kerja R-CNN [37]

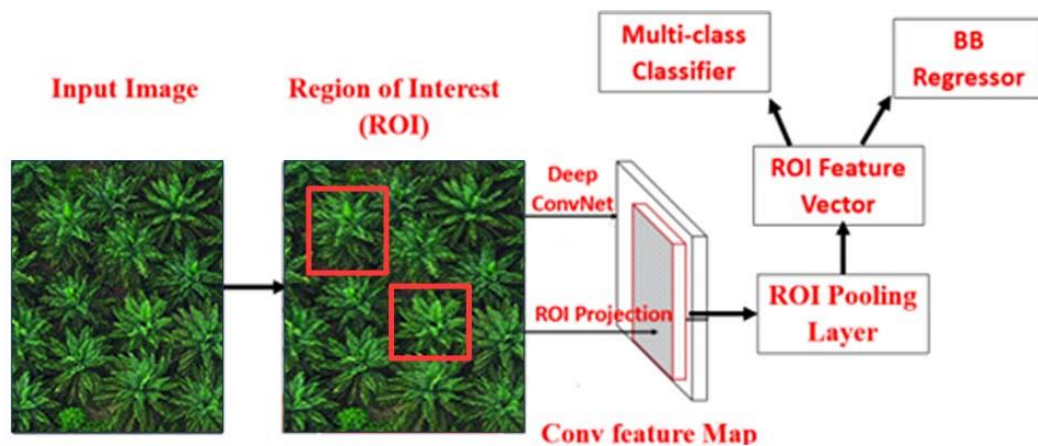
Gambar 2.8 menunjukkan alur kerja sistem ini bekerja dengan cara menerima gambar sebagai *input*, selanjutnya ekstraksi *Regional Proposal*. *Regional Proposal* merupakan teknik deteksi objek yang digunakan untuk mengidentifikasi area dalam gambar yang berpotensi mengandung objek yang ingin dideteksi. Berbagai makalah menawarkan metode untuk menghasilkan *Region Proposal*. contohnya meliputi: *objectness*, *Selective Search*, *category-independent object proposals*, *constrained parametric min-cuts (CPMC)*, *multi-scale combinatorial grouping*, dan Ciresan et al. R-CNN menggunakan *Selective Search* sebagai *Region Proposal* eksternal untuk memungkinkan perbandingan yang terkendali dengan penelitian deteksi sebelumnya. Tahapan selanjutnya proses ekstraksi fitur dilakukan dengan mengambil vektor fitur berdimensi 4096 dari setiap

Region Proposal. Fitur *Region Proposal* dihitung melalui propagasi maju pada gambar RGB berukuran 227×227 yang sudah dikurangi nilai rata-ratanya (ekstraksi), melalui lima *Layer* konvolusi dan dua *Fully Connected Layers*. Untuk menyesuaikan data gambar dari *Region Proposal* yang ke format CNN [37].

Pengujian dilakukan dengan menjalankan *Selective Search* pada gambar uji untuk mengekstrak sekitar 2000 *Region Proposal*, lalu me-warp setiap proposal dan melakukan *forward propagation* melalui CNN untuk mendapatkan fitur dari lapisan yang diinginkan. Kemudian, untuk setiap kelas *Region* tadi, fitur diekstraksi menggunakan SVM yang dilatih untuk kelas tersebut. Setelah semua *Region* diberi skor dalam sebuah gambar, menerapkan *non-maximum suppression* yang bekerja secara *greedy* di setiap kelas secara independent dan menolak sebuah *Region* jika memiliki *intersection-over-union* (IoU). IoU adalah *overlap* dengan *Region* terpilih yang memiliki skor lebih tinggi dari ambang batas yang telah dipelajari. Saat proses *training* menggunakan CNN selanjutnya dilakukan *Fine-tuning Spesifik Domain* untuk menyesuaikan CNN dengan tugas baru (deteksi) dan domain baru. Analisis ini menunjukkan bahwa R-CNN dapat diskalakan hingga ribuan kelas objek tanpa menggunakan teknik perkiraan, seperti *hashing*. Bahkan jika terdapat 100 ribu kelas, perkalian matriks yang dihasilkan hanya memerlukan 10 detik pada CPU *multi-core modern* [37].

2.7. Fast Region-based Convolutional Neural Network (Fast R-CNN)

Fast R-CNN menerapkan beberapa inovasi untuk meningkatkan kecepatan pelatihan dan pengujian sekaligus meningkatkan akurasi deteksi. *Fast R-CNN* melatih jaringan VGG16 yang sangat dalam, 9 kali lebih cepat dari pada R-CNN, 213 kali lebih cepat saat pengujian, dan mencapai *map* yang lebih tinggi pada PASCAL VOC 2012. [37]



Gambar 2. 9 Alur kerja *Fast R-CNN* [37]

Gambar 2.9 menggambarkan alur kerja *Fast R-CNN*. Jaringan *Fast R-CNN* menerima *inputan* berupa gambar utuh dan serangkaian proposal objek. Jaringan ini pertama-tama memproses seluruh gambar melalui beberapa lapisan konvolusi (conv) dan max *Pooling* untuk menghasilkan conv fitur *map*. Kemudian, untuk setiap proposal objek, lapisan *RoI Pooling* mengekstrak vektor fitur dengan panjang tetap dari peta fitur. Lapisan *RoI Pooling* menggunakan max *Pooling* untuk mengubah fitur di dalam setiap daerah minat (*RoI*) yang valid menjadi peta fitur kecil dengan ukuran spasial tetap $H \times W$ (misalnya, 7×7), di mana H dan W adalah *hiperparameter* lapisan yang tidak tergantung pada *RoI* tertentu. *RoI* adalah jendela persegi panjang ke dalam peta fitur *Convolutional*. Setiap *RoI* didefinisikan oleh empat *tuple* (r, c, h, w) yang menentukan sudut kiri atasnya (r, c) serta tinggi dan lebar (h, w).

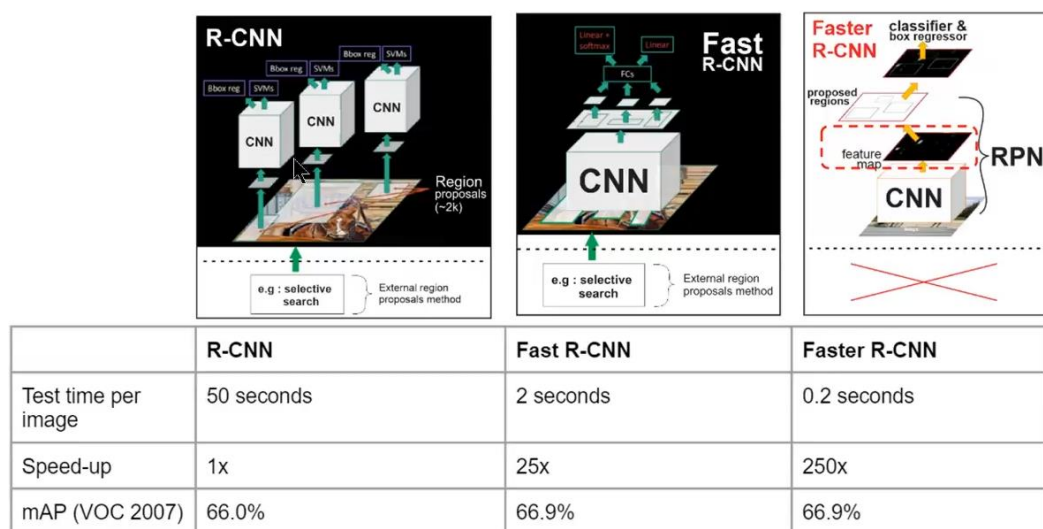
Setiap vektor fitur dimasukkan ke dalam urutan lapisan *Fully Connected* (fc) yang akhirnya bercabang menjadi dua lapisan *output* bersaudara: satu yang menghasilkan estimasi probabilitas *softmax* untuk K kelas objek ditambah satu kelas "latar belakang", dan lapisan lain yang mengeluarkan empat angka bernilai riil untuk setiap kelas K . Setiap set 4 nilai ini menyandikan posisi *bounding-box* yang diperbaiki untuk salah satu kelas K [37].

Faster R-CNN mengusulkan metode pelatihan yang lebih efisien yang memanfaatkan pembagian fitur selama pelatihan. Dalam *Fast R-CNN*, mini-batch stochastic *Gradient descent* (SGD) diambil secara hierarkis, dimulai dengan N gambar dan diikuti oleh pengambilan R/N *RoI* dari setiap gambar. *RoI* dari gambar yang sama berbagi komputasi

dan memori dalam *forward* dan *backward passes*, sehingga membuat N kecil dapat mengurangi perhitungan mini-batch sehingga mempercepat proses *training*. Selain itu, *Fast R-CNN* menerapkan proses pelatihan yang lebih ramping dengan satu *fine-tuning* yang secara bersamaan mengoptimalkan pengklasifikasi *softmax* dan *regresor bounding-box*, berbeda halnya dengan pelatihan terpisah untuk pengklasifikasi *softmax*, SVM, dan regresor dalam tiga tahap yang berbeda [38].

2.8. *Faster Region-based Convolutional Neural Network (Faster R-CNN)*

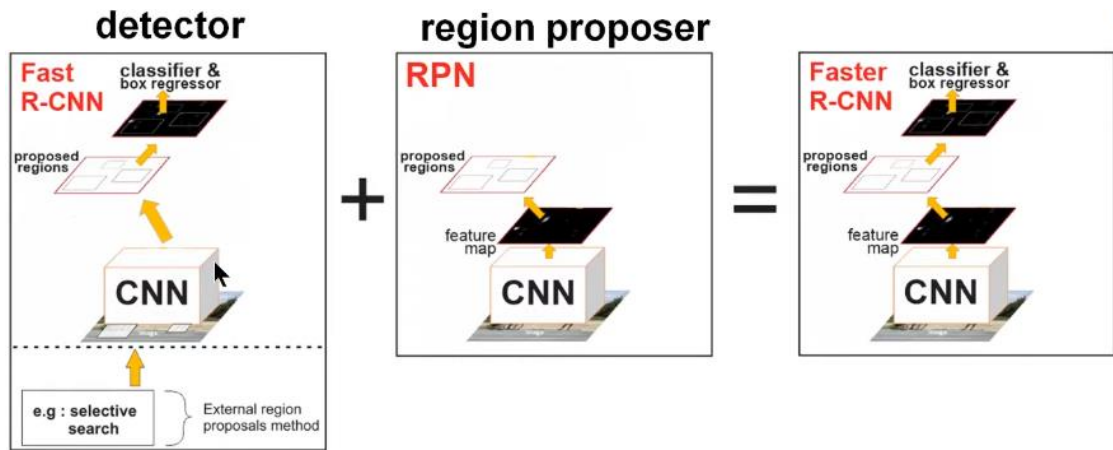
Faster R-CNN adalah metode deteksi objek yang telah terbukti efektif dalam berbagai aplikasi. Metode ini memanfaatkan konsep *Region Proposal Network* (RPN) untuk menghasilkan proposal *Region* secara otomatis dari gambar *input*. RPN bekerja dengan memanfaatkan fitur konvolusi yang diekstraksi dari *Convolutional Neural Network* (CNN) yang dalam. Dengan cara ini, *Faster R-CNN* tidak hanya mengurangi waktu yang dibutuhkan untuk menghasilkan proposal *Region*, tetapi juga meningkatkan akurasi deteksi objek. Perhatikan perbandingan 3 model pada gambar 2.10 di bawah.



Gambar 2. 10 Perbandingan R-CNN, *Fast R-CNN*, dan *Faster R-CNN* [39]

Gambar 2.10 diatas merupakan Perbandingan *R-CNN*, *Fast R-CNN*, dan *Faster R-CNN*, *Fast R-CNN* awalnya menggunakan metode *Selective Search* untuk menghasilkan proposal *Region*, yang membutuhkan waktu sekitar 2 detik per gambar. Sebaliknya, *Faster R-CNN* berhasil mengurangi waktu tersebut menjadi 0.2 detik per gambar

dengan mengintegrasikan RPN ke dalam proses deteksi, sehingga menunjukkan bahwa metode ini dapat mendeteksi objek dengan lebih efisien dibandingkan dengan R-CNN sebelumnya [38]. Lebih detail mengenai alur kerja faster R-CNN dapat di lihat pada gambar 2.11 di bawah ini.



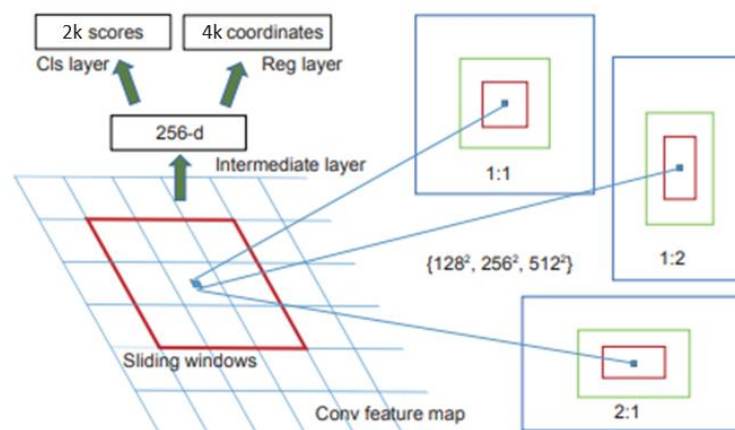
Gambar 2. 11 Alur kerja *Faster* R-CNN [39]

Sistem ini yang di sebut *Faster* R-CNN, pada gambar 2.11 terdiri dari 2 modul, RPN sebagai *deep* jaringan konvolusi yang akan di generate untuk menghasilkan *Region Proposal*. Modul kedua selanjutnya adalah fungsi detector yang akan di lakukan oleh *Fast* R-CNN. Namun dari hasil penggabungan tersebut masih banyak perbaikan yang di lakukan model *Faster* R-CNN ini, contohnya RPN menghasilkan proposal *Region* dengan menggunakan kotak "*anchor*" yang berfungsi sebagai referensi untuk berbagai skala dan aspek rasio. Proses ini memungkinkan model untuk mendeteksi objek dengan ukuran yang bervariasi dalam satu gambar. RPN dilatih secara bersamaan dengan jaringan deteksi objek, sehingga kedua model dapat saling mendukung dalam meningkatkan kinerja deteksi. Dengan pelatihan yang terintegrasi ini, *Faster* R-CNN dapat mencapai tingkat akurasi yang tinggi pada dataset yang kompleks seperti PASCAL VOC dan COCO, di mana model ini menunjukkan keunggulan signifikan dibandingkan metode konvensional lainnya [37].

2.8.1. *Region Proposal Network (RPN)*

Regional Proposal merupakan teknik deteksi objek yang digunakan untuk mengidentifikasi area dalam gambar yang berpotensi mengandung objek yang ingin

dideteksi berdasarkan dataset yang memiliki *ground truth box*. Gambar setelah di-*input*-kan di tebak dengan menggunakan “*anchor*” apakah memiliki objek atau tidak, dengan cara menghitung nilai IoU. Selanjutnya pada bagian ini dilakukan proses CNN misal menggunakan VGG-19 dengan cara gambar di *sliding windows* menggunakan ukuran skala yang bermacam-macam, misal 3x3 aspek rasio seperti terlihat pada gambar 2.12. Pada kasus ini untuk gambar berukuran 1000×600 akan terjadi banyak kemungkinan *anchor* setelah melakukan *sliding windows* sekita 2400 *anchor* (40×60). Meskipun memungkinkan untuk mengoptimalkan fungsi *loss* untuk semua *anchor*, hal ini akan cenderung memberikan bias terhadap sampel negatif karena mendominasi. Sebagai gantinya, secara acak mengambil sampel 256 *anchor* dalam sebuah gambar untuk menghitung fungsi kerugian dari mini-batch, dengan rasio *anchor* positif dan negatif yang diambil 1:1 rasio, 128 sampel *positive* dan *negative* [40].



Gambar 2. 12 RPN [40]

Gambar 2.12 diatas menunjukkan model ini menggaransi berbagai tipe skala *anchor* k yaitu sekitar 9 *anchor*. Maka, akan ada sekitar 20.000 ($\approx 60 \times 40 \times 9$) *anchor* secara keseluruhan. *Anchor* di ubah menjadi 6000 setelah *anchor* di luar objek di buang. Teknik selanjutnya menggunakan *Non-Max Suppression* (NMS) dengan memaksimalkan IoU sehingga menghasilkan 2000 *anchor* class *Layer* dan koordinat ada x dan y menjadi 4000 koordinat. *Feature map* terdiri dari enam bagian, yaitu penentuan objek dan non-objek dengan nilai 0–1, serta koordinat nilai x dan y, dan dimensi *weight* dan *height* dari *bounding box*. *Sliding window* diletakkan pada setiap *Feature map* dengan ukuran $N \times N$, sesuai dengan setiap *anchor Sliding window* yang telah dibuat. *Anchor* memiliki titik pusat yang sama, tetapi dengan rasio aspek dan faktor skala yang berbeda [40].

Penelitian ini menggunakan FPN, FPN tidak menghasilkan *Region Proposal*, tetapi digunakan untuk meningkatkan representasi fitur yang masuk ke RPN dan detektor. Dengan FPN, *Faster R-CNN* dapat memperoleh informasi dari berbagai skala dalam gambar, yang membuat model lebih sensitif terhadap objek dengan ukuran yang berbeda. FPN membangun piramida fitur dari jaringan *backbone* dengan menggunakan lapisan konvolusi pada berbagai tingkat. FPN mengambil keluaran fitur dari beberapa lapisan konvolusi *backbone* (misalnya, ResNet) pada resolusi berbeda dan menggabungkannya menjadi sebuah piramida yang mencakup fitur dari skala kecil hingga besar. FPN menunjukkan peningkatan yang signifikan dalam mendeteksi objek kecil, karena CNN membentuk piramida fitur melalui propagasi maju. Implementasi FPN + *Faster R-CNN* menghasilkan hasil deteksi yang lebih baik pada dataset MS-COCO tanpa menggunakan fitur tambahan [37].

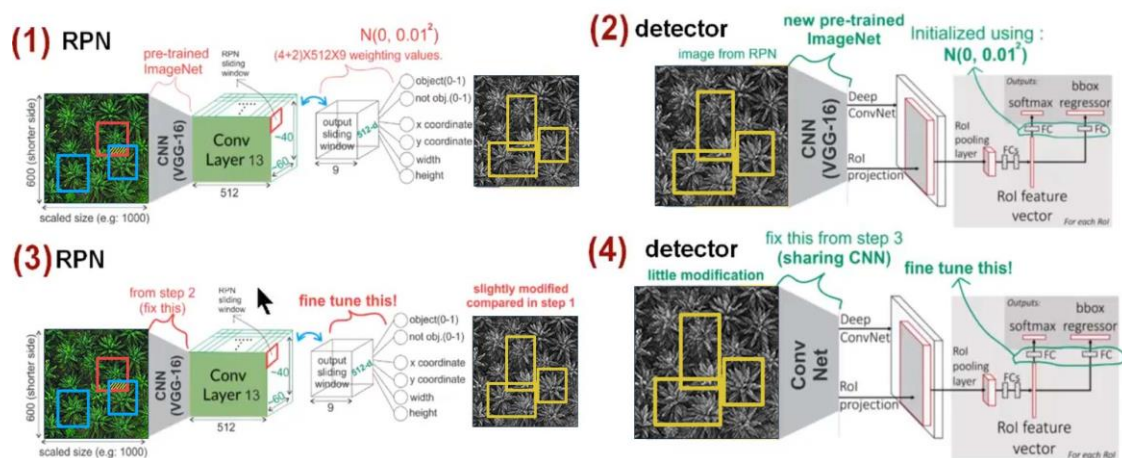
2.8.2. Sharing Features for RPN and Fast R-CNN

Penelitian ini mengembangkan metode pelatihan jaringan gabungan yang mengintegrasikan RPN (*Region Proposal Network*) dan *Fast R-CNN* untuk deteksi objek berbasis *Region* dengan lapisan konvolusi bersama. Perlu mengembangkan teknik yang memungkinkan pembagian lapisan konvolusi antara kedua jaringan ini, dari pada melatih dua jaringan terpisah. Model ini mendiskusikan tiga cara untuk melatih jaringan dengan fitur bersama:

- a. *Alternating training*: Melatih RPN terlebih dahulu, kemudian menggunakan hasilnya untuk melatih *Fast R-CNN*, dan proses ini diulang untuk menyempurnakan kmbnas jaringan.
- b. *Approximate joint training*: Pelatihan gabungan perkiraan menggabungkan jaringan RPN dan *Fast R-CNN* menjadi satu jaringan selama pelatihan, di mana setiap iterasi menghasilkan proposal wilayah yang diperlakukan seperti proposal yang sudah dihitung untuk melatih *Fast R-CNN*. Propagasi mundur dari kerugian RPN dan *Fast R-CNN* digabungkan untuk lapisan yang berbagi parameter. Meskipun mengabaikan turunan terhadap koordinat kotak proposal (sehingga sifatnya perkiraan), solusi ini menghasilkan akurasi *map* yang hampir sama dengan pelatihan bergantian tetapi mengurangi waktu pelatihan hingga 25-50%.

- c. *Non-approximate joint training*: Pelatihan gabungan non-perkiraan mempertimbangkan gradien terhadap koordinat *bounding box* yang diprediksi oleh RPN, karena koordinat ini juga merupakan fungsi dari *input*. Untuk menerapkan backpropagation yang valid secara teoretis, diperlukan lapisan *RoI Pooling* yang bisa dibedakan terhadap koordinat *bounding box*. Hal ini memerlukan lapisan “*RoI warping*” sebagai solusi, namun topik tersebut berada di luar cakupan makalah ini.

Selanjutnya pada tahapan *Training* terdapat 4 langkah *alternative* yang dapat di lihat pada gambar di bawah ini.



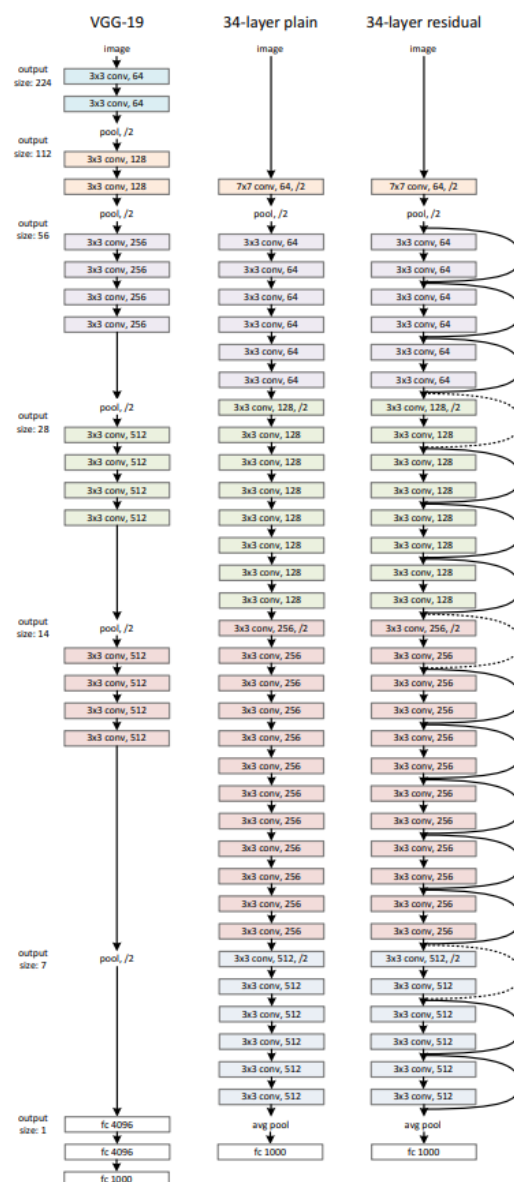
Gambar 2. 13 Pelatihan Alternatif 4-langkah [39]

Gambar 2.13 diatas menunjukkan *Training* Alternatif 4-langkah, Pelatihan ini merupakan metode pragmatis untuk melatih jaringan RPN dan *Fast R-CNN* secara bergantian sambil berbagi fitur konvolusi. Langkah pertama melatih RPN dengan model pra-latih *ImageNet*. Langkah kedua melatih jaringan deteksi *Fast R-CNN* menggunakan proposal dari RPN, tanpa berbagi lapisan konvolusi. Pada langkah ketiga, pelatihan RPN diinisialisasi ulang dengan lapisan bersama yang sudah tetap, hanya menyetel lapisan khusus RPN. Terakhir, lapisan unik *Fast R-CNN* disetel ulang untuk membentuk jaringan terintegrasi dengan lapisan konvolusi yang sama.

2.9. Residual Network (ResNet)

Residual Network (ResNet) telah memberikan kontribusi besar terhadap peningkatan pembelajaran serta performa model *deep learning*. Dalam *deep learning*, lapisan

jaringan yang dalam lebih diutamakan untuk memahami pola yang lebih rumit dan fitur tingkat tinggi, tetapi kedalaman ini dapat menyebabkan masalah gradien menghilang saat backpropagation. Untuk mengatasi masalah ini *Residual network* muncul, selain mengurangi risiko hilangnya gradien adalah untuk memaksimalkan potensi *learning* dari jaringan yang lebih mendalam. ResNet membuktikan bahwa performa jaringan dapat terus meningkat seiring bertambahnya kedalaman, dengan berbagai opsi kedalaman seperti 18, 32, 50, 101, dan 152. Untuk perbandingan Alur kerja jaringan ResNet34, yang memiliki 34 lapisan sederhana yang terinspirasi oleh VGG-19 dan digunakan oleh ResNet ditambah dengan koneksi pintas (*shortcut connection*). Alur kerja ini kemudian diubah menjadi jaringan *Residual* melalui skip koneksi tersebut, seperti yang ditunjukkan pada gambar 2.14.



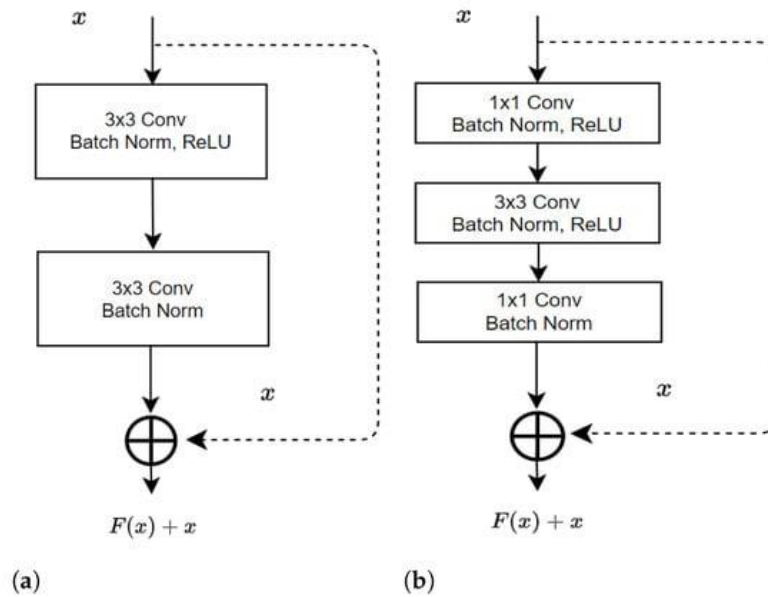
Gambar 2. 14 VGG vs ResNet [7]

Gambar 2.16 di atas membandingkan arsitektur jaringan saraf konvolusional (CNN) VGG-19 dengan dua varian yang lebih sederhana: model 34-layer plain dan 34-layer residual. VGG-19 dikenal karena kedalamannya yang signifikan, menggunakan banyak lapisan konvolusi untuk mengekstrak fitur yang kompleks. Model 34-layer plain mencoba menyederhanakan VGG-19 dengan mengurangi jumlah lapisan dan parameter, namun seringkali mengalami masalah vanishing gradient. Model 34-layer residual mengatasi masalah ini dengan menambahkan koneksi skip yang memungkinkan informasi mengalir langsung ke lapisan yang lebih dalam, sehingga meningkatkan kinerja pelatihan. Koneksi skip, yang juga disebut koneksi *Residual*, adalah mekanisme pada ResNet yang menambahkan *input* dari suatu lapisan langsung ke *output*nya. Mekanisme ini ditunjukkan oleh persamaan 2.1 dibawah ini [41].

$$y = f(x) + x \quad (2.1)$$

Dalam persamaan ini, x merupakan *input* awal atau *output* dari lapisan sebelumnya, sementara $f(x)$ adalah transformasi yang diterapkan pada x di lapisan tersebut, dan y adalah nilai *output* akhirnya. Pada ResNet, koneksi *skip* memungkinkan transfer langsung *input* x ke lapisan selanjutnya, sehingga mempertahankan informasi penting yang mungkin hilang selama pelatihan. Struktur ini memungkinkan jaringan untuk mempelajari fitur baru $F(x)$ sambil menjaga *input* asli x , sehingga mencegah masalah gradien menghilang selama backpropagation dan menjaga kestabilan pelatihan. Koneksi ini juga mengurangi perhitungan berlebih, membantu percepatan konvergensi jaringan. Jika *Residual* bernilai nol, jaringan hanya melakukan pemetaan identitas dari x , yaitu meneruskan *input* tanpa perubahan [41].

Tidak seperti ResNet-18 dan ResNet-34 yang menggunakan blok dasar, model ResNet-50 dan yang lebih dalam menggunakan *blok bottleneck* untuk menangani masalah *gradient* saat kedalaman jaringan bertambah. Perhatikan ilustrasi di bawah ini.



Gambar 2. 15 ResNet *block* diagram. (a) Basic *block* untuk ResNet-18, ResNet-34 (b) *Bottleneck block* untuk ResNet-50, ResNet-101, and ResNet-152 [41]

Gambar 2.15 diatas memperlihatkan perbedaan antara blok dasar dan blok *bottleneck* serta menunjukkan persamaan untuk koneksi *skip* berdasarkan kedalaman jaringan. Blok dasar pada bagian (a) terdiri dari dua lapisan konvolusi 3×3 dengan koneksi *skip*, sedangkan blok *bottleneck* pada bagian (b) mencakup tiga lapisan konvolusi. Lapisan konvolusi pertama berukuran 1×1 mengurangi jumlah saluran *input* sehingga biaya komputasi berkurang, sementara konvolusi 3×3 mempertahankan dimensi spasial *input* untuk ekstraksi fitur yang penting. Lapisan konvolusi kedua berukuran 1×1 mengembalikan dimensi ke ukuran awalnya. Rumus untuk struktur blok *bottleneck* ditunjukkan di bawah ini [41].

$$y = W_3(W_2(W_1)) \quad (2.2)$$

W_1 , W_2 , dan W_3 masing-masing mewakili konvolusi 1×1 , 3×3 , dan 1×1 .

2.10. Siklus Proyek AI

Untuk mengembangkan solusi berbasis AI, perlu membagi seluruh proses menjadi beberapa tahap. Kerangka kerja Siklus Proyek AI dapat membantu mempermudah

melakukannya dengan membagi proses pengembangan proyek berbasis AI dengan domain CV menjadi beberapa tahapan.



Gambar 2. 16 Tahapan Siklus Proyek AI [42]

1. *Problem Scoping*

Tahap ini melibatkan penentuan dan pemahaman permasalahan yang akan diselesaikan dengan solusi AI. Pengguna mendefinisikan pernyataan masalah dan menetapkan tujuan proyek ini. Ini membantu dalam memahami apa yang perlu dicapai dan bagaimana AI dapat membantu dalam mencapai tujuan tersebut. Dengan demikian, pengguna dapat memastikan bahwa AI dapat berfungsi secara efektif dan efisien dalam menyelesaikan permasalahan yang dihadapi terutama dalam bidang dunia sawit.

2. *Data Acquisition*

Untuk melatih model *Faster R-CNN* dengan *backbound* ResNet, diperlukan data yang benar dan memadai. Pada tahap ini, data dikumpulkan dari sumber-sumber yang autentik dan andal. Ini mencakup proses memilih sumber data yang tepat dan mengumpulkan data seperti scraping melalui internet atau mengambil langsung dari kebun melalui *Drone* UAV untuk melatih model AI atau untuk pengujian evaluasi nantinya.

3. *Data Exploration*

Setelah data dikumpulkan, data tersebut perlu dieksplorasi dan dianalisis untuk memastikan tidak ada data yang rusak atau salah. Tahap ini melibatkan *filter* manual data untuk memastikan kualitas data. Selain itu, data sering divisualisasikan untuk mengidentifikasi pola dan tren yang mungkin mempengaruhi hasil akhir.

4. *Modelling*

Pada tahap ini, pengguna memilih model AI yang sesuai dan melatihnya dengan data yang telah dikumpulkan dan dibersihkan pada tahap sebelumnya. Proses ini melibatkan pemilihan algoritma yang tepat dan pengoptimalan model untuk mencapai kinerja terbaik. Model *Faster R-CNN* dengan *backbone* ResNet yang di gunakan dalam penelitian ini.

5. *Evaluation*

Setelah model dilatih, kinerjanya dievaluasi untuk mengukur efisiensinya. Ini melibatkan pengujian model dengan data uji untuk memastikan model dapat melakukan generalisasi dengan baik. Berdasarkan hasil evaluasi, diputuskan apakah model perlu disesuaikan, dilatih ulang, perbaikan dataset atau siap digunakan. Biasanya, model akan diberi skor menggunakan parameter tertentu. Berikut adalah parameter evaluasi yang biasa di gunakan dalam proyek *Artificial Intellegent* (AI).

a. *Confusion Matriks*

Confusion matrix adalah tabel yang menggambarkan performa model klasifikasi dengan data yang memiliki label benar (*supervised*). Berdasarkan tabel ini, berbagai metrik seperti, *presisi*, *recall* dan *F-Score* dapat dihitung, serta beberapa variabel lainnya yang bergantung pada hasil prediksi atau klasifikasi data. Tabel *confusion matrix* berisi empat elemen yang umumnya disajikan dalam bentuk tabel seperti pada gambar 2.17.

		True Class	
		Positive	Negative
Predicted Class	Positive	TP	FP
	Negative	FN	TN

Gambar 2. 17 *Confution Matrix* [43]

- *True Positive* (TP) adalah kondisi ketika model memprediksi data sebagai positif (*TRUE*) dan data sebenarnya juga positif (*TRUE*).
- *True Negative* (TN) terjadi saat model memprediksi data sebagai negatif (*FALSE*) dan data sebenarnya juga negatif (*FALSE*).
- *False Positive* (FP) adalah situasi di mana model memprediks data sebagai positif (*TRUE*), padahal data sebenarnya negatif (*FALSE*).
- *False Negative* (FN) adalah kondisi ketika model memprediks data sebagai negatif (*FALSE*), sedangkan data sebenarnya positif (*TRUE*).

b. *Precision*

Presisi merupakan perbandingan jumlah data yang kategori positif yang diklasifikasikan secara benar oleh sistem dan keseluruhan data yang terklasifikasi positif. Proporsi pohon sawit yang terdeteksi dengan benar dari seluruh pohon sawit yang terdeteksi oleh sistem. Semakin tinggi *precision*, semakin sedikit kesalahan deteksi (pohon yang bukan sawit terdeteksi sebagai sawit). Presisi dapat dihitung menggunakan persamaan (2.3) berikut [43].

$$Precision = \frac{TP}{TP + FP} \quad (2.3)$$

c. *Recall*

Recall adalah ukuran yang menunjukkan seberapa banyak data berlabel positif yang berhasil diidentifikasi dengan benar oleh sistem. Proporsi pohon sawit yang terdeteksi dengan benar dari seluruh pohon sawit yang ada di citra udara. Semakin tinggi *recall*, semakin sedikit pohon sawit yang terlewatkan (tidak terdeteksi). *Recall* dapat dihitung menggunakan persamaan (2.4) berikut [43].

$$Recall = \frac{TP}{TP + FN} \quad (2.4)$$

d. *F1 Score*

F1-Score digunakan untuk menghitung kombinasi antara presisi dan *recall*. *F1-Score* memanfaatkan rata-rata harmonis dari presisi dan *recall*. Semakin tinggi *F1-score* memberikan ukuran semakin keseimbangan terbaik antara *precision* dan *recall*. *F1-Score* dapat dihitung menggunakan persamaan (2.5) berikut [43].

$$F1 - Score = \frac{Precision \cdot Recall}{Precision + Recall} \quad (2.5)$$

e. *Mean Average Precision (map)*

Mean Average Precision (map) adalah metrik yang digunakan untuk mengukur akurasi detektor objek di semua kelas dalam basis data tertentu. Semakin tinggi mAP, semakin baik kinerja model dalam mendeteksi pohon sawit dengan tepat dan lengkap. mAP hanyalah AP rata-rata di semua kelas yang dapat di formulakan seperti pada persamaan (2.6) berikut [43].

$$mAP = \frac{1}{n} \sum_{i=1}^n AP_i \quad (2.6)$$

Di mana AP_i adalah AP untuk kelas ke- i dan N adalah jumlah total kelas yang dievaluasi.

f. *Mean Squared Error (MSE)*

Mean Squared Error (MSE) adalah salah satu metrik yang digunakan untuk mengukur tingkat kesalahan antara nilai sebenarnya dan nilai prediksi yang dihasilkan oleh suatu model. MSE menghitung rata-rata kuadrat selisih antara nilai prediksi dan nilai actual di formulakan seperti pada persamaan (2.7) berikut [43].

$$MSE = \frac{1}{n} \sum_{i=1}^n (\text{nilai true} - \text{nilai prediction})^2 \quad (2.7)$$

g. *Root Mean Squared Error (RMSE)*

Root Mean Squared Error (RMSE) memberikan informasi tentang seberapa jauh prediksi model dari nilai sebenarnya, dengan memberikan bobot lebih pada kesalahan yang lebih besar. Semakin kecil nilai MSE dan RSME, semakin baik kinerja model. RMSE dihitung dengan cara mengambil akar dari MSE yang di formulakan seperti pada persamaan (2.8) berikut [43].

$$RSME = \sqrt{MSE} \quad (2.8)$$

III METODE PENELITIAN

3.1. Waktu dan Tempat Penelitian

Penelitian tugas akhir ini dilakukan dari bulan Agustus 2024 hingga Januari 2024 di Laboratorium Telekomunikasi, Prodi Teknik Elektro, Jurusan Teknik Elektro, Fakultas Teknik, Universitas Lampung. Pada tahapan *Data Acquisition* diambil dari kebun PT Sampoerna Agro, mesuji, Lampung.

3.2. Alat dan Bahan Penelitian

Adapun alat dan bahan yang digunakan dalam penelitian ini adalah sebagai berikut:

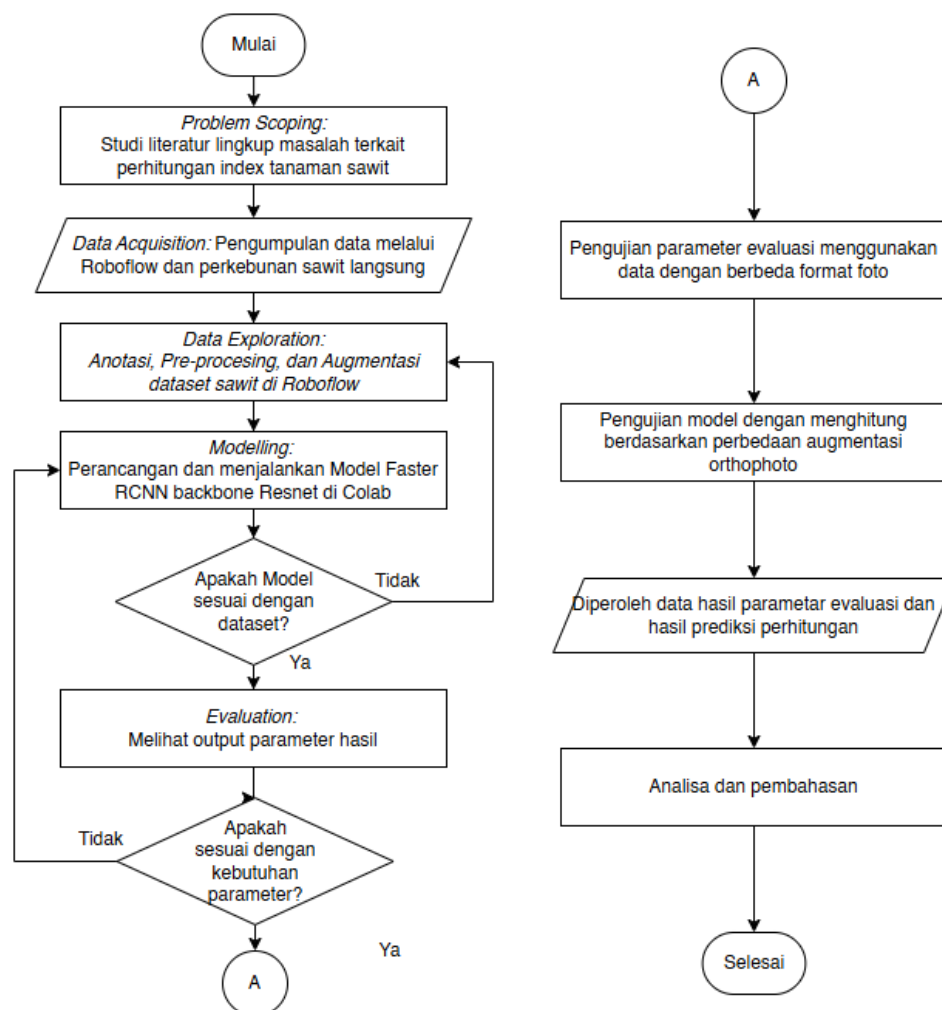
1. Satu unit pesawat UAV tipe *Fixed Wing* yang dilengkapi dengan kamera SONY A6000, digunakan untuk mengambil citra udara di perkebunan PT Sampoerna Agro di Mesuji, Lampung.
2. Satu laptop Acer dengan prosesor Intel Core i5 yang digunakan untuk melatih model dan menjalankan perangkat lunak deteksi objek.
3. Satu komputer dengan prosesor Ryzen 5 5600x dan *Graphic Card* NVIDIA GeForce RTX 2060 12G.
4. Perangkat lunak Agisoft Metashape *Professional* yang digunakan untuk memproses gambar dan data citra sawit dalam pembuatan wilayah gambar (peta).
5. Perangkat lunak Roboflow yang digunakan untuk eksplorasi data.
6. Perangkat lunak Google Colab yang digunakan untuk melatih model.
7. *Faster R-CNN* dengan *backbone* ResNet yang digunakan untuk mendeteksi dan menghitung jumlah pohon.

3.3. Pembuatan Algoritma dan Testing

Pembuatan algoritma dan melakukan testing secara keseluruhan dilakukan menggunakan Google Colab dengan *runtime* T4 GPU supaya mempercepat proses *running* program. Referensi program dari berbagai sumber seperti GitHub, ChatGPT, Gemini, YouTube, Kaggle, dan lainnya.

3.4. Diagram Alir Penelitian

Alur kerja dan metode yang diterapkan dalam penelitian ini mengikuti *AI Project Cycle*. *AI Project Cycle* untuk penelitian ini diilustrasikan dalam diagram alir yang terdapat pada gambar 3.1 berikut.



Gambar 3. 1 Diagram *Flowchart* Penelitian

Gambar 3.1 diagram alir di atas diawali menjelaskan bagaimana proses tahapan *problem scoping* untuk mencari literatur dalam mengidentifikasi masalah pada perhitungan indeks tanaman sawit, kesulitan dalam menghitung jumlah pohon sawit secara akurat menggunakan metode manual yang memakan waktu dan rentan kesalahan. Solusinya adalah dengan menggunakan teknologi drone dan kecerdasan buatan berbasis deep learning untuk memperoleh data visual yang lebih akurat dan efisien. Tahapan kedua adalah *data acquisition* dalam tahapan ini data kebun sawit di akuisis melalui *software* roboflow, untuk melakukan pengujian data parameter evaluasi di buat untuk datatest dengan project yang berbeda yang berisi *crop* data fotogrametri dari kebun PT Sampoerna Agro, Mesuji, Lampung. Selanjutnya di lakukan proses *Data Exploration* pada datatest dari kebun Sampoerna Agro untuk di lakukan pelabelan atau anotasi serta di lakukan preprosesing dan augmentasi secara otomatis pada *software* tersebut. Lalu data di *load* ke dalam *google colab* untuk di lakukan proses *modeling*.

Proses *Modeling* ini diawali dengan meng-import dataset API roboflow, manajemen *file* dan folder, membuat beberapa fungsi, dan perubahan data sehingga bisa di gunakan untuk model *Faster R-CNN* dengan dilakukan perubahan variasi *backbone* ResNet. Proses *training* di jalankan dengan menentukan parameter *compile* apa yang di gunakan dan menggunakan berapa kali proses iterasi (*epoch*). Jika iterasi berjalan dan memunculkan parameter yang sesuai maka dilanjutkan proses *evaluation*, namun jika gagal maka perlu dilakukan perbaikan dataset atau perbaikan fungsi-fungsi yang di jalankan sebelum *training* model. Selanjutnya pada tahapan *evaluation*, dapat di lihat parameter hasil datatest yang di peroleh jika telah sesuai dilanjutkan pada tahapan selanjutnya, parameter tersebut seperti *Precision*, *Recall*, *F1-score*, mAP, MSE, dan RSME. Namun jika tidak sesuai bisa di lakukan proses *Training* ulang. Selanjutnya melakukan proses pengujian untuk dataset baru untuk di deteksi dan di hitung. Proses Pengambilan data hasil parameter evaluasi di lakukan pada berbagai format untuk data *testing* yang berbeda. Selanjutnya tahapan perhitungan di lakukan dengan metode *cropping* yang dilakukan dengan membandingkan nilai hitung secara manual dengan cara prediksi oleh model *deep learning*. Proses ini di lakukan secara berulang dengan mengganti variasi *backbone* atau mengganti model untuk di lakukan proses perbandingan. Langkah terakhir setelah data hasil di peroleh maka dilakukan tahapan analisa dan pembahasan. Analisa data di lakukan dengan membandingkan data evaluasi

dan data hitung disetiap variasi *backbone* atau variasi model uji lainnya untuk di bahas secara jelas. Penjelasan lebih mendetai akan di jelaskan berikut ini:

3.4.1. *Problem Scoping*

Pada tahap awal, dilakukan studi literatur untuk memahami tantangan dan peluang dalam pendeteksian objek berbasis fotogrametri, khususnya pada perkebunan kelapa sawit. Kajian literatur ini juga membantu mengidentifikasi kesenjangan penelitian (research gap) dalam implementasi algoritma Faster R-CNN dan pengoptimalan *backbone* seperti ResNet untuk mendukung deteksi pohon kelapa sawit. Solusi berbasis Faster R-CNN dengan *backbone* ResNet diharapkan mampu menjawab kebutuhan ini. Berikut adalah rangkuman fokus penelitian dalam bentuk analisis *Problem Scoping*.

Tabel 3. 1 *Problem Scoping*

4W	Rincian
<i>What</i>	Tantangan dalam mendeteksi dan menghitung jumlah pohon kelapa sawit secara akurat pada lahan perkebunan yang luas, sehingga diperlukan metode berbasis <i>deep learning</i> yang memanfaatkan citra fotogrametri dari drone untuk hasil yang efisien dan akurat.
<i>Who</i>	Penelitian dilakukan dengan menggunakan data dari perkebunan kelapa sawit milik PT Sampoerna Agro di Mesuji, Lampung, sebagai studi kasus utama.
<i>Where</i>	Fokus penelitian adalah pada perkebunan kelapa sawit di Mesuji dengan algoritma Faster R-CNN dengan Backbone ResNet, namun metode yang dikembangkan memiliki potensi untuk diterapkan pada jenis vegetasi lain
<i>Why</i>	Masalah ini timbul karena metode tradisional yang digunakan saat ini tidak efisien dan kurang akurat, sedangkan pengelola perkebunan membutuhkan data yang presisi untuk mendukung pengambilan keputusan yang optimal dalam manajemen lahan.

Tahapan Studi literatur dilakukan dengan meninjau publikasi ilmiah yang relevan menggunakan platform seperti *Google Scholar* dan *ResearchGate*. Kata kunci utama dalam pencarian meliputi “*object detection*,” “*Faster R-CNN*,” “*ResNet*,” “*oil palm plantation*,” dan “*drone-based mapping*.” Literatur yang dipilih memenuhi kriteria berikut:

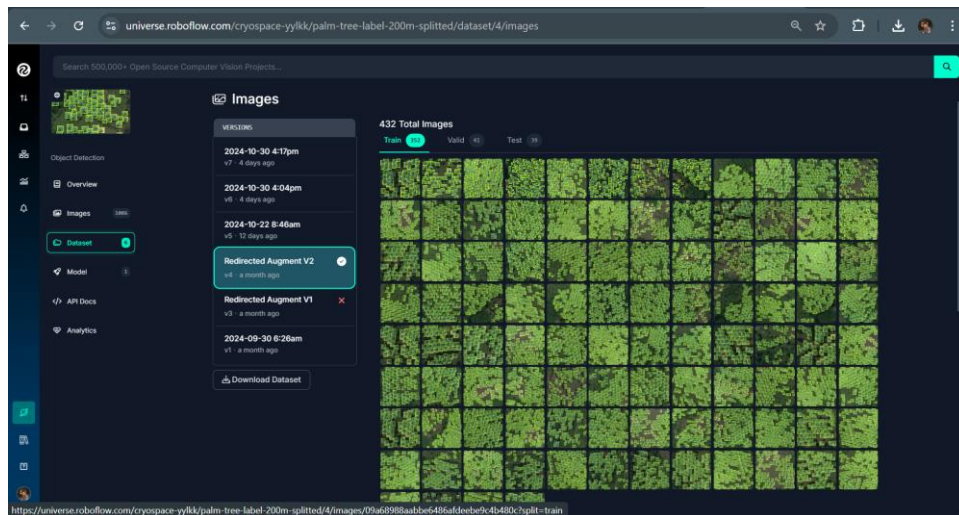
- Publikasi ilmiah yang relevan dengan deteksi objek berbasis citra udara menggunakan deep learning.
- Penelitian yang secara khusus membahas implementasi algoritma Faster R-CNN atau model deteksi objek lain untuk tanaman atau vegetasi.
- Publikasi dalam bahasa Indonesia dan Inggris, baik dalam jurnal nasional maupun jurnal internasional.
- Fokus pada studi berbasis citra udara, seperti *orthophoto* atau fotogrametri, yang digunakan untuk analisis perkebunan.

Melalui pendekatan ini, referensi yang diperoleh tidak hanya menjadi landasan teori tetapi juga memberikan wawasan tentang metodologi yang telah terbukti efektif dalam konteks yang serupa. Berdasarkan hasil kajian literatur, beberapa tantangan utama dalam pendeteksian pohon kelapa sawit menggunakan fotogrametri teridentifikasi, antara lain:

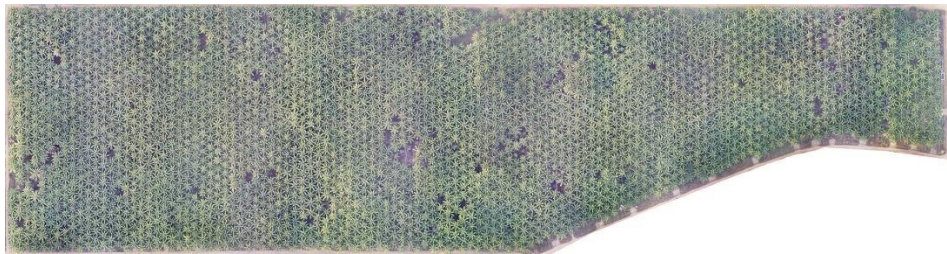
- Kepadatan dan tumpang tindih objek tanaman
- Variasi augmentasi dan kualitas gambar
- Optimalisasi model *Faster* R-CNN dengan *backbone* ResNet
- Konteks perkebunan kelapa sawit yang terbatas dalam penelitian
- Manajemen Data Skala Besar

3.4.2. *Data Acquisition*

Pada tahap ini, dataset diperoleh melalui dua langkah utama: pengambilan data secara *online* dan *offline*. Tahap pertama dilakukan dengan memperoleh dataset utama melalui website project Roboflow, dataset berjumlah 432 gambar tersebut umumnya di bagi lagi atau di-*splitt* untuk memudahkan proses selanjutnya dengan rincian *splitting* data: data train 80% dengan total 352 gambar, data valid 9% dengan total 41 gambar, dan data test 39 gambar. Namun pada akhirnya semua dataset ini tidak akan di *splitt*, melainkan di gunakan seluruhnya untuk data dalam proses training, sehingga diperoleh 432 data training. Dikarenakan Data lainnya akan di peroleh dari kebun PT Sampoerna Agro blok 44, Mesuji, Lampung.



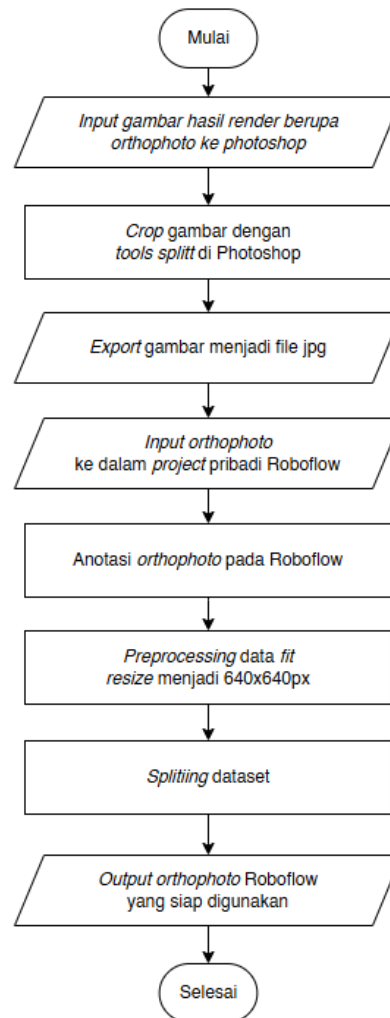
Gambar 3. 2 Dataset Roboflow [44]



Gambar 3. 3 Fotogrametri Lahan Sawit Sampoerna Agro Mesuji [13]

3.4.3. *Data Exploration*

Setelah data di akuisi selanjutnya di lakukan eksplorasi data hingga hingga data gambar dapat dilakukan proses modeling. Berikut gambar 3.4 menjelaskan alur proses untuk tahap *Data Exploration*.

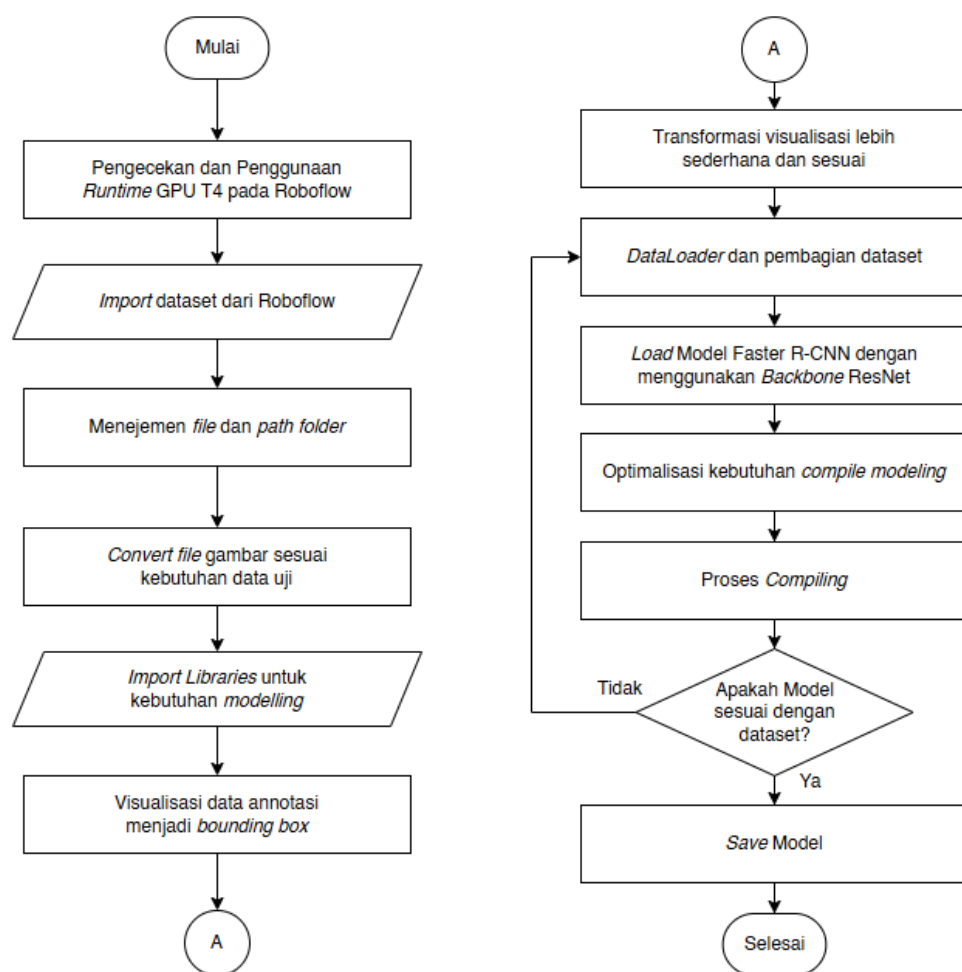


Gambar 3. 4 Diagram *Flowchart Data Exploration*

Gambar 3.4 diatas dimulai dengan fotogrametri yang sudah di *render*, dengan hasil *file* '.tif' gambar di import ke *software* adobe photoshop untuk meng-*crop* dan di *export* kebentuk *file* .jpg. Proses *crop* di lakukan agar data untuk melakukan validasi di peroleh dalam jumlah yang banyak. *Crop* di lakukan dengan melakukan dengan *tools splitt* di photoshop dengan menginputkan jumlah potongan gambar baik vertical maupun horizontal. Tahap akhir pada *software* photoshop adalah meng-*ekspert* kedalm bentuk gambar berupa format jpg. Tahapan selanjutnya pada *software* roboflow peneliti melakukan prosses pembuatan project dan menginputkan gambar yang sudah di *ekspert* diphotoshop kedalam project roboflow. Setelah gambar hasil *crop* di import ke roboflow secara bertahap gambar di lakukan proses anotasi atau pelabelan berupa *bounding box* pada setiap objek gambar pohon sawit.

Dataset yang sudah selesai di lakukan proses anotasi umumnya data akan di lakukan proses augmentasi namun tidak perlu di lakukan mengingat data ini digunakan untuk proses evaluasi. Data tersebut di lakukan *split* seluruhnya kedalam dataset *training* yang nantinya akan disesuaikan saat proses *modelling*. Selanjutnya pada tahapan preprocessing data di lakukan perubahan gambar menjadi lebih kecil dengan skala 1:1, *resize fit* ini jika tidak memenuhi skala tersebut maka gambar akan ditambah backround putih agar memenuhi skala tersebut.

3.4.4. *Modelling*



Gambar 3. 5 Diagram *Flowchart Modelling*

Gambar 3.5 ini merupakan tahapan diawali dengan pengecekan jenis runtime GPU T4 agar siap untuk di gunakan, Proses di Injutkan dengan melakukan import dua *project* dataset dari Roboflow. Setelah itu, dataset tadi di *split* untuk di lakukan

proses training dan proses validasi, data tersebut di manajemen dan penentuan *folder* supaya memudahkan untuk proses dataset selanjutnya. Untuk melakukan *modeling* perlu dilakukan meng-*impor* berbagai library penting seperti *os*, *numpy*, *Matplotlib*, *torch*, *torchVision*, dan lainnya. Banyak fungsi program yang di jalankan selanjutnya di mulai dari *generate* visualisasi *bounding box* dari bentuk *annotations XML*. Setelah di visualisasikan data dilakukan transformasi kedalam bentuk *torch* agar lebih efisien dan sesuai yang di inginkan model.

Pada bagian berikutnya, dataset dibagi menggunakan *DataLoader* untuk keperluan *training* dan evaluasi. Kemudian, model *Faster R-CNN* dengan *Backbone* *ResNet* di-*load* dan parameternya dapat dioptimalkan sesuai kebutuhan untuk *compile modeling*. Setelah itu, proses *compiling* dilakukan, diikuti dengan evaluasi untuk memastikan model sesuai dengan dataset. Jika model sudah memenuhi kriteria yang diinginkan, model tersebut disimpan untuk penggunaan selanjutnya. Tahapan ini memastikan bahwa model dapat bekerja optimal dalam mendeteksi objek sesuai dengan dataset yang digunakan. Karena Beberapa kasus data set yang tidak sesuai sehingga proses *compiling* tidak berhasil dan model tidak bisa di evaluasi.

3.4.5. *Evaluation*

Tahapan setelah model di peroleh adalah melakukan evaluasi menggunakan dataset evaluasi dari *crop orthophoto* kebun PT Sampoerna Agro untuk menguji Tingkat akurasi model diberbagai format gambar seberapa baik. Parameter evaluasi yang di gunakan adalah *Precision*, *Recall*, *F1-score*, *Accuracy*, *map*, *MSE*, dan *RSME*. Setelah format foto sudah sesuai, maka format foto tersebut di jadikan basis dalam mendeteksi dan menghitung *orthophoto* di berbagai augmentasi gambar. Setelah di peroleh parameter evaluasi dan proses perhitungan maka hasil tersebut dilakukan proses lebih lanjut untuk di analisis dan dibahas.

BAB V PENUTUP

5.1. Kesimpulan

Adapun Kesimpulan pada penelitian ini adalah sebagai berikut

1. Tahapan pengukuran bobot nilai parameter evaluasi dengan berbagai format gambar di peroleh model *Faster R-CNN Backbone* ResNet50 memiliki performa dan kesetabilan terbaik yaitu sebesar 89.02% dengan nilai *error* paling rendah disemua format gambar, namun jika dilihat dari setiap format gambar, JPG menjukan format terbaik pada *backbone* ResNet 152 sebesar 89.70%, pada format PNG dan AVIF terbaik adalah ResNet 18 (90.8% dan 87.33%), dan BMP terbaik pada ResNet50 sebesar 90.63%.
2. Akurasi deteksi dan perhitungan model Faster R-CNN dengan *backbone* ResNet diberbagai augmentasi di peroleh rata-rata akurasi tertinggi untuk format PNG pada ResNet50 sebesar 85.76%, disusul oleh ResNet 34 (84.94%), ResNet 18 (84.69%), ResNet 101 (83.84%) dan terakhir ResNet152 (81.55%). Sedangkan pengujian pada format JPG rata-rata akurasi tertinggi pada ResNet 50 juga (85.97%), di susul oleh ResNet 18 (85.96%), ResNet 34 (84.66), ResNet152 (84.63%) dan ResNet101 (82.65%).
3. Pengujian standar *orthophoto* dengan meningkatkan parameter augmentaasi *Brightness*, Hue, dan Kontras model dapat meningkatkan akurasi perhitungan deteksi, sedangkan dalam standar augmentasi *shear* sekitar 30 drajat model masih baik masih di akurasi sekitar 80%, dan pada pengujian *cut off* akurasi model tetap stabil disekitar 89 - 90%.

5.2. Saran

Adapun saran yang dapat digunakan dalam penelitian selanjutnya adalah sebagai berikut

1. Penelitian selanjutnya peneliti dapat memaksimalkan dataset baik dari jumlah objek dalam setiap gambar ataupun meningkatkan kualitas setiap gambar untuk dataset.
2. Penelitian selanjutnya jika menggunakan model yang sama, baiknya arsitektur model di modifikasi supaya model lebih efisien lagi untuk melakukan perhitungan lebih banyak pohon dalam satu gambar.
3. Model yang dirancang masih membatasi waktu dan kapasitas GPU dalam proses *running*. Penelitian selanjutnya harusnya melakukan jenis *runtime* yang lebih baik atau melakukan program di lokal dengan *device* yang lebih baik agar tidak keterbatasan memori yang dapat menghambat inferensi gambar beresolusi tinggi dengan jumlah *bounding box* yang sangat besar.

DAFTAR PUSTAKA

- [1] E. F. Himmah, M. Widyaningsih, and M. Maysaroh, "Identifikasi Kematangan Buah Kelapa Sawit Berdasarkan Warna RGB Dan HSV Menggunakan Metode K-Means Clustering," *J. Sains dan Inform.*, vol. 6, no. 2, pp. 193–202, 2020. E-ISSN: 2698-5841. <https://doi.org/10.34128/jsi.v6i2.242>
- [2] D. S. Prasvita, M. M. Santoni, R. Wirawan, and N. Trihastuti, "Klasifikasi Pohon Kelapa Sawit Pada Data Fusi Citra Lidar Dan Foto Udara Menggunakan Convolutional Neural Network," *JUPI (Jurnal Ilm. Penelit. dan Pembelajaran Inform.)*, vol. 6, no. 2, pp. 406–415, 2021. ISSN: 2540-8984. <https://doi.org/10.29100/jupi.v6i2.2437>
- [3] Kementerian Koordinator Bidang Perekonomian Republik Indonesia, "Industri kelapa sawit Indonesia: Menjaga keseimbangan aspek sosial, ekonomi, dan lingkungan," 2021. [Online]. Available: <https://www.ekon.go.id/publikasi/detail/2921/industri-kelapa-sawit-indonesia-menjaga-keseimbangan-aspek-sosial-ekonomi-dan-lingkungan>. Accessed: Dec. 1, 2024.
- [4] Devi Yana Armanto, Sahid Agustian Hudjimartsu, and E. Hermawan, "Identifikasi Perhitungan Pohon Kelapa Sawit Otomatis Dengan Menggunakan Metode Convolutional Neural Network (CNN)," *JATI (Jurnal Mahasiswa Teknik Informatika)*, vol. 8, no. 3, pp. 2648–2654, May 2024. E-ISSN: 2598828X.
- [5] I Putu Harianja Prayogo, F. J. Manoppo, and Lucia, "Pemanfaatan Teknologi Unmanned Aerial Vehicle (Uav) Quadcopter Dalam Pemetaan Digital (Fotogrametri) Menggunakan Kerangka Ground Control Point (Gcp)," *Jurnal Ilmiah Media Engineering*, vol. 10, no. 1, 2020, Accessed: Nov. 04, 2024. ISSN: 2087-9334.
- [6] J. Zheng, W. Li, M. Xia, R. Dong, H. Fu, and S. Yuan, "Large-Scale Oil Palm Tree Detection from High-Resolution Remote Sensing Images Using Faster-RCNN | IEEE Conference Publication | IEEE Xplore," ieeexplore.ieee.org, 2019.
- [7] K.-H. Lee, R.-W. Lee, J.-S. Yun, M.-S. Kim, and H.-S. Choi, "Automated Diagnosis of Knee Osteoarthritis Using ResNet101 on a DEEP:PHI: Leveraging a No-Code AI Platform for Efficient and Accurate Medical Image Analysis," *Diagnostics*, vol. 14, no. 21, pp. 2451–2451, Nov. 2024.
- [8] Nie Y, Shen Y, Chen Q. Combination Analysis of Future Polar-Type Gravity Mission and GRACE Follow-On[J]. *Remote Sensing*, 2019, 11(2): 200. ISSN: 2072-4292.

- [9] Li, W.; Dong, R.; Fu, H.; Yu, L. Large-Scale Oil Palm Tree Detection from High-Resolution Satellite Images Using Two-Stage Convolutional Neural Networks. *Remote Sens.* 2019, 11, 11. ISSN: 2072-4292.
- [10] A. Ammar, A. Koubaa, and B. Benjdira, "Deep-Learning-Based Automated Palm Tree Counting and Geolocation in Large Farms from Aerial Geotagged Images," *Agronomy*, vol. 11, no. 8, p. 1458, Jul. 2021. ISSN:1435-0645. <https://doi.org/10.3390/agronomy11081458>
- [11] P. N. Chowdhury, P. Shivakumara, L. Nandanwar, F. Samiron, U. Pal, and T. Lu, "Oil palm tree counting in drone images," *Pattern Recognition Letters*, vol. 153, pp. 1–9, Jan. 2022. ISSN: 1872-7344.
- [12] Muhammad Mujahidin Furqan and Hanifah Rahmi Fajrin, "Pemanfaatan Pesawat Tanpa Awak Mode Fixed Wing sebagai Sarana Pengiriman Obat-obatan," *Medika Teknik Jurnal Teknik Elektromedik Indonesia*, vol. 6, no. 1, pp. 44–52, Nov. 2024. ISSN : 27160505. <https://doi.org/10.18196/mt.v6i1.22105>
- [13] PT Dronila Nirwana Bentala, "Dokumentasi Perusahaan" diterima secara langsung, 1-Nov-2023. Accessed: Nov. 1, 2024.
- [14] B. K. Yakti and R. H. Prayitno, "Perbandingan Dan Analisa Gambar Pada Steganografi Berdasarkan MSE Dan PSNR," *ICIT Journal*, vol. 6, no. 2, pp. 138–152, Aug. 2020. ISSN: 2654-8704.
- [15] N. Camgöz, C. Yener, and D. Güvenç, "Effects of hue, saturation, and brightness on preference," *Color Research & Application*, vol. 27, no. 3, pp. 199–207, Apr. 2002. ISSN:1520-6378
- [16] E. M. author, "Penggunaan deep belief network-deep neural network (dbn-dnn) untuk pengenalan emosi dasar dengan Singular Value Decomposition (SVD) = Deep Belief Network-Deep Neural Network (DBN-DNN) for facial emotion recognition using Singular Value Decomposition (SVD)," Universitas Indonesia Library, 2024.
- [17] L. Santos, F. Neves Dos Santos, P. Moura Oliveira, and P. Shinde, "Deep Learning Applications in Agriculture: a Short Review," pp. 139–151, 2020.
- [18] K. Kipli et al., "Deep learning applications for oil palm tree detection and counting," *Smart Agricultural Technology*, vol. 5, p. 100241, Oct. 2023. ISSN: 2772-3755
- [19] M. Haris Diponegoro, S. Suning Kusumawardani, and I. Hidayah, "Tinjauan Pustaka Sistematis: Implementasi Metode Deep Learning pada Prediksi Kinerja Murid," *Jurnal Nasional Teknik Elektro dan Teknologi Informasi*, vol. 10, no. 2, pp. 131–138, May 2021. ISSN : 24605719
- [20] M Abdy Mulya, None Zaenul Arif, and None Syefudin, "Tinjauan Pustaka Sistematis : Penerapan Metode Gabor Wavelet Pada Computer Vision," *Journal of Computer Science and Technology*, vol. 1, no. 2, pp. 83–88, May 2023. ISSN: 1860-4749.

- [21] A. C. K. Y. S. e. a. Esteva, "Deep learning-enabled medical computer vision," *NPJ digital medicine*, vol. 4, no. 5, 2021. ISSN: 2398-6352.
- [22] Z. Li, F. Liu, W. Yang, S. Peng, and J. Zhou, "A Survey of Convolutional Neural Networks: Analysis, Applications, and Prospects," *IEEE Trans. Neural Networks Learn. Syst.*, pp. 1–21, 2021. ISSN 2691-4581. <https://doi.org/10.1109/TNNLS.2021.3084827>
- [23] M. K. Bohmrah and H. Kaur, "Classification of Covid-19 patients using efficient fine-tuned deep learning DenseNet model," *Glob. Transitions Proc.*, vol. 2, no. 2, pp. 476–483, 2021. ISSN: 2589-7918.
- [24] A. Khan, A. Sohail, U. Zahoor, and A. S. Qureshi, "A survey of the recent architectures of deep convolutional neural networks," *Artif. Intell. Rev.*, vol. 53, no. 8, pp. 5455–5516, 2020. ISSN: 1872-7921
- [25] S. J. Shri and S. Jothilakshmi, "Crowd Video Event Classification using Convolutional Neural Network," *Comput. Commun.*, vol. 147, pp. 35–39, 2019. ISSN: 1873-703X.
- [26] R. Roncancio, A. El Gamal, and J. P. Gore, "Turbulent flame image classification using Convolutional Neural Networks," *Energy AI*, vol. 10, p. 100193, 2022. ISSN: 2666-5468
- [27] Ken Ratri Wardani and L. Leonardi, "Klasifikasi Penyakit pada Daun Anggur menggunakan Metode Convolutional Neural Network," *Jurnal Tekno Insentif*, vol. 17, no. 2, pp. 112–126, Oct. 2023. E-ISSN : 2655089X.
- [28] MathWorks, "Convolutional Neural Network," [Online]. Available: <https://www.mathworks.com/discovery/convolutional-neural-network.html>. Accessed: Nov. 1, 2024.
- [29] O. A. Montesinos López, A. Montesinos López, and J. Crossa, *Multivariate Statistical Machine Learning Methods for Genomic Prediction*, no. January. 2022. <https://doi.org/10.1007/978-3-030-89010-0>
- [30] S. A. Suha and T. F. Sanam, "A deep convolutional neural network-based approach for detecting burn severity from skin burn images," *Mach. Learn. with Appl.*, vol. 9, no. April, p. 100371, 2022. ISSN: 1573-0565.
- [31] H. Chen et al., "Plant image recognition with deep learning: A review," *Computers and Electronics in Agriculture*, vol. 212, pp. 108072–108072, Sep. 2023. ISSN: 1872-7107.
- [32] R. Riad, O. Teboul, D. Grangier, and N. Zeghidour, "Learning strides in convolutional neural networks," *International Conference on Learning Representations*, pp. 1–17, 2022.
- [33] A. Nguyen, S. Choi, W. Kim, S. Ahn, J. Kim, and S. Lee, "Distribution Padding in Convolutional Neural Networks," *2019 IEEE International Conference on Image Processing (ICIP)*, pp. 4275–4279, 2019.

- [34] S. A. Singh, T. G. Meitei, and S. Majumder, "Short PCG classification based on deep learning," *Deep Learning Techniques for Biomedical and Health Informatics*, Elsevier Inc., pp. 141–164, 2020. ISSN: 2197-6511.
- [35] N. Sabri, "A Comparison between Average and Max-Pooling in Convolutional Neural Network for Scoliosis Classification," *International Journal of Advanced Trends in Computer Science and Engineering*, vol. 9, no. 1.4, pp. 689–696, Sep. 2020. ISSN: 2278 - 3091
- [36] D. Bhatt et al., "Cnn variants for computer vision: History, architecture, application, challenges and future scope," *Electron.*, vol. 10, no. 20, p. 2470, 2021. ISSN:2751-2614.
- [37] C. B. Murthy, M. F. Hashmi, N. D. Bokde, and Z. W. Geem, "Investigations of Object Detection in Images/Videos Using Various Deep Learning Techniques and Embedded Platforms—A Comprehensive Review," *Applied Sciences*, vol. 10, no. 9, p. 3280, May 2020. ISSN: 2076-3417.
- [38] A. Banerjee, E. Kumar, and Ravinder Megavath, "Learning optimal deep prototypes for video retrieval systems with hybrid SVM-softmax layer," *International Journal of Data Science and Analytics*, Jun. 2024. ISSN: 2364-4168.
- [29] Ardian Umam, "Gambar diambil dari playlist 'Deep Learning: Faster RCNN untuk Deteksi Objek,'" YouTube. [Online]. Available: <https://youtube.com/playlist?list=PLkRkKTC6HZMy8smJGhhZ4HBIQgShLaTo8&si=qFxQcIWJ8lZt78r6>. Accessed: Nov. 1, 2024.
- [40] W. WU et al., "Detection and enumeration of wheat grains based on a deep learning method under various scenarios and scales," *Journal of Integrative Agriculture*, vol. 19, no. 8, pp. 1998–2008, Aug. 2020. ISSN: 2352-3425.
- [41] T. Wi, M. Yang, S. Park, and J. Jeong, "Welcome To Zscaler Directory Authentication," Doi.org, 2024.
- [42] CBSE Skill Education, "AI project cycle class 10," CBSE Skill Education. [Online]. Available: <https://cbse-skilleducation.com/ai-project-cycle-class-10/>. Accessed: Nov. 1, 2024.
- [43] J. Terven, D. M. Cordova-Esparza, A. Ramirez-Pedraza, and E. A. Chavez-Urbiola, "Loss Functions and Metrics in Deep Learning. A Review," *arXiv.org*, Jul. 05, 2023. <https://arxiv.org/abs/2307.02694>
- [44] Roboflow, Palm Tree Label 200m Splitted Dataset. [Online]. Available: <https://universe.roboflow.com/cryospace-yyllk/palm-tree-label-200m-splitted/dataset/4>. Accessed: Nov. 1, 2024.