DIGITAL REPOSITORY UNIVERSITAS JEMBER

INTEGRASI DOMINATING SET DALAM PENERAPAN YOLO UNTUK IDENTIFIKASI PENYAKIT VSD PADA TANAMAN KAKAO

diajukan untuk memenuhi sebagian persyaratan memperoleh gelar Magister pada program studi Matematika.

TESIS

Oleh

Annisa Fitri Maghfiroh Harvyanti
201820101006

KEMENTERIAN PENDIDIKAN, KEBUDAYAAN, RISET, DAN TEKNOLOGI
UNIVERSITAS JEMBER
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
PROGRAM STUDI MATEMATIKA
2024

DIGITAL REPOSITORY UNIVERSITAS JEMBER

DIGITAL REPOSITORY UNIVERSITAS JEMBER

PERSEMBAHAN

Dengan menyebut nama Allah yang Maha Pengasih dan Maha Penyayang serta sholawat dan salam selalu tercurahkan kepada Nabi Muhammad SAW, penulis persembahkan tesis ini sebagai ungkapan kebahagiaan dan rasa terima kasih kepada:

1. Keluarga saya yaitu Ibunda Sylvie Nurul Damayanti, Ayahanda Hari Purnomo, dan Adik Ilham Hanif Firdausyah yang telah mendukung dan senantiasa memberikan doa, kasih sayang, motivasi, kepercayaan dan senyuman yang selalu menguatkan di setiap perjalanan hidup saya;
2. Dr. Ika Hesti Agustin, S.Si., M.Si. selaku Dosen Pembimbing Utama dan Prof. Bayu Taruna Widjaja Putra, S.TP., M.Eng., Ph.D. selaku Dosen Pembimbing Anggota yang telah meluangkan banyak waktu dan energinya untuk memberikan banyak arahan, bimbingan, perhatian, serta bantuannya dalam penulisan tesis;
3. Sahabat-sahabat yang selalu mendukung dan memotivasi serta sahabat seperjuangan mahasiswa prodi Magister Matematika Fakultas MIPA yang telah menemani dan memberikan dukungan;

DIGITAL REPOSITORY UNIVERSITAS JEMBER

DIGITAL REPOSITORY UNIVERSITAS JEMBER

MOTTO

"If you want to be happy, help others without a reason"

"Everytime you give up on something for the sake of Allah,
He will replace it with something better"

- Ali Banat -

"Keberhasilan bukanlah milik orang yang pintar.
Keberhasilan adalah kepunyaan mereka yang senantiasa berusaha."

- Bahruddin Jusuf Habibie -

DIGITAL REPOSITORY UNIVERSITAS JEMBER

DIGITAL REPOSITORY UNIVERSITAS JEMBER

PERNYATAAN ORISINALITAS

Saya yang bertanda tangan di bawah ini :

Nama : Annisa Fitri Maghfiroh Harvyanti

NIM : 211820101006

Menyatakan dengan sesungguhnya bahwa skripsi yang berjudul: *Integrasi Dominating Set dalam Penerapan YOLO untuk Identifikasi Penyakit VSD pada Tanaman Kakao* adalah benar-benar hasil karya sendiri, kecuali jika dalam pengutipan substansi disebutkan sumbernya, dan belum pernah diajukan pada institusi manapun, serta bukan karya jiplakan. Saya bertanggung jawab atas keabsahan dan kebenaran isinya sesuai dengan skripsi ilmiah yang harus dijunjung tinggi.

Demikian pernyataan ini saya buat dengan sebenarnya, tanpa adanya tekanan dan paksaan dari pihak manapun serta bersedia mendapat sanksi akademik jika ternyata di kemudian hari pernyataan ini tidak benar.

Jember, 23 Januari 2024

Yang menyatakan,

Materai 10.000

Annisa Fitri Maghfiroh Harvyanti

NIM 211820101006

DIGITAL REPOSITORY UNIVERSITAS JEMBER

DIGITAL REPOSITORY UNIVERSITAS JEMBER

HALAMAN PERSETUJUAN

Tesis berjudul *Integrasi Dominating Set dalam Penerapan YOLO untuk Identifikasi Penyakit VSD pada Tanaman Kakao* telah diuji dan disahkan oleh Fakultas Matematika dan Ilmu Pengetahuan Universitas Jember pada:

Hari : Selasa

Tanggal : 23 Januari 2024

Tempat : Fakultas Matematika dan Ilmu Pengetahuan Universitas Jember

Pembimbing — Tanda Tangan

1. Pembimbing Utama
   Nama : Dr. Ika Hesti Agustin, S.Si., M.Si. (…….)
   NIP : 198408012008012006
2. Pembimbing Anggota
   Nama : Prof. Bayu Taruna Widjaja Putra, S.TP., M.Eng., Ph.D. (…….)
   NIP : 198410082008121002

Penguji

1. Penguji I
   Nama : Dr. Kristiana Wijaya, S.Si., M.Si. (…….)
   NIP : 197408132000032004
2. Penguji II
   Nama : Drs. Moh. Hasan, M.Sc., Ph.D. (…….)
   NIP : 196404041988021001

DIGITAL REPOSITORY UNIVERSITAS JEMBER

DIGITAL REPOSITORY UNIVERSITAS JEMBER

ABSTRACT

This research is an application of machine learning in the field of precision agriculture. It aims to obtain an object detection model to detect VSD disease in cocoa plants. The dataset used is primary data, which consists of 2 classes, namely the healthy class and the VSD class, with 1250 images in each class. The method used is the YOLOv1-YOLOv8 network model. Each YOLO network is applied in the training data process with 4000 iterations. The mean average precision (mAP) value and the number of iterations where the early stopping point (ESP) occurs in each YOLO network model are used as model evaluations. The best results were obtained from the YOLOv5 and YOLOv8 network models. YOLOv5 has the highest mAP value of 99.006% with an early stop point (ESP) in the 75th iteration, while YOLOv8 has the highest mAP value of 98.929% with ESP in the 20th iteration. The results of the model analysis show that YOLOv8 is superior to YOLOv5 because in the 20th iteration, YOLOv8 has achieved ESP, while YOLOv5 has achieved ESP in the 75th iteration, with a MAP value that is not much different. The next stage is to identify the level of spread of VSD disease in cocoa plants in a field. The graph dominating set theory is used. Field is represented as a king graph $P_m \boxtimes P_n$. Based on the results of the observations, the severity level on land 1 was low, land 2 and land 3 was high so the treatment must be carried out immediately on land 2 and 3.

Keywords: object detection, VSD cocoa disease, YOLO, graph dominating set, severity plant disease

DIGITAL REPOSITORY UNIVERSITAS JEMBER

DIGITAL REPOSITORY UNIVERSITAS JEMBER

RINGKASAN

Integrasi *Dominating Set* dalam Penerapan YOLO untuk Identifikasi Penyakit VSD pada Tanaman Kakao; Annisa Fitri Maghfiroh Harvyanti; 211820101006; 2024; 60 halaman; Program Studi S2 Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Jember.

Penelitian ini mengeksplorasi aplikasi deteksi objek menggunakan YOLOv1 sampai YOLOv8. Model jaringan YOLO diimplementasikan dalam konteks pertanian untuk mendeteksi penyakit VSD pada daun tanaman kakao. Penyakit VSD merupakan salah satu penyakit paling mematikan pada tanaman kakao karena penyakit ini menginfeksi pembuluh kayu dan penyebarannya melalui angin sehingga cepat menyebar pada suatu lahan. Masalah deteksi penyakit VSD pada kakao menjadi hal yang krusial sehingga dalam penelitian ini model objek detektor YOLO diimplementasikan untuk mengatasi masalah tersebut. Performa dari tiap model jaringan YOLO, yaitu YOLOv1 sampai dengan YOLOv8 dibandingkan berdasarkan nilai *mean Average Precision* (mAP). Model dengan performa paling baik kemudian diterapkan untuk mendeteksi penyakit VSD pada kakao di suatu lahan.

Tujuan dari penelitian ini adalah membandingkan dan mendapatkan model objek detektor YOLOv1 sampai dengan YOLOv8 yang dapat mendeteksi penyakit VSD dan menganalisis tingkat sebaran penyakit VSD pada suatu lahan. Dataset gambar kakao yang digunakan berjumlah 2500 merupakan data primer yang diambil menggunakan kamera berkapasitas 48 MP.

Proses training data dengan model jaringan YOLOv1 sampai dengan YOLOv8 menggunakan jumlah iterasi yang sama, yaitu sebanyak 4000. Hasil dari proses training data diperoleh nilai mAP terbaik pada masing-masing dari YOLOv1 sampai YOLOv8 sebesar 46.41% pada iterasi ke-2050, 40.59% pada iterasi ke-2390, 95.95% pada iterasi ke-2350, 98.61% pada iterasi ke-1200, 99.006% pada iterasi ke-200, 98.16% pada iterasi ke-117, 98.4% pada iterasi ke-225, dan 98.929% pada iterasi ke-172.

DIGITAL REPOSITORY UNIVERSITAS JEMBER

DIGITAL REPOSITORY UNIVERSITAS JEMBER

Diperoleh kesimpulan bahwa model jaringan YOLOv5 dan YOLOv8 merupakan metode yang paling baik. YOLOv5 menghasilkan nilai mAP tertinggi sebesar 99.006% dengan *early stopping point* (ESP) pada iterasi ke-75, sedangkan pada YOLOv8 menghasilkan nilai mAP tertinggi sebesar 98.929% dengan ESP pada iterasi ke-20. Proses training YOLOv8 lebih cepat dalam mencapai ESP dan dapat mendeteksi daun sehat dan VSD pada video dengan baik. Oleh karena itu, YOLOv8 dipilih menjadi model terbaik untuk selanjutnya diterapkan dalam mendeteksi penyakit VSD pada pohon anggota himpunan dominasi. Hasil deteksi tersebut digunakan untuk mengidentifikasi tingkat sebaran penyakit VSD pada lahan. Berdasarkan hasil pengamatan, didapat tingkat penyebaran VSD pada lahan 1 cukup rendah, tetapi tindakan pencegahan penyebaran tetap perlu dilakukan. Pada lahan 2 penyebaran VSD cukup parah karena tingkat penyebarannya melebihi setengah lahan sehingga langkah penanganan harus segera dilakukan. Pada lahan 3 tingkat penyakit VSD menyebar hampir setengah lahan sehingga perlu dilakukan tindakan penanganan.

DIGITAL REPOSITORY UNIVERSITAS JEMBER

DIGITAL REPOSITORY UNIVERSITAS JEMBER

PRAKATA

Puji syukur atas kehadirat Allah SWT yang telah memberi limpahan rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan tesis dengan berjudul "*Integrasi Dominating Set dalam Penerapan YOLO untuk Identifikasi Penyakit VSD pada Tanaman Kakao*". Tesis ini disusun untuk memenuhi salah satu syarat menyelesaikan pendidikan magister pada Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Jember.

Penyusunan tesis ini tidak lepas dari bantuan berbagai pihak yang terlibat. Oleh karena itu, penulis menyampaikan terimakasih kepada:

1. Dr. Alfian Futuhul Hadi, S.Si., M.Si. selaku Dosen Pembimbing Akademik yang telah membimbing selama menjadi mahasiswa S2;
2. Dr. Ika Hesti Agustin, S.Si., M.Si. dan Prof. Bayu Taruna Widjaja Putra, S.TP., M.Eng., Ph.D. selaku Dosen Pembimbing yang telah meluangkan banyak waktu dan energinya untuk memberikan banyak arahan, bimbingan, perhatian, serta bantuannya dalam penulisan tesis;
3. Dr. Kristiana Wijaya, S.Si., M.Si. dan Drs. Moh. Hasan, M.Sc., Ph.D. selaku Dosen Penguji yang telah memberikan kritik dan saran yang membangun demi kesempurnaan tesis;
4. seluruh dosen dan karyawan Jurusan Matematika FMIPA Universitas Jember;
5. keluarga tercinta diantaranya Mama, Ayah, dan adek yang senantiasa memberikan do'a, support, dan motivasi;
6. sahabat-sahabat yang selalu mendukung dan memotivasi serta sahabat seperjuangan mahasiswa S2 Matematika murni dan pendidikan yang telah memberikan dukungan;
7. semua pihak yang tidak dapat disebutkan satu per satu.

Penulis menerima segala bentuk kritik dan saran dari semua pihak demi kesempurnaan tesis ini. Penulis berharap semoga tesis ini dapat bermanfaat.

Jember, 23 Januari 2023

Penulis

DIGITAL REPOSITORY UNIVERSITAS JEMBER

DIGITAL REPOSITORY UNIVERSITAS JEMBER

DAFTAR ISI

DIGITAL REPOSITORY UNIVERSITAS JEMBER

DIGITAL REPOSITORY UNIVERSITAS JEMBER

DIGITAL REPOSITORY UNIVERSITAS JEMBER

DIGITAL REPOSITORY UNIVERSITAS JEMBER

DAFTAR TABEL

DIGITAL REPOSITORY UNIVERSITAS JEMBER

DIGITAL REPOSITORY UNIVERSITAS JEMBER

DAFTAR GAMBAR

DIGITAL REPOSITORY UNIVERSITAS JEMBER

DIGITAL REPOSITORY UNIVERSITAS JEMBER

DAFTAR LAMPIRAN

DIGITAL REPOSITORY UNIVERSITAS JEMBER

DIGITAL REPOSITORY UNIVERSITAS JEMBER

DAFTAR ISTILAH DAN SINGKATAN

Singkatan/Istikal	Arti dan keterangan
AI	*Artificial Intelligence*
ANN	*Artificial Neural Network*
CNN	*Convolutional Neural Network*
ESP	*Early Stopping Point*
mAP	*mean Average Precision*
MSE	*Mean Squared Error*
NLP	*Natural Language Processing*
NMS	*Non-Maximum Supression*
R-CNN	*Region-based CNN*
ROI	*Region of Interest*
SSD	*Single Shot Detector*
VSD	*Vascular Streak Disease*
YOLO	*You Only Look Once*

DIGITAL REPOSITORY UNIVERSITAS JEMBER

DIGITAL REPOSITORY UNIVERSITAS JEMBER

BAB 1. PENDAHULUAN

Bagian ini berisi tentang latar belakang dilakukannya penelitian, rumusan masalah, batasan, manfaat, dan tujuan penelitian.

1.1 Latar Belakang

Pengembangan teknologi deteksi objek merupakan aspek penting dalam berbagai bidang, seperti pertanian presisi, pengenalan wajah, kendaraan, dan sistem keamanan. Dalam beberapa tahun terakhir, terjadi lonjakan inovasi di bidang ini dengan munculnya berbagai model objek detektor yang semakin canggih. Model-model ini memiliki karakteristik yang berbeda, termasuk akurasi, kecepatan inferensi, dan kemampuan mengatasi tantangan lingkungan yang kompleks. Oleh karena itu, penelitian yang membandingkan performa model objek detektor menjadi krusial untuk menjadi acuan pemilihan model yang optimal dalam berbagai konteks aplikasi. Dengan adanya perbandingan model objek detektor, memungkinkan peneliti untuk memilih model yang paling sesuai dengan kebutuhan spesifik tugas deteksi objek. Selain itu, pemilihan model yang tepat juga dapat memberikan dampak positif terhadap efisiensi, keandalan, dan akurasi sistem yang mengimplementasikan teknologi deteksi objek.

Model objek detektor berbasis *Convolutional Neural Network* (CNN) telah menjadi landasan penting dalam kemajuan *computer vision* dan pengenalan objek. Salah satu yang terkenal adalah *Region-based* CNN (R-CNN), yang diperkenalkan oleh Ross Girshick dan tim pada tahun 2014 (Girshick et al., 2014). R-CNN menyajikan pendekatan inovatif dengan menggabungkan algoritma *selective search* untuk menghasilkan *region proposal* dengan model CNN untuk mengidentifikasi dan mengklasifikasikan objek dalam setiap proposal. Pada tahun 2015, *Faster* R-CNN dikembangkan oleh Shaoqing Ren dan timnya (Ren et al., 2016) yang mulai memperkenalkan *Region Proposal Network* (RPN) sebagai bagian dari arsitektur, menyatukan tahap proposal, dan pengenalan objek. Dengan

DIGITAL REPOSITORY UNIVERSITAS JEMBER

DIGITAL REPOSITORY UNIVERSITAS JEMBER

melakukan ini, Faster R-CNN berhasil meningkatkan kecepatan dan akurasi deteksi objek secara signifikan. Pada tahun berikutnya, model *You Only Look Once* (YOLO) pertama kali diusulkan oleh Joseph Redmon dan Santosh Divvala dalam paper yang berjudul "*You Only Look Once: Unified, Real-Time Object Detection*" (Redmon et al., 2016). YOLO membawa terobosan baru dengan menyelesaikan masalah deteksi objek dengan mengubahnya menjadi masalah regresi yang dapat diselesaikan dalam satu langkah. Seiring berjalannya waktu, model YOLO terus dikembangkan sehingga model YOLO yang pertama kali dikembangkan oleh Redmon selanjutnya biasa disebut dengan YOLOv1.

Perkembangan selanjutnya adalah YOLOv2 atau YOLO9000, yang diusulkan oleh Joseph Redmon dan Ali Farhadi pada tahun 2017 (Redmon & Farhadi, 2016). YOLO9000 dirancang untuk memprediksi lebih dari 9000 kategori objek berbeda dengan melatihnya bersama pada kumpulan data MS COCO dan ImageNet. Pada tahun 2018, YOLOv3 muncul dengan peningkatan signifikan dan kembali diusulkan oleh Joseph Redmon dan Ali Farhadi dalam makalah yang berjudul "YOLOv3: *An Incremental Improvement*" (Redmon & Farhadi, 2018). YOLOv3 memperkenalkan Darknet-53 sebagai arsitektur backbone yang lebih canggih dan juga memanfaatkan teknik *feature pyramid* untuk meningkatkan deteksi objek pada skala yang berbeda. Selanjutnya, YOLOv4 yang diperkenalkan oleh Alexey Bochkovskiy pada tahun 2020 dengan peningkatan yang signifikan dalam kecepatan dan akurasi, serta kemampuan untuk mengatasi skenario deteksi objek yang lebih kompleks (Bochkovskiy et al., 2020). Pengembangan YOLO terus berlanjut sampai pada tahun 2023 diperkenalkan YOLOv8 yang dikembangkan oleh tim Ultralytics.

Pada penelitian ini model objek detektor yang akan dibandingkan adalah model jaringan YOLOv1 sampai dengan YOLOv8. Penelitian tentang perbandingan performa model objek detektor dapat menjadi acuan dalam aplikasi yang komprehensif. Riset unggulan pertanian seperti pertanian presisi membutuhkan aplikasi praktis dalam mengatasi tantangan di dunia pertanian. Penelitian juga mengacu pada riset yang berwawasan lingkungan dan pertanian industrial. Selain itu, keunggulan sumberdaya alam di Indonesia menjadi

DIGITAL REPOSITORY UNIVERSITAS JEMBER

DIGITAL REPOSITORY UNIVERSITAS JEMBER

pendorong peneliti untuk fokus pada penelitian ekplorasi sumberdaya alam. Riset unggulan tersebut diarahkan untuk berkontribusi pada pemecahan masalah nasional dan berkontribusi pada perkembangan keilmuan.

Perekonomian Indonesia khususnya kegiatan ekspor didominasi oleh sektor agrikultur, salah satunya adalah kakao. Pada tahun 2022 Indonesia menjadi produsen kakao terbesar dunia ketiga dengan total produksi 739.483 ton. Pantai Gading di urutan pertama memproduksi total 2,2 juta ton dan Ghana sebesar 800 ribu ton (Ghosh, 2022). Kakao berperan penting bagi perekonomian nasional, terutama sebagai penyedia lapangan kerja, dan sumber pendapatan serta devisa negara (Ploetz, 2016). Kualitas kakao Indonesia tidak kalah dengan kakao dunia karena memiliki keunggulan tidak mudah meleleh. Sejalan dengan keunggulan tersebut, peluang pasar kakao Indonesia cukup terbuka, baik untuk ekspor maupun permintaan dalam negeri. Namun, sejak tahun 2015 produktivitas kakao Indonesia mulai menurun. Penurunan jumlah produksi kakao di Indonesia dikarenakan berkurangnya luas area lahan kakao, meningkatnya jumlah tanaman tidak produktif, dan penurunan produktivitas tanaman. Penyebab utama tanaman tidak lagi produktif dan penurunan produktivitas adalah penyakit tanaman.

Penyakit tanaman yang paling sering dialami oleh tanaman kakao di Indonesia adalah *Vascular Streak Disease* (VSD). Penyakit VSD disebabkan oleh jamur *Oncobasidium theobromae (Ceratobasidiales: Ceratobasidiaceae)* (Samuels et al., 2012). Gejala yang pertama kali muncul biasanya daun menguning terutama pada daun kedua atau ketiga dari ujung. Pada daun muncul bintik-bintik hijau kecil yang menyebar pada daerah daun yang menguning. Penyakit ini menyebar sangat cepat dan bisa mematikan. Hal ini menjadi tantangan bagi perkebunan kakao untuk meningkatkan produktivitas kakao.

Banyak penelitian yang telah dilakukan terkait klasifikasi otomatis dan identifikasi penyakit tanaman. Dalam sebuah studi oleh Brahimi dan tim, mereka mengklasifikasikan sembilan penyakit daun tomat dengan tingkat akurasi 99,18% menggunakan Alexnet dan Googlenet (Brahimi et al., 2017). Darwish dan tim mengaplikasikan VGG-16 dan VGG-19 untuk merancang metode untuk mengidentifikasi tiga penyakit jagung dan menemukan akurasi rata -rata 98,2%

DIGITAL REPOSITORY UNIVERSITAS JEMBER

DIGITAL REPOSITORY UNIVERSITAS JEMBER

(Darwish et al., 2020). Liu dan tim mengusulkan arsitektur baru CNN untuk mengidentifikasi penyakit daun apel dengan akurasi 97,62% (Liu et al., 2018). Penelitian serupa dilakukan oleh penulis yang menggunakan arsitektur CNN untuk menemukan dan mengenali penyakit dalam jagung (DeChant et al., 2017). Berdasarkan artikel Tugrul dkk., masih belum ada yang melakukan penelitian yang menggunakan CNN untuk klasifikasi citra penyakit tanaman daun kakao. Unsur-unsur pendukung dalam penelitian ini adalah metode pengambilan data, membangun model YOLOv1 sampai dengan YOLOv8, evaluasi menggunakan tingkat akurasi model, dan penentuan model yang paling optimal untuk masalah identifikasi penyakit pada daun kakao.

Selanjutnya, peneliti menggunakan teori graf *dominating set* untuk menganalisis tingkat sebaran penyakit VSD pada tanaman kakao di suatu lahan, *Dominating set* merupakan suatu konsep penentuan himpunan titik seminimal mungkin pada suatu graf dengan ketentuan setiap titik yang merupakan anggota dari himpunan tersebut menjangkau titik lain di luar himpunan (Goddard dan Henning, 2013). Tanaman kakao dalam suatu lahan direpresentasikan sebagai titik pada sebuah graf yang selanjutnya dapat diterapkan teori *dominating set* untuk mencari himpunan dominasi. Himpunan dominasi merupakan tanaman yang akan menjadi data identifikasi. Dari hal tersebut dapat dianalisis sebaran penyakit VSD pada tanaman kakao. Kardinalitas himpunan dominasi adalah jumlah minimum sampel pohon kakao yang akan diidentifikasi. Jumlah sampel yang minimum dalam pengambilan data akan memudahkan analisis sebaran penyakit VSD pada suatu lahan menjadi efektif dan efisien.

1.2 Rumusan Masalah

Rumusan masalah dalam penelitian ini adalah sebagai berikut:

1. Bagaimana membandingkan dan mendapatkan model terbaik dari metode jaringan YOLOv1 – YOLOv8 untuk mendeteksi objek penyakit VSD pada daun kakao?
2. Bagaimana menganalisis tingkat sebaran penyakit VSD menggunakan teori dominating set?

DIGITAL REPOSITORY UNIVERSITAS JEMBER

DIGITAL REPOSITORY UNIVERSITAS JEMBER

1.3 Batasan Penelitian

Penelitian ini mencakup tentang penerapan YOLOv1 sampai YOLOv8 pada identifikasi penyakit VSD pada tanaman kakao. Selanjutnya proses identifikasi tingkat keparahan penyakit VSD suatu lahan dilakukan dengan memperhatikan faktor utama penyebarannya, yaitu angin dan menerapkan teori *dominating set*.

1.4 Tujuan Penelitian

Tujuan dari penelitian ini adalah sebagai berikut:

1. Membandingkan dan mendapatkan model terbaik dari metode jaringan YOLOv1 – YOLOv8 untuk mendeteksi objek penyakit VSD pada daun kakao.
2. Menganalisis tingkat sebaran penyakit VSD menggunakan teori dominating set.

1.5 Manfaat Penelitian

Manfaat dari penulisan tugas akhir ini adalah sebagai sumber literasi dalam penelitian terkait aplikasi YOLOv1 sampai dengan YOLOv8 untuk mengidentifikasi penyakit pada daun tanaman kakao. Hasil dari penelitian ini dapat mengidentifikasi penyakit VSD secara dini pada tanaman kakao sehingga dapat dilakukan penanganan segera. Lebih lanjut, penerapan YOLO dan teori dominating set dalam identifikasi penyebaran penyakit pada tanaman dapat dilakukan secara otomatis dan diterapkan pada berbagai jenis tanaman.

DIGITAL REPOSITORY UNIVERSITAS JEMBER

DIGITAL REPOSITORY UNIVERSITAS JEMBER

BAB 2. TINJAUAN PUSTAKA

2.1 Kajian Literatur

Pada tahun 2015, YOLOv1 pertama kali diperkenalkan oleh Joseph Redmon, Santosh Divvala, Ross Girshick, dan Ali Farhadi pada papernya yang berjudul "You Only Look Once: Unified, Real-Time Object Detection" (Redmon, et al., 2016). YOLOv1 muncul dengan inovasi besarnya dalam menyelesaikan masalah deteksi objek menggunakan regresi. Namun, kelemahan dari YOLOv1 adalah kurangnya kemampuan dalam mendeteksi objek kecil sehingga pada akhir tahun 2016, muncullah YOLOv2 yang dapat mengatasi masalah tersebut. (Redmon & Farhadi, 2016). Pada April tahun 2018, mereka menyempurnakan lagi karyanya yaitu YOLOv3 dengan perkembangan yang signifikan dalam hal performa dan tingkat akurasi (Redmon & Farhadi, 2018). Lama berselang, Alexey Bochkovskiy, Chien-Yao Wang, dan Hong-Yuan Mark Liao merilis YOLOv4 pada tahun 2020 (Bochkovskiy et al., 2020). YOLOv4 menjadi terobosan baru yang menggabungkan berbagai teknik canggih, seperti CSPDarknet53, PANet, SAM, dan lainnya untuk meningkatkan akurasi dan kecepatan deteksi objek yang luar biasa. Perkembangan peelitian yang menggunaan metode YOLO untuk mendeteksi objek dapat dilihat pada Tabel 2.1 di bawah ini.

Tabel 2.1 Hasil literatur aplikasi YOLO

Penulis	Judul	Objek	Pokok Bahasan
Ye A, Pang B, Jin Y, dan Cui J, 2020	A YOLO-Based Neural Network With VAE for Intelligent Garbage Detection and Classification	Video sampah	Penelitian ini menggunakan metode YOLOv1 dengan *Variational Autoencoder* untuk meningkatkan akurasi proses pemilahan sampah daur ulang secara otomatis.
Loey M, Manogaran G, Taha M, dan Khalifa, 2021	Fighting Against COVID-19: A Novel Deep Learning Model Based on YOLO-V2 with Resnet-50 for Medical Face Mask Detection	Foto penggunaan masker di tempat umum	Penelitian ini menerapkan YOLOv2 dengan ResNet-50 sebagai *transfer learning* untuk mendeteksi masker wajah.
Li J, Gu J, Huang Z, dan Wen J, 2019	Application Research of Improved YOLO V3 Algorithm in PCB Electronic Component Detection	Foto PCB *(Printed Circuit Board)*	Penelitian ini bertujuan mendeteksi komponen pada PCB menggunakan YOLOv3 dengan mengaplikasikan

DIGITAL REPOSITORY UNIVERSITAS JEMBER

DIGITAL REPOSITORY UNIVERSITAS JEMBER

			DarkNet-53 sebagai *feature extractor*.
Dongyao, 2022	Detection of cervical cancer cells in complex situation based on improved YOLOv3 network	Gambar sel pada screening pasien kanker serviks	Penelitian ini menerapkan algoritma k-means++ pada YOLOv3 untuk mendeteksi sel kanker serviks.
Kannadaguli P, 2020	YOLOv4 Based Human Detection System Using Aerial Thermal Imaging for UAV Based Surveillance Applications	Gambar dan video termal	Penelitian ini menerapkan YOLOv4 dengan deteksi objek multi-skala menggunakan dataset termal untuk mendeteksi manusia.
Jiang J, Fu X, Qin R, Wang X, dan Ma Z, 2021	High-Speed Lightweight Ship Detection Algorithm Based on YOLO-V4 for Three-Channels RGB SAR Image	Gambar *Synthetic Aperture Radar* (SAR)	Penelitian ini bertujuan mendeteksi kapal dengan kecepatan tinggi pada gambar SAR menggunakan metode YOLOv4.
Wu D, Lv S, Jiang M, dan Song H, 2020	Using Channel Pruning-Based YOLO V4 Deep Learning Algorithm for The Real-Time and Accurate Detection of Apple Flowers in Natural Environments	Gambar bunga apel	Penelitian ini menerapkan YOLOv4 untuk mendeteksi 3 jenis bunga apel, yaitu bunga apel fuji, *red love*, dan gala.
Fouad, 2023	Robust Classification and Detection of Big Medical Data Using Advanced Parallel K-Means Clustering, YOLOv4, and Logistic Regression	Dataset besar di bidang kesehatan	Penelitian ini menerapkan algoritma k-means paralel dan regresi logistik pada YOLOv4 untuk dipalikasikan pada dataset bidang kesehatan.

2.2 Tanaman Kakao dan Penyakit VSD

Kakao atau *Theobroma cacao L.* adalah tanaman budidaya perkebunan yang berasal dari hutan tropis di Amerika Tengah dan Amerika Selatan. Di Indonesia kakao banyak ditanam di Pulau Sulawesi, Sumatera, dan Jawa. Kakao merupakan salah satu hasil perkebunan terbaik Indonesia yang menjadi komoditas ekspor. Indonesia termasuk dalam negara penghasil kakao terbesar di dunia. Penyakit pada kakao yang paling umum di Indonesia adalah penyakit *Vascular Streak Disease* (VSD). Penyakit VSD disebabkan oleh jamur *Oncobasidium theobromae.* Jamur membentuk basidiopsora yang merupakan sumber infeksi dari jamur ini dan disebarkan oleh angin pada malam hari saat kelembaban tinggi. Gejala awal yang dapat dilihat adalah klorosis (menguning) pada satu daun, biasanya pada daun muda (*flush*) kedua atau ketiga, dengan bercak hijau tersebar sekitar 2 - 5 mm. Gejala pada bibit muda muncul setelah beberapa minggu,

DIGITAL REPOSITORY UNIVERSITAS JEMBER

DIGITAL REPOSITORY UNIVERSITAS JEMBER

sedangkan pada cabang di pohon dewasa, gejala baru muncul setelah 2-3 bulan. Infeksi awal hanya terjadi pada daun muda yang belum mengeras, selanjutnya infeksi menyebar mengikuti jaringan pembuluh *xylem*. VSD adalah penyakit yang mematikan, menular, dan penyebarannya sangat cepat sehingga penting untuk identifikasi lebih awal dan harus segera ditangani.

Gambar 2.1 Perbandingan daun kakao yang sehat dan terinfeksi VSD

Citra digital merupakan sebuah larik (array) yang berisi nilai-nilai real maupun kompleks yang direpresentasikan dengan deretan bit tertentu. Citra digital dapat didefinisikan secara matematis sebagai fungsi intensitas dalam 2 variabel x dan y. Fungsi tersebut dapat dituliskan sebagai $f(x,y)$ dimana (x, y) merepresentasikan koordinat spasial pada bidang 2 dimensi dan $f(x,y)$ merupakan intensitas cahaya pada kordinat tersebut. Formula $f(x,y)$ dituliskan pada Persamaan 2.1 sampai 2.3 dan dapat didefinisikan sebagai perkalian dari dua komponen, yaitu $i(x,y)$ yang merupakan jumlah cahaya dari sumber dan $r(x,y)$ sebagai reflektansi atau proporsi cahaya yang dipantulkan (Gonzales et al., 2018).

$$f(x,y) = i(x,y)r(x,y) \tag{2.1}$$

dimana

$$0 \leq i(x,y) < \infty \tag{2.2}$$

dan

$$0 \leq r(x,y) \leq 1 \tag{2.3}$$

Sebuah citra juga dapat dipresentasikan menjadi sebuah matriks berukuran $m \times n$ seperti pada Persamaan 2.4. Setiap elemen berisi nilai yang disebut dengan piksel. Nilai piksel merepresentasikan intensitas kanal warna pada citra. Ilustrasi citra digital dalam bentuk matriks dapat direpresentasikan seperti matriks di bawah ini dengan x menunjukkan baris dan y menunjukkan kolom.

DIGITAL REPOSITORY UNIVERSITAS JEMBER

DIGITAL REPOSITORY UNIVERSITAS JEMBER

$$f(x,y) = \begin{bmatrix} f(0,0) & f(0,1) & \cdots & f(0,n-1) \\ f(1,0) & f(1,1) & \cdots & f(1,n-1) \\ \vdots & \vdots & \vdots & \vdots \\ f(m-1,0) & f(m-1,1) & \cdots & f(m-1,n-1) \end{bmatrix} \quad (2.4)$$

2.3 Machine Learning

Istilah *machine learning* atau pembelajaran mesin pertama kali didefinisikan oleh Arthur Samuel pada tahun 1959. Menurut Arthur Samuel, *machine learning* adalah suatu bidang ilmu komputer yang memberikan kemampuan pembelajaran kepada komputer untuk mengetahui sesuatu tanpa pemrograman yang jelas. Berikut akan dijelaskan 3 jenis *machine learning* dilihat dari metode pembelajarannya.

1. *Supervised Learning*

Pada *supervised learning* mesin dilatih untuk mempelajari pola antara input data dan lebel output. Contoh aplikasi metode ini adalah regresi dan klasifikasi.

2. *Unsupervised Learning*

Mesin tidak perlu melatih algoritma komputer untuk mengenali pola data, dimana data input tidak diberi label. Contohnya adalah klasterisasi dan mengubah dimensi.

3. *Reinforcement Learning*

Pembelajaran mesin dilakukan berdasarkan umpan balik dengan tujuan mesin dapat membuat keputusan sendiri. Contoh aplikasinya adalah *real-time decisions, robot navigation*, dan game AI.

2.4 Deep Learning

Deep learning merupakan bagian dari *machine learning* untuk mempelajari representasi dari data yang menekankan pada pembelajaran *layer* (lapisan). Kata *'deep'* bermakna semakin banyak *layer* berurutan yang digunakan untuk mempelajari data tersebut. Pada perkembangannya *deep learning* menggunakan puluhan sampai ratusan *layer*, kemudian algoritma mempelajari secara otomatis dari data pelatihan. Dengan mengumpulkan data dalam jumlah besar dan menganalisisnya, *deep learning* membuat beberapa model prediktif untuk memahami pola dan tren dalam data sehingga dapat memberikan hasil yang

DIGITAL REPOSITORY UNIVERSITAS JEMBER

DIGITAL REPOSITORY UNIVERSITAS JEMBER

informatif dan bermakna. Penggunaan deep learning ini dapat dikelompokkan menjadi empat bidang, yaitu *computer vision* (penglihatan computer), *speech recognition* (pengenalan ucapan), *natural language processing/NLP* (pemrosesan bahasa alami), dan *recommendation engines* (mesin rekomendasi).

2.5 Deteksi Objek

Deteksi objek merupakan salah satu aplikasi dari *supervised learning* dan banyak mengaplikasikan *deep learning*. Model yang digunakan untuk mendeteksi objek (model detektor) dibagi menjadi 2 jenis, yaitu model detektor 1 tahap dan 2 tahap.

2.5.1 Model Detektor Dua Tahap

Model yang populer antara lain, *Region based Convolutional Neural Networks* (RCNN), *Fast* CNN, dan *Faster* CNN. Berikut merupakan tahapan pada model ini.

1. Ekstraksi fitur pada citra

Proses ini mengidentifikasi atau mengekstrak fitur penting pada suatu citra untuk meningkatkan kinerja dan akurasi model. Tahap ini menghasilkan *Region of Interest* (ROI). Pada RCNN ekstraksi fitur dilakukan pada lapisan konvolusi.

2. Klasifikasi dan/atau lokalisasi

Metode yang sering digunakan untuk klasifikasi antara lain, ANN, *Support Vector Machine* (SVM), *decision trees*, dan *K-Nearest Neighbor* (KNN). Setelah itu dilakukan lokalisasi untuk menentukan posisi objek di dalam sebuah citra. Pada tahap ini dibuat kotak pembatas (*bounding box*) di sekitar objek.

2.5.2 Model Detektor Satu Tahap

Model ini mendeteksi objek dengan memperhatikan keseluruhan daerah pada citra, tanpa adanya daerah usulan. Dalam sekali pengambilan, model ini menghasilkan kotak pembatas dengan nilai probabilitas kelas tertentu untuk suatu objek. Contoh model ini antara lain, *Single Shot Detector* (SSD), *Deconvolution Single Shot Detector* (DSSD), RetinaNet, M2Det, RefineDet, dan YOLO.

2.6 *Convolutional Neural Network* (CNN)

CNN merupakan bagian dari *Deep Learning* yang banyak digunakan dalam pemrosesan dan pengenalan citra. Secara umum, terdapat tiga *layer* di CNN, yaitu

DIGITAL REPOSITORY UNIVERSITAS JEMBER

DIGITAL REPOSITORY UNIVERSITAS JEMBER

input layer, *hidden layer*, dan *output layer* (lapisan keluaran). Operasi dalam *hidden layer* mengubah data untuk mempelajari fitur-fitur khusus yang ada pada data citra. Empat *layer* yang paling umum di dalam *hidden layer* antara lain, *convolution layer* (lapisan konvolusi), *activation layer* (lapisan aktivasi), *pooling layer* (lapisan penyatuan), dan *fully connected layer*. Berikut akan dijelaskan fungsi dari ketiga jenis *layer* tersebut. Ilustrasi dan penjelasan proses dalam CNN akan dijelaskan dalam Lampiran 1.

2.7 *You Only Look Once* (YOLO)

YOLO pertama kali diperkenalkan oleh Joseph Redmon pada tahun 2015 di dalam jurnalnya yang berjudul "*You Only Look Once: Unified, Real-Time Object Detection*". YOLO merupakan algoritma pendeteksi objek secara real-time yang mampu memprediksi posisi objek dan probabilitas kelas untuk semua objek pada suatu gambar. Algoritma ini mengidentifikasi objek dan posisinya dengan sekali melihat sehingga disebut dengan *You Only Look Once*.

2.7.1 Perkembangan YOLO

YOLOv1 mampu menyelesaikan masalah deteksi sebagai masalah regresi tunggal, setelah sebelumnya masalah deteksi dianggap sebagai masalah klasifikasi. Algoritma ini dapat memproses gambar dengan sangat cepat, yaitu sebanyak 45 frame gambar per detik. Namun, kekurangan dari YOLOv1 adalah banyaknya deteksi palsu pada objek kecil dan objek yang saling tumpang tindih. Kemudian YOLOv2 hadir dengan kecepatan yang lebih tinggi dan dapat mendeteksi objek kecil dengan menerapkan *bounding box clustering*. Kemudian muncul YOLOv3 yang mampu meningkatkan akurasi dengan menggunakan fitur penggabungan skala dan *non-maximum suppression*. Kemudian YOLOv4 hadir dengan peningkatan akurasi deteksi objek secara signifikan. YOLO v4 juga mampu mengatasi masalah deteksi objek kecil dan objek yang saling tumpang tindih.

2.7.2 Bagian YOLO

YOLO memiliki 3 bagian dalam arsitekturnya, yaitu *backbone*, *neck*, dan *head*. Berikut adalah penjelasan dari ketiga bagian tersebut.

DIGITAL REPOSITORY UNIVERSITAS JEMBER

DIGITAL REPOSITORY UNIVERSITAS JEMBER

1. *Backbone : feature extractor*

Bagian ini bertugas untuk mengekstraksi fitur dari gambar input. YOLO menerapkan algoritma CNN sebagai *backbone* dalam arsitekturnya. Arsitektur ini terdiri dari beberapa *convolution layer*, *pooling layer*, dan *upsampling layer*.

2. *Neck : feature aggregator*

Neck bertugas untuk mengumpulkan berbagai fitur/informasi dari level yang berbeda pada *feature map* yang diperoleh dari proses ekstraksi fitur pada *backbone*.

3. *Head : object detector*

Bagian ini terdiri dari beberapa *convolution layer* untuk menghasilkan *bounding box* dan *confidence score* untuk setiap kelas objek. YOLO menggunakan teknik *non-maximum suppression* (NMS) dengan menetapkan *treshold* sebagai batas tingkat *confidence* suatu *bounding box* yang akan ditraining.

2.7.3 Tahapan YOLO

Algoritma YOLO memiliki arsitektur yang cukup rumit. Berikut adalah tahapan YOLO secara umum.

1. Gambar input diubah menjadi bentuk grid berukuran $S \times S$ (ukuran piksel).
2. Setiap *grid cell* mempunyai *bounding box* yang berisi 5 nilai parameter yang berbeda (x, y, w, h, cf) untuk setiap sel. nC menyatakan banyaknya kelas dan nB meyatakan banyaknya *bounding box* (bbox).

   x : lokasi koordinat x berdasarkan baris w : lebar *grid cell*

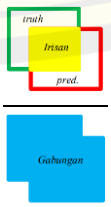
   y : lokasi koordinat y berdasarkan baris h : tinggi *grid cell*

   cf : nilai *confidence*, $cf = 0$ jika tidak mengandung objek

   Nilai cf dapat dihitung menggunakan Persamaan 2.5. Sedangkan Persamaan 2.6 merupakan persamaan Intersection Over Union $\left(IoU_{pred}^{truth}\right)$.

$$Confidence\ (cf) = P(object) * IoU \tag{2.5}$$

$$IoU = \frac{Intersection\ (Irisan)}{Union\ (Gabungan)} = \frac{}{} \tag{2.6}$$



DIGITAL REPOSITORY UNIVERSITAS JEMBER

DIGITAL REPOSITORY UNIVERSITAS JEMBER

Lakukan *Non-Maximum Supression* (NMS). NMS digunakan untuk mengatasi deteksi yang tumpang tindih dan mempertahankan deteksi yang lebih akurat. Berikut adalah Langkah-langkah dalam melakukan NMS.

- Daftar semua bbox
- Set bbox dengan skor maksimum sebagai bbox_maks.
- Bandingkan bbox_max dengan bbox lainnya sebagai bbox_temp.
- Jika nilai IoU(bbox_maks, bbox_temp) > 0.5, maka set skor bbox_temp menjadi nol.
- Jika nilai IoU(bbox_max, bbox_temp) ≤ 0.5, maka skor bbox_temp tetap.

3. Langkah terakhir, plot *bounding box* berdasarkan hasil NMS.

2.7.4 *Loss Function* YOLO

Loss function pada YOLO digunakan untuk mengukur sejauh mana prediksi model mendekati nilai sebenarnya (*ground truth*) dari deteksi objek. YOLO menggunakan beberapa komponen *loss function* untuk melatih model dengan efektif. Berikut adalah beberapa komponen utama *loss function* pada jaringan YOLO.

1. *Localization Loss*

Komponen ini bertanggung jawab untuk mengukur kesalahan dalam prediksi koordinat *bounding box*. *Loss* ini diformulasikan dalam Persamaan 2.7 untuk diterapkan pada setiap *bounding box* yang memiliki objek dan dihitung menggunakan *mean squared error* (MSE).

$$Local\ Loss = \lambda_{coord} \sum_{i=0}^{s^2} \sum_{j=0}^{B} 1_{ij}^{obj} [(x_i - \hat{x_i})^2 + (y_i - \hat{y}_i)^2] + \lambda_{coord} \sum_{i=0}^{s^2} \sum_{j=0}^{B} 1_{ij}^{obj} \left[\left(\sqrt{w_i} - \sqrt{\hat{w_i}}\right)^2 + \left(\sqrt{h_i} - \sqrt{\hat{h_i}}\right)^2 \right] \quad (2.7)$$

2. *Confidence Loss*

Komponen ini mengukur kesalahan prediksi tingkat kepercayaan (*confidence*) apakah suatu *bounding box* berisi objek atau tidak. Jika suatu bounding box

DIGITAL REPOSITORY UNIVERSITAS JEMBER

DIGITAL REPOSITORY UNIVERSITAS JEMBER

mengandung objek, tetapi memiliki kepercayaan rendah, *loss confidence* akan tinggi. *Loss* ini diformulasikan dalam Persamaan 2.8.

$$Confidence\ Loss = \sum_{i=0}^{s^2}\sum_{j=0}^{B} 1_{ij}^{obj}\left(C_i - \hat{C}_\iota\right)^2 + \lambda_{noobj}\sum_{i=0}^{s^2}\sum_{j=0}^{B} 1_{ij}^{noobj}\left(C_i - \hat{C}_\iota\right)^2 \quad (2.8)$$

3. *Classification Loss*

Komponen ini mengukur kesalahan dalam memprediksi kelas objek. *Loss* ini diterapkan pada setiap *bounding box* yang mengandung objek untuk memperbarui bobot yang berkaitan dengan kelas objek yang benar. *Loss* ini diformulasikan dalam Persamaan 2.9.

$$Classification\ Loss = \sum_{i=0}^{s^2} 1_{ij}^{obj} \sum_{c\in classes}^{B}\left(p_i(c) - \hat{p_\iota}(c)\right)^2 \quad (2.9)$$

2.8 Confusion Matrix

Confusion matrix digunakan untuk menggambarkan seberapa baik kinerja sistem klasifikasi. Output dari algoritma klasifikasi ditampilkan dan diringkas dalam *confusion matrix*.

Tabel 2.2 Tabel confusion matrix

	True	**False**
True	True Positive (TP)	False Positive (FP)
False	False Negative (FN)	True Negative (TN)

Matriks ini dapat digunakan untuk mengevaluasi kinerja suatu model, seperti akurasi, presisi, dan *recall*. Ketiga nilai tersebut dapat dihitung menggunakan Persamaan 2.10, 2.11, dan 2.12.

$$Akurasi = \frac{TP + TN}{TP + TN + FP + FN} \times 100\% \quad (2.10)$$

$$Presisi = \frac{TP}{TP + FP} \times 100\% \quad (2.11)$$

$$Recall = \frac{TP}{TP + FN} \times 100\% \quad (2.12)$$

DIGITAL REPOSITORY UNIVERSITAS JEMBER

DIGITAL REPOSITORY UNIVERSITAS JEMBER

2.9 *Mean Average Precision* (mAP)

Average Precision (AP) menghitung nilai rata-rata presisi model dalam mendeteksi objek untuk berbagai nilai ambang batas (*threshold*). Nilai AP didapat dengan menghitung luasan di bawah kurva *presisi-recall* yang dihasilkan dengan memvariasikan nilai ambang batas.

Langkah-langkah dalam menghitung nilai mAP adalah sebagai berikut.

1. Implementasi model pada suatu dataset.
2. Untuk setiap kelas objek, dihitung nilai presisi dan *recall* menggunakan nilai yang didapat pada *confusion matrix*.
3. Kurva presisi-*recall* dibuat dengan memvariasikan nilai ambang batas.
4. Luas area di bawah kurva presisi-*recall* dihitung untuk mendapatkan nilai AP tiap kelas objek.
5. Nilai mAP adalah nilai rata-rata nilai AP setiap kelas objek yang ada.

Nilai mAP merupakan ukuran kualitas model detektor objek secara menyeluruh. Semakin tinggi nilai mAP, semakin baik performa deteksi objek suatu model.

2.10 *Early Stopping Point*

Early stopping dalam machine learning adalah suatu teknik di mana proses pelatihan model dihentikan lebih awal untuk mencegah overfitting dan memastikan bahwa model dapat berkinerja dengan baik pada data baru yang belum pernah dilihat sebelumnya. Hal ini dilakukan dengan memantau kinerja model pada data validasi selama pelatihan. Jika kinerja pada data validasi tidak meningkat atau bahkan menurun seiring berjalannya waktu, pelatihan dihentikan.

Early stopping membantu mengidentifikasi titik di mana model mencapai tingkat kompleksitas yang optimal, tanpa melibatkan overfitting. Berikut adalah langkah-langkah dalam melakukan *early stopping* (Prechelt, n.d.).

1. Pisahkan data latih menjadi data latih dan data validasi, misal dengan perbandingan 2:1.
2. Proses training dilakukan hanya pada data latih. Lakukan evaluasi nilai error model menggunakan data evaluasi setiap n iterasi, misal $n = 10$.

DIGITAL REPOSITORY UNIVERSITAS JEMBER

DIGITAL REPOSITORY UNIVERSITAS JEMBER

3. Hentikan proses training segera setelah nilai error data validasi lebih tinggi dari nilai error sebelumnya, terakhir kali dievaluasi.
4. Gunakan bobot jaringan pada langkah sebelumnya sebagai hasil dari proses training.

2.11 Graf *Dominating Set*

Dominating set merupakan suatu konsep penentuan titik seminimal mungkin pada graf dengan ketentuan titik sebagai dominating set menjangkau titik yang ada di sekitarnya. Kardinalitas terkecil dari dominating set disebut *domination number*.

Definisi 2.1

G adalah sebuah graf terhubung yang berorder n dengan himpunan titik $V(G)$ dan himpunan sisi $E(G)$. Himpunan titik $S \subseteq V(G)$ disebut sebagai himpunan dominasi (*dominating set*) dari graf G, jika untuk setiap titik x di $V(G) \backslash S$, ada minimal satu titik $u \in S$ sehingga $x \sim u$. Notasi $x \sim u$ berarti titik x berdekatan dengan titik u. Kardinalitas minimum dari himpunan dominasi pada graf G disebut bilangan dominasi (*domination number*) dan dinotasikan dengan $\gamma(G)$ (Susilowati et al., 2020). Berikut diberikan contoh bilangan dominasi pada Gambar 2.2. Diperhatikan bahwa x_1 dan x_3 merupakan titik dominator dan 2 merupakan jumlah minimum dari himpunan dominasi sehingga didapat bilangan dominasi $\gamma(P_4) = 2$.

Gambar 2.2 Himpunan dominasi pada graf lintasan P_4

Teorema 2.1 (Arshad et al., 2023) Untuk setiap 2 graf lintasan P_m dan P_n dengan $m, n \geq 2$ berlaku $\gamma(P_m \boxtimes P_n) = \left\lceil \frac{m}{3} \right\rceil \left\lceil \frac{n}{3} \right\rceil$.

Misalkan $m = 3q_m + r_m$ dan $n = 3q_n + r_n$ dan label pada graf lintasan $P_m = \{u_1, u_2, \dots, u_m\}$ dan $P_n = \{v_1, v_2, \dots, v_n\}$. Berdasarkan nilai dari r_m dan r_n, terdapat tiga kasus sebagai berikut.

DIGITAL REPOSITORY UNIVERSITAS JEMBER

DIGITAL REPOSITORY UNIVERSITAS JEMBER

Kasus 1 : Jika $r_m = r_n = 0$, maka $m = 3q_m$ dan $n = 3q_n$ didapat himpunan dominan S dari $P_m \boxtimes P_n$ sebagai berikut.

$$S = \{u_2, u_5, u_8, \dots, u_{m-1}\} \times \{v_2, v_5, v_8, \dots, v_{n-1}\}.$$

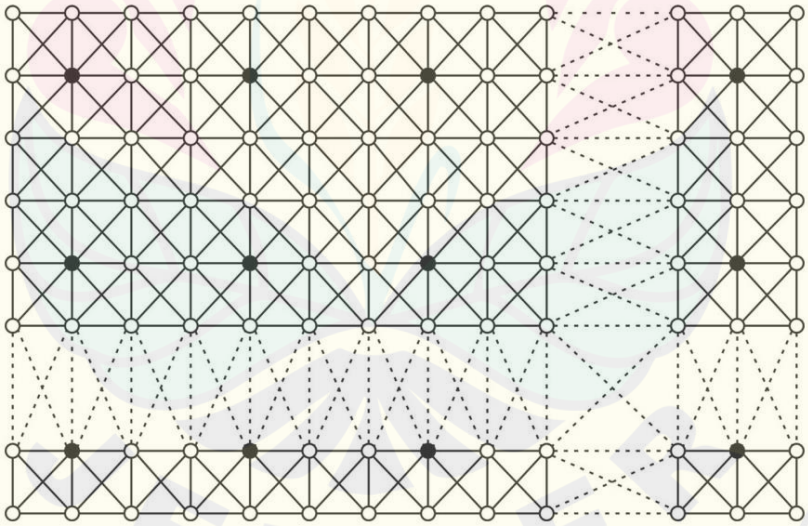
Kasus 2 : Jika salah satu dari r_m dan r_n bernilai nol dan lainnya tidak nol, tanpa mengurangi keumuman dimisalkan $r_m = 0$ dan $r_n \neq 0$, maka $m = 3q_m$ dan $n = 3q_n + r_n$. Didapat himpunan dominan S dari $P_m \boxtimes P_n$ sebagai berikut.

$$S = \{u_2, u_5, u_8, \dots, u_{m-1}\} \times \{v_2, v_5, v_8, \dots, v_{n-1}, v_{n+1}\}.$$

Kasus 3 : Jika r_m dan r_n keduanya tidak nol, maka $m = 3q_m + r_m$ dan $n = 3q_n + r_n$ dimana r_m dan r_n bernilai 1 atau 2. Didapat himpunan dominan S dari $P_m \boxtimes P_n$ sebagai berikut.

$$S = \{u_2, u_5, u_8, \dots, u_{m-1}, u_{m+1}\} \times \{v_2, v_5, v_8, \dots, v_{m-1}, v_{s+1}\}.$$

Himpunan dominasi dari graf $P_m \boxtimes P_n$ diilustrasikan pada Gambar 2.3 sebagai berikut.



Gambar 2.3 Ilustrasi himpunan dominasi graf $P_m \boxtimes P_n$

DIGITAL REPOSITORY UNIVERSITAS JEMBER

DIGITAL REPOSITORY UNIVERSITAS JEMBER

BAB 3. METODOLOGI PENELITIAN

Pada bab ini akan dijelaskan metodologi penelitian yang dilakukan untuk memperoleh tujuan dari penelitian ini, yaitu membandingkan performa model jaringan YOLOv1 sampai dengan YOLOv8 dalam mendeteksi objek penyakit VSD pada tanaman kakao dan mengidentifikasi tingkat keparahan lahan.

3.1 Lokasi dan Waktu Penelitian

Pengambilan data dilakukan di Pusat Penelitian Kopi dan Kakao Jember, Jawa Timur pada bulan Januari 2023. Kemudian dilanjutkan dengan penelitian selama 6 bulan.

3.2 Populasi dan Sampel/Subyek Penelitian

Populasi target yang digunakan dalam penelitian ini adalah tanaman kakao yang ditanam di suatu lahan. Sedangkan sampel yang digunakan adalah tanaman kakao pada suatu lahan yang ditanam di Pusat Penelitian Kopi dan Kakao Indonesia.

3.3 Pengumpulan Data Penelitian

Penelitian objek berfokus pada tanaman kakao. Data gambar daun diambil dari tanaman kakao yang ditanam di Institut Penelitian Kopi dan Kakao Indonesia di Jember, Jawa Timur. Proses deteksi objek menggunakan *deep learning* memerlukan sejumlah data besar untuk proses pelatihan. Selain itu, dalam proses pengujian juga diperlukan data untuk menguji tingkat akurasi model yang digunakan. Hal yang diperhatikan terkait data yang ada dalam proses deteksi objek yang dilakukan, di antaranya yaitu data input dan persiapan data.

Data yang digunakan dalam penelitian ini adalah data primer berupa data gambar daun yang terdiri dari dua kelas, yaitu kelas VSD dan kelas sehat. Terdapat tiga jenis data yang akan diuji coba dalam proses deteksi objek, yaitu data *training*, *validation*, dan *testing*. Jumlah dataset gambar yang terkumpul adalah 1250 gambar, terdiri dari 625 kelas VSD dan 625 kelas sehat. Selanjutnya dilakukan augmentasi data berupa *rotate* dan *flip*. Proses augmentasi menghasilkan data input

DIGITAL REPOSITORY UNIVERSITAS JEMBER

DIGITAL REPOSITORY UNIVERSITAS JEMBER

sejumlah 2500 gambar dengan masing-masing kelas berjumlah 1250 gambar. Augmentasi data mencegah overfitting dengan memodifikasi kumpulan data yang terbatas sehingga memiliki karakteristik seperti kumpulan data yang besar (Nachtigal dkk., 2016). Studi sebelumnya menunjukkan efektivitas augmentasi data menggunakan transformasi sederhana seperti augmentasi ruang warna, membalik, dan memutar gambar (Shorten, 2019).

Data yang telah diaugmentasi dibagi menjadi 80% untuk data *training* dan 20% data *validation*. Sedangkan data *testing* digunakan untuk menguji coba pada data yang benar-benar baru agar dapat mengetahui kinerja YOLOv4 dalam mendeteksi daun kakao yang sehat dan terdeteksi VSD. Data *testing* berupa video pohon kakao. Daftar data input dalam penelitian ini disajikan pada Tabel 3.1.

Tabel 3.1 Data input

Data Input	Jumlah Data		Total
	Kelas VSD	Kelas Sehat	
Data Latih	1000	1000	2000
Data Uji	250	250	500
Total	1250	1250	2500

3.4 Alat/Instrumen Penelitian

Gambar diambil menggunakan kamera smartphone dengan kamera 48 MP. Semua gambar disimpan dalam format JPG dengan ukuran gambar 3000 × 3000 piksel. Untuk mengimplementasikan algoritma ini akan digunakan perangkat lunak Python dan Google Colab.

3.5 Prosedur Penelitian

Prosedur penelitian berfungsi sebagai acuan sehingga penelitian dapat berjalan sistematis. Langkah-langkah yang dilakukan dibagi menjadi 2 tahap, yaitu membandingkan performa model jaringan YOLO dan identifikasi tingkat keparahan lahan.

3.5.1 Membandingkan Performa Model Jaringan YOLO

Berikut adalah langkah-langkah yang dilakukan untuk membandingkan performa model jaringan YOLOv1 sampai dengan YOLOv8.

DIGITAL REPOSITORY UNIVERSITAS JEMBER

DIGITAL REPOSITORY UNIVERSITAS JEMBER

1. Studi Literatur

Tahap studi literatur yaitu melakukan pendalaman materi tentang tahap pengolahan citra, pengklasifikasian citra yaitu metode CNN, pendeteksi objek yaitu metode YOLO, dan teori graf *dominating set*. Literatur yang digunakan yaitu berdiskusi dengan dosen pembimbing, membaca buku, jurnal, artikel, dan website yang membahas tentang materi tersebut.

2. Pengumpulan Data

Penelitian objek berfokus pada kakao yang ditanam secara luas di Indonesia, khususnya Jember, Jawa Timur. Data gambar daun diambil dari tanaman kakao yang ditanam di Institut Penelitian Kopi dan Kakao Indonesia di Jember. Citra yang diambil terdiri dari dua kelas, yaitu kelas sehat dan kelas VSD. Daun yang terinfeksi diamati berdasarkan kondisi daun berdasarkan informasi dari literatur dan petani di sana. Gambar diambil menggunakan kamera smartphone dengan kamera 48 MP. Semua gambar disimpan dalam format JPG dengan ukuran gambar 3000 × 3000 piksel. Dataset awal mencakup 343 gambar daun sehat dan 327 yang terinfeksi VSD.

3. Penerapan Algoritma

Algoritma yang digunakan dalam penelitian ini untuk mengidentikasi penyakit VSD pada kakao melalui citra adalah YOLOv1 sampai dengan YOLOv8. Pada tahap *training* digunakan dataset berupa gambar foto daun. Selanjutnya, pada tahap *testing* akan digunakan data berupa video yang menyorot daun dan mengelilingi pohon kakao. Untuk mengimplementasikan algoritma ini akan digunakan perangkat lunak MatLab.

4. Analisis Hasil

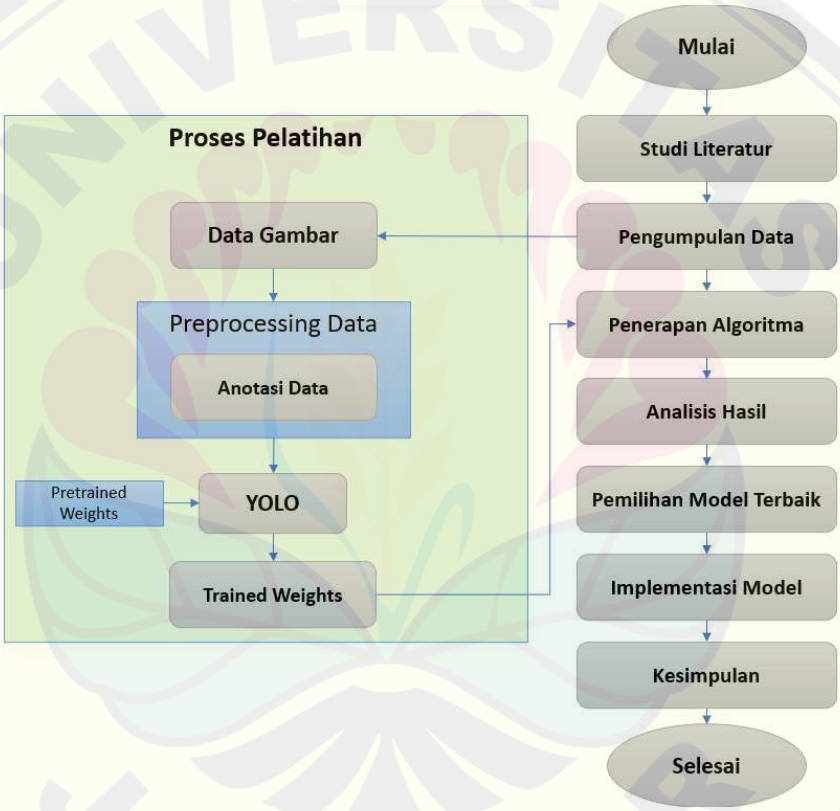
Proses analisis hasil pengujian algoritma dapat dilakukan menggunakan MAP (*Mean Average Precision*) dan c*onfusion matrix.* Matriks ini digunakan untuk mencari nilai akurasi, presisi, dan *recall*. Ketiga nilai tersebut digunakan untuk mengevaluasi kinerja suatu model. Semakin besar nilai ketiga kriteria tersebut (semakin mendekati 100%), maka semakin baik kinerja model yang digunakan.

DIGITAL REPOSITORY UNIVERSITAS JEMBER

DIGITAL REPOSITORY UNIVERSITAS JEMBER

5. Pemilihan Model Terbaik

Berdasarkan nilai dari keempat kriteria, yaitu MAP, akurasi, presisi, dan recall dapat dipilih model terbaik yang memberikan hasil paling akurat. Model terbaik memiliki nilai rata-rata MAP, akurasi, presisi, dan recall tertinggi yang paling mendekati 100%. Di samping itu, diperhatikan pula waktu yang diperlukan untuk melatih *(training)* data. Model yang paling efisien memiliki tingkat akurasi yang tinggi dengan waktu pelatihan yang cepat. Alur penelitian dalam membandingkan performa model YOLO dapat dilihat pada Gambar 3.1.



Gambar 3.1 *Flowchart* alur penelitian

3.5.2 Identifikasi Tingkat Sebaran VSD pada Lahan

Langkah selanjutnya setelah diperoleh model jaringan YOLO terbaik dalam mendeteksi objek adalah melakukan identifikasi tingkat keparahan lahan menggunakan teori dominating set. Berikut adalah langkah-langkah dalam mengidentifikasi tingkat keparahan lahan.

DIGITAL REPOSITORY UNIVERSITAS JEMBER

DIGITAL REPOSITORY UNIVERSITAS JEMBER

1. Konstruksi Graf Berdasarkan Kondisi Lahan

Langkah pertama yang dilakukan adalah pengamatan kondisi lahan dan karakteristik penyebaran penyakit. Graf dikonstruksi berdasarkan pola tanam pohon kakao dan faktor yang paling mempengaruhi penyebaran penyakit VSD, yaitu angin.

2. Penentuan Dominating Set dan Kardinalitasnya

Berdasarkan bentuk graf yang telah dikonstruksi dan studi literatur dilakukan, selanjutnya dapat ditentukan dominating set dan kardinalitasnya. Langkah ini dilakukan agar proses identifikasi penyakit dapat dilakukan secara efisien.

3. Identifikasi Penyakit VSD pada Dominating Set

Selanjutnya proses identifikasi penyakit VSD hanya diperhatikan pada tanaman kakao yang merupakan anggota himpunan dominating. Dengan ini proses identifikasi tidak perlu dilakukan pada semua tanaman kakao yang ada di sebuah lahan.

4. Analisa Tingkat Penyebaran Penyakit

Selanjutnya dari analisis tersebut dapat diidentifikasi tingkat sebaran penyakit VSD pada lahan tersebut berdasarkan presentase tanaman yang terinfeksi VSD yang merupakan anggota himpunan dominating.

DIGITAL REPOSITORY UNIVERSITAS JEMBER

DIGITAL REPOSITORY UNIVERSITAS JEMBER

BAB 4. HASIL DAN PEMBAHASAN

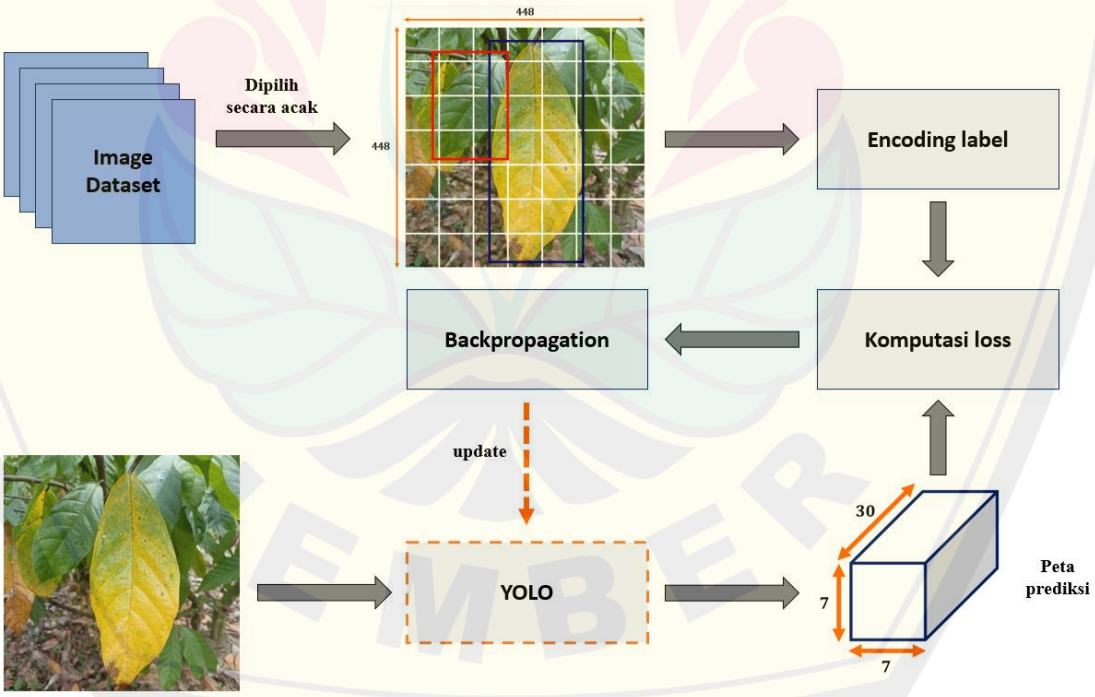
Pada bab ini akan dijelaskan mengenai parameter training, proses training data menggunakan YOLOv1 sampai dengan YOLOv8, pemrosesan output, implementasi sistem, proses uji coba, analisis hasil, evaluasi model, dan penggunaan graf *dominating set* dalam mengidentifikasi penyakit.

4.1 Proses Perancangan Algoritma dan Input Data

Pada subbab ini akan dijelaskan mengenai struktur dari algoritma YOLO, menentukan parameter yang akan digunakan dalam proses training, dan proses input data dari data gambar menjadi data berbentuk matriks.

4.1.1 Algoritma YOLO

Secara garis besar, algoritma deteksi objek menggunakan YOLO dapat diilustrasikan dalam Gambar 4.1.



Gambar 4.1 Algoritma deteksi objek menggunakan YOLO

Darknet digunakan sebagai *backbone* pada algoritma YOLO. Struktur YOLO terdiri atas layer-layer sebagai berikut.

DIGITAL REPOSITORY UNIVERSITAS JEMBER

DIGITAL REPOSITORY UNIVERSITAS JEMBER

a. Ektraksi fitur : 24 layer konvolusi dengan rincian pada Tabel 4.1.

Tabel 4.1 Tabel layer pada YOLO

	Layer	Ukuran	Filter	Stride	Output
	Conv.	$7 \times 7 \times 3$	64	2	$224 \times 224 \times 64$
	Max pool	2×2			$112 \times 112 \times 64$
	Conv.	$3 \times 3 \times 64$	192	1	$112 \times 112 \times 192$
	Max pool	2×2			$56 \times 56 \times 192$
	Conv.	$1 \times 1 \times 192$	128	1	$56 \times 56 \times 128$
	Conv.	$3 \times 3 \times 128$	256	1	$56 \times 56 \times 256$
	Conv.	$1 \times 1 \times 256$	256	1	$56 \times 56 \times 256$
	Conv.	$3 \times 3 \times 256$	512	1	$56 \times 56 \times 512$
	Max pool	2×2			$28 \times 28 \times 512$
4 ×	Conv.	$1 \times 1 \times 512$	256	1	$28 \times 28 \times 256$
	Conv.	$3 \times 3 \times 256$	512	1	$28 \times 28 \times 512$
	Conv.	$1 \times 1 \times 512$	512	1	$28 \times 28 \times 512$
	Conv.	$3 \times 3 \times 512$	1024	1	$28 \times 28 \times 1024$
	Max pool	2×2			$14 \times 14 \times 1024$
2 ×	Conv.	$1 \times 1 \times 1024$	512	1	$14 \times 14 \times 512$
	Conv.	$3 \times 3 \times 512$	1024	1	$14 \times 14 \times 1024$
	Conv.	$3 \times 3 \times 1024$	1024	1	$14 \times 14 \times 1024$
	Conv.	$3 \times 3 \times 1024$	1024	2	$7 \times 7 \times 1024$
	Conv.	$3 \times 3 \times 1024$	1024	1	$7 \times 7 \times 1024$
	Conv.	$3 \times 3 \times 1024$	1024	1	$7 \times 7 \times 1024$

b. Layer reduksi menggunakan filter 1×1

c. Layer reduksi diikuti dengan layer konvolusi 3×3

d. Prediksi parameter : 2 *fully connected layer*

e. Flatten layer : feature map $7 \times 7 \times 1024$ diubah menjadi 50.176 feature vector

f. Melalui 2 layer fully connected

g. Reshape layer : mengubah bentuk menjadi $7 \times 7 \times 30$ parameter feature map

DIGITAL REPOSITORY UNIVERSITAS JEMBER

DIGITAL REPOSITORY UNIVERSITAS JEMBER

Proses deteksi objek menggunakan algoritma YOLO diawali dengan menentukan berbagai parameter yang digunakan dalam proses training. Berikut akan dijelaskan mengenai parameter yang digunakan dalam penelitian ini.

4.1.2 Parameter Training

Parameter yang digunakan dalam proses training data pada YOLO dapat dilihat pada Tabel 4.2.

Tabel 4.2 Parameter YOLO

Parameter	Nilai	Parameter	Nilai	Parameter	Nilai
batch	64	decay	0.0005	angle	0
subdivisions	64	learning rate	0.001	saturation	0
width	416	burn in	1000	exposure	0
height	416	max batches	4000	hue	0
channels	3	steps	3200,3600		
momentum	0.9	scales	0.1,0.1		

Nilai parameter *batch* dan *subdivisions* menentukan jumlah sampel data yang akan dilatih dalam satu iterasi. Semua data input berupa citra akan dikonversi ukurannya menjadi $416 \times 416 \times 3$. Nilai parameter channels menunjukkan banyaknya kanal warna pada gambar, yaitu RGB (Red, Green, Blue). Parameter *momentum* diatur sebesar 0.9 untuk mencegah lonjakan perubahan pada saat pembaharuan bobot, sedangkan *decay* diatur sebesar 0.0005 untuk menghindari *overfitting*. Nilai parameter *learning rate* diatur lebih kecil untuk mengantisipasi kemungkinan data input untuk proses training sangat berbeda antara satu data dengan yang lain sehingga dapat mencegah *overfitting* di awal. Parameter *burn in* diatur 1000 sehingga didapat nilai *learning rate* saat iterasi di bawah 1000 ($i < 1000$) sebesar $learning\ rate_{new} = learning\ rate_{old} \times \left(\frac{i}{burn\ in}\right)^2$. Untuk mencapai bobot yang optimum, nilai *learning rate* harus diatur lebih kecil sehingga saat mencapai iterasi ke 3200 dan 3600, *learning rate* akan diperbaharui menjadi $learning\ rate_{new} = learning\ rate_{old} \times scales$. Jumlah *max batches* merupakan jumlah iterasi, yaitu sebesar $max\ batches = jumlah\ kelas \times 2000$

DIGITAL REPOSITORY UNIVERSITAS JEMBER

DIGITAL REPOSITORY UNIVERSITAS JEMBER

sehingga diperoleh nilai *max batches* adalah 4000 karena dataset memiliki 2 kelas. Selanjutnya tidak dilakukan perubahan warna pada data dengan mengatur *exposure, saturation, dan hue* bernilai nol karena aspek warna merupakan hal yang penting sebagai pembeda kelas pada dataset yang dimiliki.

4.1.3 Proses Input Data

Langkah awal dalam proses deteksi objek menggunakan YOLO adalah persiapan data dan proses input data. Data input dalam algoritma YOLO berupa gambar dengan anotasinya. Langkah dalam anotasi data gambar yaitu membuat *bounding box* secara manual pada setiap objek yang ada pada suatu gambar dan diberi label sesuai kelas objek tersebut. Data anotasi disimpan dalam suatu file *.txt* yang berisi nilai $(c, (x, y), (w, h))$ yang menyatakan kelas objek, koordinat titik pusat *bounding box*, dan ukuran dimensi *bounding box*. Data anotasi yang disimpan digunakan sebagai *ground truth* dalam menghitung nilai IoU dan perhitungan *loss*.

Objek yang ada pada gambar dilakukan proses *encoding* data bounding box $\big((x, y), (w, h)\big)$ menggunakan Persamaan 4.1 – 4.4. Nilai titik tengah (x, y) relatif terhadap *anchor box*. Sedangkan nilai lebar dan tinggi (w, h) relatif terhadap keseluruhan gambar.

$$\Delta x = x - x_a/64 \tag{4.1}$$

$$\Delta y = y - y_a/64 \tag{4.2}$$

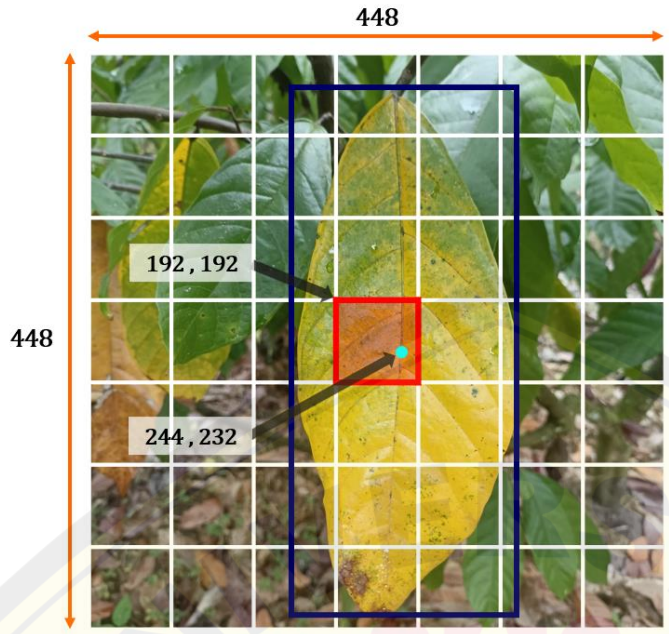
$$\Delta w = w/448 \tag{4.3}$$

$$\Delta h = h/448 \tag{4.4}$$

Selanjutnya, *c* yang menyatakan kelas objek diberi nilai $c = 1$. Dataset memiliki 2 kelas, yaitu kelas sehat $(c = 0)$ dan kelas sakit $(c = 1)$. Diperoleh anotasi data pada gambar 4.2 adalah $(c, (\Delta x, \Delta y), (\Delta w, \Delta h)) = (1, 0.81, 0.625, 0.40, 0.91)$.

DIGITAL REPOSITORY UNIVERSITAS JEMBER

DIGITAL REPOSITORY UNIVERSITAS JEMBER



$$\Delta x = \frac{x - x_a}{64} = \frac{244 - 192}{64} \approx 0.81$$

$$\Delta y = \frac{y - y_a}{64} = \frac{232 - 192}{64} \approx 0.625$$

$$\Delta w = \frac{w}{448} = \frac{180}{448} \approx 0.40$$

$$\Delta h = \frac{h}{448} = \frac{410}{448} \approx 0.91$$

Gambar 4.2 Proses anotasi data input

4.2 Proses Deteksi Objek dengan YOLO

Jaringan YOLOv1 mempunyai 26 layer yang terdiri dari layer konvolusi, batch mormalization layer, Leaky ReLU layer, maxpooling layer, flatten layer route. Diasumsikan bahwa gambar input merupakan frame ROI yang telah didefinisikan sebelumnya. Algoritma jaringan YOLOv1 secara matematis akan diilustrasikan sebagai berikut.

4.2.1 Convolutional Layer

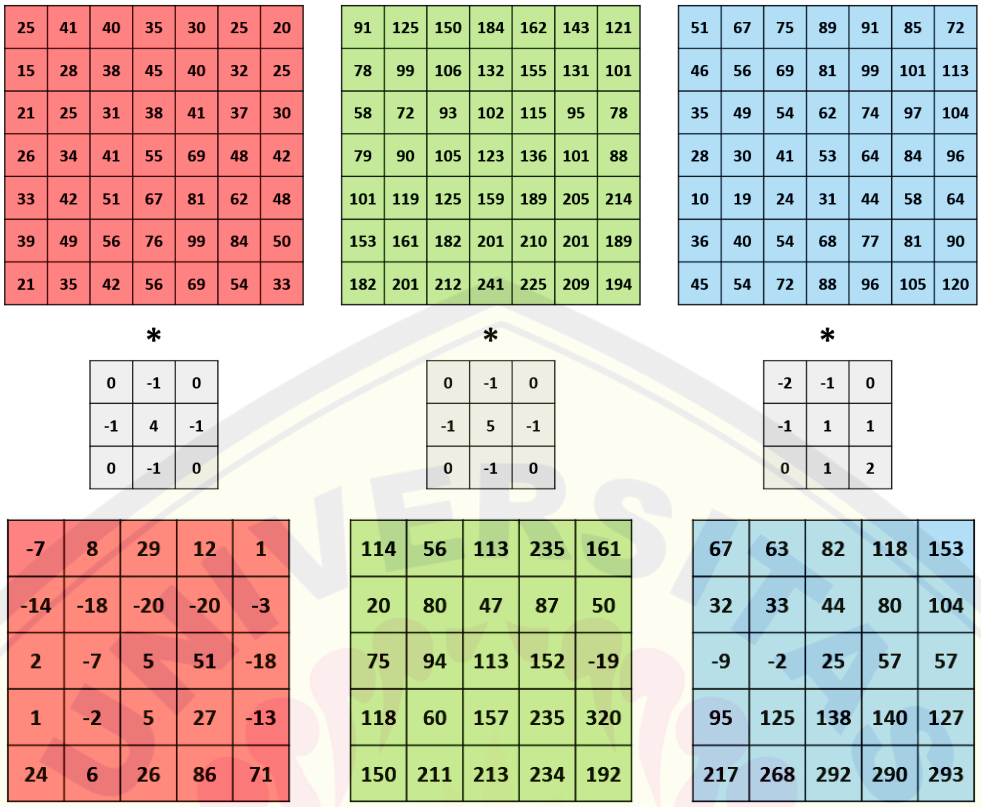
Pada convolutional layer terdapat 3 proses di dalamnya, yaitu Convolutional 2D Layer, Batch Normalization Layer, dan Leaky ReLU Layer.

1. *Convolutional 2D Layer*

Pada layer ini akan dilakukan proses konvolusi untuk setiap channel menggunakan filter konvolusi berukuran persegi pada keseluruhan gambar. Berikut adalah ilustrasi proses konvolusi dengan input berukuran 7×7, *stride* = 1, *filter* berukuran 3×3 menghasilkan output berukuran 5×5.

DIGITAL REPOSITORY UNIVERSITAS JEMBER

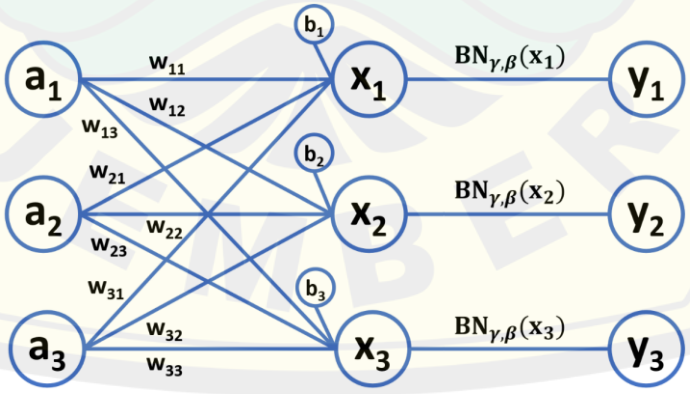
DIGITAL REPOSITORY UNIVERSITAS JEMBER



Gambar 4.3 Ilustrasi proses konvolusi

2. *Batch Normalization Layer*

Layer ini berfungsi untuk meningkatkan stabilitas jaringan dan kecepatan proses training. Proses ini dilakukan pada input di setiap layer konvolusi untuk membantu menjaga distribusi input tetap dalam rentang yang stabil. Berikut adalah ilustrasi proses yang terjadi pada layer pertama.



Gambar 4.4 Ilustrasi proses pada layer pertama

DIGITAL REPOSITORY UNIVERSITAS JEMBER

DIGITAL REPOSITORY UNIVERSITAS JEMBER

$a_1, a_2,$ dan a_3 secara berturut-turut merepresentasikan matriks dari *red image, green image,* dan *blue image* dari suatu gambar. Misalkan ketiga matriks tersebut sebagai berikut.

$$a_1 = \begin{pmatrix} 1 & 4 & 5 & 0 & 8 & 1 \\ 5 & 3 & 0 & 4 & 6 & 3 \\ 2 & 8 & 5 & 0 & 7 & 5 \\ 0 & 9 & 1 & 5 & 4 & 7 \\ 4 & 2 & 6 & 2 & 0 & 9 \\ 5 & 3 & 8 & 2 & 1 & 9 \end{pmatrix}, \; a_2 = \begin{pmatrix} 7 & 6 & 5 & 1 & 0 & 8 \\ 1 & 2 & 4 & 0 & 7 & 6 \\ 6 & 4 & 7 & 9 & 1 & 4 \\ 2 & 0 & 7 & 6 & 2 & 3 \\ 5 & 1 & 4 & 3 & 0 & 1 \\ 6 & 2 & 8 & 5 & 3 & 8 \end{pmatrix}, a_3 = \begin{pmatrix} 9 & 0 & 2 & 5 & 3 & 5 \\ 8 & 6 & 2 & 4 & 4 & 6 \\ 0 & 4 & 1 & 3 & 5 & 3 \\ 4 & 8 & 2 & 6 & 0 & 4 \\ 7 & 2 & 5 & 0 & 7 & 8 \\ 8 & 2 & 7 & 5 & 1 & 4 \end{pmatrix}$$

Selanjutnya bobot (*weights*) dan bias dimisalkan sebagai berikut.

$$w_{11} = w_{23} = w_{31} = \begin{pmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{pmatrix}, \qquad w_{12} = w_{21} = w_{33} = \begin{pmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{pmatrix},$$

$$w_{13} = w_{22} = w_{32} = \begin{pmatrix} -2 & -1 & 0 \\ -1 & 1 & -1 \\ 0 & -1 & -2 \end{pmatrix}, \qquad \text{dan} \quad b_1 = b_2 = b_3 = 1$$

Selanjutnya dilakukan zero padding pada matriks $a_1, a_2,$ dan a_3 agar didapat feature map dengan ukuran yang sama dengan matriks awal. Besar *zero padding* (P) yang diterapkan dihitung dengan Persamaan 4.5 dimana F menyatakan ukuran *weights*.

$$P = \frac{F-1}{2} \tag{4.5}$$

Diperoleh :

$$P = \frac{3-1}{2} = 1$$

Diperoleh matriks $a_1, a_2,$ dan a_3 sebagai berikut.

$$a_1 = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 4 & 5 & 0 & 8 & 1 & 0 \\ 0 & 5 & 3 & 0 & 4 & 6 & 3 & 0 \\ 0 & 2 & 8 & 5 & 0 & 7 & 5 & 0 \\ 0 & 0 & 9 & 1 & 5 & 4 & 7 & 0 \\ 0 & 4 & 2 & 6 & 2 & 0 & 9 & 0 \\ 0 & 5 & 3 & 8 & 2 & 1 & 9 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \; a_2 = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 7 & 6 & 5 & 1 & 0 & 0 \\ 0 & 1 & 2 & 4 & 0 & 7 & 0 \\ 0 & 6 & 4 & 7 & 9 & 1 & 0 \\ 0 & 2 & 0 & 7 & 6 & 2 & 0 \\ 0 & 5 & 1 & 4 & 3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}, a_3 = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 9 & 0 & 2 & 5 & 3 & 5 & 0 \\ 0 & 8 & 6 & 2 & 4 & 4 & 6 & 0 \\ 0 & 0 & 4 & 1 & 3 & 5 & 3 & 0 \\ 0 & 4 & 8 & 2 & 6 & 0 & 4 & 0 \\ 0 & 7 & 2 & 5 & 0 & 7 & 8 & 0 \\ 0 & 8 & 2 & 7 & 5 & 1 & 4 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Selanjutnya akan dihitung $x_1, x_2,$ dan x_3 menggunakan Persamaan 4.6 dengan $*$ menyatakan operasi konvolusi dan n merupakan ordo matriks a_i.

$$x_j = \sum_{i=1}^{n} (a_i * w_{ij} + b) \tag{4.6}$$

DIGITAL REPOSITORY UNIVERSITAS JEMBER

DIGITAL REPOSITORY UNIVERSITAS JEMBER

$$x_1 = a_1 * w_{11} + a_2 * w_{21} + a_3 * w_{31} + b =$$

$$\begin{pmatrix} -5 & 7 & 16 & -17 & 25 & -7 \\ 14 & -5 & -17 & 10 & 2 & 0 \\ -5 & 13 & 11 & -21 & 13 & 3 \\ -15 & 25 & -21 & 13 & -3 & 10 \\ 9 & -14 & 11 & -5 & -16 & 20 \\ 13 & -3 & 21 & -3 & -7 & 26 \end{pmatrix} + \begin{pmatrix} 47 & 29 & 27 & -8 & -22 & 51 \\ -17 & -24 & -2 & -34 & 27 & 28 \\ 39 & 3 & 24 & 38 & -29 & 13 \\ 0 & -36 & 22 & 15 & -11 & 16 \\ 29 & -26 & 0 & -11 & -31 & -8 \\ 40 & -8 & 49 & 22 & 7 & 60 \end{pmatrix} +$$

$$\begin{pmatrix} 28 & -17 & 1 & 11 & -2 & 11 \\ 17 & 10 & -5 & 2 & -2 & 12 \\ -16 & 1 & -7 & -4 & 10 & -3 \\ 1 & 20 & -12 & 19 & -22 & 5 \\ 14 & -14 & 9 & -23 & 19 & 17 \\ 23 & -9 & 16 & 12 & -12 & 7 \end{pmatrix} + \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix} =$$

$$\begin{pmatrix} 71 & 20 & 45 & -13 & 2 & 56 \\ 15 & -18 & -23 & -21 & 28 & 41 \\ 19 & 18 & 29 & 14 & -5 & 14 \\ -13 & 10 & -10 & 48 & -35 & 32 \\ 53 & -53 & 21 & -38 & -27 & 30 \\ 77 & -19 & 87 & 32 & -11 & 94 \end{pmatrix}$$

$$x_2 = a_1 * w_{12} + a_2 * w_{22} + a_3 * w_{32} + b =$$

$$\begin{pmatrix} -4 & 18 & 29 & -23 & 50 & -9 \\ 22 & -6 & -29 & 1 & 20 & -3 \\ -9 & 39 & 10 & -32 & 22 & 13 \\ -25 & 44 & -29 & 15 & -3 & 31 \\ 13 & -20 & 16 & -11 & -39 & 51 \\ 31 & -1 & 49 & -1 & -14 & 62 \end{pmatrix} + \begin{pmatrix} -4 & -16 & -6 & -18 & -28 & 2 \\ -22 & -41 & -40 & -33 & 10 & -13 \\ -1 & -27 & -33 & -17 & -27 & -20 \\ -11 & -34 & -24 & -29 & -28 & -6 \\ -8 & -30 & -25 & -32 & -37 & -14 \\ -1 & -23 & -5 & -17 & -16 & 4 \end{pmatrix} +$$

$$\begin{pmatrix} -11 & -21 & -13 & -12 & -23 & -4 \\ -15 & -28 & -17 & -24 & -30 & -12 \\ -32 & -31 & -34 & -17 & -21 & -20 \\ -15 & -14 & -26 & -15 & -44 & -17 \\ -11 & -42 & -32 & -29 & -22 & -7 \\ -1 & -29 & -9 & -13 & -15 & 19 \end{pmatrix} + \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix} =$$

$$\begin{pmatrix} -18 & -18 & 11 & -52 & 0 & -10 \\ -14 & -74 & -85 & -55 & -19 & -27 \\ -41 & -18 & -56 & -65 & -25 & -26 \\ -50 & -3 & -78 & -28 & -74 & 9 \\ -5 & -91 & -40 & -71 & -97 & 31 \\ 30 & -52 & 36 & -30 & -44 & 48 \end{pmatrix}$$

$$x_3 = a_1 * w_{13} + a_2 * w_{23} + a_3 * w_{33} + b =$$

$$\begin{pmatrix} -14 & -5 & -7 & -29 & -5 & -10 \\ -17 & -26 & -25 & -26 & -26 & -25 \\ -29 & -23 & -20 & -29 & -30 & -24 \\ -19 & -18 & -44 & -12 & -33 & -25 \\ -9 & -36 & -29 & -15 & -44 & -15 \\ -2 & -20 & -7 & -21 & -14 & -1 \end{pmatrix} + \begin{pmatrix} 21 & 10 & 9 & -1 & -16 & 26 \\ -11 & -7 & 2 & -21 & 21 & 5 \\ 17 & 1 & 4 & 22 & -18 & 6 \\ -3 & -14 & 11 & 3 & -2 & 5 \\ 11 & -7 & -3 & -3 & -9 & -7 \\ 17 & -7 & 21 & 6 & -1 & 28 \end{pmatrix} +$$

$$\begin{pmatrix} 58 & -27 & -1 & 25 & 0 & 27 \\ 45 & 22 & -9 & 7 & -2 & 28 \\ -30 & 1 & -27 & 0 & 10 & 5 \\ 11 & 39 & -13 & 25 & -36 & 9 \\ 32 & -27 & 8 & -33 & 28 & 48 \\ 53 & -13 & 42 & 20 & -16 & 16 \end{pmatrix} + \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix} =$$

$$\begin{pmatrix} 66 & -21 & 2 & -4 & -20 & 44 \\ 18 & -10 & -31 & -39 & -6 & 9 \\ -41 & -20 & -42 & -6 & -37 & -12 \\ -10 & 8 & -45 & 17 & -70 & -10 \\ 35 & -69 & -23 & -50 & -24 & 27 \\ 69 & -39 & 57 & 6 & -30 & 44 \end{pmatrix}$$

Kemudian dihitung nilai dari μ_B dan σ_B^2 menggunakan Persamaan 4.7 dan 4.8.

DIGITAL REPOSITORY UNIVERSITAS JEMBER

DIGITAL REPOSITORY UNIVERSITAS JEMBER

$$\mu_B = \frac{1}{m}\sum_{i=1}^{m} x_i \tag{4.7}$$

$$\sigma_B^2 = \frac{1}{m}\sum_{i=1}^{m}(x_i - \mu_B)^2 \tag{4.8}$$

Diperoleh

$$\mu_B = \frac{1}{3}\sum_{i=1}^{3} x_i = \frac{1}{3}(x_1 + x_2 + x_3)$$

$$= \frac{1}{3}\times\begin{pmatrix} 119 & -19 & 58 & -69 & -18 & 90 \\ 19 & -102 & -139 & -115 & 3 & 23 \\ -63 & -20 & -69 & -57 & -67 & -24 \\ -73 & 15 & -133 & 37 & -179 & 31 \\ 83 & -213 & -42 & -159 & -148 & 88 \\ 176 & -110 & 180 & 8 & -85 & 186 \end{pmatrix}$$

$$= \begin{pmatrix} 39.67 & -6.33 & 19.33 & -23 & -6 & 30 \\ 6.33 & -34 & -46.33 & -38.33 & 1 & 7.67 \\ -21 & -6{,}67 & -23 & -19 & -22.33 & -8 \\ -24.33 & 5 & -44.33 & 12.33 & -59.67 & 10.33 \\ 27.67 & -71 & -14 & -53 & -49.33 & 29.33 \\ 58.67 & -36.67 & 60 & 2.67 & -28.33 & 62 \end{pmatrix}$$

$$\sigma_B^2 = \frac{1}{3}\sum_{i=1}^{3}(x_i - \mu_B)^2 = \frac{1}{3}((x_1 - \mu_B)^2 + (x_2 - \mu_B)^2 + (x_3 - \mu_B)^2)$$

$$= \begin{pmatrix} 1666.9 & 348.2 & 342.9 & 434 & 98.7 & 82.4 \\ 208.2 & 810.7 & 758.2 & 192.9 & 392.7 & 771.6 \\ 800 & 304.9 & 1384.7 & 1124.7 & 174.2 & 274.7 \\ 330.9 & 32.7 & 770.9 & 973.6 & 306.9 & 294.9 \\ 587.6 & 242.7 & 660.7 & 186 & 1137.6 & 2.9 \\ 421.6 & 184.2 & 438 & 646.2 & 182.9 & 514.7 \end{pmatrix}$$

Selanjutnya akan dicari nilai dari $\hat{x}_1, \hat{x}_2$, dan $\hat{x}_3$ menggunakan Persamaan 4.9. Diasumsikan $\epsilon = 0.001$ dengan ϵ merupakan sebarang bilangan positif yang nilainya mendekati nol untuk mempertahankan stabilitas numerik.

$$\hat{x}_i = \frac{x_i - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}} \tag{4.9}$$

$$\hat{x}_1 = \frac{x_1 - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}} = \begin{pmatrix} 0.767 & 1.411 & 1.386 & 0.480 & 0.805 & 0.906 \\ 0.600 & 0.562 & 0.847 & 1.248 & 1.362 & 1.200 \\ 1.414 & 1.412 & 1.397 & 0.984 & 1.313 & 1.327 \\ 0.623 & 0.874 & 1.237 & 1.143 & 1.408 & 1.262 \\ 1.045 & 1.155 & 1.361 & 1.099 & 0.662 & 0.392 \\ 0.893 & 1.302 & 1.290 & 1.154 & 1.282 & 1.410 \end{pmatrix}$$

$$\hat{x}_2 = \frac{x_2 - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}} = \begin{pmatrix} -1.412 & -0.625 & -0.450 & -1.392 & 0.604 & -1.393 \\ -1.409 & -1.404 & -1.404 & -1.200 & -1.009 & -1.248 \\ -0.707 & -0.649 & -0.886 & -1.371 & -0.202 & -1.086 \\ -1.411 & -1.399 & -1.212 & -1.292 & -0.818 & -0.077 \\ -1.347 & -1.284 & -1.011 & -1.319 & -1.413 & 0.980 \\ -1.396 & -1.129 & -1.147 & -1.285 & -1.158 & -0.617 \end{pmatrix}$$

DIGITAL REPOSITORY UNIVERSITAS JEMBER

DIGITAL REPOSITORY UNIVERSITAS JEMBER

$$\hat{x}_3 = \frac{x_3 - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}} = \begin{pmatrix} 0.645 & -0.786 & -0.936 & 0.912 & -1.409 & 0.487 \\ 0.808 & 0.842 & 0.556 & -0.048 & -0.353 & 0.048 \\ -0.707 & -0.763 & -0.510 & 0.387 & -1.111 & -0.241 \\ 0.788 & 0.524 & -0.024 & 0.149 & -0.589 & -1.184 \\ 0.302 & 0.128 & -0.350 & 0.220 & 0.751 & -1.373 \\ 0.503 & -0.172 & -0.143 & 0.131 & -0.123 & -0.793 \end{pmatrix}$$

Pada batch normalization layer, parameter yang dipelajari atau diperbarui nilainya adalah parameter β dan γ. Diasumsikan nilai awal untuk β dan γ yaitu $\beta = 0.9$ dan $\gamma = 0.0005$. Nilai $y_1, y_2,$ dan y_3 dapat dicari menggunakan Persamaan 4.10 sebagai berikut.

$$y_i = \gamma \hat{x}_i + \beta \tag{4.10}$$

Parameter β dan γ dapat diatur menjadi $\beta = \mu_B$ dan $\gamma = \sqrt{\sigma_B^2 + \epsilon}$ untuk memperoleh nilai aktivasi yang asli jika itu adalah hal yang optimal untuk dilakukan. Dengan begitu, diperoleh nilai output pada layer ini yaitu $y_i = x_i$. Layer berikutnya adalah Leaky ReLU Layer yang akan dijelaskan pada poin 3.

3. Leaky ReLU Layer

Fungsi aktivasi yang digunakan pada jaringan YOLOv1 adalah Leaky ReLU yang dinyatakan dalam Persamaan 4.11 berikut.

$$f(x) = \begin{cases} x & x \geq 0 \\ 0.01x & x < 0 \end{cases} \tag{4.11}$$

Diperoleh hasil aktivasi dengan Leaky ReLU sebagai berikut.

$y_i = LeakyReLU(x_i)$

$$y_1 = \begin{pmatrix} 71 & 20 & 45 & -0.13 & 2 & 56 \\ 15 & -0.18 & -0.23 & -0.21 & 28 & 41 \\ 19 & 18 & 29 & 14 & -0.05 & 14 \\ -0.13 & 10 & -0.10 & 48 & -0.35 & 32 \\ 53 & -0.53 & 21 & -0.38 & -0.27 & 30 \\ 77 & -0.19 & 87 & 32 & -0.11 & 94 \end{pmatrix}, y_2 = \begin{pmatrix} -0.18 & -0.18 & 11 & -0.52 & 0 & -0.10 \\ -0.14 & -0.74 & -0.85 & -0.55 & -0.19 & -0.27 \\ -0.41 & -0.18 & -0.56 & -0.65 & -0.25 & -0.26 \\ -0.50 & -0.03 & -0.78 & -0.28 & -0.74 & 9 \\ -0.05 & -0.91 & -0.4 & -0.71 & -0.97 & 31 \\ 30 & -0.52 & 36 & -0.30 & -0.44 & 48 \end{pmatrix}$$

$$y_3 = \begin{pmatrix} 66 & -0.21 & 2 & -0.04 & -0.20 & 44 \\ 18 & -0.10 & -0.31 & -0.39 & -0.06 & 9 \\ -0.41 & -0.20 & -0.42 & -0.06 & -0.37 & -0.12 \\ -0.10 & 8 & -0.45 & 17 & -0.70 & -0.10 \\ 35 & -0.69 & -0.23 & -0.50 & -0.24 & 27 \\ 69 & -0.39 & 57 & 6 & -0.30 & 44 \end{pmatrix}$$

4.2.2 MaxPooling Layer

Pada layer berikutnya terdapat Maxpooling Layer dengan ukuran 2×2 dengan *stride* bernilai 2. Diperoleh sebagai berikut.

$$y_1' = \begin{pmatrix} 71 & 45 & 56 \\ 19 & 48 & 32 \\ 77 & 87 & 94 \end{pmatrix}, \quad y_2' = \begin{pmatrix} -0.14 & 11 & 0 \\ -0.03 & -0.28 & 9 \\ 30 & 36 & 48 \end{pmatrix}, \quad y_3' = \begin{pmatrix} 66 & 2 & 44 \\ 8 & 17 & -0.1 \\ 69 & 57 & 44 \end{pmatrix}$$

DIGITAL REPOSITORY UNIVERSITAS JEMBER

DIGITAL REPOSITORY UNIVERSITAS JEMBER

4.2.3 Flatten Layer

Pada layer ini matriks output dari layer sebelumnya diubah bentuknya menjadi vektor.

$$\hat{y}_1 = Flatten(y_1') = (71 \quad 45 \quad 56 \quad 19 \quad 48 \quad 32 \quad 77 \quad 87 \quad 94)^T$$

$$\hat{y}_2 = Flatten(y_2') = (-0.14 \quad 11 \quad 0 \quad -0.03 \quad -0.28 \quad 9 \quad 30 \quad 36 \quad 48)^T$$

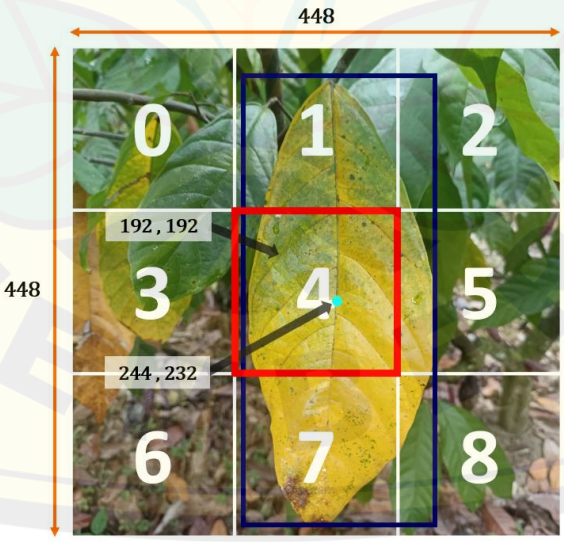
$$\hat{y}_3 = Flatten(y_3') = (66 \quad 2 \quad 44 \quad 8 \quad 17 \quad -0.1 \quad 69 \quad 57 \quad 44)^T$$

4.2.4 Fully Connected Layer

Convolutiunal layer awal dari jaringan mengekstrak fitur dari citra, sementara *fully connected layer* memprediksi probabilitas dan koordinat output. Gambar input diproses pada semua fitur objek detektor yang telah dipelajari oleh convolutional layer. Kemudian dilakukan prediksi dengan menggunakan fully connected layer berukuran 1×1. Sehingga diperoleh output dari YOLO adalah *feature map*. Karena YOLO telah menggunakan konvolusi 1 × 1, maka ukuran *feature map* persis sama dengan ukuran peta fitur sebelumnya. Selanjutnya, output yang diperoleh dilakukan perhitungan *loss* terhadap *ground truth*.

4.2.5 *Loss Function*

Pada langkah ini akan dicari nilai loss function dari model YOLOv1.



Gambar 4.5 Input gambar proses perhitungan *loss*

DIGITAL REPOSITORY UNIVERSITAS JEMBER

DIGITAL REPOSITORY UNIVERSITAS JEMBER

Perhitungan loss function dari model jaringan YOLOv1 diterapkan pada gambar yang dibagi menjadi 3×3 grid sel. Pada Gambar 4.5 dapat dilihat bahwa objek gambar diwakili oleh grid sel ke-4. Dimisalkan pada sel 4 memiliki parameter untuk ground truth dan bounding box (kotak prediksi) sebagai berikut.

- *Ground truth* $= (1, 0.81245, 0.62541, 0.40821, 0.91001)$ yang menyatakan kelas objek, koordinat center box, dan ukuran box.
- *Bounding box* $= (0.95, 0.80998, 0.62777, 0.41035, 0.90565, 0.856)$ yang menyatakan kelas objek, koordinat center box, dan probabilitas kelas objek.
- Kotak prediksi pada sel $i = 4$ yang secara berurutan menyatakan *confidence,* koordinat *center box*, ukuran box, dan kelas objek.
- Kotak prediksi 1 $(B_1) = (0.8, 0.8055, 0.6541, 0.4235, 0.8966,\ 0.95)$
- Kotak prediksi 2 $(B_2) = (0.8, 0.8025, 0.6197, 0.4199, 0.8987, 0.95)$

Selanjutnya dapat dihitung nilai dari loss pada model jaringan YOLOv1 sebagai berikut.

a. *Localization Loss*

Objek hanya berada pada sel grid ke-4 $(i = 4)$ sehingga nilai $1_{ij}^{obj} = 1$ pada $i = 4$ dan $1_{ij}^{obj} = 0$ pada sel $i \neq 4$, maka dengan menggunakan Persamaan 2.7 diperoleh *localization loss* sebagai berikut.

$$Local.\ Loss = \lambda_{coord} \sum_{j=0}^{2} 1_{4j}^{obj} [(x_4 - \hat{x}_4)^2 + (y_4 - \hat{y}_4)^2]$$

$$+ \lambda_{coord} \sum_{j=0}^{2} 1_{4j}^{obj} \left[\left(\sqrt{w_4} - \sqrt{\hat{w}_4}\right)^2 + \left(\sqrt{h_i} - \sqrt{\hat{h}_4}\right)^2 \right]$$

$$= 5 \times [(0.80998 - 0.81245)^2 + (0.62777 - 0.62541)^2]$$

$$+ 5 \times \left[\left(\sqrt{0.41035} - \sqrt{0.40821}\right)^2 + \left(\sqrt{0.90565} - 91001\right)^2 \right]$$

$$= 5 \times (0.00285 + 0.00042) = 5 \times 0.003292 = 0.01646.$$

b. *Confidence Loss*

Objek hanya berada pada sel grid ke-4 $(i = 4)$ sehingga nilai $C_i = \hat{C}_i = 0$ pada $i \neq 4$, maka dengan menggunakan Persamaan 2.8 diperoleh *confidence loss* sebagai berikut.

$$Conf.Loss = \sum_{j=0}^{2} 1_{4j}^{obj} \left(C_4 - \hat{C}_4\right)^2 = 2 \times (0.8 - 1)^2 = 0.08.$$

DIGITAL REPOSITORY UNIVERSITAS JEMBER

DIGITAL REPOSITORY UNIVERSITAS JEMBER

c. *Classification Loss*

Objek hanya berada pada sel grid ke-4 ($i = 4$) sehingga nilai $1_{ij}^{obj} = 0$ pada sel $i \neq 4$, , maka dengan menggunakan Persamaan 2.9 diperoleh *classification loss* sebagai berikut.

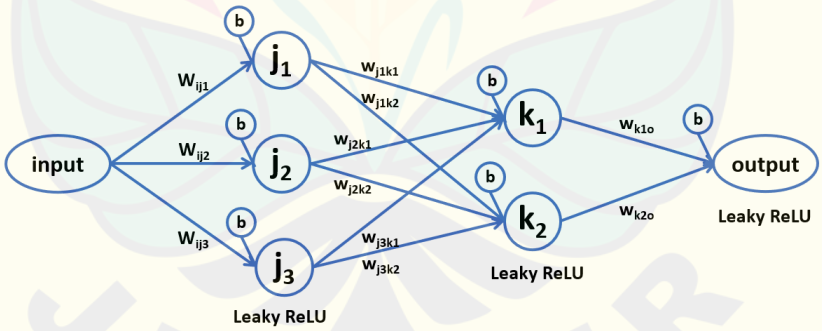
$p_i(c) = 0.95 \times 0.856 = 0.8132$

$$Class.Loss = \sum_{c=0}^{1} 1_{4j}^{obj} (p_i(c) - \hat{p}_i(c))^2 = (0.8132 - 0.91001)^2 = (-0.09681)^2 = 0.00937.$$

Diperoleh nilai total loss function dari YOLOv1 adalah $0.01646 + 0.08 + 0.00937 = 0.10583$

4.2.6 Back Propagation

Algoritma backpropagation merupakan salah satu langkah dalam proses training data untuk memperbarui nilai parameter. Langkah ini dilakukan dengan menyesuaikan bobot untuk memperkecil nilai error berdasarkan perbedaan nilai output dan target yang diinginkan. Tahap ini memperbarui nilai *weights* dan *bias* pada saat *forward pass.* Misalkan jaringan YOLOv1 dinyatakan sebagai jaringan pada Gambar 4.6. Jaringan ini memiliki 2 hidden layer dengan fungsi aktivasi Leaky ReLU. Weight dilambangkan dengan w dan bias diasumsikan bernilai sama dan dilambangkan sebagai b.



Gambar 4.6 Iustrasi jaringan YOLO dalam proses *backpropagation*

$$input = [4.0]\,,\ b = [1.0]\,,\ output = [2.5],$$

$$w_{ij} = [w_{ij1} \quad w_{ij2} \quad w_{ij3}] = [0.1 \quad 0.2 \quad 0.3]$$

$$w_{jk} = \begin{bmatrix} w_{j1k1} & w_{j2k1} \\ w_{j2k1} & w_{j2k2} \\ w_{j3k1} & w_{j2k3} \end{bmatrix} = \begin{bmatrix} 0.3 & 0.1 \\ 0.2 & 0.2 \\ 0.1 & 0.3 \end{bmatrix}, \qquad w_{ko} = \begin{bmatrix} w_{k_1o} \\ w_{k_2o} \end{bmatrix} = \begin{bmatrix} 0.2 \\ 0.4 \end{bmatrix}$$

DIGITAL REPOSITORY UNIVERSITAS JEMBER

DIGITAL REPOSITORY UNIVERSITAS JEMBER

Selanjutnya akan diilustrasikan proses perhitungan yang terjadi dalam proses backpropagation yang terdiri dari *forward pass, backward pass,* dan *update stochastic gradient descent (SGD)*.

a. *Forward Pass* 1 : Input → Hidden Layer 1

Pada tahap ini terjadi perkalian dot product antara input dan weight kemudian dijumlahkan dengan bias.

$$[j_{1in} \quad j_{2in} \quad j_{3in}] = [input] \times [w_{ij1} \quad w_{ij2} \quad w_{ij3}] + [b \quad b \quad b]$$
$$= [4.0] \times [0.1 \quad 0.2 \quad 0.3] + [1.0 \quad 1.0 \quad 1.0]$$
$$= [0.4 \quad 0.8 \quad 1.2] + [1.0 \quad 1.0 \quad 1.0]$$
$$= [1.4 \quad 1.8 \quad 2.2]$$

Selanjutnya, masing-masing nilai dari j_{in} akan dikenakan dengan fungsi aktivasi, yaitu Leaky ReLU seperti pada Persamaan 4.11 sebagai berikut.

$$[j_{1out} \quad j_{2out} \quad j_{3out}] = LeakyReLU([j_{1in} \quad j_{2in} \quad j_{3in}])$$
$$= LeakyReLU([1.4 \quad 1.8 \quad 2.2])$$
$$= [1.4 \quad 1.8 \quad 2.2]$$

b. *Forward Pass* 2 : Hidden Layer 2 → Output

Pada tahap ini terjadi perkalian dot product antara input dan weight kemudian dijumlahkan dengan bias.

$$[k_{1in} \quad k_{2in}] = [j_{1out} \quad j_{2out} \quad j_{3out}] \times \begin{bmatrix} w_{j1k1} & w_{j2k1} \\ w_{j2k1} & w_{j2k2} \\ w_{j3k1} & w_{j2k3} \end{bmatrix} + [b \quad b]$$
$$= [1.4 \quad 1.8 \quad 2.2] \times \begin{bmatrix} 0.3 & 0.1 \\ 0.2 & 0.2 \\ 0.1 & 0.3 \end{bmatrix} + [1.0 \quad 1.0]$$
$$= [1.0 \quad 1.16] + [1.0 \quad 1.0]$$
$$= [2.0 \quad 2.26]$$

Selanjutnya, masing-masing nilai dari j_{in} akan dikenakan dengan fungsi aktivasi Leaky ReLU pada Persamaan 4.11 sebagai berikut.

$$[k_{1out} \quad k_{2out}] = LeakyReLU([k_{1in} \quad k_{2in}])$$
$$= LeakyReLU([2.0 \quad 2.26])$$
$$= [2.0 \quad 2.26]$$

c. *Forward Pass* 3 : Hidden Layer 1 → Hidden Layer 2

Pada tahap ini terjadi perkalian dot product antara input dan weight kemudian dijumlahkan dengan bias.

DIGITAL REPOSITORY UNIVERSITAS JEMBER

DIGITAL REPOSITORY UNIVERSITAS JEMBER

$$[o_{in}] = [k_{1out} \quad k_{2out}] \times \begin{bmatrix} w_{k_1o} \\ w_{k_2o} \end{bmatrix} + [b]$$

$$= [2.0 \quad 2.26] \times \begin{bmatrix} 0.2 \\ 0.4 \end{bmatrix} + [1.0]$$

$$= [1.304] + [1.0] = [2.304]$$

Selanjutnya, masing-masing nilai dari j_{in} akan dikenakan dengan fungsi aktivasi, yaitu Leaky ReLU sebagai berikut.

$[o_{out}] = LeakyReLU([o_{in}]) = LeakyReLU([2.304]) = [2.304]$

d. *Loss : Squared Error*

Selanjutnya akan dicari nilai *loss* menggunakan *squared error*.

$$loss = \frac{1}{2}(Prediksi - Target)^2 = \frac{1}{2}(2.304 - 2.5)^2 = \frac{1}{2}(-0.196)^2 = 0.0192$$

e. Derivatif Fungsi Aktivasi

Kemudian dicari derivatif dari fungsi aktivasi Leaky ReLU pada Persamaan 4.11 sehingga diperoleh derivatif fungsi aktivasi sebagai berikut.

$$\frac{\partial y}{\partial x} = \begin{cases} 1 & x \geq 0 \\ 0.01 & x < 0 \end{cases}$$

f. *Backward Pass* : Output → Hidden Layer 2

Proses ini bertujuan untuk mengupdate nilai parameter. Akan dicontohkan untuk mengupdate nilai dari w_{k1o} dengan menggunakan prinsip aturan rantai sebagai berikut.

$$\frac{\partial loss}{\partial w_{k1o}} = \frac{\partial loss}{\partial o_{out}} \times \frac{\partial o_{out}}{\partial o_{in}} \times \frac{\partial o_{in}}{\partial w_{k1o}}$$

Pertama akan dicari *gradient loss function* terhadap output, yaitu derivatif parsial dari *loss function* terhadap output.

$$loss = \frac{1}{2}(output - o_{out})^2$$

$$\frac{\partial loss}{\partial o_{out}} = \frac{\partial \left(\frac{1}{2}(output - o_{out})^2\right)}{\partial o_{out}}$$

$$= 2 \times \frac{1}{2} \times (-1) \times (output - o_{out})$$

$$= o_{out} - output$$

$$= 2.304 - 2.5 = -0.196$$

Kemudian akan dicari gradient o_{out} terhadap o_{in}. Pada o_{out} fungsi yang digunakan adalah fungsi aktivasi Leaky ReLU sehingga diperoleh sebagai berikut.

DIGITAL REPOSITORY UNIVERSITAS JEMBER

DIGITAL REPOSITORY UNIVERSITAS JEMBER

$$\frac{\partial o_{out}}{\partial o_{in}} = \begin{cases} 1 & o_{in} \geq 0 \\ 0.01 & o_{in} < 0 \end{cases}$$

Nilai $o_{in} = 2.304 \geq 0$ sehingga diperoleh

$$\frac{\partial o_{out}}{\partial o_{in}} = 1$$

Selanjutnya akan dicari gradient o_{in} terhadap w_{k1o}, w_{k1o}, dan bias b.

$$o_{in} = w_{k1o} k_{1out} + w_{k2o} k_{2out} + b$$

$$\frac{\partial o_{in}}{\partial w_{k1o}} = k_{1out}$$

$$\frac{\partial o_{in}}{\partial w_{k2o}} = k_{2out}$$

$$\frac{\partial o_{in}}{\partial b} = 1$$

Kemudian akan dicari *gradient loss* terhadap *weights* dan bias dengan menerapkan aturan rantai.

$$\frac{\partial loss}{\partial w_{k1o}} = \frac{\partial loss}{\partial o_{out}} \times \frac{\partial o_{out}}{\partial o_{in}} \times \frac{\partial o_{in}}{\partial w_{k1o}} = -0.196 \times 1 \times 2.0 = -0.392$$

$$\frac{\partial loss}{\partial w_{k2o}} = \frac{\partial loss}{\partial o_{out}} \times \frac{\partial o_{out}}{\partial o_{in}} \times \frac{\partial o_{in}}{\partial w_{k2o}} = -0.196 \times 1 \times 2.26 = -0.44296$$

$$\frac{\partial loss}{\partial b} = \frac{\partial loss}{\partial o_{out}} \times \frac{\partial o_{out}}{\partial o_{in}} \times \frac{\partial o_{in}}{\partial b} = -0.196 \times 1 \times 1 = -0.196$$

g. Update SGD : Output → Hidden Layer 2

Algoritma *Stochastic Gradient Descent* (SGD) ini digunakan untuk update nilai parameter dalam training, yaitu weights dan bias. Hyperparameter learning rate atau alpha (α) diterapkan dalam algoritma ini. Pada YOLOv1 ditetapkan nilai parameter $\alpha = 0.01$. Nilai parameter yang diupdate dapat dihitung menggunakan Persamaan 4.12 sebagai berikut.

$$p' = p - \alpha \left(\frac{\partial loss}{\partial p} \right) \tag{4.12}$$

$$w_{k1o}{}' = w_{k1o} - \alpha \left(\frac{\partial loss}{\partial w_{k1o}} \right) = 0.2 - \left(0.01 \times (-0.392)\right) = 0.204$$

$$w_{k2o}{}' = w_{k2o} - \alpha \left(\frac{\partial loss}{\partial w_{k2o}} \right) = 0.4 - \left(0.01 \times (-0.44296)\right) = 0.405$$

$$b' = b - \alpha \left(\frac{\partial loss}{\partial b} \right) = 1.0 - \left(0.01 \times (-0.196)\right) = 1.002$$

h. *Backward Pass* : Hidden Layer 2 → Hidden Layer 1

Dalam mencari nilai *gradient loss* terhadap w_{jk} digunakan aturan rantai pada Persamaan 4.13 sebagai berikut.

DIGITAL REPOSITORY UNIVERSITAS JEMBER

DIGITAL REPOSITORY UNIVERSITAS JEMBER

$$\frac{\partial loss}{\partial w_{jk}} = \frac{\partial loss}{\partial k_{out}} \times \frac{\partial k_{out}}{\partial k_{in}} \times \frac{\partial k_{in}}{\partial w_{jk}} \quad (4.13)$$

Pertama, akan dicari nilai *gradient loss* terhadap k_{1out} dan k_{2out} menggunakan aturan rantai sebagai berikut.

$$\frac{\partial loss}{\partial k_{1out}} = \frac{\partial loss}{\partial o_{out}} \times \frac{\partial o_{out}}{\partial o_{in}} \times \frac{\partial o_{in}}{\partial K_{1out}} = -0.196 \times 1 \times w_{k1o}$$
$$= -0.196 \times 1 \times 0.2 = -0.0392$$
$$\frac{\partial loss}{\partial k_{2out}} = \frac{\partial loss}{\partial o_{out}} \times \frac{\partial o_{out}}{\partial o_{in}} \times \frac{\partial o_{in}}{\partial k_{2out}} = -0.196 \times 1 \times w_{k2o}$$
$$= -0.196 \times 1 \times 0.4 = -0.0784$$

$$\begin{bmatrix} \frac{\partial loss}{\partial k_{1out}} & \frac{\partial loss}{\partial k_{2out}} \end{bmatrix} = [-0.0392 \quad -0.0784]$$

Kemudian akan dicari gradient k_{out} terhadap k_{in}. Pada k_{out} fungsi yang digunakan adalah fungsi aktivasi Leaky ReLU sehingga diperoleh sebagai berikut.

$$\begin{bmatrix} \frac{\partial k_{1out}}{\partial k_{1in}} & \frac{\partial k_{2out}}{\partial k_{2in}} \end{bmatrix} = [0.01 \quad 0.01]$$

Selanjutnya, dicari gradient k_{in} terhadap w_{jk} dan b sebagai berikut.

$$k_{1in} = w_{j1k1} j_{1out} + w_{j2k1} j_{2out} + w_{j3k1} j_{3out} + b$$
$$k_{2in} = w_{j1k2} j_{1out} + w_{j2k2} j_{2out} + w_{j3k2} j_{3out} + b$$

$$\begin{bmatrix} \frac{\partial k_{1in}}{\partial w_{j1k1}} & \frac{\partial k_{1in}}{\partial w_{j2k1}} & \frac{\partial k_{1in}}{\partial w_{j3k1}} \\ \frac{\partial k_{2in}}{\partial w_{j1k2}} & \frac{\partial k_{2in}}{\partial w_{j2k2}} & \frac{\partial k_{2in}}{\partial w_{j3k2}} \end{bmatrix} = \begin{bmatrix} j_{1out} & j_{2out} & j_{3out} \\ j_{1out} & j_{2out} & j_{3out} \end{bmatrix} = \begin{bmatrix} 1.4 & 1.8 & 2.2 \\ 1.4 & 1.8 & 2.2 \end{bmatrix}$$

$$\begin{bmatrix} \frac{\partial k_{1in}}{\partial w_b} & \frac{\partial k_{2in}}{\partial w_b} \end{bmatrix} = [1.0 \quad 1.0]$$

Kemudian dapat dihitung *gradient loss* terhadap w_{jk} dengan menerapkan aturan rantai pada Persamaan 4.11.

$$\begin{bmatrix} \frac{\partial loss}{\partial w_{j1k1}} & \frac{\partial loss}{\partial w_{j2k1}} & \frac{\partial loss}{\partial w_{j3k1}} \\ \frac{\partial loss}{\partial w_{j1k2}} & \frac{\partial loss}{\partial w_{j2k2}} & \frac{\partial loss}{\partial w_{j3k2}} \end{bmatrix}$$
$$= \begin{bmatrix} \frac{\partial loss}{\partial k_{1out}} \times \frac{\partial k_{1out}}{\partial k_{1in}} \times \frac{\partial k_{1in}}{\partial w_{j1k1}} & \frac{\partial loss}{\partial k_{1out}} \times \frac{\partial k_{1out}}{\partial k_{1in}} \times \frac{\partial k_{1in}}{\partial w_{j2k1}} & \frac{\partial loss}{\partial k_{1out}} \times \frac{\partial k_{1out}}{\partial k_{1in}} \times \frac{\partial k_{1in}}{\partial w_{j3k1}} \\ \frac{\partial loss}{\partial k_{2out}} \times \frac{\partial k_{2out}}{\partial k_{2in}} \times \frac{\partial k_{2in}}{\partial w_{j1k2}} & \frac{\partial loss}{\partial k_{2out}} \times \frac{\partial k_{2out}}{\partial k_{2in}} \times \frac{\partial k_{2in}}{\partial w_{j2k2}} & \frac{\partial loss}{\partial k_{2out}} \times \frac{\partial k_{2out}}{\partial k_{2in}} \times \frac{\partial k_{2in}}{\partial w_{j3k2}} \end{bmatrix}$$

DIGITAL REPOSITORY UNIVERSITAS JEMBER

DIGITAL REPOSITORY UNIVERSITAS JEMBER

$$= \begin{bmatrix} -0.0392 \times 0.01 \times 1.4 & -0.0392 \times 0.01 \times 1.8 & -0.0392 \times 0.01 \times 2.2 \\ -0.0784 \times 0.01 \times 1.4 & -0.0784 \times 0.01 \times 1.8 & -0.0784 \times 0.01 \times 2.2 \end{bmatrix}$$

$$= 10^{-3} \times \begin{bmatrix} 0.5488 & 0.7056 & 0.8624 \\ 1.0976 & 1.4112 & 1.7248 \end{bmatrix}$$

Dapat dihitung pula *gradient loss* terhadap b sebagai berikut.

$$\begin{bmatrix} \frac{\partial loss}{\partial b} \\ \frac{\partial loss}{\partial b} \end{bmatrix} = \begin{bmatrix} \frac{\partial loss}{\partial k_{1out}} \times \frac{\partial k_{1out}}{\partial k_{1in}} \times \frac{\partial k_{1in}}{\partial w_b} \\ \frac{\partial loss}{\partial k_{2out}} \times \frac{\partial k_{2out}}{\partial k_{2in}} \times \frac{\partial k_{1in}}{\partial w_b} \end{bmatrix} = \begin{bmatrix} -0.0392 \times 0.01 \times 1 \\ -0.0784 \times 0.01 \times 1 \end{bmatrix} = 10^{-3} \times \begin{bmatrix} 0.392 \\ 0.784 \end{bmatrix}$$

i. Update SGD : Hidden Layer 2 → Hidden Layer 1

Nilai *weights* w_{jk} diupdate menggunakan Persamaan 4.13 Sehingga diperoleh nilai *weights* yang baru sebagai berikut.

$$\begin{bmatrix} w_{j1k1}' & w_{j2k1}' & w_{j3k1}' \\ w_{j1k2}' & w_{j2k2}' & w_{j3k2}' \end{bmatrix} = \begin{bmatrix} 0.3 & 0.2 & 0.1 \\ 0.1 & 0.2 & 0.3 \end{bmatrix}$$

Diperoleh pula nilai bias b yang telah diperbarui sebagai berikut.

$$\begin{bmatrix} b_{jk1}' \\ b_{jk2}' \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

j. Backward Pass : Hidden Layer 1 → Input Layer

$$\frac{\partial loss}{\partial k_{out}} = \frac{\partial loss}{\partial k_{1out}} + \frac{\partial loss}{\partial k_{2out}} = -0.0392 + (-0.0784) = -0.1176$$

$$\frac{\partial k_{out}}{\partial k_{in}} = \frac{\partial k_{1out}}{\partial k_{1in}} + \frac{\partial k_{2out}}{\partial k_{2in}} = 0.01 + 0{,}01 = 0.02$$

$$\frac{\partial k_{in}}{\partial w_{j1k}} = \frac{\partial k_{1in}}{\partial w_{j1k1}} + \frac{\partial k_{2in}}{\partial w_{j1k2}} = 1.4 + 1.4 = 2.8$$

$$\frac{\partial w_{j1k}}{\partial j_{1out}} = \frac{\partial w_{j1k1}}{\partial j_{1out}} + \frac{\partial w_{j1k2}}{\partial j_{1out}} = 1$$

$$\frac{\partial loss}{\partial j_{1out}} = \frac{\partial loss}{\partial k_{out}} \times \frac{\partial k_{out}}{\partial k_{in}} \times \frac{\partial k_{in}}{\partial w_{j1k}} \times \frac{\partial w_{j1k}}{\partial j_{1out}} = -0.1176 \times 0.02 \times 2.8 \times 1 = -0.00658$$

Kemudian akan dicari gradient j_{1out} terhadap j_{1in}.

$$j_{1out} = LeakyReLU(j_{1in}) = max(0.001 j_{1in}, j_{1in})$$

$$\frac{\partial j_{1out}}{\partial j_{1in}} = \begin{cases} 1 & j_{1in} \geq 0 \\ 0.001 & j_{1in} < 0 \end{cases}$$

Karena $j_{1in} = 1.4 \geq 0$, maka diperoleh

$$\frac{\partial j_{1out}}{\partial j_{1in}} = 1$$

Selanjutnya, akan dicari gradient j_{1in} terhadap w_{ij1}

$$j_{1in} = w_{ij1}.input + b$$

DIGITAL REPOSITORY UNIVERSITAS JEMBER

DIGITAL REPOSITORY UNIVERSITAS JEMBER

$$\frac{\partial j_{1in}}{\partial w_{ij1}} = input = 4.0$$

Sehingga dengan menggunakan aturan rantai dapat dihitung gradient loss terhadap w_{ij1}.

$$\frac{\partial loss}{\partial w_{ij1}} = \frac{\partial loss}{\partial j_{1out}} \times \frac{\partial j_{1out}}{\partial j_{1in}} \times \frac{\partial j_{1in}}{\partial w_{ij1}} = -0.00658 \times 1 \times 4 = -0.02632$$

Perhitungan ini akan diterapkan untuk menghitung gradient loss terhadap semua parameter sehingga diperoleh hasil sebagai berikut.

$$\begin{bmatrix} \frac{\partial loss}{\partial w_{ij1}} & \frac{\partial loss}{\partial w_{ij2}} & \frac{\partial loss}{\partial w_{ij3}} \end{bmatrix} = [-0.0263 \quad -0.0338 \quad -0.0414]$$

$$\begin{bmatrix} \frac{\partial loss}{\partial b_{ij1}} & \frac{\partial loss}{\partial b_{ij2}} & \frac{\partial loss}{\partial b_{ij3}} \end{bmatrix} = [-0.00658 \quad -0.00846 \quad -0.01035]$$

k. Update SGD : Hidden Layer 1 → Input Layer

Diperoleh nilai parameter *weights* dan bias yang baru sebagai berikut.

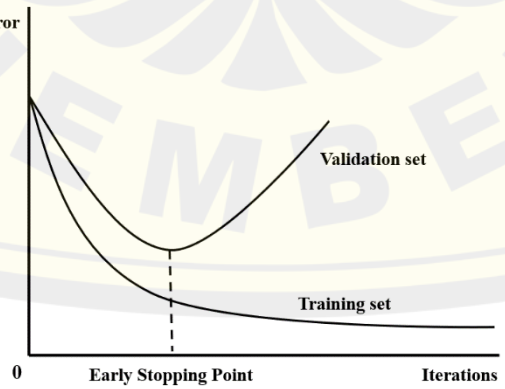
$$[w_{ij1}' \quad w_{ij2}' \quad w_{ij3}'] = [0.100263 \quad 0.200338 \quad 0.300414]$$

$$[b_{ij1}' \quad b_{ij2}' \quad b_{ij3}'] = [1.0000658 \quad 1.0000846 \quad 1.0001035]$$

Semua proses di atas dilakukan secara terus-menerus hingga mencapai iterasi yang telah ditentukan atau nilai error yang diinginkan.

4.1.2 Hasil Uji Coba

Pada tahap ini dilakukan pengujian terhadap proses training untuk memperoleh *trained weights* terbaik. Untuk memperoleh hasil yang sesuai, maka dilakukan pengujian pada proses training dengan menggunakan data *validation*. Proses training pada YOLOv1 sampai dengan YOLOv8 dilakukan untuk memperoleh *trained weights* pada *Early Stopping Point* agar tidak terjadi *overfitting*. *Early Stopping Point* yang dimaksud dapat dilihat pada Gambar 4.7.



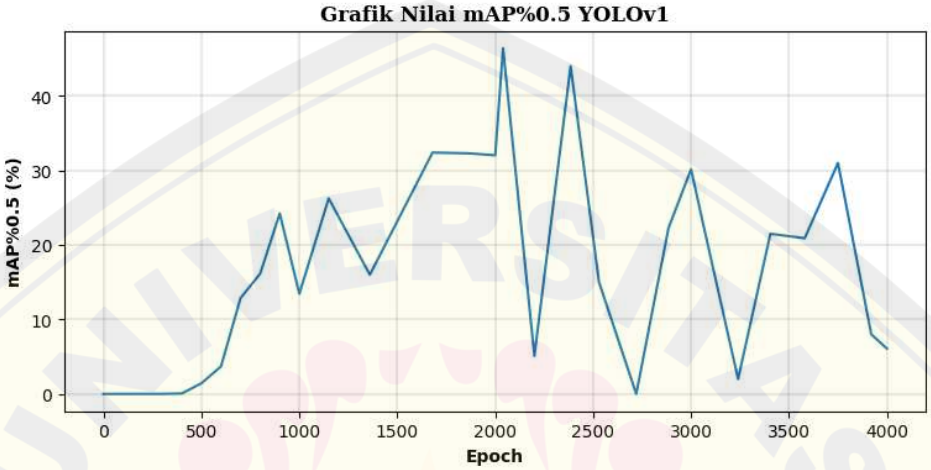
Gambar 4.7 Early stopping point

DIGITAL REPOSITORY UNIVERSITAS JEMBER

DIGITAL REPOSITORY UNIVERSITAS JEMBER

1. YOLOv1

Hasil uji coba proses training dengan menggunakan data *validation* pada penelitian ini menggunakan iterasi sebanyak 4000 dan confidence threshold 0.25. Dari banyaknya iterasi tersebut akan dicari *mean average precision* (mAP%0.50) terbaik. Hasil uji coba tersebut ditunjukkan grafik pada Gambar di bawah ini.



Gambar 4.8 Grafik nilai mAP YOLOv1

Berdasarkan grafik pada Gambar di atas diperoleh nilai mAP%0,5 tertinggi adalah 46,41% pada iterasi ke-2050. Pada grafik tersebut mulai iterasi 2050 sampai 4000 diperoleh rata-rata mAP%0,5 di bawah 46,41%. Hal ini dapat diartikan bahwa pada iterasi 2050 merupakan *Early Stopping Point* pada model yang dilatih. Jika diteruskan iterasinya model yang dibuat akan mengalami *overfitting*. Sehingga model *trained weights* yang dibuat sampai pada iterasi 2050 sudah cukup baik dalam melakukan pendeteksian penyakit VSD. Setelah dilakukan proses uji coba *training* terhadap data *validation*, kemudian dilakukan evaluasi dengan menghitung presisi, recall, dan IoU akurasi. Berikut ini merupakan ilustrasi perhitungan tiap iterasi yang disediakan pada Tabel .

Tabel 4.3 Hasil uji coba YOLOv1

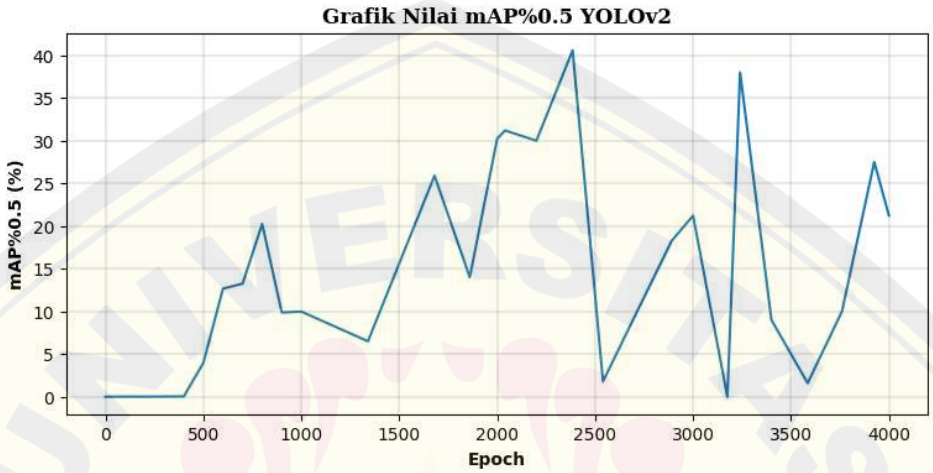
Iterasi	Sehat		VSD		mAP (%)	Precision	Recall	F1 Score	IoU (%)
	TP	FP	TP	FP					
1000	0	0	0	0	13.45	-	0	-	0
2000	0	0	0	0	32.03	-	0	-	0
2050	**0**	**0**	**0**	**0**	**46.41**	**-**	**0**	**-**	**0**
3000	174	472	129	475	30.16	0.24	0.71	0.36	15.21
4000	0	1	0	1	6. 08	0.50	0	0	32.71

DIGITAL REPOSITORY UNIVERSITAS JEMBER

DIGITAL REPOSITORY UNIVERSITAS JEMBER

2. YOLOv2

Hasil uji coba proses training dengan menggunakan data *validation* pada penelitian ini menggunakan iterasi sebanyak 4000 dan confidence threshold 0.25. Dari banyaknya iterasi tersebut akan dicari *mean average precision* (mAP%0.50) terbaik. Hasil uji coba tersebut ditunjukkan grafik pada Gambar di bawah ini.



Gambar 4.9 Grafik nilai mAP YOLOv2

Berdasarkan grafik pada Gambar di atas diperoleh nilai mAP%0,5 tertinggi adalah 40,59% pada iterasi ke-2390. Pada grafik tersebut mulai iterasi 2390 sampai 4000 diperoleh rata-rata mAP%0,5 di bawah 40,59%, Hal ini dapat diartikan bahwa pada iterasi 2390 merupakan *Early Stopping Point* pada model yang dilatih. Jika diteruskan iterasinya model yang dibuat akan mengalami *overfitting*. Sehingga model *trained weights* yang dibuat sampai pada iterasi 2390 sudah cukup baik dalam melakukan pendeteksian penyakit VSD. Setelah dilakukan proses uji coba *training* terhadap data *validation*, kemudian dilakukan evaluasi dengan menghitung presisi, recall, dan IoU akurasi. Berikut ini merupakan ilustrasi perhitungan tiap iterasi yang disediakan pada Tabel .

Tabel 4.4 Hasil uji coba YOLOv2

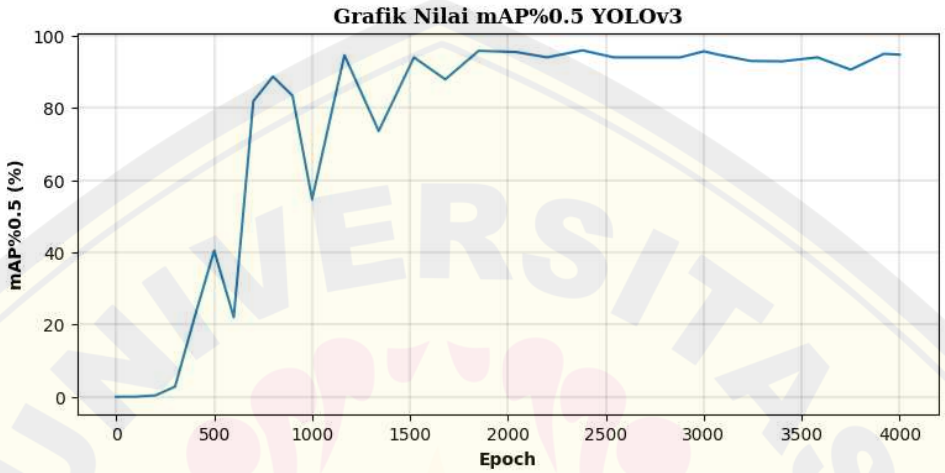
Iterasi	Sehat		VSD		mAP (%)	Precision	Recall	F1 Score	IoU (%)
	TP	FP	TP	FP					
1000	0	0	0	0	9.98	-	0	-	0
2000	0	0	0	0	30.25	-	0	-	0
2390	**0**	**0**	**0**	**0**	**40.59**	**-**	**0**	**-**	**0**
3000	38	103	65	69	21.22	0.37	0.24	0.29	23.32
4000	2	14	12	2	21.23	0.47	0. 03	0. 06	32.72

DIGITAL REPOSITORY UNIVERSITAS JEMBER

DIGITAL REPOSITORY UNIVERSITAS JEMBER

3. YOLOv3

Hasil uji coba proses training dengan menggunakan data *validation* pada penelitian ini menggunakan iterasi sebanyak 4000 dan confidence threshold 0.25. Dari banyaknya iterasi tersebut akan dicari *mean average precision* (mAP%0.50) terbaik. Hasil uji coba tersebut ditunjukkan grafik pada Gambar di bawah ini.



Gambar 4.10 Grafik nilai mAP YOLOv3

Berdasarkan grafik pada Gambar di atas diperoleh nilai mAP%0,5 tertinggi adalah 95,95% pada iterasi ke-2350. Pada grafik tersebut mulai iterasi 2350 sampai 4000 diperoleh rata-rata mAP%0,5 di bawah 95,95%, Hal ini dapat diartikan bahwa pada iterasi 2350 merupakan *Early Stopping Point* pada model yang dilatih. Jika diteruskan iterasinya model yang dibuat akan mengalami *overfitting*. Sehingga model *trained weights* yang dibuat sampai pada iterasi 2350 sudah cukup baik dalam melakukan pendeteksian penyakit VSD.

Setelah dilakukan proses uji coba *training* terhadap data *validation*, kemudian dilakukan evaluasi dengan menghitung presisi, recall, dan IoU akurasi. Berikut perhitungan tiap iterasi disajikan pada Tabel 4.5.

Tabel 4.5 Hasil uji coba YOLOv3

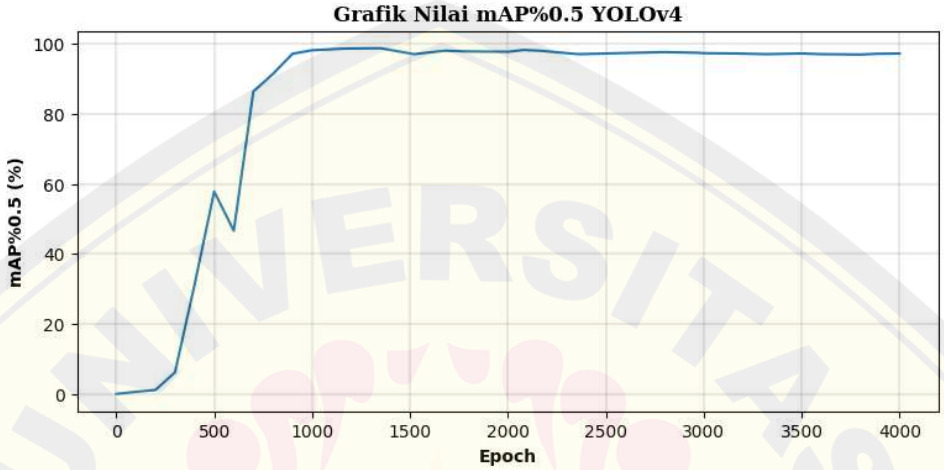
Iterasi	Sehat		VSD		mAP (%)	Precision	Recall	F1 Score	IoU (%)
	TP	FP	TP	FP					
1000	199	65	94	77	54.73	0.67	0.69	0.68	44.02
2000	153	61	120	144	55.93	0.57	0.64	0.61	34.83
2350	**235**	**18**	**164**	**13**	**95.95**	**0.93**	**0.94**	**0.93**	**76.37**
3000	235	15	167	25	95.68	0.91	0.95	0.93	68.58
4000	234	33	160	10	94.75	0.90	0.93	0.92	70.33

DIGITAL REPOSITORY UNIVERSITAS JEMBER

DIGITAL REPOSITORY UNIVERSITAS JEMBER

4. YOLOv4

Hasil uji coba proses training dengan menggunakan data *validation* pada penelitian ini menggunakan iterasi sebanyak 4000 dan confidence threshold 0.25. Dari banyaknya iterasi tersebut akan dicari *mean average precision* (mAP%0.50) terbaik. Hasil uji coba tersebut ditunjukkan grafik pada Gambar di bawah ini.



Gambar 4.11 Grafik nilai mAP YOLOv4

Berdasarkan grafik pada Gambar di atas diperoleh nilai mAP%0,5 tertinggi adalah 98,61% pada iterasi ke-1200. Pada grafik tersebut mulai iterasi 1200 sampai 4000 diperoleh rata-rata mAP%0,5 di bawah 98,61%, Hal ini dapat diartikan bahwa pada iterasi 1200 merupakan *Early Stopping Point* pada model yang dilatih. Jika diteruskan iterasinya model yang dibuat akan mengalami *overfitting*. Sehingga model *trained weights* yang dibuat sampai pada iterasi 1200 sudah cukup baik dalam melakukan pendeteksian penyakit VSD. Setelah dilakukan proses uji coba *training* terhadap data *validation*, kemudian dilakukan evaluasi dengan menghitung presisi, recall, dan IoU akurasi. Berikut ini merupakan ilustrasi perhitungan tiap iterasi yang disediakan pada Tabel .

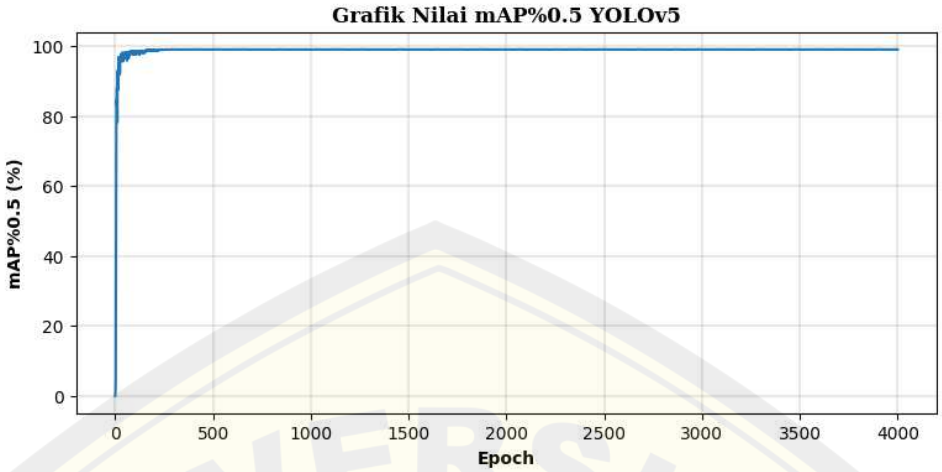
Tabel 4.6 Hasil uji coba YOLOv4

Iterasi	Sehat		VSD		mAP (%)	Precision	Recall	F1 Score	IoU (%)
	TP	FP	TP	FP					
1000	232	29	152	44	90.12	0.84	0.91	0.87	55.38
1200	**240**	**12**	**173**	**15**	**98.61**	**0.94**	**0.97**	**0.96**	**79.97**
2000	238	5	173	9	97.69	0.97	0.97	0.97	84.48
3000	237	14	172	9	97.28	0.95	0.96	0.96	79.91
4000	236	10	173	9	97.19	0.96	0.96	0.96	86.01

DIGITAL REPOSITORY UNIVERSITAS JEMBER

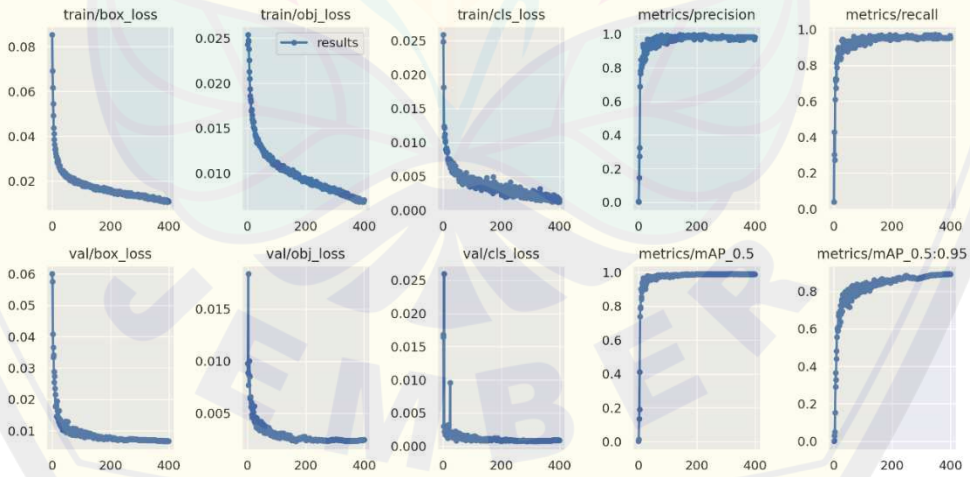
DIGITAL REPOSITORY UNIVERSITAS JEMBER

5. YOLOv5



Gambar 4.12 Grafik nilai mAP YOLOv5

Berdasarkan grafik pada Gambar di atas diperoleh nilai mAP%0,5 tertinggi adalah 99.006% pada iterasi ke-117. Pada grafik tersebut, tidak diperoleh hasil yang berubah signifikan dalam 50 iterasi terakhir, mulai iterasi 75. Oleh karena itu, proses pelatihan model berhenti secara otomatis karena terjadi *Early Stopping Point* pada iterasi ke-75. Jika diteruskan iterasinya model yang dibuat akan mengalami *overfitting*. Sehingga model *trained weights* yang dibuat sampai pada iterasi 100 sudah sangat baik dalam melakukan pendeteksian penyakit VSD.

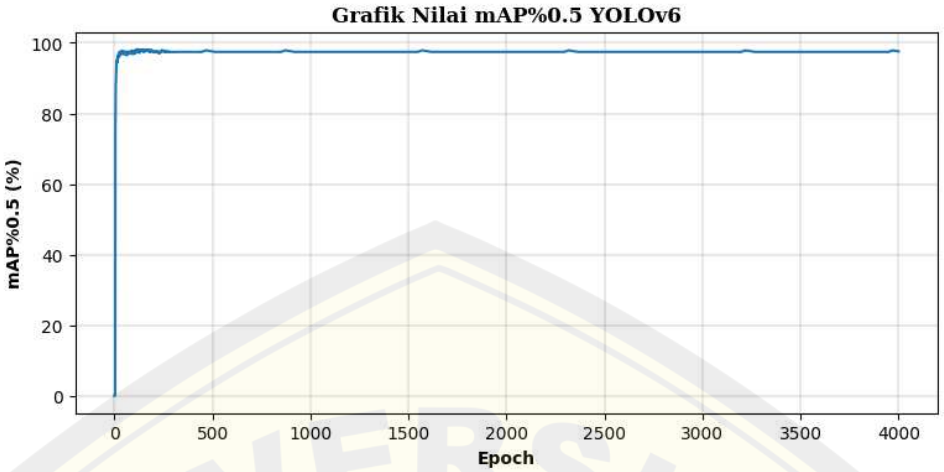


Gambar 4.13 Grafik hasil analisis YOLOv5

DIGITAL REPOSITORY UNIVERSITAS JEMBER

DIGITAL REPOSITORY UNIVERSITAS JEMBER

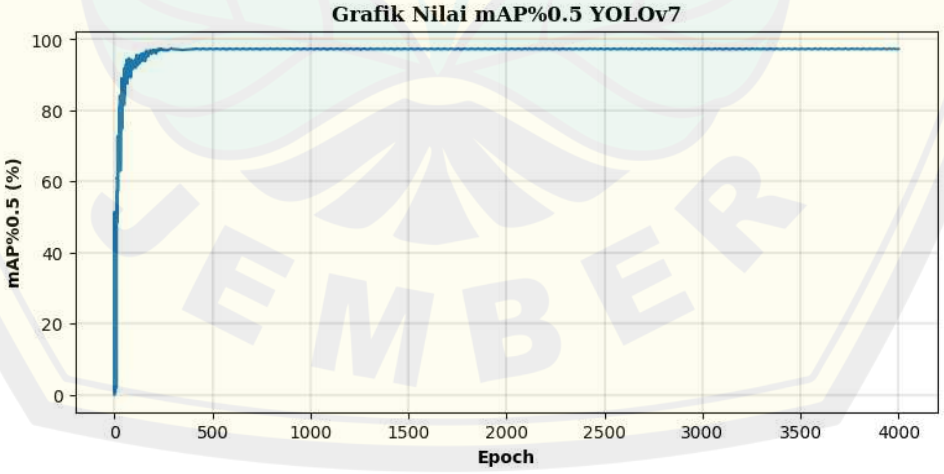
6. YOLOv6



Gambar 4.14 Grafik nilai mAP YOLOv6

Berdasarkan grafik pada Gambar di atas diperoleh nilai mAP%0,5 tertinggi adalah 98.16% pada iterasi ke-200. Pada grafik tersebut, tidak diperoleh hasil yang berubah signifikan dalam 50 iterasi terakhir, mulai iterasi 200 sampai 250. Oleh karena itu, proses pelatihan model berhenti secara otomatis karena terjadi *Early Stopping Point* pada iterasi ke-200. Jika diteruskan iterasinya model yang dibuat akan mengalami *overfitting*. Sehingga model *trained weights* yang dibuat sampai pada iterasi 200 sudah sangat baik dalam melakukan pendeteksian penyakit VSD.

7. YOLOv7



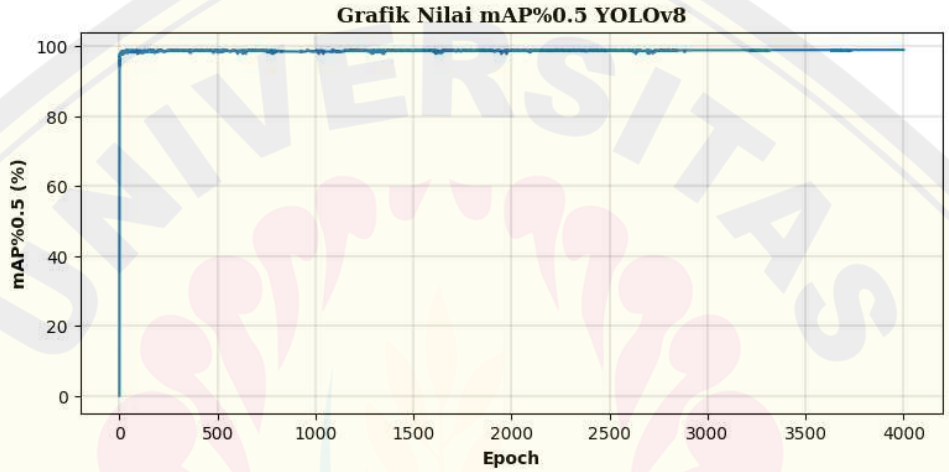
Gambar 4.15 Grafik nilai mAP YOLOv7

DIGITAL REPOSITORY UNIVERSITAS JEMBER

DIGITAL REPOSITORY UNIVERSITAS JEMBER

Berdasarkan grafik pada Gambar 4.15 diperoleh nilai mAP%0,5 tertinggi adalah 98,4% pada iterasi ke-225. Pada grafik tersebut, tidak diperoleh hasil yang berubah signifikan dalam 25 iterasi terakhir, mulai iterasi 225 sampai 250. Oleh karena itu, proses pelatihan model berhenti secara otomatis karena terjadi *Early Stopping Point* pada iterasi ke-250. Jika diteruskan iterasinya model yang dibuat akan mengalami *overfitting*. Sehingga model *trained weights* yang dibuat sampai pada iterasi 250 sudah sangat baik dalam melakukan pendeteksian penyakit VSD.

8. YOLOv8

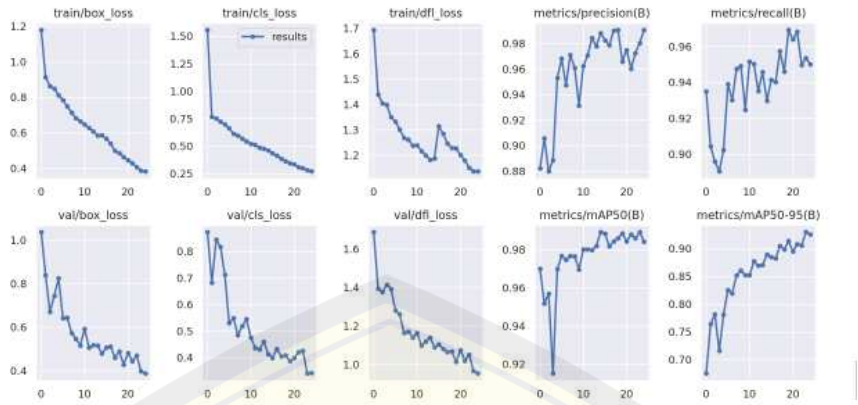


Gambar 4.16 Grafik nilai mAP YOLOv8

Berdasarkan grafik pada Gambar 4.16 diperoleh nilai mAP%0,5 tertinggi adalah 98.929% pada iterasi ke-172. Pada grafik tersebut, Early Stopping Point terjadi saat iterasi ke-20 dan tidak diperoleh hasil yang berubah signifikan dalam iterasi selanjutnya. Oleh karena itu, proses pelatihan model berhenti secara otomatis karena terjadi *Early Stopping Point* pada iterasi ke-25. Jika diteruskan iterasinya model yang dibuat akan mengalami *overfitting*. Sehingga model *trained weights* yang dibuat sampai pada iterasi 172 sudah sangat baik dalam melakukan pendeteksian penyakit VSD.

DIGITAL REPOSITORY UNIVERSITAS JEMBER

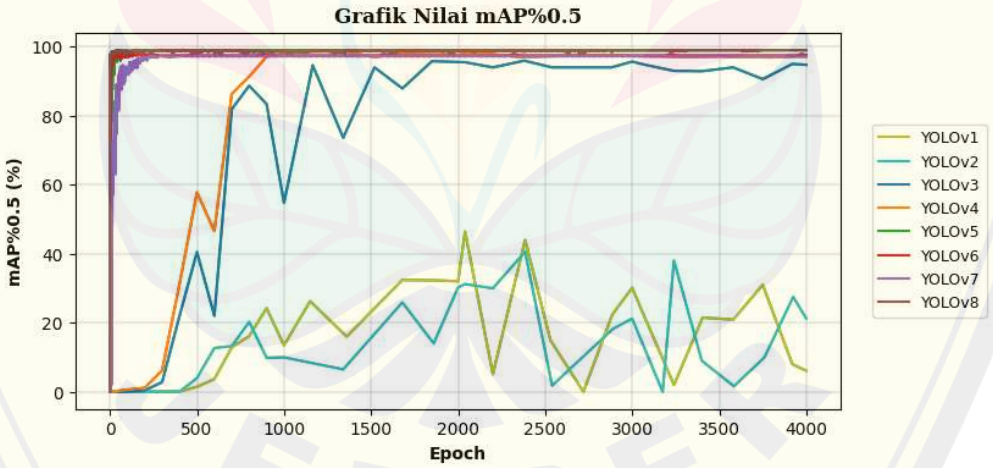
DIGITAL REPOSITORY UNIVERSITAS JEMBER



Gambar 4.17 Grafik hasil analisis YOLOv8

9. Analisa Perbandingan Hasil YOLOv1 sampai YOLOv8

Berdasarkan analisis hasil dapat disimpulkan bahwa model jaringan YOLOv5 dan YOLOv8 merupakan metode yang paling baik. YOLOv5 menghasilkan nilai mAP tertinggi sebesar 99.006% dengan early stopping point (ESP) pada iterasi ke-75, sedangkan pada YOLOv8 menghasilkan nilai mAP tertinggi sebesar 98.929% dengan ESP pada iterasi ke-20. Perbandingan nilai MAP tiap YOLO dapat dilihat pada Gambar 4.18.



Gambar 4.18 Perbandingan nilai mAP tiap YOLO

Diperoleh 2 model tebaik, yaitu YOLOv5 dan YOLOv8. Dalam menentukan model yang akan digunakan dalam proses pengujian, diperhatikan beberapa kriteria lain yag dijabarkan dalam Lampiran 3. Berdasarkan hal tersebut, YOLOv8 ditetapkan menjadi model terbaik dan digunakan untuk proses pengujian.

DIGITAL REPOSITORY UNIVERSITAS JEMBER

DIGITAL REPOSITORY UNIVERSITAS JEMBER

4.3 Identifikasi Tingkat Keparahan Lahan Menggunakan Dominating Set

Langkah selanjutnya adalah identifikasi tingkat keparahan penyakit VSD pada lahan tanaman kakao. Penyakit VSD merupakan penyakit pada tanaman kakao yang penyebarannya sangat cepat. Penyakit ini menyebar ke tanaman lain melalui angin. Pada penelitian ini diasumsikan bahwa penyakit VSD hanya menyebar melalui angin yang direpresentasikan dengan 8 arah mata angin. Dalam mengkonstruksi graf, dilakukan pengamatan pada lahan dimana pola tanam tanaman kakao berbentuk grid dengan jarak yang sama. Setiap tanaman kakao dalam sebuah lahan direpresentasikan sebagai sebuah titik dalam graf dan 8 arah angin sebagai simpul yang menghubungkan setiap titiknya. Diperoleh representasi graf dari lahan berupa graf king yang diilustrasikan pada Teorema 2.1. Kemudian dicari himpunan dominating pada graf tersebut. Selanjutnya proses identifikasi penyakit VSD hanya diperhatikan pada tanaman kakao yang merupakan anggota himpunan dominating. Dari analisis tersebut dapat diidentifikasi tingkat sebaran penyakit VSD pada lahan tersebut berdasarkan presentase tanaman yang terinfeksi VSD yang merupakan anggota himpunan dominating. Pada penelitian ini dilakukan pengamatan pada 2 lahan dan diterapkan teori dominating set sebagai berikut.

4.3.1 Hasil Pengamatan Lahan 1

Lahan 1 memiliki 81 pohon yang direpresentasikan sebagai titik dan dihubungkan oleh simpul sedemikian rupa sehingga membentuk graf $P_9 \boxtimes P_9$. Bentuk dari graf $P_9 \boxtimes P_9$ dan *dominating set* S diilustrasikan pada Gambar 4.19

Himpunan titik pada graf $P_9 \boxtimes P_9$ adalah sebagi berikut

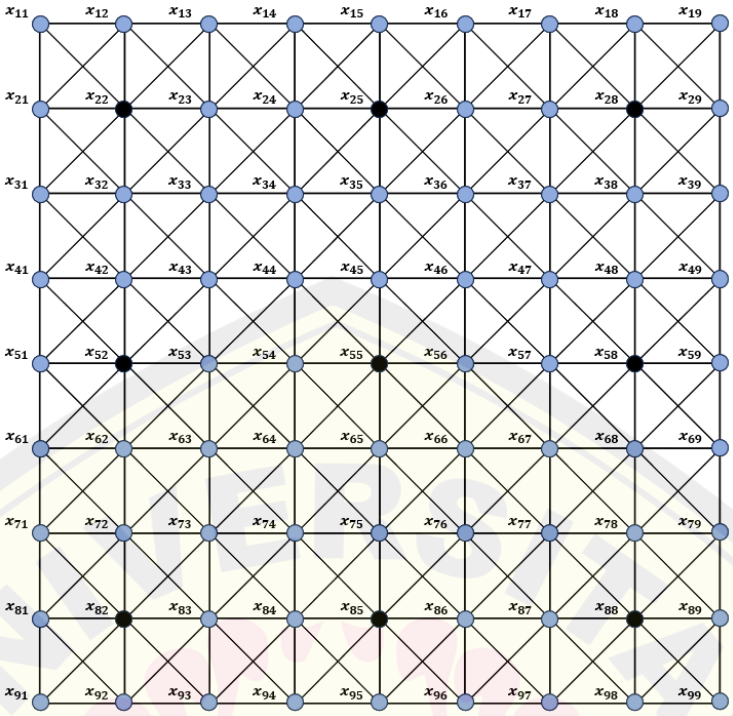
$V(P_9 \boxtimes P_9) = \{x_{ij} | 1 \leq i \leq 9, 1 \leq j \leq 9\}$.

Berdasarkan Teorema 2.1, diperoleh himpunan dominasi S sebagai berikut

$S = \{x_{22}, x_{25}, x_{28}, x_{52}, x_{55}, x_{58}, x_{82}, x_{85}, x_{88}\}$ sehingga didapat $|S| = 9$.

DIGITAL REPOSITORY UNIVERSITAS JEMBER

DIGITAL REPOSITORY UNIVERSITAS JEMBER



Gambar 4.19 Ilustrasi graf $P_9 \boxtimes P_9$ dengan $|S| = 9$.

Selanjutnya proses identifikasi penyakit VSD hanya diperhatikan pada tanaman kakao yang merupakan anggota himpunan dominating. Dengan ini proses identifikasi tidak perlu dilakukan pada semua tanaman kakao yang ada di sebuah lahan. Proses identifikasi dilakukan menggunakan model objek detektor YOLOv8 yang diperoleh dari proses training sebelumnya. Hasil pengamatan pada lahan 1 disajikan pada Tabel 4.7 sebagai berikut.

Tabel 4.7 Hasil pengamatan pada lahan 1

No.	Pohon	Persentase VSD	VSD	Sehat
1.	x_{22}	2%		✓
2.	x_{25}	2%		✓
3.	x_{28}	2%		✓
4.	x_{52}	10%		✓
5.	x_{55}	10%		✓
6.	x_{58}	5%		✓
7.	x_{82}	35%	✓	
8.	x_{85}	20%		✓
9.	x_{88}	10%		✓

DIGITAL REPOSITORY UNIVERSITAS JEMBER

DIGITAL REPOSITORY UNIVERSITAS JEMBER

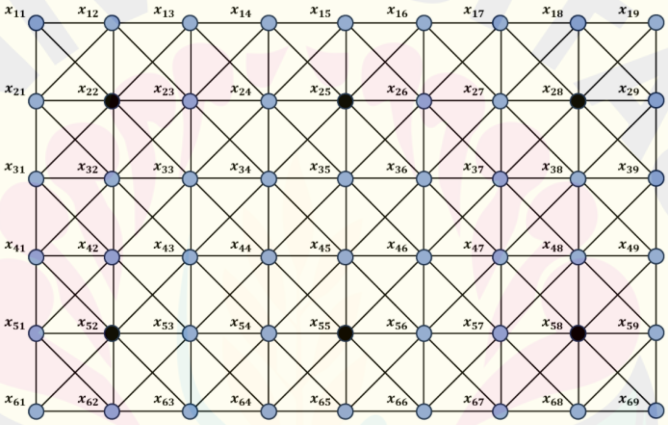
Berdasarkan ilustrasi lahan 1 di atas, dapat dihitung tingkat keparahan penyakit pada lahan 1 dengan membandingkan jumlah pohon yang terjangkit VSD dengan jumlah seluruh pohon yang merupakan anggota himpunan dominasi.

$$Tingkat\ keparahan = \frac{1}{9} \times 100\% = 11.11\%$$

Diperoleh nilai rata-rata sebesar 11.11% menunjukkan bahwa tingkat keparahan penyakit pada lahan tersebut cukup rendah.

4.3.2 Hasil Pengamatan Lahan 2

Lahan 2 memiliki 54 pohon yang direpresentasikan sebagai graf $P_6 \boxtimes P_9$. Bentuk dari graf $P_6 \boxtimes P_9$ dan *dominating set* S diilustrasikan pada Gambar 4.20.



Gambar 4.20 Ilustrasi graf $P_6 \boxtimes P_9$ dengan $|S| = 6$

Himpunan dominasi $S = \{x_{22}, x_{25}, x_{28}, x_{52}, x_{55}, x_{58}\}$ sehingga didapat $|S| = 6$. Hasil pengamatan pada lahan 2 diilustrasikan pada Tabel 4.8 sebagai berikut.

Tabel 4.8 Hasil pengamatan pada lahan 2

No.	Pohon	Persentase VSD	VSD	Sehat
1.	x_{22}	40%	✓	
2.	x_{25}	30%	✓	
3.	x_{28}	15%		✓
4.	x_{52}	50%	✓	
5.	x_{55}	30%	✓	
6.	x_{58}	15%		✓

Dapat dihitung tingkat keparahan penyakit pada lahan 1 sebagai berikut.

$$Tingkat\ keparahan = \frac{4}{6} \times 100\% = 66.67\%$$

DIGITAL REPOSITORY UNIVERSITAS JEMBER

DIGITAL REPOSITORY UNIVERSITAS JEMBER

Diperoleh nilai rata-rata sebesar 66.67% menunjukkan bahwa tingkat keparahan penyakit pada lahan tersebut tinggi. Penyebaran penyakit VSD harus segera ditangani agar penyakit tidak menyerang seluruh pohon yang ada.

4.3.3 Hasil Pengamatan Lahan 3

Lahan memiliki 81 pohon yang direpresentasikan sebagai graf $P_9 \boxtimes P_9$. Bentuk dari graf $P_9 \boxtimes P_9$ dan *dominating set* S diilustrasikan pada Gambar 4.19. Hasil pengamatan pada lahan 3 diilustrasikan pada Tabel 4.8 sebagai berikut.

Tabel 4.9 Hasil pengamatan pada lahan 3

No.	Pohon	Persentase VSD	VSD	Sehat
1.	x_{22}	20%		✓
2.	x_{25}	30%	✓	
3.	x_{28}	50%	✓	
4.	x_{52}	10%		✓
5.	x_{55}	30%	✓	
6.	x_{58}	20%		✓
7.	x_{82}	10%		✓
8.	x_{85}	10%		✓
9.	x_{88}	30%	✓	

Berdasarkan ilustrasi lahan 3 di atas, dapat dihitung tingkat keparahan penyakit pada lahan 1 dengan membandingkan jumlah pohon yang terjangkit VSD dengan jumlah seluruh pohon yang merupakan anggota himpunan dominasi.

$$Tingkat\ keparahan = \frac{4}{9} \times 100\% = 44.44\%$$

Diperoleh nilai rata-rata sebesar 44.44% menunjukkan bahwa tingkat keparahan penyakit pada lahan tersebut parah sehingga perlu segera ditangani secara menyeluruh pada satu lahan tersebut.

Pengamatan pada lahan 1, 2, dan 3 di perkebunan yang sama menghasilkan nilai tingkat sebaran penyakit VSD yang berbeda-beda. Hal ini dikarenakan penyakit VSD menyebar pada pohon yang terletak di sekitarnya saja. Oleh karena itu, untuk mengidentifikasi tingkat sebarannya, perlu dilakukan pengamatan untuk setiap lahan. Sedangkan untuk lahan yang sangat luas dapat dilakukan pengampilan sampel lahan berukuran sama yang terletak di tengah setiap petak lahan dengan ukuran tertentu.

DIGITAL REPOSITORY UNIVERSITAS JEMBER

DIGITAL REPOSITORY UNIVERSITAS JEMBER

BAB 5. KESIMPULAN DAN SARAN

Bab ini berisi kesimpulan yang menjelaskan hasil dari penelitian terkait tujuan penelitian yang telah ditentukan di awal serta saran untuk pengembangan penelitian selanjutnya.

5.1 Kesimpulan

Berikut merupakan kesimpulan yang diperoleh berdasarkan penelitian yang telah dilakukan dan analisis hasil penelitian.

1. Hasil dari proses training data diperoleh nilai Mean Average Precision (mAP) terbaik pada masing-masing dari YOLOv1 sampai dengan YOLOv8 sebesar 46.41% pada iterasi ke-2050, 40.59% pada iterasi ke-2390, 95.95% pada iterasi ke-2350, 98.61% pada iterasi ke-1200, 99.006% pada iterasi ke-200, 98.16% pada iterasi ke-117, 98.4% pada iterasi ke-225, dan 98.929% pada iterasi ke-172.
2. Berdasarkan analisis hasil dapat disimpulkan bahwa model jaringan YOLOv5 dan YOLOv8 merupakan metode yang paling baik. YOLOv5 menghasilkan nilai mAP tertinggi sebesar 99.006% dengan early stopping point (ESP) pada iterasi ke-75, sedangkan pada YOLOv8 menghasilkan nilai mAP tertinggi sebesar 98.929% dengan ESP pada iterasi ke-20.
3. Tingkat sebaran penyakit VSD dapat dianalisis dengan menerapkan teori graf dominating set. Penyebaran VSD pada lahan 1 cukup rendah, tetapi tindakan pencegahan penyebaran tetap perlu dilakukan. Pada lahan 2 penyebaran VSD cukup parah karena tingkat penyebarannya melebihi setengah lahan sehingga langkah penanganan harus segera dilakukan. Pada lahan 3 tingkat penyakit VSD menyebar hampir setengah lahan sehingga perlu dilakukan tindakan penanganan.

5.2 Saran

Beberapa saran yang perlu diperhatikan dalam pengembangan penelitian selanjutnya diantaranya sebagai berikut.

DIGITAL REPOSITORY UNIVERSITAS JEMBER

DIGITAL REPOSITORY UNIVERSITAS JEMBER

1. Dalam menerapkan metode machine learning khususnya deep learning, semakin banyak data yang digunakan dalam proses training, maka hasil deteksi objek yang diperoleh akan semakin baik dapat diterapkan dalam berbagai kondisi.
2. Proses training data dalam deep learning merupakan proses yang membutuhkan mesin berkapasitas besar. Penggunaan GPU sangat disarankan agar dapat menghasilkan model dengan tingkat performa yang tinggi dan mempercepat proses training data.
3. Untuk penelitian selanjutnya dapat dikembangkan untuk mendeteksi beberapa penyakit pada kakao atau dilakukan pada tanaman lain sehingga semakin banyak objek yang dapat terdeteksi.

DIGITAL REPOSITORY UNIVERSITAS JEMBER

DIGITAL REPOSITORY UNIVERSITAS JEMBER

DAFTAR PUSTAKA

Awad, F. H., Hamad, M. M., & Alzubaidi, L. (2023). Robust Classification and Detection of Big Medical Data Using Advanced Parallel K-Means Clustering, YOLOv4, and Logistic Regression. *Life*, *13*(3), 691. https://doi.org/10.3390/life13030691

Arshad, M., Hayat, S., & Jamil, H. (2023). The domination number of the king's graph. Computational and Applied Mathematics, 42(6), 251. https://doi.org/10.1007/s40314-023-02386-8

Bingham, G., & Miikkulainen, R. (2022). Discovering Parametric Activation Functions. *Neural Networks*, *148*, 48–65. https://doi.org/10.1016/j.neunet.2022.01.001

Bochkovskiy, A., Wang, C.-Y., & Liao, H.-Y. M. (2020). *YOLOv4: Optimal Speed and Accuracy of Object Detection* (arXiv:2004.10934). arXiv. http://arxiv.org/abs/2004.10934

Brahimi, M., Boukhalfa, K., & Moussaoui, A. (2017). Deep Learning for Tomato Diseases: Classification and Symptoms Visualization. *Applied Artificial Intelligence*, *31*(4), 299–315. https://doi.org/10.1080/08839514.2017.1315516

Darwish, A.A., Ezzat, D., & Hassanien, A.E. (2020). An Optimized Model Based on Convolutional Neural Networks and Orthogonal Learning Particle Swarm Optimization Algorithm for Plant Diseases Diagnosis. Swarm Evol. Comput., 52.

DeChant, C., Wiesner-Hanks, T., Chen, S., Stewart, E.L., Yosinski, J., Gore, M.A., Nelson, R.J. Lipson, H. (2017) Automated Identification of Northern Leaf 57 Blight-Infected Maize Plants from Field Imagery Using Deep Learning. *Phytopathology,* 107, 1426–1432.

Ghosh, D. (2022). The Top Cocoa Producing Countries in The World. https://www.worldatlas.com/industries/the-top-cocoa-producing-countries-in-the-world.html.

Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2014). *Rich feature hierarchies for accurate object detection and semantic segmentation* (arXiv:1311.2524). arXiv. http://arxiv.org/abs/1311.2524

Goddard, W., & Henning, M.A. (2013). Independent domination in graphs: A survey and recent results. *Discrete Mathematics*, 313: 839-854.

Gonzalez, R.C. and Woods, R.E. (2018) Digital Image Processing. 4th Edition, Pearson Education, New York, 1022.

Jia, D., He, Z., Zhang, C. et al. (2022). Detection of cervical cancer cells in complex situation based on improved YOLOv3 network. *Multimed Tools Appl*, 81 : 8939–8961. https://doi.org/10.1007/s11042-022-11954-9

DIGITAL REPOSITORY UNIVERSITAS JEMBER

DIGITAL REPOSITORY UNIVERSITAS JEMBER

Jiang, J., Fu X, Qin R, Wang X, Ma Z. 2021. High-speed lightweight ship detection algorithm based on YOLO-V4 for three-channels RGB SAR image. Remote Sens 13 (10):1909

Jiang, P., Ergu, D., Liu, F., Cai, Y., & Ma, B. (2022). A Review of Yolo Algorithm Developments. *Procedia Computer Science*, *199*, 1066–1073. https://doi.org/10.1016/j.procs.2022.01.135

Kannadaguli, P. 2020. YOLO v4 based human detection system using aerial thermal imaging for UAV based surveillance applications. In 2020 international conference on decision aid sciences and application (DASA). 1213-1219

Li J, Gu J, Huang Z, Wen J. 2019. Application research of improved YOLO V3 algorithm in PCB electronic component detection. Appl Sci 9(18):3750

Liu, B., Zhang, Y., He, D., & Li, Y. (2018). Identification of Apple Leaf Diseases Based on Deep Convolutional Neural Networks. *Symmetry*, 10(11). https://doi.org/10.3390/sym10010011

Loey, M., G. Manogaran, M. H. N. Taha, & N. E. M. Khalifa. 2021. Fighting against COVID-19: a novel deep learning model based on YOLO-v2 with ResNet-50 for medical face mask detection. Sustain Cities Soc 65: 102600

Ploetz, R. 2016. The impact of diseases on cacao production: A Global Overview. In: Bailey A B and Meinhardt W L, eds. Cacao Diseases: A History of Old Enemies and New Encounters (New York: Springer International Publishing). 307-35

Prechelt, L. (n.d.). *Early Stopping | but when?*

Redmon, J., & Farhadi, A. (2016). *YOLO9000: Better, Faster, Stronger* (arXiv:1612.08242). arXiv. http://arxiv.org/abs/1612.08242

Redmon, J., & Farhadi, A. (2018). *YOLOv3: An Incremental Improvement* (arXiv:1804.02767). arXiv. http://arxiv.org/abs/1804.02767

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). *You Only Look Once: Unified, Real-Time Object Detection* (arXiv:1506.02640). arXiv. http://arxiv.org/abs/1506.02640

Ren, S., He, K., Girshick, R., & Sun, J. (2016). *Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks* (arXiv:1506.01497). arXiv. http://arxiv.org/abs/1506.01497

Samuels, G. J., Ismaiel, A., Rosmana, A., Junaid, M., Guest, D., McMahon, P., Keane, P., Purwantara, A., Lambert, S., Rodriguez-Carres, M., & Cubeta, M. A. (2012). Vascular Streak Dieback of cacao in Southeast Asia and Melanesia: in planta detection of the pathogen and a new taxonomy. *Fungal biology*, 116(1), 11–23. https://doi.org/10.1016/j.funbio.2011.07.009

DIGITAL REPOSITORY UNIVERSITAS JEMBER

DIGITAL REPOSITORY UNIVERSITAS JEMBER

Shorten, C., Khoshgoftaar, T.M. A survey on Image Data Augmentation for Deep Learning. *J Big Data 6* , 60 (2019).

Susilowati, L., Sa'adah, I., Fauziyyah, R. Z., Erfanian, A., & Slamin. (2020). The dominant metric dimension of graphs. Heliyon, 6(3), e03633. https://doi.org/10.1016/j.heliyon.2020.e03633

Terven, J., & Cordova-Esparza, D. (2023). A Comprehensive Review of YOLO Architectures in Computer Vision: From YOLOv1 to YOLOv8 and YOLO-NAS. *Machine Learning and Knowledge Extraction*, *5*(4), 1680–1716. https://doi.org/10.3390/make5040083

Wu, D., Lv S, Jiang M, Song H. 2020. Using channel pruning-based YOLO v4 deep learning algorithm for the real-time and accurate detection of apple flowers in natural environments. Comput Electron Agric 178: 105742

Ye, .A, Pang, B., Jin, Y., & Cui, J. 2020. A YOLO-based neural network with VAE for intelligent garbage detection and classification. In 2020 3rd international conference on algorithms computing and artificial intelligence pp 1-7

DIGITAL REPOSITORY UNIVERSITAS JEMBER

DIGITAL REPOSITORY UNIVERSITAS JEMBER

LAMPIRAN-LAMPIRAN

2.1 Lampiran Ilustrasi *Convolutional Neural Network* (CNN)

Proses klasifikasi menggunakan *Convolutional Neural Network* yang terdiri empat layer, yaitu *convolution layer* (lapisan konvolusi), *activation layer* (lapisan aktivasi), *pooling layer* (lapisan penyatuan), dan *fully connected layer* akan diilustrasikan sebagai berikut.

1. Convolution layer

Proses konvolusi terdiri atas serangkaian filter/kernel konvolusi yang dikerjakan pada input gambar, mengaktifkan berbagai aspek atau fitur pada gambar dengan masing-masing filter. Operasi konvolusi dilakukan secara berulang dimana output dari layer sebelumnya menjadi input untuk layer setelahnya. Konvolusi yang melibatkan input 1 dimensi disebut dengan 1D-Convolution atau konvolusi 1 dimensi. Sedangkan, konvolusi dengan input berupa grid yaitu matriks 2 dimensi merupakan konvolusi 2 dimensi. Berikut akan diilustrasikan proses konvolusi 2 dimensi. Persamaan konvolusi $g(x)$ dapat ditunjukkan pada Persamaan 1.

$$g(x) = w(x).f(x) \tag{1}$$

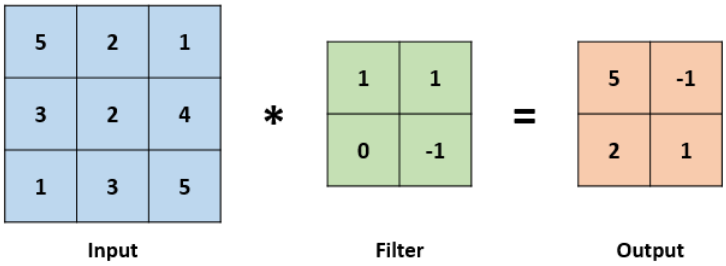
Pada Persamaan 1, $f(x)$ merupakan input citra dan $w(x)$ merupakan filter konvolusi. Operasi konvolusi menghasilkan nilai pada posisi tertentu pada feature map. Nilai tersebut merupakan hasil perkalian untuk setiap nilai pada filter dan nilai piksel gambar yang letaknya bersesuaian dengan sel kernel (Khan et al., 2018). Operasi perkalian tersebut dinyatakan dalam Persamaan 2.

$$h_{i,j} = \sum_{k=1}^{m} \sum_{l=1}^{m} W_{k,l} X_{i+k-1,j+l-1} \tag{2}$$

Pada Persamaan 2, h menyatakan output, W merupakan filter konvolusi, X sebagai input, dan m adalah ukuran filter. Proses konvolusi diilustrasikan pada Gambar 1.

DIGITAL REPOSITORY UNIVERSITAS JEMBER

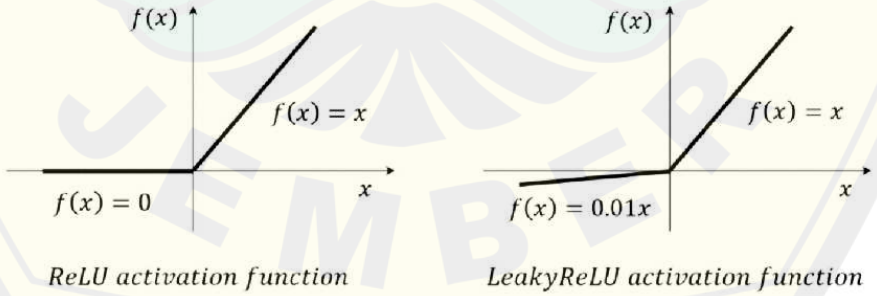
DIGITAL REPOSITORY UNIVERSITAS JEMBER



Gambar 1. Ilustrasi proses konvolusi

6. Activation Layer

Pada tahap ini hasil konvolusi dari *convolution layer* akan dikenakan fungsi aktivasi. Tujuan dari fungsi aktivasi adalah untuk menambahkan non-linearitas pada jaringan saraf. Ada beberapa fungsi aktivasi yang banyak dikenal, di antaranya yaitu fungsi *Rectified linear unit* (ReLU), Leaky ReLU, *sigmoid/logistic*, *hyperbolic tangent* (tanh), dan *Exponential Linear Unit* (ELU). ReLU adalah fungsi aktivasi yang paling sering digunakan pada jaringan saraf (Bingham, et al., 2022). ReLU mempertahankan nilai positif dan menerjemahkan nilai negatif menjadi nol. Fungsi Leaky ReLU berperilaku sama seperti fungsi ReLU saat x bernilai positif. Namun, ketika x negatif, fungsi Leaky ReLU mengembalikan nilai negatif yang proposional dengan input x. Keuntungan utama Leaky ReLU dibandingkan fungsi ReLU adalah dapat membantu meningkatkan kinerja jaringan saraf dalam dengan mengatasi masalah *dying* pada ReLU. Dengan menggunakan nilai kemiringan yang kecil untuk nilai x negatif, Leaky ReLU memastikan bahwa semua neuron di jaringan dapat berkontribusi pada output, meskipun inputnya negatif. Grafik dari fungsi aktivasi ReLU dan Leaky ReLU dapat dilihat pada Gambar 2.



Gambar 2. Fungsi aktivasi ReLU dan Leaky ReLU

DIGITAL REPOSITORY UNIVERSITAS JEMBER

DIGITAL REPOSITORY UNIVERSITAS JEMBER

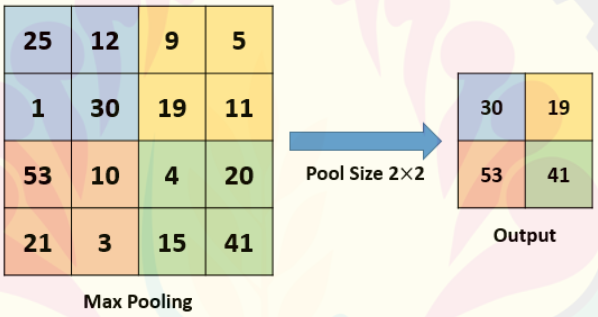
Persamaan 3 merupakan fungsi aktivasi ReLU dan Persamaan 4 merupakan fungsi aktivasi Leaky ReLU.

$$f(x) = \begin{cases} x & x \geq 0 \\ 0 & x < 0 \end{cases} \quad (3)$$

$$f(x) = \begin{cases} x & x \geq 0 \\ 0.01x & x < 0 \end{cases} \quad (4)$$

7. *Pooling Layer*

Pooling bertujuan untuk mengurangi jumlah parameter yang perlu dipelajari jaringan dengan melakukan downsampling nonlinear pada output. Terdapat dua jenis pooling yang paling umum digunakan, yaitu *max pooling* dan *average pooling*. Pada *max pooling* diambil nilai maksimal, sedangkan *average pooling* diambil nilai rata-rata. Sebagai contoh diilustrasikan proses *max pooling* menggunakan *pool size* 2×2 seperti pada Gambar 3.



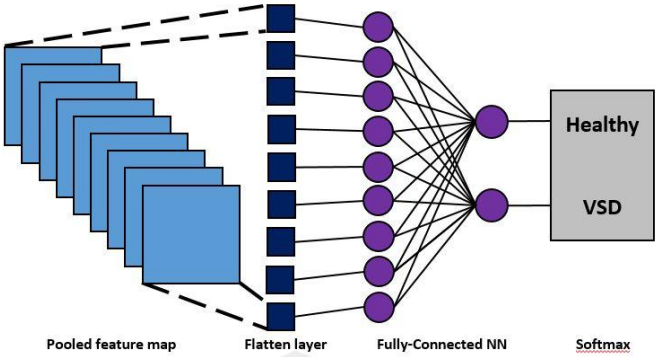
Gambar 3. Ilustrasi *max pooling*

8. *Fully Connected Layer*

Layer terakhir pada CNN adalah fully connected layer (FCNN) dengan inputnya merupakan output dari layer sebelumnya yang diubah bentuknya menjadi *array*. FCNN juga disebut sebagai *hidden layer*. Banyaknya *hidden layer* dan neuron di setiap layernya bebas dan tidak ada batasan. Algoritma FCNN diilustrasikan pada Gambar 4.

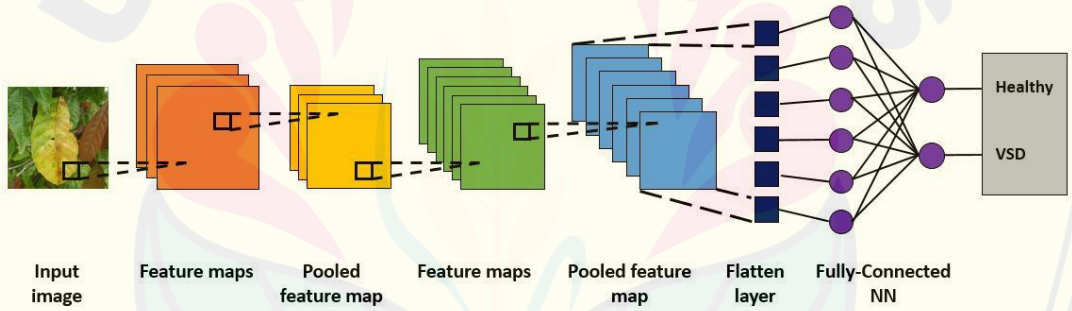
DIGITAL REPOSITORY UNIVERSITAS JEMBER

DIGITAL REPOSITORY UNIVERSITAS JEMBER



Gambar 4. Ilustrasi Fully Connected Layer

Algoritma pembelajaran fitur data pada CNN diilustrasikan pada gambar di bawah ini.



Gambar 5. Ilustrasi algoritma CNN

DIGITAL REPOSITORY UNIVERSITAS JEMBER

DIGITAL REPOSITORY UNIVERSITAS JEMBER

Lampiran 2. Perbedaan YOLOv1-YOLOv8

YOLO pertama kali dirilis pada tahun 2015 oleh Joseph Redmon. Seiring waktu YOLO terus dikembangkan dengan berbagai fitur untuk meningkatkan akurasi. Sampai pada tahun 2023, terdapat variasi model YOLOv1 sampai dengan YOLOv8. Setiap versi YOLO dibuat dengan tujuan meningkatkan kinerja dan efisiensi dari versi sebelumnya (Jiang et al., 2022). Peningkatan tersebut dapat mencakup peningkatan akurasi deteksi objek, peningkatan kecepatan pelatihan, atau peningkatan dalam hal arsitektur model. Berikut adalah perbedaan secara umum variasi model YOLO dari YOLOv1 sampai dengan YOLOv8 yang disajikan dalam Tabel 1 (Terven & Cordova-Esparza, 2023).

Tabel 1. Perbedaan YOLOv1 sampai dengan YOLOv8

Versi YOLO	Tahun Rilis	Anchor	Frame work	Backbone	Evolusi
YOLOv1	2015	Tidak	Darknet	Darknet24	Inovasi penyelesaian masalah deteksi objek dengan mengubahnya menjadi masalah regresi.
YOLOv2	2016	Ya	Darknet	Darknet24	Menggunakan *batch normalization*, *anchor box*, dan dapat mendeteksi objek kecil.
YOLOv3	2018	Ya	Darknet	Darknet53	Memprediksi objek pada 3 skala yang berbeda dengan backbone yang lebih efisien, namun memerlukan komputasi yang lebih berat.
YOLOv4	2020	Ya	Darknet	CSPDarknet53	Backbone yang lebih canggih dan beban komputasi yang lebih rendah dan efisien.
YOLOv5	2020	Ya	Pytorch	YOLOv5-CSPDarknet	Lebih ramah pengguna dengan instalasi yang lebih

DIGITAL REPOSITORY UNIVERSITAS JEMBER

DIGITAL REPOSITORY UNIVERSITAS JEMBER

					mudah dengan Pytorch dan waktu pelatihannya yang cepat.
YOLOv6	2022	Tidak	Pytorch	EfficientRep	Model yang cocok digunakan untuk aplikasi industri *real-time* dengan adanya kuantisasi dan *knowledge distillation*.
YOLOv7	2022	Tidak	Pytorch	YOLOv7-Backbone	Model mudah beradaptasi dengan dataset berjumlah sedikit karena model mempelajari fitur yang secara khusus disesuaikan dengan data yang ada.
YOLOv8	2023	Tidak	Pytorch	YOLOv8-CSPDarknet	Menerapkan berbagai inovasi, seperti backbone YOLOv8CSPDarknet, anchor-free split head, dan fungsi loss baru.

DIGITAL REPOSITORY UNIVERSITAS JEMBER

DIGITAL REPOSITORY UNIVERSITAS JEMBER

Lampiran 3. Perbandingan Performa YOLOv5 dan YOLOv8

YOLOv5 diperkenalkan pada tahun 2020 oleh Ultralytics, pengembang YOLOv3, dan dibangun dalam kerangka PyTorch. Performa model YOLOv5 cepat, mudah digunakan, dan mampu mencapai hasil yang optimal dalam menyelesaikan masalah deteksi objek. Hasilnya juga lebih akurat dan lebih mudah untuk dilatih dibandingkan pendahulunya, menjadikannya model yang populer dan banyak digunakan.

YOLOv8 adalah model terbaru dalam variasi model YOLO dan diperkenalkan pada tahun 2022 oleh Ultralytics. YOLOv8 dibangun pada kerangka YOLOv5 dan mempunyai beberapa peningkatan arsitektur dan pengembangan fitur. Model ini lebih cepat dan lebih akurat dibanding YOLOv5. YOLOv8 telah dilengkapi dengan berbagai kerangka kerja terpadu untuk melakukan deteksi objek, segmentasi instan, dan klasifikasi gambar. Perbandingan performa nilai mAP YOLOv5 dan YOLOv8 dalam mendeteksi objek menggunakan dataset COCO yang berukuran 640 piksel disajikan pada Tabel 2.

Tabel 2. Tabel Performa mAP YOLOv5 dan YOLOv8

Ukuran Model	YOLOv5	YOLOv8	Perbedaan
Nano	28	37.3	+33.21%
Small	37.4	44.9	+20.05%
Medium	45.4	50.2	+10.57%
Large	49	52.9	+7.96%
Xtra Large	50.7	53.9	+6.31%

Dalam memilih model deteksi objek terbaik, ada beberapa faktor yang perlu dipertimbangkan. Beberapa faktor tersebut antara lain kecepatan, keakuratan, kemudahan penggunaan, dan pengalaman pengguna.

Kecepatan

YOLOv8 dan YOLOv5 adalah model pendeteksi objek yang cepat dan mampu memproses gambar secara real-time. Namun, YOLOv8 lebih cepat dibandingkan YOLOv5 sehingga menjadikannya pilihan yang lebih baik untuk aplikasi yang memerlukan deteksi objek secara *real-time*.

Akurasi

DIGITAL REPOSITORY UNIVERSITAS JEMBER

DIGITAL REPOSITORY UNIVERSITAS JEMBER

Akurasi merupakan faktor penting untuk dipertimbangkan ketika memilih model deteksi objek. Dalam hal ini, YOLOv8 lebih akurat daripada YOLOv5 dikarenakan beberapa peningkatan yang dilakukan pada arsitekturnya.

Pengalaman pengembang

YOLOv8 menyediakan berbagai kerangka kerja terpadu untuk training model dalam melakukan tugas deteksi objek, segmentasi instan, dan klasifikasi gambar. Hal ini menjadi poin penting bagi pengguna yang menginginkan perangkat yang lebih komprehensif.

DIGITAL REPOSITORY UNIVERSITAS JEMBER