

BAB II

TINJAUAN PUSTAKA

2.1 Justifikasi Solusi

2.1.1 Implementasi UNet++ untuk Segmentasi Peta Kawasan Kebun Salak

Penelitian Zaini [7] meletakkan dasar pemrosesan citra drone untuk komoditas salak dengan menerapkan arsitektur UNet++ dan *backbone* MobileNetV2 yang telah dilatih sebelumnya (*pre-trained*) pada ImageNet. Penelitian ini menggunakan strategi *patching* citra berukuran 256×256 piksel dengan *steps* sebesar 220 untuk menangani resolusi tinggi dari citra drone. Dalam evaluasinya, model ini menunjukkan performa yang solid untuk klasifikasi biner (salak vs non-salak), dengan pencapaian nilai *Dice Coefficient* sebesar 0,8361 dan IoU (*Intersection over Union*) sebesar 0,7747. Meskipun akurasinya tinggi, model ini memiliki keterbatasan dalam membedakan objek non-salak secara spesifik, seperti memisahkan antara vegetasi inang alternatif, bangunan, dan objek-objek lainnya.

Acuan yang dapat diambil dari penelitian ini adalah:

- Strategi pemotongan citra (*patching*) berukuran 256×256 piksel dengan *steps* sebesar 220 terbukti efektif mempertahankan detail tekstur tanaman salak tanpa membebani memori komputasi, sehingga metode ini akan diadopsi kembali untuk dataset *multi-class* dalam penelitian ini.
- Penggunaan arsitektur UNet++ dengan *backbone* MobileNetV2 (*pre-trained*) dijadikan *baseline* karena efektivitasnya yang telah teruji untuk segmentasi citra drone perkebunan salak.

- Ketidakmampuan model biner dalam memetakan kompleksitas lingkungan menjadi landasan kuat perlunya pengembangan model baru dengan label ekologis yang lebih rinci (7 kelas).

2.1.2 Implementasi DeepLabV3+ untuk Segmentasi Kebun Salak

Tjandra [10] mengeksplorasi arsitektur DeepLabV3+ dengan membandingkan tiga variasi *backbone*, yaitu Xception, ResNet-101, dan EfficientNet-B3. Hasil eksperimen menunjukkan bahwa konfigurasi DeepLabV3+ dengan *backbone* Xception menghasilkan performa terbaik, mencapai nilai IoU 0,8196, F1-Score 0,8549, dan Recall 0,9089. Sebaliknya, penggunaan *backbone* yang lebih dalam seperti ResNet-101 dan EfficientNet-B3 justru menunjukkan indikasi *overfitting*, di mana selisih antara *training loss* dan *validation loss* cukup signifikan (misalnya selisih 0,12 pada ResNet-101), yang disebabkan oleh ketidakseimbangan antara kompleksitas model yang tinggi dengan variasi data yang terbatas.

Acuan yang dapat diambil dari penelitian ini adalah:

- Kemampuan DeepLabV3+ dalam menangkap konteks spasial melalui modul *Atrous Spatial Pyramid Pooling* (ASPP) dinilai potensial untuk menangani variasi skala objek di perkebunan, sehingga arsitektur ini dipilih sebagai model pembanding serta menjadi kerangka dasar bagi pengembangan metode DiverseNet dalam penelitian ini, berhubung DiverseNet pada dasarnya menggunakan arsitektur DeepLabV3+ dengan *backbone* ResNet-50.
- Temuan mengenai *overfitting* pada model berkapasitas besar (seperti ResNet-101) menegaskan bahwa sekadar memperdalam jaringan tidak efektif jika data terbatas. Hal ini menjadi catatan untuk memilih kompleksitas *backbone*

yang proporsional dengan volume dan kerumitan pola atau tekstur data yang tersedia guna menjaga generalisasi model.

2.1.3 DiverseNet: *Decision Diversified Semi-supervised Semantic Segmentation Networks for Remote Sensing Imagery*

Penelitian ini menggunakan DiverseNet, sebuah *framework Semi-Supervised Learning* (SSL) yang dirancang spesifik untuk karakteristik citra penginderaan jauh. Metode ini menggunakan arsitektur DiverseHead, yaitu satu jaringan *encoder-decoder* tunggal yang dimodifikasi dengan *decision heads* ringan. Dalam publikasi aslinya, metode ini divalidasi secara komprehensif menggunakan empat dataset *benchmark* penginderaan jauh (ISPRS Potsdam, DFC2020, RoadNet, dan Massachusetts Buildings). Hasil evaluasi rata-rata menunjukkan bahwa varian DiverseHead (dengan *Dynamic Freezing*) mampu mencapai performa unggul dengan *Overall Accuracy* (OA) sebesar 0,89 dan *Producer's Accuracy* (PA) sebesar 0,8583, serta mencatatkan nilai mIoU (mean *Intersection over Union*) sebesar 0,7128. Angka ini mengungguli metode kompetitor konvensional seperti Mean Teacher (MT), Cross-Consistency Training (CCT), dan Cross Pseudo Supervision (CPS) [11].

Keunggulan utama DiverseNet terletak pada efisiensi arsitekturnya. Alih-alih menduplikasi seluruh jaringan, arsitektur ini hanya memodifikasi bagian akhir jaringan dengan menambahkan beberapa *decision heads* paralel yang sangat ringan, di mana setiap *head* hanya terdiri dari dua lapisan konvolusi (*2 convolutional layers*). Untuk menciptakan variasi keputusan yang diperlukan dalam pembelajaran *semi-supervised*, metode ini menerapkan strategi perturbasi unik berupa *dynamic freezing* yang secara acak membekukan parameter pada sebagian *head* selama iterasi pelatihan dan *dropout* pada fitur spasial. Hasil prediksi yang beragam dari *multi-head* ini kemudian disatukan melalui mekanisme *dual voting*

(kombinasi *mean voting* dan *max voting*) untuk menghasilkan *pseudo-label* yang robust dan memiliki fidelitas tinggi, sehingga meminimalkan *noise* saat model belajar dari data tidak berlabel [11].

Acuan yang dapat diambil dari penelitian ini adalah:

- Arsitektur *multi-head* memungkinkan model mempelajari representasi fitur yang beragam dari setiap *head*. Hal ini berdasarkan validasi pada dataset ISPRS Potsdam dan DFC2020 yang merupakan dataset segmentasi multi-kelas.
- Strategi perturbasi parameter melalui *dynamic freezing* dan mekanisme *dual voting* menjadi solusi untuk menangani ambiguitas antar-kelas. Mengingat penelitian ini melibatkan 7 label ekologis yang kompleks (seperti membedakan jalan, bangunan, tanaman inang, dan salak), strategi ini mencegah model terjebak pada kesalahan yang berulang dengan cara memvalidasi prediksi melalui *voting* dari berbagai *decision heads*, sehingga menghasilkan *pseudo-label* yang lebih akurat.
- Validitas metode ini telah teruji pada domain citra *remote sensing* yang memiliki karakteristik sudut pandang *top-down* menyerupai citra drone. Efektivitasnya terbukti stabil meskipun menggunakan proporsi data berlabel yang sangat minim, yaitu rasio 1/4 (25%) pada dataset ISPRS Potsdam (total 3.456 data), RoadNet (410 data), dan Massachusetts Buildings (137 data), serta rasio 1/5 (20%) pada dataset DFC2020 yang memiliki volume data lebih besar (6.112 data). Fakta ini menjadi landasan kuat bagi penelitian ini untuk merancang skenario eksperimen dengan efisiensi anotasi tinggi (misalnya mulai dari 5% atau 10% data).

2.1.4 Exploring the Limits of Weakly Supervised Pretraining

Penelitian fundamental oleh Mahajan dkk. (2018) mengeksplorasi batas kemampuan Deep Learning pada dataset berskala besar dengan distribusi kelas yang sangat timpang (*Zipfian/Long-tailed*). Dalam eksperimen yang melibatkan miliaran citra media sosial, mereka menunjukkan bahwa strategi penyeimbangan distribusi menggunakan pendekatan akar kuadrat (*square-root sampling*) menghasilkan performa *transfer learning*, dengan peningkatan akurasi 5-6% dibandingkan metode distribusi alami. Temuan ini menjadi landasan teoritis bagi penerapan metode *Sqrt-Damped Class Weighting* dalam penelitian ini, karena pendekatan berbasis akar kuadrat tersebut terbukti efektif dalam memitigasi dominasi kelas mayoritas pada total *loss*, sekaligus menjaga stabilitas pelatihan dibandingkan dengan penyeimbangan frekuensi linear yang ekstrem [35].

Acuan yang dapat diambil dari penelitian ini adalah:

- Penelitian ini menunjukkan bahwa penggunaan metode pembobotan berbasis akar kuadrat yang diadaptasi dalam penelitian ini, cukup optimal untuk menangani data yang sangat timpang. Dalam konteks kebun salak, pendekatan ini memvalidasi keputusan untuk tidak menggunakan pembobotan linear yang terlalu agresif, melainkan menggunakan versi yang lebih halus (*smoothing* dengan akar kuadrat) agar model dapat memberikan perhatian proporsional pada kelas mayoritas maupun minoritas.
- Pembobotan standar sering kali menyebabkan model mengalami lonjakan perubahan bobot yang drastis saat menemukan kelas yang jarang muncul. Dengan strategi akar kuadrat, lonjakan tersebut diredam sehingga model dapat belajar mengenali objek kecil dengan aman tanpa merusak pengetahuan yang sudah ada mengenai objek dominan.

2.1.5 FlexMatch: *Boosting Semi-Supervised Learning with Curriculum Pseudo Labeling*

Zhang et al. melalui pengembangan algoritma FlexMatch mengidentifikasi bahwa penggunaan ambang batas kepercayaan statis pada metode seperti FixMatch cenderung mengabaikan perbedaan tingkat kesulitan pembelajaran antar kelas yang berbeda. Mereka mengusulkan pendekatan *Curriculum Pseudo Labeling* (CPL) melalui metode FlexMatch, yang secara fleksibel menyesuaikan ambang batas kepercayaan untuk setiap kelas berdasarkan hasil pembelajaran model. Selain meningkatkan akurasi pada skenario label yang sangat terbatas, mekanisme ini terbukti secara signifikan mempercepat kecepatan konvergensi model hingga lima kali lebih cepat dibandingkan metode konvensional [36].

Acuan yang dapat diambil dari penelitian ini adalah:

- Prinsip penyesuaian ambang batas setiap kelas dipakai untuk menggantikan *fixed threshold* pada semua kelas dalam arsitektur DiverseNet, memungkinkan model untuk mempelajari fitur kelas sulit.
- Mekanisme ini digunakan untuk menyeimbangkan kontribusi *pseudo-label* dari data tidak berlabel, mencegah model hanya memperkuat bias pada kelas-kelas yang mudah diprediksi di awal pelatihan.

2.1.6 Penerapan Inductive Semi-Supervised Learning dengan Evaluasi Terpisah

Dalam paradigma *Machine Learning* klasik dengan jumlah sampel terbatas, *rule of thumb* seperti rasio Pareto 80:20 atau 70:30 sering dijadikan standar baku untuk pembagian data latih dan data uji. Namun, seiring dengan evolusi *Deep Learning* dan ketersediaan data berskala masif, Ng [22] menegaskan bahwa ketergantungan pada rasio persentase tetap menjadi tidak relevan dan bahkan kontra-produktif.

Menurut Ng, prioritas utama dalam pembagian data bukan lagi memenuhi kuota persentase tertentu (misalnya 20% untuk pengujian), melainkan memastikan bahwa himpunan data uji memiliki ukuran absolut yang cukup untuk mencapai signifikansi statistik. Pada dataset dengan volume jutaan titik data, alokasi 1% atau bahkan kurang untuk pengujian sudah sangat memadai untuk merepresentasikan distribusi populasi, asalkan jumlah sampel tersebut cukup besar (misalnya >1.000 sampel) untuk memberikan estimasi performa model yang stabil dan tidak bias.

Landasan teoritis mengenai maksimisasi data latih ini divalidasi secara matematis oleh Joseph [29] melalui formulasi rasio pembagian data optimal. Dalam studi statistiknya, Joseph membuktikan bahwa rasio ideal antara himpunan data latih dan data uji mengikuti proporsi $p : 1$, di mana p merepresentasikan jumlah parameter model. Formulasi ini mengimplikasikan hubungan invers di mana proporsi data yang dialokasikan untuk pengujian (γ) harus menurun seiring dengan lonjakan kompleksitas model atau peningkatan nilai p . Joseph menegaskan bahwa perilaku ini merupakan konsekuensi logis dalam pemodelan statistik, mengingat bahwa model dengan kompleksitas tinggi, seperti arsitektur *Deep Learning* yang memuat jutaan parameter, membutuhkan alokasi data pelatihan yang maksimal untuk menjamin akurasi estimasi parameter, sembari mempertahankan data uji pada batas kecukupan statistik semata.

Acuan yang dapat diambil dari penelitian ini adalah:

- Berdasarkan pandangan Ng [22], validitas evaluasi model dalam penelitian ini tidak ditentukan oleh pemenuhan kuota persentase data uji (seperti 20% atau 30%), melainkan oleh kecukupan jumlah absolut sampel untuk mencapai signifikansi statistik. Dalam konteks citra resolusi tinggi, himpunan uji yang secara persentase kecil ($\pm 2\%$) tetap valid secara ilmiah karena proses *patching* menghasilkan ribuan

sampel evaluasi independen yang melampaui ambang batas signifikansi statistik (>1.000 sampel). Hal ini menjustifikasi penggunaan set data uji terpisah dengan jumlah terbatas, tapi representatif demi mengalokasikan sumber daya data yang lebih besar untuk pelatihan.

- Berdasarkan formulasi Joseph [29], penggunaan arsitektur *Deep Learning* dengan kompleksitas parameter tinggi menuntut alokasi data pelatihan yang maksimal. Oleh karena itu, strategi meminimalkan proporsi data uji hingga batas kecukupan statistik adalah langkah optimasi yang terjustifikasi secara matematis untuk menjamin model dapat mempelajari jutaan parameter secara akurat, terutama dalam skenario kelangkaan data berlabel yang dihadapi.
- Pendekatan pembagian data ini selaras dengan tujuan *Inductive Semi-Supervised Learning*, di mana data uji diambil dari kumpulan citra yang dipisahkan sepenuhnya dari proses pelatihan. Hal ini menjamin bahwa metrik akurasi mencerminkan kemampuan model dalam mengenali objek pada citra-citra belum pernah dilihat sebelumnya, memvalidasi bahwa prediksi didasarkan pada pembelajaran fitur morfologi tanaman, bukan sekadar menghafal posisi atau pola visual dari data latih (*overfitting*).

2.2 Tinjauan Teori

2.2.1 *Deep Learning*

Deep learning merupakan cabang dari *machine learning* yang memungkinkan model komputasi yang terdiri dari *multiple processing layers* untuk mempelajari representasi data dengan berbagai tingkat abstraksi [12]. Metode ini telah meningkatkan performa *state-of-the-art* dalam berbagai domain, termasuk pengenalan objek visual dan deteksi objek. Berbeda dengan metode *machine learning* konvensional yang memiliki keterbatasan dalam memproses data

alami dalam bentuk mentahnya, *deep learning* mampu menemukan struktur yang rumit dalam dataset besar.

Mekanisme pembelajaran dalam *deep learning* melibatkan penggunaan algoritma *backpropagation* untuk mengindikasikan bagaimana mesin harus mengubah parameter internalnya guna meminimalkan kesalahan. Parameter yang dapat disesuaikan ini, sering disebut sebagai bobot (*weights*), dimodifikasi untuk mengurangi selisih pada fungsi objektif yang mengukur jarak antara skor *output* dan target yang diinginkan. Untuk tugas klasifikasi citra, lapisan representasi yang lebih tinggi berfungsi memperkuat aspek *input* yang penting untuk diskriminasi dan menekan variasi yang tidak relevan, seperti variasi posisi atau pencahayaan.

Dalam konteks pengolahan citra visual, arsitektur *Deep Convolutional Networks* (ConvNets) telah membawa terobosan besar. ConvNets dirancang khusus untuk memproses data yang hadir dalam bentuk *multiple arrays*, seperti citra warna 2D. Arsitektur ini memanfaatkan empat ide kunci yang mengambil keuntungan dari properti sinyal alami, yaitu *local connections*, *shared weights*, *pooling*, dan penggunaan banyak lapisan [12]. Struktur ini terinspirasi langsung oleh hierarki visual dalam neurosains, di mana fitur tingkat rendah digabungkan membentuk motif, dan motif membentuk bagian objek yang lebih kompleks.

Mengingat kemampuan *deep learning* yang sangat baik dalam menemukan struktur rumit pada data berdimensi tinggi serta meminimalkan kebutuhan rekayasa fitur manual, pendekatan ini menjadi sangat relevan untuk diterapkan dalam penelitian ini. Kompleksitas morfologi tanaman salak atau objek lainnya juga pada citra drone membutuhkan model yang mampu mempelajari fitur invarian secara otomatis, yang mana arsitektur berbasis *deep learning* menawarkan solusi yang lebih *robust* dibandingkan metode ekstraksi fitur konvensional.

2.2.2 Segmentasi

Segmentasi citra merupakan salah satu tugas dalam *computer vision* yang bertujuan untuk mempartisi sebuah gambar digital menjadi beberapa segmen atau *region*. Setiap segmen merepresentasikan objek atau bagian tertentu dari citra, yang kemudian dapat digunakan untuk analisis lebih lanjut seperti deteksi objek dan pengenalan pola. Proses ini penting untuk menyederhanakan representasi citra menjadi sesuatu yang lebih bermakna dan lebih mudah untuk dianalisis.

Dalam praktiknya, objek pada gambar dapat dikategorikan menjadi dua jenis, yaitu *things* dan *stuff*. *Things* merujuk pada objek-objek yang dapat dihitung, seperti pohon, manusia, atau kendaraan. Sementara itu, *stuff* merujuk pada wilayah amorf yang tidak dapat dihitung dan cenderung memiliki tekstur yang seragam, seperti langit, rumput, air, atau tanah [13].

Berdasarkan perbedaan perlakuan terhadap *things* dan *stuff*, tugas segmentasi citra terbagi menjadi tiga kategori utama:

1. Segmentasi Semantik (*Semantic Segmentation*) yang bertujuan untuk memberikan label kelas pada setiap piksel dalam citra. Segmentasi semantik memperlakukan semua piksel sebagai *stuff*, sehingga semua objek dari kelas yang sama akan digabungkan menjadi satu segmen tunggal tanpa adanya pemisahan antar-instansi (misalnya, semua pohon salak dalam satu area dianggap sebagai satu kelas "pohon salak").
2. Segmentasi Instansi (*Instance Segmentation*), berbeda dengan segmentasi semantik, segmentasi instansi berfokus pada deteksi dan segmentasi setiap instansi objek yang termasuk dalam kategori *things*. Tujuannya adalah untuk membedakan satu objek dari objek lainnya, bahkan jika

mereka berasal dari kelas yang sama (misalnya, mengidentifikasi "pohon salak 1", "pohon salak 2", dan seterusnya).

3. Segmentasi Panoptik (*Panoptic Segmentation*), diperkenalkan oleh Kirillov et al. (2019) untuk menggabungkan keunggulan dari segmentasi semantik dan instansi. Segmentasi panoptik memberikan label semantik dan *instance ID* yang unik untuk setiap piksel pada citra. Hasilnya adalah pemahaman adegan yang komprehensif, di mana *stuff* diberi label kelas dan *things* diberi label kelas sekaligus ID unik [14].

Secara metodologis, segmentasi dapat dilakukan dengan teknik tradisional berbasis pemrosesan citra (seperti *thresholding*, *clustering*, atau *watershed*). Namun, seiring berkembangnya *deep learning*, berbagai arsitektur *Convolutional Neural Network* (CNN) seperti *Fully Convolutional Networks* (FCN), U-Net, dan DeepLab telah menunjukkan kinerja yang jauh lebih unggul dalam tugas segmentasi semantik yang kompleks pada citra penginderaan jauh (Zhu et al., 2017) [15].

Mengingat kompleksitas citra *drone* perkebunan salak yang memiliki variasi tekstur, pencahayaan, dan kerapatan tajuk yang tinggi, serta tantangan dalam ketersediaan data berlabel anotasi, penelitian ini akan berfokus pada penerapan dan adaptasi arsitektur *deep learning* untuk segmentasi semantik dengan pendekatan *semi-supervised*.

2.2.3 *Semi-Supervised Learning*

Dalam pengembangan model pembelajaran mesin, pendekatan pelatihan umumnya diklasifikasikan berdasarkan ketersediaan label. Pendekatan yang paling umum, *Supervised Learning*, melatih sistem

menggunakan himpunan data besar di mana setiap contoh memiliki label kategori. Tujuannya adalah meminimalkan kesalahan antara output prediksi dan target yang diinginkan. Namun, pendekatan ini sangat bergantung pada keberadaan data berlabel yang masif. Di sisi lain, *Unsupervised Learning* bertujuan menangkap struktur data tanpa label, menyerupai cara manusia dan hewan belajar melalui pengamatan pasif terhadap dunia [15].

Semi-Supervised Learning (SSL) hadir sebagai solusi untuk mengatasi keterbatasan data berlabel dengan memanfaatkan data tidak berlabel yang melimpah. SSL bekerja dengan menggabungkan sejumlah kecil data berlabel dan sejumlah besar data tidak berlabel untuk meningkatkan kinerja model [16].

Agar data tidak berlabel dapat efektif membantu proses pembelajaran, algoritma SSL bersandar pada tiga asumsi distribusi data fundamental [17]:

1. *Smoothness Assumption*

Jika dua titik data x_1 dan x_2 berdekatan dalam ruang input yang padat, maka output (label) keduanya seharusnya serupa. Asumsi ini menyiratkan bahwa keputusan klasifikasi tidak boleh berubah secara drastis di area yang padat data.

2. *Cluster Assumption*

Titik-titik data cenderung membentuk kelompok atau klaster yang terpisah. Titik-titik yang berada dalam klaster yang sama kemungkinan besar memiliki kelas yang sama. Implikasinya, *decision boundary* model seharusnya melewati daerah dengan kepadatan data rendah, bukan memotong klaster yang padat.

3. *Manifold Assumption*

Data berdimensi tinggi (seperti citra) sebenarnya terletak pada struktur *manifold* berdimensi lebih rendah. Dengan

mempelajari struktur *manifold* ini dari data tidak berlabel, model dapat melakukan generalisasi yang lebih baik meskipun data berlabelnya sedikit.

Berdasarkan asumsi-asumsi di atas, metode SSL modern dalam *Deep Learning* umumnya menerapkan dua mekanisme utama:

A. *Pseudo-Labeling*

Metode ini menerapkan *Cluster Assumption* dengan mendorong model untuk membuat keputusan yang tegas. Model dilatih pada data berlabel, kemudian digunakan untuk memprediksi kelas pada data tidak berlabel. Prediksi dengan *confidence* di atas ambang batas tertentu (misalnya $> 0,95$) dianggap sebagai *pseudo-label* dan digunakan kembali untuk melatih model [18].

B. *Consistency Regularization*

Metode ini menerapkan *Smoothness Assumption*. Teknik ini didasarkan pada prinsip bahwa prediksi model harus konsisten terhadap gangguan (*perturbation*) pada input yang tidak mengubah semantik objek. Jika sebuah citra x diberi gangguan (seperti rotasi atau *noise*) menjadi x' , model dipaksa untuk menghasilkan prediksi yang sama untuk keduanya. Metode modern seperti FixMatch menggabungkan augmentasi lemah untuk pembuatan label semu dan augmentasi kuat untuk menegakkan konsistensi ini [19].

2.2.4 DiverseNet

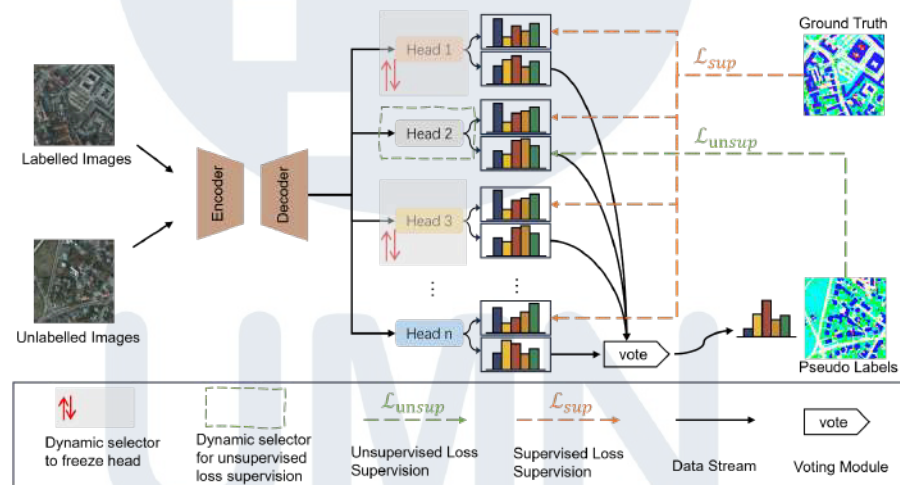
DiverseNet merupakan sebuah *framework semi-supervised learning* yang dikembangkan secara spesifik untuk menangani tantangan segmentasi semantik pada citra *remote sensing*. Metode ini hadir sebagai solusi atas keterbatasan sumber daya komputasi yang

sering dihadapi ketika menerapkan algoritma SSL konvensional berbasis *Teacher-Student* yang cenderung berat. Dalam literatur utamanya, DiverseNet sebenarnya memayungi dua pendekatan strategi yang berbeda, yaitu DiverseModel yang menggunakan pendekatan multi-model untuk sumber daya komputasi tinggi, dan DiverseHead yang dirancang sebagai arsitektur ringan dengan prinsip efisiensi ekstrem. Arsitektur ini memungkinkan model untuk mencapai keragaman keputusan yang tinggi tanpa perlu melatih dua jaringan saraf tiruan secara penuh, melainkan cukup dengan memodifikasi bagian akhir jaringan menjadi cabang-cabang keputusan yang majemuk.

Secara fundamental, arsitektur DiverseNet (dengan implementasi DiverseHead) dibangun di atas struktur *Encoder-Decoder* standar yang kemudian diekspansi pada bagian keluarannya. Aliran pemrosesan data dimulai dari komponen Encoder yang bertugas sebagai *backbone* untuk mengekstraksi representasi fitur mendalam dari citra masukan, baik itu citra berlabel maupun tidak berlabel. Fitur-fitur hasil ekstraksi ini kemudian diteruskan ke komponen Decoder, yang bertanggung jawab untuk memulihkan resolusi spasial fitur tersebut agar sesuai dengan dimensi asli citra. Perbedaan fundamental DiverseNet dibandingkan arsitektur segmentasi standar terletak pada apa yang terjadi setelah tahap *decoder*; alih-alih menggunakan satu lapisan klasifikasi tunggal, arsitektur ini mendistribusikan *feature maps* ke dalam komponen Diverse Heads, yaitu sekumpulan *decision head* yang disusun secara paralel untuk menghasilkan prediksi yang bervariasi dari sumber fitur yang sama.

Untuk menjamin efisiensi komputasi, struktur setiap *decision head* pada DiverseNet dirancang agar bersifat sangat ringan. Setiap *head* hanya terdiri dari dua lapisan konvolusi, sehingga penambahan jumlah *head* tidak akan membebani memori GPU secara signifikan

dibandingkan dengan menduplikasi seluruh jaringan *decoder*. Meskipun memiliki struktur yang sederhana, keragaman keputusan yang dihasilkan oleh masing-masing *head* berbobot. Keragaman ini dicapai melalui mekanisme *network perturbation* yang diterapkan selama proses pelatihan, yaitu strategi *dynamic freezing* dan penggunaan *dropout* pada fitur tingkat tinggi. Dalam mekanisme *dynamic freezing*, parameter pada sebagian *head* akan dibekukan secara acak dan bergantian pada setiap iterasi pelatihan, sementara *dropout* memberikan gangguan stokastik pada representasi fitur. Kombinasi kedua strategi ini memaksa setiap *head* untuk mempelajari representasi yang sedikit berbeda namun saling melengkapi, mencegah terjadinya penyeragaman prediksi yang sering terjadi pada pelatihan model tunggal. Arsitektur DiverseNet berdasarkan yang sudah dijelaskan dapat dilihat pada Gambar 2.1.



Gambar 2.1 Arsitektur *framework* DiverseNet [11]

Dalam konteks pelatihan *semi-supervised*, keberadaan *Multiple Decision Heads* ini dimanfaatkan untuk memperkuat kualitas *pseudo-label*. Data berlabel digunakan untuk melatih setiap *head* secara independen menggunakan fungsi *supervised loss*. Sementara itu, untuk data tidak berlabel, prediksi dari seluruh *head* yang

berbeda-beda tersebut diagregasi untuk menghasilkan satu *pseudo-label* yang lebih robust dan akurat dibandingkan jika hanya mengandalkan satu prediksi tunggal. Melalui pendekatan ini, DiverseNet mampu mengeksplorasi informasi dari data tidak berlabel secara maksimal melalui konsistensi antar-*head*, sekaligus menjaga biaya komputasi tetap rendah karena mayoritas parameter pada *encoder* dan *decoder* digunakan secara bersama, bukan diduplikasi.

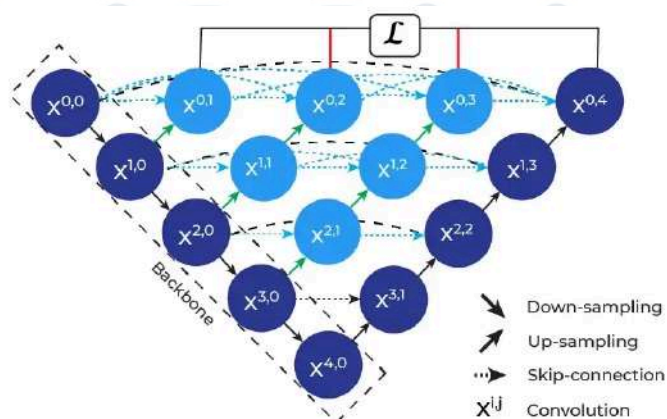
2.2.5 UNet++

UNet++ merupakan pengembangan arsitektur *Convolutional Neural Network* (CNN) berbasis U-Net yang diperkenalkan oleh Zhou et al. (2018) [30]. Arsitektur ini dirancang untuk mengatasi kelemahan pada *skip connections* standar U-Net dengan memperkenalkan strategi koneksi yang didesain ulang, yaitu *Nested Skip Pathways*. Secara visual, arsitektur ini dapat direpresentasikan sebagai susunan node yang membentuk matriks segitiga, di mana baris direpresentasikan oleh indeks i menandakan tingkat *down-sampling* pada *encoder* dan kolom oleh indeks j menandakan tingkat kedalaman blok konvolusi pada *decoder*. Arsitektur UNet++ dapat dilihat pada Gambar 2.2.

Unit dasar pembangun arsitektur ini adalah node $x^{i,j}$. Berbeda dengan U-Net konvensional yang hanya meneruskan fitur dari *encoder* ke *decoder* secara langsung, setiap node pada UNet++ terhubung melalui mekanisme *dense convolution block*. Dalam mekanisme ini, sebuah node $x^{i,j}$ tidak hanya menerima masukan dari lapisan di bawahnya (hasil *up-sampling* dari $x^{\{i,j-1\}}$), tetapi juga menerima akumulasi fitur dari seluruh node *skip pathway* sebelumnya pada baris yang sama (yaitu $x^{i,0}, x^{i,1}, \dots, x^{i,j-1}$). Secara matematis, operasi penggabungan fitur ini memastikan bahwa *semantic gap* antara fitur *encoder* dan *decoder* dapat dijembatani secara bertahap, bukan secara mendadak.

Logika konektivitas ini membuat jumlah lapisan konvolusi pada jalur *skip connection* menjadi adaptif terhadap kedalaman jaringan. Semakin besar nilai indeks j , semakin banyak blok konvolusi yang dilalui oleh fitur sebelum mencapai tahap akhir *decoder*. Sebagai contoh, informasi yang mengalir pada jalur $x^{0,j}$ akan melalui serangkaian transformasi non-linear yang padat sebelum digabungkan dengan fitur hasil *up-sampling*. Proses ini menjamin bahwa fitur yang diterima oleh *decoder* memiliki resolusi spasial yang kaya sekaligus informasi semantik yang telah dimurnikan.

Selain struktur koneksi yang padat, UNet++ juga mengimplementasikan *Deep Supervision*. Fitur ini memungkinkan model menghasilkan luaran segmentasi dari setiap cabang node $x^{0,j}$ (di mana $j \in \{1, 2, 3, 4\}$). Hal ini memberikan fleksibilitas mode operasi pada saat inferensi, mode akurasi yang merata-rata hasil dari seluruh cabang untuk performa maksimal, atau mode cepat yang melakukan *pruning* dengan hanya mengambil luaran dari salah satu cabang awal untuk mempercepat waktu komputasi. Kemampuan ini menjadikan UNet++ sangat adaptif untuk diterapkan pada skenario dengan batasan perangkat keras yang bervariasi.

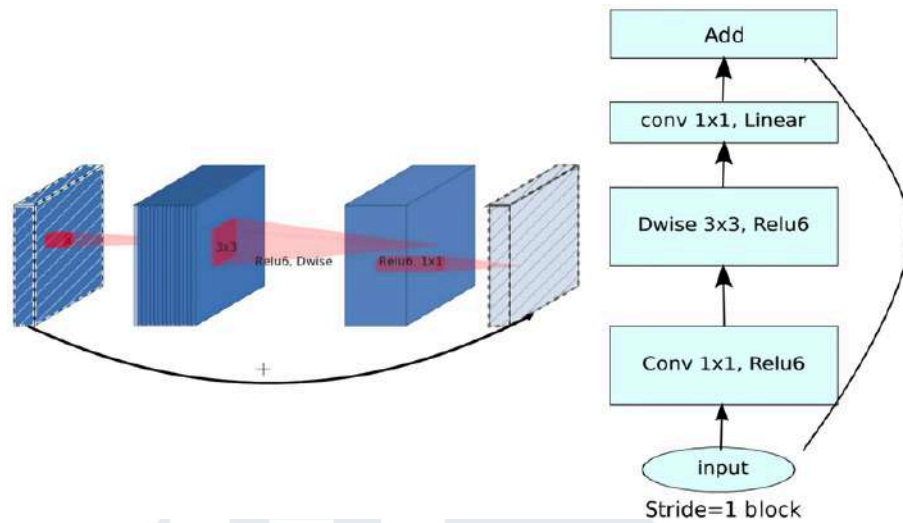


Gambar 2.2 Arsitektur UNet++ [31]

2.2.6 MobileNetV2

MobileNetV2 merupakan arsitektur *Convolutional Neural Network* (CNN) yang diperkenalkan oleh Sandler et al. (2018) dan difungsikan sebagai *backbone* untuk tugas ekstraksi fitur pada sistem dengan sumber daya komputasi terbatas. Arsitektur ini dikembangkan dari pendahulunya, MobileNetV1, dengan tetap mempertahankan efisiensi melalui penggunaan *Depthwise Separable Convolution*. Namun, MobileNetV2 menghadirkan inovasi struktural yang signifikan untuk menangani hilangnya informasi pada operasi non-linier di dimensi rendah, yaitu melalui pengenalan mekanisme *Inverted Residuals* dan *Linear Bottlenecks*. Dalam konteks segmentasi semantik, MobileNetV2 bertugas mengubah citra masukan menjadi representasi fitur yang kaya namun padat, yang kemudian akan diproses lebih lanjut oleh bagian *decoder* atau *head* jaringan [32].

Komponen fundamental yang menyusun struktur *backbone* ini adalah blok *Inverted Residual*. Berbeda dengan arsitektur residual klasik seperti ResNet yang menghubungkan lapisan dengan jumlah kanal besar melalui mekanisme *Wide-Narrow-Wide*, MobileNetV2 membalik logika tersebut menjadi *Narrow-Wide-Narrow*, seperti yang diilustrasikan pada Gambar 2.3. Dalam konfigurasi ini, koneksi *skip* menghubungkan lapisan-lapisan *bottleneck* yang memiliki dimensi rendah. Pendekatan arsitektural ini memungkinkan model untuk mentransmisikan informasi mentah melalui jalur pintas dengan penggunaan memori yang sangat efisien, sementara proses ekspansi fitur untuk mempelajari pola-pola visual yang kompleks dilakukan di dalam blok internal yang memiliki dimensi lebih besar.



Gambar 2.3 Struktur Inverted Residual Block [32]

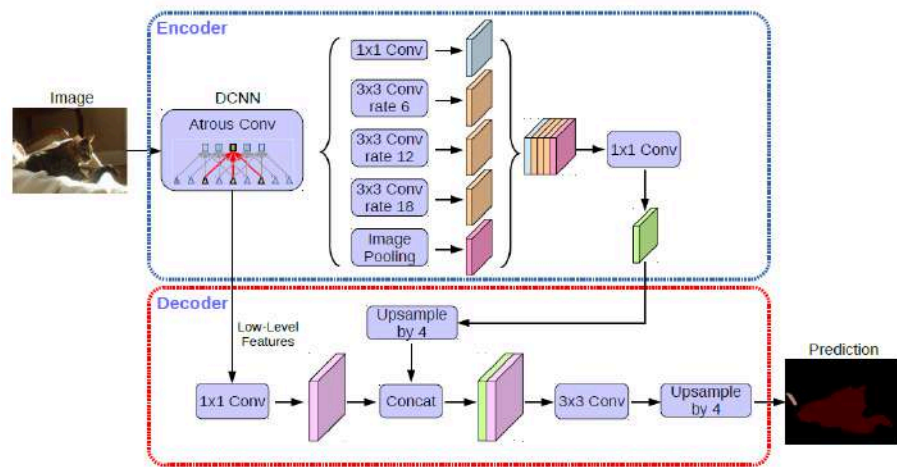
Secara operasional, setiap blok *Inverted Residual* memproses aliran data melalui tiga tahapan konvolusi yang berurutan. Tahap pertama adalah ekspansi, di mana fitur masukan berdimensi rendah diproyeksikan ke dimensi yang lebih tinggi menggunakan konvolusi 1×1 yang diikuti oleh fungsi aktivasi ReLU6. Penggunaan ReLU6, yang membatasi nilai aktivasi maksimum di angka 6, bertujuan untuk menjaga stabilitas numerik saat model dijalankan pada perangkat dengan presisi rendah. Setelah fitur diekspansi, tahap kedua melakukan penyaringan spasial menggunakan konvolusi 3×3 *depthwise*, yang memproses setiap kanal secara terpisah untuk meminimalkan beban komputasi. Tahap terakhir adalah proyeksi kembali ke dimensi rendah menggunakan konvolusi 1×1 . Pada tahap akhir ini, fungsi aktivasi non-linier dihilangkan dan digantikan dengan fungsi *Linear Bottleneck*. Hal ini didasarkan pada temuan bahwa penerapan fungsi non-linier seperti ReLU pada tensor berdimensi rendah berpotensi menghancurkan informasi penting, sehingga linearitas diperlukan untuk mempertahankan keutuhan informasi yang akan diteruskan ke lapisan berikutnya. Rincian spesifikasi operasi dan urutan lapisan pada arsitektur ini dapat dilihat pada Tabel 2.1.

Tabel 2.1 Spesifikasi Arsitektur MobileNetV2

Input	Operator	t	c	n	s
224×3	conv2d	-	32	1	2
112×32	bottleneck	1	16	1	1
112×16	bottleneck	6	24	2	2
56×24	bottleneck	6	32	3	2
28×32	bottleneck	6	64	4	2
14×64	bottleneck	6	96	3	1
14×96	bottleneck	6	160	3	2
7×160	bottleneck	6	320	1	1
7×320	conv2d 1x1	-	1280	1	1
7×1280	avgpool 7x7	-	-	1	-
1×1×1280	conv2d 1x1	-	k	-	-

2.2.7 DeepLabV3+

DeepLabV3+ merupakan varian dari model segmentasi semantik DeepLab yang dikembangkan oleh Google Research. Diperkenalkan oleh Chen et al. [33], model ini dirancang untuk mengatasi keterbatasan pendahulunya, DeepLabV3, dalam merekonstruksi informasi batas objek yang mendetail. Secara fundamental, DeepLabV3+ berstruktur *encoder-decoder* yang bertujuan untuk menyeimbangkan penangkapan konteks global dengan pelestarian resolusi spasial, sebuah mekanisme yang diilustrasikan secara skematis pada Gambar 2.4.



Gambar 2.4 Arsitektur DeepLabV3+ [33]

Mekanisme kerja DeepLabV3+ dimulai pada bagian *Encoder*, di mana tulang punggung (*backbone*) jaringan mengekstrak fitur semantik tingkat tinggi. Pada tahap ini, modul ASPP menerapkan konvolusi berlubang (*atrous convolution*) dengan berbagai tingkat dilasi untuk memperluas *receptive field* tanpa menurunkan resolusi secara drastis. Output dari *encoder* ini mengandung informasi kategori objek yang kuat namun memiliki resolusi spasial yang rendah (biasanya 1/16 dari citra asli).

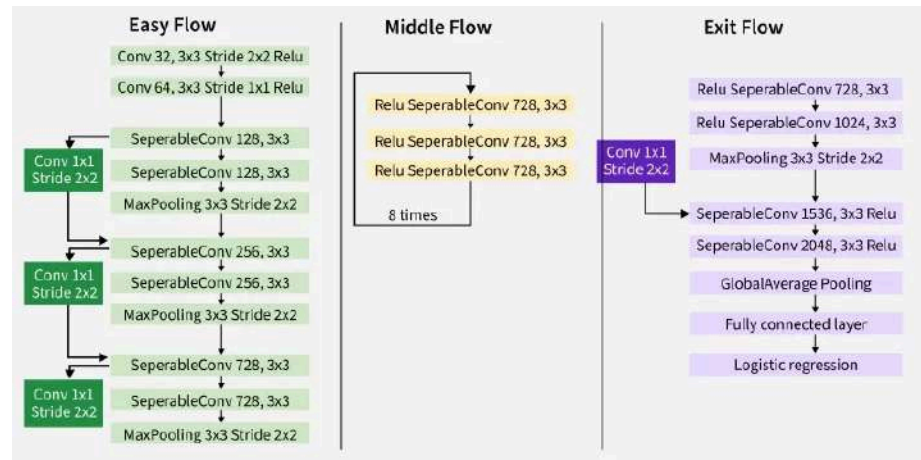
Peran selanjutnya dijalankan oleh modul *Decoder*, yang menjadi fitur pembeda utama arsitektur ini. Alih-alih langsung mengembalikan ukuran citra ke resolusi asli, fitur dari encoder terlebih dahulu di-*upsample* sebesar $4\times$ dan digabungkan dengan fitur tingkat rendah yang diambil dari lapisan awal *backbone*. Fitur tingkat rendah ini dipilih karena masih menyimpan informasi detail spasial yang kaya, seperti tekstur dan garis batas objek. Sebelum penggabungan, fitur tingkat rendah diproses dengan konvolusi 1×1 untuk menyeimbangkan jumlah kanal. Kombinasi antara fitur semantik yang kaya dari *encoder* dan detail spasial dari *decoder* inilah yang

memungkinkan DeepLabV3+ menghasilkan batas segmentasi yang jauh lebih tajam dibandingkan pendahulunya.

Untuk menunjang kompleksitas struktur *encoder-decoder* tersebut tanpa membebani sumber daya komputasi, DeepLabV3+ menggunakan teknik *Depthwise Separable Convolution* (DSC). Teknik ini memodifikasi operasi konvolusi standar menjadi dua tahap terpisah, yaitu konvolusi *depthwise* (per kanal) diikuti oleh konvolusi *pointwise* (1×1). Penerapan DSC, yang terinspirasi dari arsitektur Xception, memungkinkan model untuk mengurangi jumlah parameter dan biaya komputasi (*FLOPs*) secara signifikan sambil mempertahankan performa akurasi yang setara. Efisiensi ini menjadikan DeepLabV3+ sangat relevan untuk diterapkan pada skenario dengan sumber daya terbatas namun tetap menuntut hasil segmentasi vegetasi yang presisi.

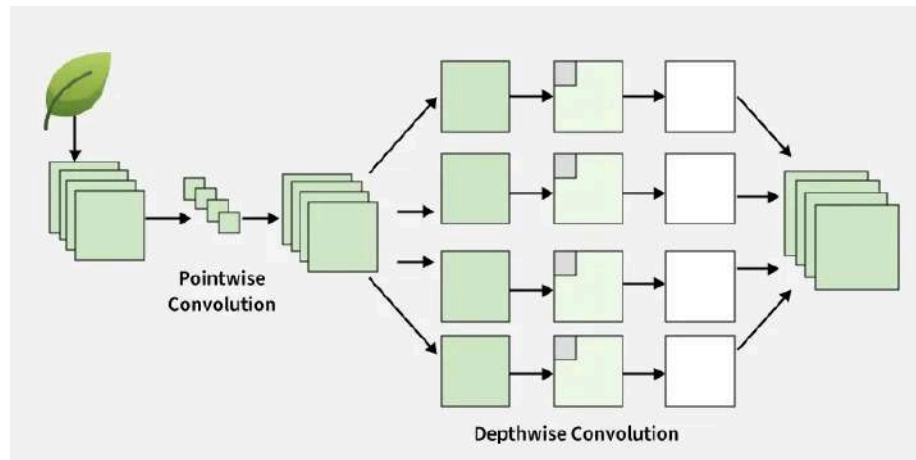
2.2.8 Xception

Xception (*Extreme Inception*) merupakan arsitektur *Convolutional Neural Network* (CNN) yang diperkenalkan oleh François Chollet (2017) [37]. Arsitektur yang dapat dilihat pada Gambar 2.5 ini dirancang sebagai pengembangan evolusioner dari keluarga Inception, dengan tujuan memaksimalkan efisiensi penggunaan parameter tanpa mengorbankan performa model. Nama Xception sendiri berasal dari konsep *Extreme Inception*, di mana arsitektur ini mendorong hipotesis Inception ke titik ekstremnya. Jika modul Inception standar mencoba memisahkan konvolusi spasial dan korelasi lintas-saluran (*cross-channel correlations*) secara parsial, Xception didasarkan pada asumsi yang lebih kuat bahwa pemetaan korelasi lintas-saluran dan korelasi spasial dapat dipisahkan sepenuhnya (*fully decoupled*).



Gambar 2.5 Arsitektur *Xception* [38]

Inovasi fundamental yang membedakan arsitektur *Xception* dari jaringan konvolusi standar seperti VGG atau ResNet awal terletak pada penggunaan *depthwise separable convolution* sebagai pengganti konvolusi konvensional yang dapat dilihat pada Gambar 2.6. Mekanisme ini memecah proses konvolusi menjadi dua tahap terpisah, yaitu *depthwise convolution* dan *pointwise convolution*. Pada tahap *depthwise convolution*, operasi konvolusi spasial (misalnya kernel 3×3) diterapkan secara independen pada setiap channel masukan, sehingga hanya menangkap korelasi spasial tanpa mencampurkan informasi antar-channel. Selanjutnya, *pointwise convolution* menggunakan kernel 1×1 untuk memproyeksikan hasil tersebut ke ruang channel yang baru, memungkinkan model untuk mempelajari korelasi lintas-saluran (*cross-channel correlations*) secara efisien. Pemisahan ini memungkinkan ekstraksi fitur yang lebih hemat parameter dan komputasi, sekaligus mempertahankan kemampuan representasi yang tinggi.



Gambar 2.6 Mekanisme *Depthwise Separable Convolution* [38]

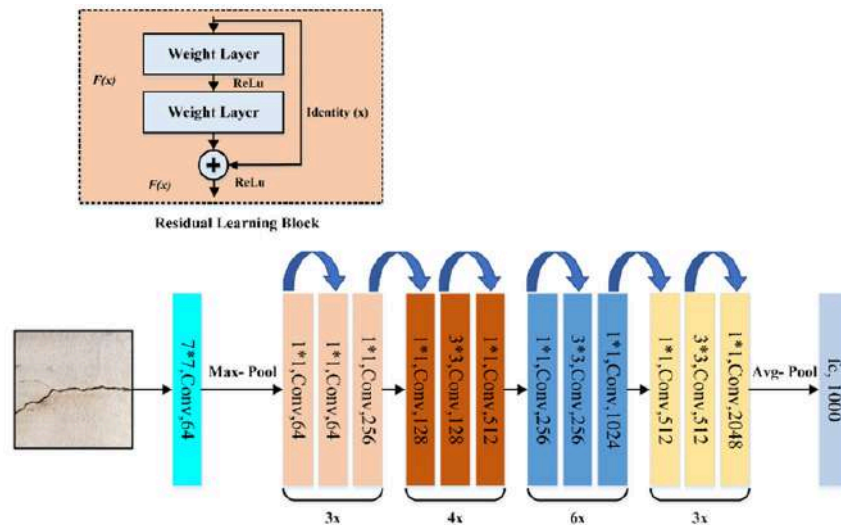
Secara matematis, pemisahan ini secara drastis mengurangi beban komputasi dan jumlah parameter model. Jika konvolusi standar dengan kernel $K \times K$, C_{in} input channel, dan C_{out} output channel membutuhkan operasi sebanyak $K^2 \times C_{in} \times C_{out}$, maka *depthwise separable convolution* hanya membutuhkan $(K^2 \times C_{in}) + (K^2 \times C_{out})$. Pengurangan kompleksitas ini memungkinkan Xception untuk memiliki arsitektur yang sangat dalam namun tetap ringan dan efisien untuk dijalankan, yang sangat krusial dalam pemrosesan citra resolusi tinggi seperti citra drone.

Struktur jaringan Xception tersusun secara linear sebagai tumpukan lapisan *depthwise separable convolution* yang dilengkapi dengan koneksi residual (*residual connections*). Arsitektur ini terbagi menjadi tiga aliran utama. Entry Flow (alur masuk untuk ekstraksi fitur awal dan *downsampling*), Middle Flow (blok pengulangan utama untuk memperdalam representasi fitur), dan Exit Flow (alur keluar untuk konsolidasi fitur akhir sebelum klasifikasi atau rekonstruksi). Penggunaan koneksi residual pada setiap blok konvolusi (mirip dengan ResNet) memastikan aliran gradien yang lancar selama proses

backpropagation, mencegah masalah *vanishing gradient*, dan mempercepat konvergensi pelatihan. Dalam konteks segmentasi semantik, Xception sering diadopsi sebagai *backbone* (enkoder) yang kuat karena kemampuannya mengekstrak fitur hierarkis yang kaya dengan efisiensi komputasi yang superior dibandingkan Inception V3.

2.2.9 ResNet-50

ResNet-50 adalah salah satu varian arsitektur *Residual Network* (ResNet) yang diperkenalkan oleh He et al. [34] sebagai solusi atas fenomena degradasi performa pada pelatihan *Deep Neural Networks* (DNN). Sebelum kemunculan ResNet, memperdalam jaringan saraf sering kali berujung pada masalah *vanishing gradient*, di mana sinyal gradien menjadi sangat kecil saat dipropagasikan balik ke lapisan awal, menyebabkan model gagal belajar secara efektif. Untuk mengatasi hal ini, ResNet memperkenalkan konsep *Residual Learning* melalui mekanisme *skip connection* atau *shortcut connection*, seperti yang ditunjukkan pada Gambar 2.7. Mekanisme ini memungkinkan sinyal input diteruskan secara langsung ke lapisan berikutnya dengan melewati beberapa lapisan konvolusi, sehingga jaringan hanya perlu mempelajari fungsi residual alih-alih fungsi pemetaan utuh dari nol. Pendekatan ini terbukti mempermudah proses optimasi dan memungkinkan pelatihan jaringan yang jauh lebih dalam hingga ratusan lapisan.



Gambar 2.7 *Residual Learning Block* dan *Bottleneck Block* pada ResNet-50 [39]

Secara spesifik, ResNet-50 dibedakan dari varian yang lebih dangkal (seperti ResNet-18 atau ResNet-34) melalui penggunaan struktur blok "Bottleneck". Blok ini dirancang untuk efisiensi komputasi tinggi dengan menyusun tiga lapisan konvolusi berurutan: konvolusi 1×1 , 3×3 , dan 1×1 . Konvolusi 1×1 pertama berfungsi untuk mereduksi dimensi kanal input, konvolusi 3×3 melakukan pemrosesan spasial pada dimensi yang lebih rendah, dan konvolusi 1×1 terakhir mengembalikan dimensi ke jumlah kanal semula. Desain *bottleneck* ini memungkinkan ResNet-50 memiliki kapasitas ekstraksi fitur yang jauh lebih kaya namun dengan kompleksitas parameter yang tetap terkendali dibandingkan jika menggunakan konvolusi 3×3 standar secara terus-menerus.

Dalam implementasi segmentasi semantik, ResNet-50 sering difungsikan sebagai *backbone* atau tulang punggung ekstraktor fitur pada arsitektur *encoder*. Struktur layernya yang terdiri dari lima tahapan (conv1 hingga conv5_x) mampu menghasilkan representasi fitur hierarkis yang kuat, mulai dari deteksi tepi dan tekstur dasar di

lapisan awal hingga pengenalan bentuk objek kompleks di lapisan akhir. Detail konfigurasi lapisan pada arsitektur ResNet-50 dapat dilihat pada Gambar 2.7 di atas, yang menunjukkan susunan filter dan jumlah pengulangan blok residual pada setiap tahapan pemrosesan. Kekuatan representasi fitur inilah yang menjadikan ResNet-50 pilihan standar untuk mendukung tugas *Computer Vision*.

2.2.10 Pest Risk Assessment

Pest Risk Assessment (PRA) merupakan proses fundamental yang menyediakan landasan ilmiah bagi Organisasi Perlindungan Tanaman Nasional (NPPO) dalam menetapkan status hama, mendukung sertifikasi fitosanitari, dan membangun area bebas hama [20], [21]. Dalam konteks surveilans, PRA adalah pendekatan berbasis sains yang digunakan untuk menentukan tata letak dan kepadatan perangkap (*trap density*) agar strategi monitoring berjalan efisien. Mengacu pada *Trapping Guidelines for Area-Wide Fruit Fly Programmes* (FAO/IAEA, 2018), strategi pengendalian hama modern tidak lagi mengandalkan estimasi kasar, melainkan harus didasarkan pada penilaian risiko kuantitatif. Metodologi ini menuntut konversi berbagai faktor lingkungan menjadi bobot nilai numerik terukur hingga mencapai total kumulatif 100 poin. Area dengan akumulasi skor risiko tertinggi mewajibkan pemasangan densitas perangkap yang paling padat, sementara area dengan risiko rendah dapat dikurangi atau ditiadakan, memungkinkan optimalisasi sumber daya monitoring [6]. Contoh tabel PRA dapat dilihat pada Gambar 2.8 di bawah.

Table 5. Risk assessment as a decision-making tool for trap placement and densities

Risk factor	Risk value	Assessed risk		
1. Distance to infested areas	12.0	Area I	Area II	Area III
0–50 Km	7 to 12		12	
51–100 km	4 to 6			
101–150 km	0 to 3	3		3
2. Host availability	20.0			
High	11 to 20		11	
Medium	6 to 10	6		8
Low	0 to 5			
3. Climatic factors (temp., rain, winds)	15.0			
Highly suitable	7.6 to 15			9
Suitable	3.9 to 7.5	5	6	
Unsuitable	0 to 3.8			
4. Host movement	23.0			
Frequent	11.6 to 23		23	
Sporadic	5.9 to 11.5			11
Rear	0 to 5.8	3		
5. Pest historical profile	30.0			
2009–2010	16 to 30		30	
2008–2007	7.6 to 15			10
2006–2004	0 to 7.5	0		
Total	100.0			
High risk: 51–100; medium risk: 26–50; low risk: 0–25		17	82	31
Traps/square kilometer (0 to 2 traps/km²)		0.5	2	1

Gambar 2.8 Contoh Tabel *Pest Risk Assessment* [6]

Berbagai faktor risiko umum yang dikaji meliputi ketersediaan inang (jenis, kelimpahan, dan distribusi spasial dan temporal), preferensi inang (primer dan sekunder), kondisi iklim (seperti suhu, hujan, dan angin), pergerakan inang (komersial dan non-komersial), permukiman manusia (*urban*, *sub-urban*, pedesaan), jarak ke area terinfestasi, dan profil historis kemunculan hama. Penilaian risiko yang berhasil dari faktor-faktor ini akan diplot ke dalam peta untuk menghasilkan peta tematik area risiko yang kemudian digunakan sebagai dasar penyebaran perangkat di lapangan. Pemanfaatan konsep risiko ini telah terbukti efektif dalam restrukturisasi dan mengoptimalkan jaringan penjeratan, memungkinkan penambahan perangkat di area berisiko tinggi dan pengurangan atau penghilangan perangkat di area berisiko rendah. Secara umum, kepadatan perangkat ditentukan berdasarkan tingkat risiko yang dinilai, di mana

area berisiko tertinggi memerlukan kepadatan perangkap yang paling tinggi [6].

2.2.11 *Fruit Fly Trap*

Sistem penjeratan lalat buah merupakan alat manajemen hama yang sangat terspesialisasi dan efisien, menjadi prasyarat untuk pengambilan keputusan yang efektif dalam program pengendalian *area-wide*. Penggunaan perangkap yang efektif bergantung pada kombinasi yang tepat antara perangkat perangkap, atraktan, dan agen pembunuh/pengawet untuk menarik, menangkap, membunuh, dan mengawetkan spesies lalat target untuk dianalisis. Perangkap diklasifikasikan menjadi tiga jenis utama, di antaranya yaitu perangkap kering, perangkap basah, dan perangkap yang dapat digunakan kering atau basah. Untuk surveilans hama genus *Bactrocera*, digunakan Methyl Eugenol (ME), yaitu paraferomon spesifik jantan (*male-specific lure*), yang efektif menangkap spesies seperti *B. dorsalis* dan *B. carambolae*. Karena ME merupakan atraktan yang sangat mudah menguap, penggunaannya dalam perangkap kering harus dikombinasikan dengan toksikan, seperti *malation* atau *deltamethrin*, sebagai agen pembunuh [6], [21].

Salah satu perangkat perangkap kering yang umum digunakan untuk spesies *Bactrocera* dan atraktan ME adalah Steiner Trap (ST). Perangkap Steiner konvensional dideskripsikan sebagai silinder plastik bening horizontal dengan bukaan di setiap ujungnya. Atraktan berupa *cotton wick* atau dispenser digantung dari bagian tengah di dalam perangkap. Meskipun perangkap Steiner termasuk perangkap standar yang direkomendasikan, perangkap ini merupakan salah satu perangkap yang paling ekonomis dan mudah ditangani di lapangan, yang memungkinkan operator melayani jumlah perangkap yang lebih banyak per jam kerja dibandingkan perangkap jenis lain [6], [21].

Penempatan perangkap yang efektif di lapangan memerlukan informasi dasar mengenai inang lalat buah, yang meliputi inang primer, inang sekunder, dan inang sesekali, serta fenologi, distribusi, dan kelimpahannya. Perangkap feromon harus ditempatkan di area perkawinan, yaitu di bagian tengah hingga atas kanopi tanaman inang, memilih tempat yang semi-teduh, dan idealnya di sisi *upwind* mahkota pohon. Selain itu, perangkap harus ditempatkan di pohon inang primer (misalnya, pohon salak) selama periode pematangan buah, dan jika inang primer tidak tersedia, inang sekunder harus digunakan. Strategi penempatan harus mencakup relokasi perangkap secara sistematis dengan mengikuti fenologi pematangan buah inang yang ada di area tersebut agar populasi lalat buah dapat dilacak sepanjang tahun. Sangat penting juga untuk memastikan pintu masuk perangkap bersih dari ranting, daun, atau penghalang lain seperti sarang laba-laba, untuk memungkinkan aliran udara yang tepat dan akses mudah bagi lalat buah. Penempatan beberapa perangkap dengan atraktan berbeda (misalnya ME dan atraktan berbasis protein) di pohon yang sama harus dihindari karena dapat menyebabkan interferensi antar atraktan dan mengurangi efisiensi penangkapan. Terakhir, untuk pengelolaan yang efisien, lokasi setiap perangkap wajib di-*geo-reference* menggunakan GPS dan dicatat dalam database GIS, yang menyediakan peta beresolusi tinggi untuk perencanaan dan pengawasan kegiatan surveilans [6], [21], [41].

2.2.12 *Fruit Fly per Trap per Day* (FTD)

Flies per Trap per Day (FTD) merupakan indeks populasi standar yang digunakan secara internasional untuk mengukur kepadatan populasi lalat buah dewasa di suatu wilayah dalam periode waktu tertentu,. Indeks ini berfungsi sebagai data dasar yang bersifat komparatif untuk mengevaluasi fluktuasi populasi sebelum, selama, dan setelah pelaksanaan program manajemen hama, baik dalam skala

supresi maupun eradikasi,. Secara matematis, FTD dihitung dengan membagi total lalat yang tertangkap dengan hasil perkalian antara jumlah perangkat yang diperiksa dan rata-rata durasi paparan perangkat di lapangan,. Rumus perhitungan FTD adalah sebagai berikut:

$$FTD = \frac{T}{D \cdot F}$$

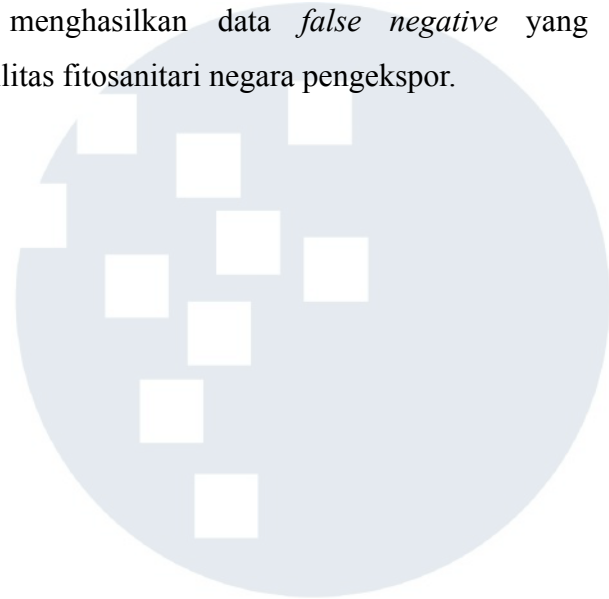
Keterangan:

- F: Total jumlah lalat buah target yang tertangkap
- T: Jumlah perangkat yang diperiksa
- D: Rata-rata jumlah hari perangkat dipasang di lapangan di antara dua waktu pemeriksaan

Dalam praktik manajemen hama terpadu area luas (AW-IPM), FTD digunakan sebagai ambang batas tindakan untuk pengambilan keputusan operasional. Apabila nilai FTD melampaui ambang batas ekonomi tertentu, maka tindakan pengendalian supresif, seperti penyemprotan umpan protein atau teknik jantan mandul, harus segera diintensifkan untuk mencegah ledakan populasi. Sebaliknya, tren penurunan nilai FTD menjadi indikator keberhasilan intervensi yang telah dilakukan. Bagi Paguyuban Mitra Turindo, pencapaian indeks FTD yang konsisten di angka harapan 0,00 sangat penting sebagai bukti ilmiah yang diakui oleh WTO. Data ini menjadi modal vital dalam diplomasi perdagangan untuk menegosiasikan penurunan beban sampling inspeksi karantina yang saat ini mencapai 10%, sehingga dapat meningkatkan daya saing ekspor dan kesejahteraan petani.

Lebih jauh lagi, dalam kerangka perdagangan internasional, FTD bukan sekadar angka statistik, melainkan instrumen legalitas ekspor. Untuk komoditas bernilai tinggi seperti salak yang menargetkan pasar dengan regulasi karantina ketat (seperti Tiongkok atau Australia), pemeliharaan nilai FTD pada level yang sangat rendah (mendekati atau sama dengan nol) adalah prasyarat mutlak. Status Area Bebas

Hama (PFA) atau Area dengan Prevalensi Hama Rendah (ALPP) hanya dapat diakui secara internasional apabila didukung oleh data historis surveilans yang menunjukkan konsistensi nilai FTD di bawah ambang batas yang disepakati dalam protokol bilateral selama periode waktu tertentu [6], [21]. Oleh karena itu, akurasi dalam penempatan perangkat, yang didukung oleh peta risiko presisi, sangat menentukan validitas nilai FTD yang dihasilkan, karena penempatan yang salah dapat menghasilkan data *false negative* yang membahayakan kredibilitas fitosanitari negara pengekspor.



UMN
UNIVERSITAS
MULTIMEDIA
NUSANTARA