

## BAB II

### TINJAUAN PUSTAKA

#### 2.1 Penelitian Terdahulu

##### 2.1.1 *Comparison of CNN-Based Architectures for Detection of Different Object Classes*

Penelitian yang dilakukan oleh Natalia Bilous, et al. Tujuan penelitian tersebut adalah, membandingkan arsitektur berbasis CNN untuk mendeteksi berbagai kelas objek. beberapa model yang digunakan adalah YOLOv4, YOLOv5, YOLOv6, YOLOv7, YOLOv8, *Faster R-CNN*, SSD, dan *EficientDet*. Kedelapan model tersebut dibandingkan dengan metrik *Precision*, *Recall*,  $mAP@0.5$ , dan *F1-Score*. Hasilnya adalah YOLOv8 menunjukkan performa tertinggi di antara semua model, menunjukkan hasil terbaik di semua metrik utama, tetapi *Faster R-CNN* unggul dengan akurasi tinggi tetapi kalah dengan YOLOv8 dalam kecepatan pemrosesan [8].

Dari penelitian tersebut, hal yang dijadikan acuan adalah:

- Penggunaan arsitektur *Faster R-CNN* terbukti lebih unggul dalam metrik akurasi, oleh karena itu arsitektur *Faster R-CNN* dipilih menjadi arsitektur yang digunakan pada penelitian ini.

##### 2.1.2 *Handling Class Imbalance in Intrusion Detection Dataset Using Undersampling and Oversampling Techniques*

Penelitian yang dilakukan oleh Emmanuel Chinanu Uwazie, Afolayan A. Obiniyi, dan Morufu Olalere bertujuan untuk mengatasi masalah ketidakseimbangan kelas pada dataset deteksi intrusi jaringan, yang dapat menyebabkan bias model terhadap kelas mayoritas dan menurunkan akurasi pada kelas minoritas. Salah satu pendekatan yang digunakan adalah *undersampling*, yaitu dengan mengurangi jumlah data pada kelas mayoritas menggunakan algoritma *RandomUnderSampler*. Model klasifikasi yang digunakan dalam penelitian ini adalah K-Nearest

Neighbors (KNN), dan evaluasi dilakukan menggunakan metrik *Precision*, *Recall*, *F1-Score*, dan akurasi.

Hasil dari penelitian menunjukkan bahwa *undersampling* mampu meningkatkan kinerja model dalam mendeteksi serangan minor dan juga secara signifikan mengurangi waktu pelatihan dan prediksi. *F1-Score* pada beberapa kelas meningkat secara drastis setelah dilakukan *undersampling* terhadap kelas mayoritas seperti "Benign" yang mendominasi dataset [9].

Dari penelitian tersebut, hal yang dijadikan acuan adalah:

- Teknik *undersampling* terbukti efektif dalam mengurangi dominasi kelas mayoritas dan meningkatkan akurasi model terhadap kelas minoritas. Oleh karena itu, dalam penelitian ini, pendekatan *undersampling* diterapkan untuk menyeimbangkan distribusi kelas dan menghindari bias dalam proses pelatihan model.

### 2.1.3 *Smart Agriculture Applications Using Deep learning Technologies: A Survey*

Penelitian yang dilakukan oleh Maha Altalak dan rekan-rekannya bertujuan untuk mengkaji penerapan teknologi *deep learning* dalam bidang pertanian cerdas, khususnya untuk klasifikasi dan deteksi penyakit tanaman berdasarkan citra daun. Studi ini merupakan survei komprehensif terhadap lebih dari 40 artikel terkini, yang mengadopsi berbagai metode augmentasi data untuk meningkatkan akurasi klasifikasi, terutama dalam kondisi dataset terbatas. Teknik augmentasi yang digunakan antara lain *Random Brightness*, *Contrast Adjustment*, *Gaussian Noise*, *Color Jitter*, rotasi, *flipping*, dan *affine transformations*, yang diaplikasikan pada berbagai model CNN seperti VGGNet, ResNet, dan EfficientNet. Teknik augmentasi ini terbukti mampu meningkatkan performa model secara signifikan, dengan beberapa model mencapai akurasi di atas 99%. Misalnya, penggunaan *Gaussian Noise* dan pengaturan kecerahan meningkatkan ketahanan model terhadap perubahan kondisi pencahayaan dan kualitas citra, sedangkan *Color Jitter* membuat model lebih robust dalam mengidentifikasi gejala

penyakit pada daun dengan warna bervariasi. Studi ini menyimpulkan bahwa strategi augmentasi data merupakan komponen esensial dalam *pipeline* klasifikasi citra pertanian, terutama ketika mengandalkan dataset dari sumber terbatas atau pencitraan UAV beresolusi rendah [10].

Dari penelitian ini, hal yang dijadikan acuan adalah:

- Penerapan teknik augmentasi berbasis *noise*, *blur*, dan *color transform* terbukti efektif dalam meningkatkan performa model ringan dalam kondisi dataset yang tidak seimbang. Oleh karena itu, dalam penelitian ini digunakan pendekatan augmentasi visual seperti *Random Brightness*, *Contrast*, *Gaussian Blur*, dan *Color Jitter* untuk memperkuat representasi kelas minoritas dan meningkatkan generalisasi model dalam proses pelatihan.

#### **2.1.4 *Assessing the Impact of Deep learning on Grey Urban Infrastructure Systems: A Comprehensive Review***

Penelitian yang dilakukan oleh A. Oulahyane, M. Kodad, dan A. Bouazza bertujuan untuk mengevaluasi dampak penerapan model *deep learning* dalam sistem infrastruktur perkotaan, khususnya melalui pendekatan *object detection* menggunakan arsitektur *Faster R-CNN*. Studi ini memberikan tinjauan komprehensif terhadap berbagai metode deteksi objek dalam konteks infrastruktur kota yang kompleks dan tidak terstruktur (*grey infrastructure systems*), seperti jalan, bangunan tua, dan fasilitas publik lainnya.

Evaluasi performa model dalam penelitian ini dilakukan dengan menggunakan metrik yang meliputi *Precision*, *Recall*, *F1-Score*, mean Average *Precision* (mAP), dan *Intersection over Union* (IoU). Penggunaan metrik ini bertujuan untuk memberikan gambaran menyeluruh tentang akurasi deteksi objek serta ketepatan model dalam menangani variasi bentuk dan ukuran objek di lingkungan urban.

Hasil dari tinjauan menunjukkan bahwa *Faster R-CNN* secara konsisten menunjukkan performa tinggi dalam deteksi objek pada citra infrastruktur, terutama ketika dikombinasikan dengan strategi optimisasi arsitektural dan *Preprocessing* yang sesuai. Metrik IoU dan mAP

menjadi indikator kunci dalam mengukur konsistensi dan kualitas prediksi spasial model terhadap objek nyata [11].

Dari penelitian tersebut, hal yang dijadikan acuan adalah:

- Penggunaan metrik evaluasi seperti *Precision*, *Recall*, *F1-Score*, *mAP*, dan *IoU* merupakan pendekatan yang tepat untuk mengukur kinerja model deteksi objek seperti *Faster R-CNN* secara komprehensif. Oleh karena itu, dalam penelitian ini, metrik tersebut digunakan sebagai dasar evaluasi untuk memastikan akurasi dan ketepatan model dalam mendeteksi objek minoritas maupun mayoritas dalam data visual.

### **2.1.5 Image Augmentation in Agriculture Using the Alumentations Library**

Penelitian yang dilakukan oleh Dr. Eray Önler bertujuan untuk meningkatkan akurasi dan generalisasi model pembelajaran mesin dalam bidang pertanian dengan menggunakan teknik augmentasi citra. Salah satu pendekatan utama yang digunakan adalah penerapan pustaka *Alumentations*, yang dikenal karena efisiensinya dan fleksibilitasnya dalam memperluas dataset gambar melalui berbagai transformasi seperti rotasi, *flipping*, pencahayaan, kontras, dan *noise*.

Dalam studi ini, augmentasi dilakukan pada dataset gambar pertanian untuk mendukung tugas klasifikasi dan segmentasi citra tanaman dan hasil panen. *Alumentations* dipilih karena kemampuannya menangani augmentasi kompleks dengan kecepatan tinggi tanpa mengorbankan kualitas citra yang dihasilkan. Transformasi ini membantu menciptakan variasi dalam data latih dan mengurangi risiko *overfitting*.

Penelitian ini tidak hanya menjelaskan berbagai teknik augmentasi yang tersedia dalam pustaka *Alumentations*, tetapi juga menekankan pentingnya pemilihan strategi augmentasi yang sesuai dengan konteks pertanian. Hasilnya menunjukkan bahwa penerapan augmentasi *Alumentations* dapat meningkatkan performa model *deep learning* secara signifikan dalam mendeteksi objek dan klasifikasi gambar berbasis pertanian [12].

Dari penelitian tersebut, hal yang dijadikan acuan adalah:

- *Albumentations* terbukti sebagai pustaka augmentasi citra yang efektif dan efisien dalam meningkatkan kualitas dataset dan performa model, terutama dalam konteks pertanian. Oleh karena itu, dalam penelitian ini, *Albumentations* akan digunakan untuk melakukan augmentasi citra, guna memperkaya dataset pelatihan dan meningkatkan kemampuan generalisasi model dalam mendeteksi objek atau klasifikasi citra pada lingkungan nyata yang bervariasi.

#### **2.1.6 Rethinking Object detection Framework through the Lens of Proposal Refinement**

Penelitian yang dilakukan oleh Jingjing Zhao bertujuan untuk mengevaluasi kembali struktur arsitektur *Faster* R-CNN melalui lensa proposal *refinement* dan menyoroti kontribusi dari berbagai komponen terhadap peningkatan akurasi deteksi objek. Salah satu fokus utama dari studi ini adalah analisis dampak dari penggantian *backbone* serta modul refiner terhadap performa model *Faster* R-CNN. Dalam transformasi bertahap dari *Faster* R-CNN ke *Deformable* DETR, dilakukan penggantian *neck network* dari *Feature Pyramid Network* (FPN) ke *deformable encoder* serta proposal refiner dari RoI Align ke *deformable attention*, yang masing-masing memberikan peningkatan signifikan pada nilai Average Precision (AP), dari 36.1 menjadi 44.0 dan 44.2. Melalui studi ablation, penelitian ini membuktikan bahwa *backbone* yang lebih kuat dan modul refiner yang lebih canggih secara substansial meningkatkan representasi fitur dan akurasi deteksi. Kombinasi *deformable attention*, *dynamic convolution*, dan *regional cross attention* yang diimplementasikan dalam modul *Hybrid Proposal Refiner* (HPR) memberikan peningkatan performa yang konsisten hingga mencapai AP sebesar 54.9 pada *benchmark* COCO [13].

Dari penelitian tersebut, hal yang dijadikan acuan adalah:

- Penggantian *backbone* dan *refiner* dalam arsitektur *Faster* R-CNN secara signifikan berdampak pada performa deteksi objek. Oleh

karena itu, dalam penelitian ini, modifikasi pada *backbone* digunakan sebagai pendekatan utama untuk meningkatkan akurasi model.

## 2.2 Tinjauan Teori

### 2.2.1 Penyakit Garis Kuning

Penyakit garis kuning pada daun kelapa sawit merupakan salah satu ancaman serius yang dapat menurunkan produktivitas tanaman secara signifikan. Penyakit ini tidak hanya berdampak pada pertumbuhan tanaman, tetapi juga dapat mengurangi hasil panen tandan buah segar (TBS) secara drastis. Jika tidak segera ditangani, garis kuning yang muncul pada daun dapat menyebabkan kerusakan permanen, melemahkan tanaman, dan dalam jangka panjang berpotensi menghambat keberlanjutan perkebunan kelapa sawit.

Langkah pertama dalam mengatasi penyakit ini adalah melakukan diagnosis yang akurat agar dapat menentukan metode pengendalian yang tepat. Penyakit garis kuning, yang juga dikenal sebagai *patch yellow*, merupakan masalah umum yang sering dihadapi oleh petani kelapa sawit di Indonesia. Penyakit ini ditandai dengan munculnya garis-garis berwarna kuning yang melintang di permukaan daun, terutama pada daun muda. Seiring waktu, bercak coklat dapat muncul di bagian tengah garis kuning tersebut, menandakan tingkat keparahan infeksi yang semakin meningkat. Gangguan ini dapat menghambat proses fotosintesis, yang berperan penting dalam produksi energi bagi tanaman. Akibatnya, pertumbuhan kelapa sawit menjadi terhambat, dan jika tidak segera ditangani, penyakit ini dapat menyebar ke seluruh bagian daun bahkan menular ke pohon lain di sekitarnya, memperburuk kondisi perkebunan secara keseluruhan.

Penyebab utama munculnya penyakit garis kuning pada tanaman kelapa sawit umumnya berkaitan dengan defisiensi unsur hara tertentu, seperti magnesium dan zat besi. Kekurangan kedua unsur ini dapat melemahkan daya tahan tanaman, membuatnya lebih rentan terhadap serangan penyakit. Selain faktor nutrisi, infeksi jamur *Fusarium*

oxysporum juga berperan dalam penyebaran penyakit ini. Jamur ini dapat menginfeksi jaringan tanaman dan mempercepat perkembangan garis kuning pada daun. Selain itu, kondisi tanah dengan pH yang terlalu rendah juga dapat menjadi faktor pemicu munculnya gejala penyakit ini, karena tanah yang bersifat asam dapat menghambat penyerapan nutrisi penting yang dibutuhkan tanaman untuk tumbuh dengan optimal [4].



Gambar 2.1 Tanaman Kelapa Sawit yang Terserang Penyakit Garis Kuning

Oleh karena itu, untuk mencegah dan mengendalikan penyakit garis kuning pada kelapa sawit, diperlukan strategi manajemen yang komprehensif. Upaya pencegahan dapat dilakukan dengan menjaga keseimbangan unsur hara dalam tanah melalui pemupukan yang tepat, memastikan kondisi lingkungan yang mendukung pertumbuhan tanaman, serta menerapkan teknik budidaya yang baik agar kelapa sawit tetap sehat dan produktif.

### 2.2.2 Pemupukan

Penyakit garis kuning dapat menyebar dengan cepat ke seluruh bagian daun dan bahkan menular ke pohon lain di sekitarnya. Oleh karena itu, salah satu upaya pengendalian yang dapat dilakukan adalah melalui pemupukan yang tepat pada tanaman kelapa sawit. Namun, pemupukan tidak boleh dilakukan sembarangan atau diberikan secara terus-menerus setiap hari. Waktu yang ideal untuk pemupukan adalah saat curah hujan rendah, sedangkan pada musim hujan, pemupukan sebaiknya dihindari. Hal ini dikarenakan air hujan dapat melarutkan pupuk sehingga nutrisi yang seharusnya diserap tanaman justru terbuang sia-sia.



Gambar 2.2 Pupuk yang Sudah diaplikasikan

Agar pemupukan efektif, dosis pupuk harus diberikan sesuai dengan takaran yang telah ditentukan guna mencegah gangguan pada tanaman kelapa sawit yang sehat [14]. Selain dosis yang tepat, pemilihan jenis pupuk juga harus diperhatikan oleh petani kelapa sawit. Unsur-unsur penting yang perlu dipertimbangkan dalam memilih pupuk yang baik meliputi fosfor, kalium, magnesium, tembaga, boron, seng (zink), nitrogen, serta kandungan mikroorganisme yang bermanfaat bagi pertumbuhan tanaman [15].

### 2.2.3 Deep learning

*Deep learning*, merupakan cabang dari *machine learning* yang menggunakan algoritma untuk membangun model abstraksi tingkat tinggi dari data. Proses ini dilakukan melalui serangkaian transformasi

non-linear yang disusun dalam beberapa lapisan yang saling berhubungan. *Deep learning* dapat diterapkan dalam berbagai pendekatan pembelajaran, seperti *supervised learning*, *unsupervised learning*, *semi-supervised learning*, serta *reinforcement learning*. Teknologi ini banyak digunakan dalam berbagai aplikasi, termasuk pengenalan citra dan suara, klasifikasi teks, serta berbagai bidang lainnya.

*Deep learning* memanfaatkan fitur berbentuk hierarki yang dapat disesuaikan dengan kompleksitas permasalahan yang dihadapi. Salah satu keunggulan utama algoritma *deep learning* adalah kemampuannya dalam mengekstraksi fitur secara otomatis dari data mentah dengan tingkat kedetailan yang tinggi. Hal ini dimungkinkan karena *deep learning* menggunakan struktur eksploitasi yang mampu mengidentifikasi pola dan fitur yang tidak terlihat secara kasat mata. Proses ini melibatkan transformasi fitur dari tingkat yang lebih kompleks ke tingkat yang lebih sederhana agar dapat dipahami oleh mesin pembelajaran. Dengan pendekatan ini, *deep learning* menjadi efektif dalam menghasilkan representasi fitur yang optimal. Selain itu, algoritma *deep learning* juga menggabungkan strategi eksploitasi, yang bertujuan untuk menggali fitur secara mendalam, dengan eksplorasi, yang berfokus pada pencarian ruang target atau wilayah yang lebih luas.

Dalam penerapannya, algoritma *deep learning* melibatkan beberapa tahapan penting yang harus diperhatikan. Pertama, diperlukan pemilihan arsitektur yang tepat, yang terdiri dari *input layer*, *hidden layer* (juga dikenal sebagai *darknet* atau lapisan tersembunyi), serta *output layer*. Jumlah dan kompleksitas lapisan ini harus disesuaikan dengan tingkat kesulitan kasus yang akan diolah. Selanjutnya, perlu dilakukan persiapan data untuk proses pelatihan dan pengujian model *deep learning*. Faktor lain yang juga harus dipertimbangkan adalah lingkungan implementasi, apakah dilakukan secara lokal atau menggunakan layanan *cloud*. Selain itu, pendekatan dalam pengembangan kode juga perlu diperhatikan, apakah akan dibuat dari nol

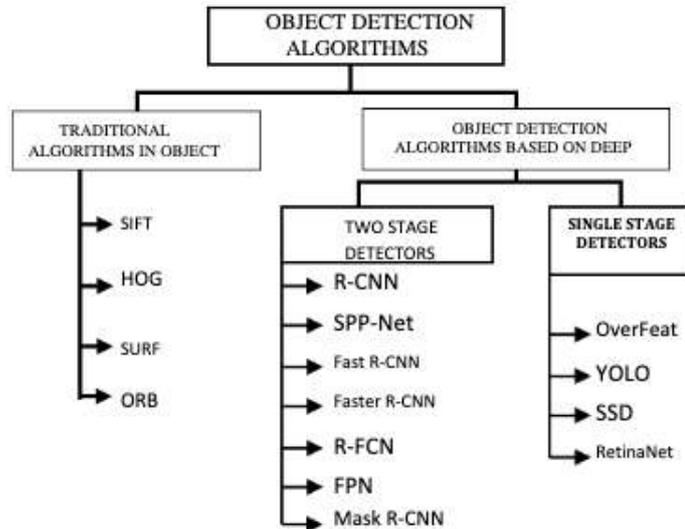
(*from scratch*), dengan menggabungkan beberapa kode dari *library*, atau sepenuhnya menggunakan *library* yang telah tersedia.

Pengujian model harus dilakukan sesuai dengan standar algoritma *deep learning*, mulai dari pemilihan parameter hingga penyesuaian arsitektur. Terdapat dua pendekatan utama dalam pengujian, yaitu metode *waterfall* (proses sekuensial tanpa perulangan) dan metode *recycle* (proses sekuensial dengan perulangan tertentu). Pendekatan *recycle* memungkinkan siklus pelatihan diulang beberapa kali untuk mengoptimalkan parameter hingga diperoleh tingkat akurasi yang stabil. Pada tahap akhir, diperlukan strategi tambahan untuk terus mengoptimalkan algoritma *deep learning* agar performanya semakin meningkat dan sesuai dengan kebutuhan aplikasi yang dikembangkan [16].

#### **2.2.4 Object detection**

*Object detection* atau deteksi objek adalah salah satu tugas dalam *computer vision* yang bertujuan untuk mengidentifikasi dan menemukan objek tertentu dalam gambar atau video. Tujuan utama dari deteksi objek adalah mengembangkan model atau teknik komputasi yang dapat menyediakan informasi esensial bagi aplikasi *computer vision*, seperti jenis objek, lokasi, serta karakteristik lainnya. Dalam beberapa tahun terakhir, penelitian di bidang deteksi objek semakin berkembang dan menjadi topik yang aktif dikaji.

Terdapat berbagai algoritma deteksi objek yang dapat digunakan sesuai dengan kebutuhan spesifik. Secara umum, metode deteksi objek terbagi menjadi dua kategori utama, yaitu metode tradisional dan metode berbasis *deep learning*, sebagaimana ditunjukkan pada Gambar 2.3.



Gambar 2.3 Algoritma *Object detection*

Salah satu pendekatan dalam deteksi objek adalah *two-stage detector*, di mana proses deteksi dilakukan dalam dua langkah. Salah satu algoritma yang termasuk dalam *two-stage detector* adalah *Faster R-CNN* seperti yang ditunjukkan pada gambar 2.3. Algoritma *Faster R-CNN* menggunakan lapisan konvolusional penuh yang disebut RPN (Region Proposal Network) yang berjalan pada gambar acak dengan berbagai ukuran dan mungkin menghasilkan serangkaian proposal di setiap lokasi peta fitur. Vektor fitur yang diperoleh dari peta fitur turunan kemudian dimasukkan ke dalam lapisan klasifikasi, diikuti oleh lapisan regresi kotak pembatas untuk pelokalan objek, yang pada akhirnya mengarah pada deteksi objek. R-CNN yang lebih cepat dapat dianggap sebagai model deteksi *deep learning end-to-end* dan *quasi-real-time* yang pertama [17].

### 2.2.5 *Faster R-CNN*

*Faster R-CNN* (*Region-based Convolutional Neural Network*) adalah algoritma deteksi objek berbasis *deep learning* yang dikenal karena akurasi dan efisiensinya. Algoritma ini merupakan pengembangan dari R-CNN dan *Fast R-CNN* dengan peningkatan utama pada kecepatan deteksi. *Faster R-CNN* bekerja dengan memanfaatkan

*Convolutional Neural Network* (CNN) untuk mengekstraksi fitur dari gambar, kemudian menggunakan *Region Proposal Network* (RPN) untuk menghasilkan wilayah kandidat objek secara otomatis tanpa perlu metode pencarian eksternal seperti *Selective Search*. Setelah itu, wilayah kandidat diproses melalui *Region of Interest* (ROI) *Pooling* agar ukurannya seragam sebelum dikirim ke jaringan klasifikasi dan regresi *bounding box* untuk menentukan jenis objek serta memperbaiki koordinatnya.

Keunggulan utama *Faster R-CNN* terletak pada efisiensinya dalam menghasilkan wilayah kandidat objek secara langsung melalui RPN, yang membuatnya lebih cepat dibandingkan metode R-CNN sebelumnya. Selain itu, algoritma ini memiliki akurasi tinggi dalam mendeteksi berbagai jenis objek, menjadikannya pilihan ideal untuk aplikasi yang membutuhkan presisi, seperti pengenalan wajah, kendaraan, dan analisis citra medis. Pemilihan *Faster R-CNN* dalam penelitian ini didasarkan pada kebutuhan untuk mendeteksi objek daun kelapa sawit yang kompleks, tumpang tindih, dan bervariasi ukurannya dengan tingkat akurasi tinggi. Dibandingkan dengan R-CNN dan *Fast R-CNN*, *Faster R-CNN* menawarkan arsitektur yang lebih efisien dan terintegrasi secara *end-to-end*. Pada R-CNN, proses deteksi dilakukan dengan pendekatan yang terpisah antara ekstraksi fitur, klasifikasi, dan *bounding box regression*, sehingga sangat lambat dan tidak efisien. *Fast R-CNN* memperbaiki hal ini dengan mengintegrasikan proses klasifikasi dan regresi ke dalam satu jaringan, namun masih mengandalkan *Selective Search* untuk menghasilkan region proposal, yang merupakan proses eksternal dan memakan waktu. Sementara itu, *Faster R-CNN* mengatasi kelemahan ini dengan memperkenalkan *Region Proposal Network* (RPN) yang secara otomatis dan efisien menghasilkan proposal area langsung dari feature map, sehingga mempercepat proses dan memungkinkan pelatihan yang lebih stabil dan akurat. Dengan kemampuannya tersebut, *Faster R-CNN* menjadi pilihan yang lebih tepat untuk mendeteksi dan mengklasifikasikan intensitas penyakit daun sawit

dalam citra *drone* yang kompleks . Namun, dibandingkan dengan metode *single-stage detector* seperti YOLO (*You Only Look Once*) atau SSD (*Single Shot MultiBox Detector*), *Faster R-CNN* masih lebih lambat karena proses deteksinya dilakukan dalam dua tahap. Selain itu, implementasinya lebih kompleks dan memerlukan sumber daya komputasi yang lebih besar.

Secara keseluruhan, *Faster R-CNN* adalah algoritma deteksi objek yang sangat andal dan cocok untuk tugas-tugas yang memprioritaskan akurasi dibandingkan kecepatan. Dengan kombinasi antara *Convolutional Neural Network* (CNN) untuk ekstraksi fitur dan *Region Proposal Network* (RPN) untuk menghasilkan area-area calon objek, algoritma ini mampu mengoptimalkan proses deteksi dengan hasil yang lebih presisi dan terstruktur.

Pendekatan *Faster R-CNN* termasuk dalam kategori *two-stage*, di mana proses deteksi dilakukan dalam dua langkah: pertama menghasilkan proposal lokasi objek, lalu mengklasifikasikannya secara lebih mendalam. Hal ini berbeda dengan metode *single-stage* seperti YOLO yang melakukan deteksi dan klasifikasi dalam satu langkah sekaligus. Meskipun pendekatan *single-stage* jauh lebih cepat dan cocok untuk aplikasi *real-time*, akurasinya bisa menurun terutama saat mendeteksi objek kecil, samar, atau saling tumpang tindih. Oleh karena itu, *Faster R-CNN* menjadi pilihan ideal untuk kasus seperti deteksi penyakit pada daun kelapa sawit dari citra *drone*, yang memerlukan presisi tinggi dalam mengenali objek kecil dengan pola visual yang kompleks [18].