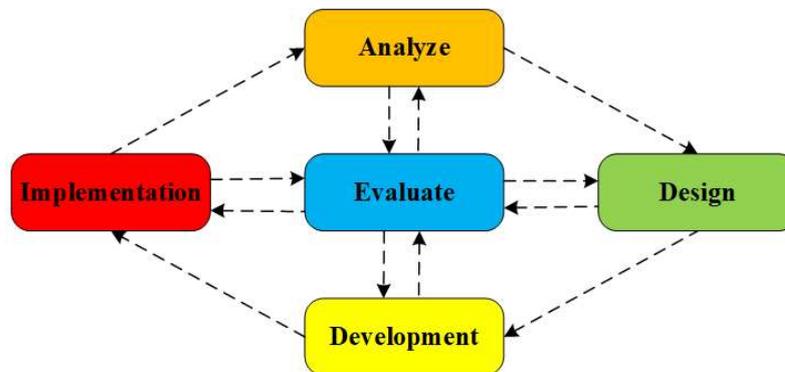


BAB III METODE PENELITIAN

3.1. Desain Penelitian

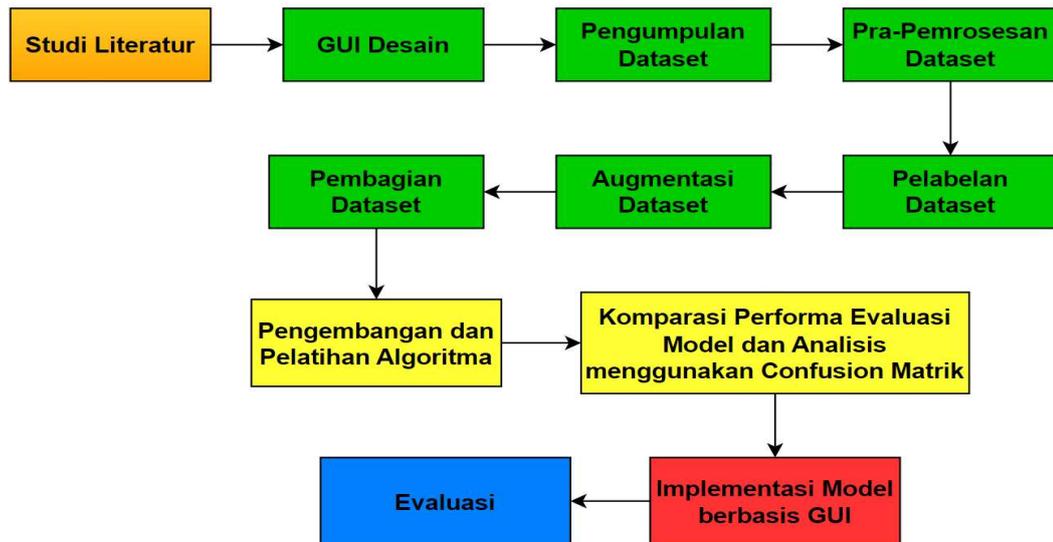
Berlandaskan pada fokus dan tujuan penelitian yang telah dirumuskan sebelumnya, penelitian ini dimaksudkan untuk mengetahui algoritma yang terbaik antara penggunaan Faster R-CNN dengan YOLOV8 untuk diimplementasikan mendeteksi cacat yang ditemukan pada PCB layout. Penelitian ini menggunakan metode penelitian dan pengembangan atau *Research and Development (R&D)*. Dengan menggunakan metode R&D ini, kedua algoritma yang dianalisis dan dilakukan komparasi dapat diimplementasikan dari hasil model yang terbaik.

Model pengembangan yang digunakan dalam penelitian ini menggunakan pendekatan ADDIE (*Analysis, Design, Development, Implementation, and Evaluation*). ADDIE dikembangkan oleh Reiser dan Mollenda yang digagas pada tahun 1990-an. Alasan peneliti menggunakan model pengembangan ini karena ADDIE memiliki prosedur kerja yang mengacu pada tahapan penelitian dan pengembangan (R&D) tetapi lebih sederhana dan sistematis, yang memungkinkan pengembangan dan analisis algoritma menjadi lebih efisien (Jaja Kustija et al., 2022).



Gambar 3. 1 Model Pengembangan ADDIE
(Sumber : (Fahrizal et al., 2024))

Adapun langkah-langkah penelitian dan pengembangan model ADDIE yang merujuk pada tahapan *Research and Development (R&D)*, yaitu dijelaskan sebagai berikut.



Gambar 3. 2 Prosedur Penelitian berdasar Model Pengembangan ADDIE
(Sumber : Dokumen pribadi)

1. Tahapan Analisis (Analysis)

Pada tahap ini, masalah diidentifikasi melalui studi literatur dan observasi masalah yang ada. Salah satu masalah yang diidentifikasi adalah masalah mendeteksi cacat pada layout PCB secara manual. Dilakukan juga analisis bibliometrik untuk mengetahui kebaruan penelitian yang akan dilakukan terkait dengan solusi untuk mendeteksi cacat pada PCB layout dengan metode deep learning.

2. Tahapan Desain (Design)

Tahapan ini berfokus pada perancangan infrastruktur data, desain GUI, dan teknis seperti hardware dan software. Data yang diperlukan termasuk dataset gambar PCB dan lingkungan pengembangan yang menggunakan Google Colab dan PyTorch.

3. Tahapan Pengembangan (Development)

Pada tahapan ini, dataset yang telah dikumpulkan diproses menggunakan teknik seperti anotasi, normalisasi, dan augmentasi. Selanjutnya, arsitektur model dirancang, backbone algoritma ditentukan, dan hyperparameter diatur sebelum melakukan pelatihan. Model dibangun dan dilatih dengan menggunakan pendekatan dua tipe dataset, yaitu dataset yang tidak dilakukan augmentasi dan dataset yang dilakukan augmentasi. Setelah dilakukan pelatihan, masing-masing model dianalisis dari segi performa menggunakan *confusion matrix* untuk

dilakukan analisis komparatif, menentukan algoritma dengan model yang terbaik dari segi performa kecepatan dan akurasi mendeteksi serta ketepatan objek cacat yang dideteksi.

4. Tahapan Implementasi (Implementation)

Algoritma yang telah dilakukan analisis komparatif dan menghasilkan model yang terbaik dari segi performa, kemudian dikembangkan dan diintegrasikan ke dalam desain GUI yang telah dirancang dengan PyQt5. Implementasi ini diuji untuk memastikan bahwa model beroperasi dengan baik dalam lingkungan nyata dan mampu mendeteksi cacat dengan akurasi yang tinggi.

5. Tahapan Evaluasi (Evaluation)

Tahap akhir ini mengevaluasi efisiensi dan efektivitas model terbaik yang diintegrasikan melalui GUI dan dilakukan pengujian untuk menganalisis kinerja performa aplikasi GUI yang dirancang dalam menemukan berbagai cacat PCB.

3.2. Proses Perancangan dan Pengembangan

3.2.1. Pengumpulan Dataset

Penelitian ini menggunakan dataset HRIPCB (*High Resolution Image PCB*), sebuah kumpulan data gambar PCB sintesis yang dikembangkan oleh (Huang et al., 2020) khusus untuk studi di bidang deteksi dan klasifikasi cacat PCB. Dataset ini, dibuat untuk mendukung komunitas penelitian dengan menyediakan data yang mewakili kondisi nyata PCB yang semakin kompleks dan padat.



Gambar 3. 3 Pengembangan dataset HRIPCB
(Sumber : google.com)

Proses pengembangan dataset ini melibatkan simulasi sistem Inspeksi Optik Otomatis (AOI) yang umum di industri. Ini dilakukan dengan menggunakan kamera industri HD 16-megapiksel dengan sensor CMOS dan pencahayaan

Diki Fahrizal, 2024

ANALISIS KOMPARATIF ALGORITMA FASTER R-CNN DAN YOLOV8 UNTUK DETEKSI OBJEK CACAT PCB LAYOUT (IMPLEMENTASI GRAPHICAL USER INTERFACE)

Universitas Pendidikan Indonesia | repository.upi.edu | perpustakaan.upi.edu

canggih, menghasilkan gambar berkualitas tinggi dengan resolusi asli 4608 x 3456 piksel. Gambar-gambar ini disesuaikan berdasarkan ukuran setiap papan PCB yang diperiksa.

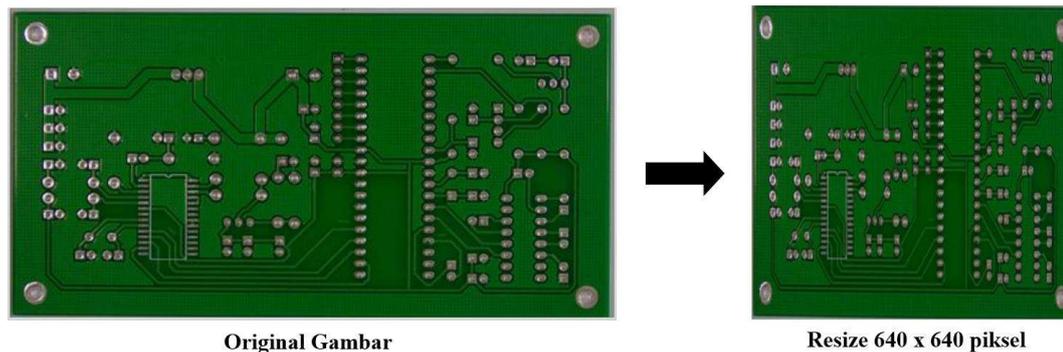
Untuk menciptakan variasi kondisi cacat, dataset ini termasuk enam jenis cacat yang sering terjadi pada PCB, yaitu missing hole, mouse bite, open circuit, short, spur, dan spurious copper. Setiap cacat ini dibuat secara artifisial menggunakan Photoshop, untuk meniru kondisi nyata seakurat mungkin. Referensi gambar PCB yang digunakan untuk pembuatan cacat ini terdapat dalam bagian **Lampiran 3**.

3.2.2. Pra-Pemrosesan Dataset

Untuk memastikan dataset siap untuk digunakan dalam pelatihan model, proses pre-processing data merupakan langkah selanjutnya setelah proses pengumpulan dataset. ada dua metode pre-processing yang digunakan dalam penelitian ini yaitu Auto Orient dan Streect to 640 x 640 piksel dengan bantuan platform Roboflow.

Langkah pertama dalam pre-processing adalah penerapan proses Auto-Orient karena dataset yang digunakan terdiri dari gambar gambar yang memiliki orientasi berbeda. Proses ini memastikan bahwa semua gambar memiliki orientasi yang sama dengan mengikuti standar yang telah diterapkan dan memastikan model algoritma selama proses pelatihan tidak terganggu oleh variansi orientasi yang ada pada gambar dataset.

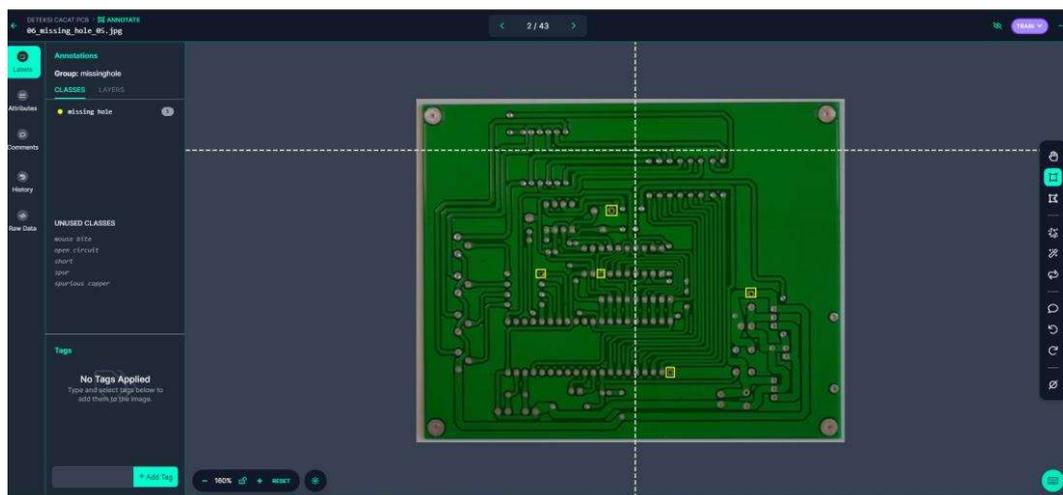
Setelah mengubah orientasi gambar, langkah selanjutnya mengubah ukuran semua gambar dalam kumpulan dataset menjadi 640 x 640 piksel. Proses ini dilakukan untuk menstandarisasi ukuran gambar untuk memastikan model dapat belajar secara maksimal tanpa harus menyesuaikan diri dengan berbagai ukuran gambar yang luas. Teknik stretching digunakan untuk mengubah ukuran gambar sehingga seluruh gambar tetap terlihat tanpa memotong bagian penting dari gambar tersebut.



Gambar 3. 4 Proses Resizing Dataset Gambar
(Sumber : Dokumen Pribadi)

3.2.3. Dataset Anotasi / Labeling

Pada tahap anotasi, setiap gambar dataset diberi bounding box untuk menandai lokasi dan jenis cacat pada PCB. Proses ini dilakukan menggunakan platform Roboflow, yang memastikan bahwa setiap cacat ditandai dengan koordinat x dan y secara jelas. Untuk deteksi cacat yang efektif, keakuratan anotasi sangat penting.



Gambar 3. 5 Proses Pelabelan Dataset Gambar
(Sumber : Dokumen Pribadi)

Setiap gambar tidak hanya dianotasi tetapi juga diberi label berdasarkan jenis cacatnya. Label ini sangat penting untuk model untuk mengklasifikasikan dan menemukan cacat dengan benar, seperti missing hole, mouse bite, open circuit, short, spur, dan spurious copper.

Untuk mempermudah penyimpanan dan akses dalam penelitian, dataset mengadopsi konvensi penamaan yang konsisten untuk gambar dan file anotasi. Misalnya, gambar dinamai sebagai *[Referensi gambar PCB]_[Jenis cacat yang*

Diki Fahrizal, 2024

ANALISIS KOMPARATIF ALGORITMA FASTER R-CNN DAN YOLOV8 UNTUK DETEKSI OBJEK CACAT PCB LAYOUT (IMPLEMENTASI GRAPHICAL USER INTERFACE)

Universitas Pendidikan Indonesia | repository.upi.edu | perpustakaan.upi.edu

teridentifikasi]_[Nomor urut].jpg, dan file anotasi dinamai sama, hanya dengan ekstensi yang berbeda untuk membedakan format antara XML untuk Faster R-CNN dan TXT untuk YOLOv8, contohnya [*Referensi gambar PCB]_[Jenis cacat yang teridentifikasi]_[Nomor urut].xml / .txt*.

3.2.4. Dataset Splitting

Setelah melakukan proses pre-processing dan anotasi dataset, langkah selanjutnya dalam mempersiapkan dataset yaitu membaginya menjadi tiga subset, diantaranya data pelatihan, data validasi, dan data pengujian. Pembagian dataset ini sangat penting untuk melatih model, mengevaluasi kinerja selama proses pelatihan dan pembelajaran, serta menguji kemampuan untuk memprediksi data baru dengan akurat. Dalam penelitian ini, dataset dibagi dengan proporsi seperti pada **Tabel 3.1** di bawah ini.

Tabel 3. 1 Proporsi Pembagian Dataset

Data Pelatihan (Train)	Data Validasi (Val)	Data Pengujian (Test)
85%	10%	5%

Pada **Tabel 3.1**, data pelatihan digunakan untuk melatih model dengan memberikan contoh data membantu model memahami pola dan fitur terkait dengan berbagai jenis cacat pada PCB. Data validasi digunakan untuk mengevaluasi kinerja model selama proses pelatihan dan menyempurnakan parameter model. Selain itu, membantu dalam menemukan masalah seperti overfitting, underfitting, sehingga dapat menyesuaikan hyperparameter yang digunakan. Sedangkan data pengujian, digunakan untuk menguji kinerja model setelah proses pelatihan selesai. Data pengujian memberikan evaluasi yang tidak bias tentang seberapa baik model dapat menggeneralisasi pada data yang sebelumnya tidak diamati selama pelatihan.

3.2.5. Dataset Augmentation

Augmentasi data adalah salah satu tahapan strategi penting dalam penelitian ini untuk meningkatkan generalisasi model dengan meningkatkan variasi data latih tanpa harus mengumpulkan lebih banyak data nyata. Hal ini penting untuk mengurangi risiko overfitting, dimana model menghafal terlalu banyak data pelatihan tetapi tidak dapat menggeneralisasikan ke data baru. Dalam penelitian ini untuk pelatihan model deteksi cacat pada layout PCB, teknik

augmentasi yang digunakan yaitu *90° Rotate (Clockwise, Counter-Clockwise, Upside Down)*, *Rotation (Between -15° and +15°)*, dan *Shear ($\pm 10^\circ$ Horizontal, $\pm 10^\circ$ Vertical)*.

3.2.6. Pengembangan dan Training Model

3.2.6.1. Faster R-CNN

Model Faster R-CNN yang akan digunakan dalam pengembangan dan pelatihan pada penelitian ini bersumber pada dokumentasi framework Pytorch yang dapat diakses pada https://pytorch.org/vision/main/models/faster_rcnn.html. Backbone yang digunakan dalam model ini ResNet50 FPN dan ResNet50 FPN V2 yang dilatih dengan sumber dataset yang telah disiapkan dan hasil pelatihan diambil bobot inisiasi yang terbaik diantara keduanya. Adapun arsitektur model yang digunakan secara umum seperti pada **Gambar 2.12**, untuk detail setiap lapisannya seperti pada **Tabel 3.2** di bawah ini.

Tabel 3. 2 Komponen Arsitektur Faster R-CNN yang Digunakan

Komponen	Input Shape	Output Shape	Param#	Keterangan
Faster R-CNN	[8, 3, 640, 640]	[100, 4]	-	Model keseluruhan
Generalized R-CNN Transform	[8, 3, 640, 640]	[8, 3, 640, 640]	-	Transformasi umum gambar
Backbone with FPN	[8, 3, 640, 640]	[8, 256, 10, 10]	-	Backbone dengan FPN
Intermediate Layer Getter	[8, 3, 640, 640]	[8, 2048, 20, 20]	-	Ekstraktor fitur utama
Conv2d + BatchNorm2d + ReLu	[8, 3, 640, 640]	[8, 64, 320, 320]	9536	Konvolusi awal, normalisasi, dan aktivasi
MaxPool2d	[8, 64, 320, 320]	[8, 64, 160, 160]	-	Pooling maksimum

Komponen	Input Shape	Output Shape	Param#	Keterangan
Sequential (layers 1 – 4)	[8, 64, 160, 160]	[8, 2048, 20, 20]	23498496	Empat layer konvolusi berurutan
Feature Pyramid Network (FPN)	[8, 256, 160, 160]	[8, 256, 10, 10]	-	FPN untuk ekstraksi fitur skala multi
Region Proposal Network (RPN)	[8, 3, 640, 640]	[1000, 4]	-	Jaringan proposal region
RPN Head	Ukuran variasi	Ukuran Variasi	1184015	Kepala RPN untuk prediksi
Anchor Generator	[8, 3, 640, 640]	[102300, 4]	-	Generator anchor untuk RPN
RoI Heads	[8, 256, 160, 160]	[100, 4]	-	Kepala RoI untuk deteksi objek
Multi Scale RoI Align	[8, 256, 160, 160]	[8000, 256, 7, 7]	-	RoI Align pada berbagai skala
Fast R-CNN Conv Fully Connected Head	[8000, 256, 7, 7]	[8000, 1024]	14216088	Kepala Fast R- CNN
Fast R-CNN Predictor	[8000, 1024]	[8000, 7] (classes)	7175	Prediktor untuk klasifikasi
		[8000, 28] (bbox)	28700	Prediktor untuk bounding box

3.2.6.2. YOLOv8

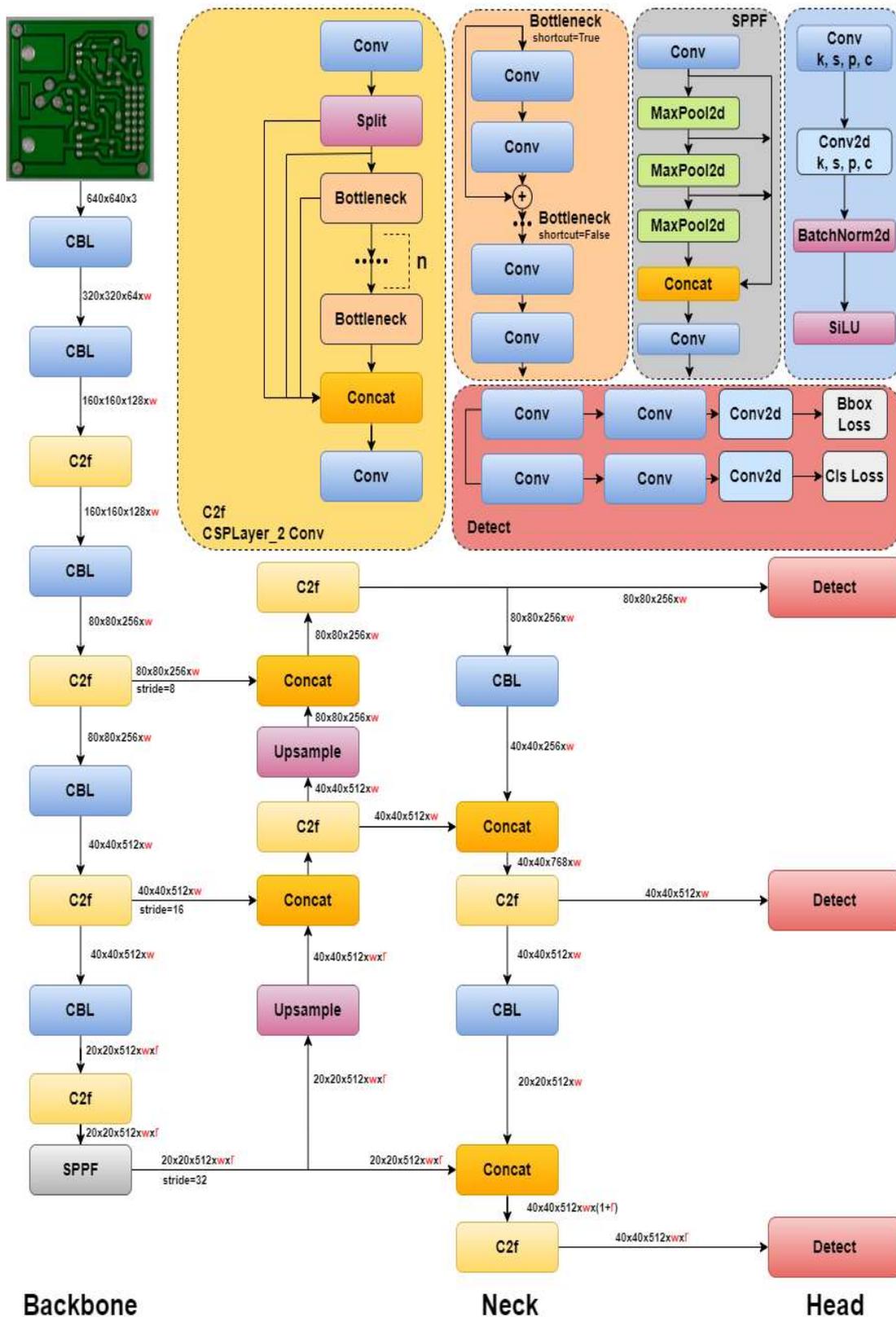
Model YOLOv8 yang akan digunakan dalam pengembangan dan pelatihan pada penelitian ini menggunakan model yang telah dirancang sebelumnya menggunakan framework Pytorch yang diciptakan oleh Ultralytics, sebuah perusahaan pengembangan AI dan machine learning khususnya dalam bidang visi

Diki Fahrizal, 2024

ANALISIS KOMPARATIF ALGORITMA FASTER R-CNN DAN YOLOV8 UNTUK DETEKSI OBJEK CACAT PCB LAYOUT (IMPLEMENTASI GRAPHICAL USER INTERFACE)

Universitas Pendidikan Indonesia | repository.upi.edu | perpustakaan.upi.edu

komputer. Dokumentasi YOLOv8 dapat diakses melalui website resmi Ultralytics yaitu <https://docs.ultralytics.com/> yang menyediakan berbagai kebutuhan selama proses pengembangan model. Pelatihan model YOLOv8 menggunakan teknik transfer learning dengan cara *fine tuning* dari bobot inisiasi yang telah disediakan oleh Ultralytics, nantinya bobot hasil pelatihan diambil yang terbaik dan diberi label file *best.pt*. Adapun arsitektur YOLOv8 yang digunakan dalam penelitian ini seperti pada **Gambar 3.6**.



Gambar 3. 6 Arsitektur Algoritma YOLOv8 yang Digunakan (Sumber : Dokumen Pribadi)

Pada **Gambar 3.6** merupakan arsitektur model YOLOv8 yang digunakan dalam penelitian ini dengan mengadaptasi dari dokumentasi Ultralytics. YOLOv8 menggunakan CSPDarkNet53 sebagai extractor fitur dengan perubahan pada lapisan CSP, modul C2f yang merupakan *bottleneck cross-stage partial* dengan dua konvolusi menggabungkan informasi kontekstual dengan fitur tingkat tinggi. Model YOLOv8 menggunakan pendekatan *anchor-free* dan memiliki *head* deteksi terpisah untuk menangani tugas *objectness*, klasifikasi, dan regresi secara independen. Sementara *loss bounding box* menggunakan fungsi *loss* CIOU dan DFL, sementara untuk *loss classification* menggunakan *binary cross-entropy*. Dalam penelitian ini menggunakan lima varian model YOLOv8, dengan tingkat karakteristik setiap varian berbeda-beda untuk setiap arsitektur. Berikut **Tabel 3.3** menunjukkan varian model YOLOv8 yang digunakan dalam penelitian ini.

Tabel 3. 3 Varian Model YOLOv8

Model	d (<i>depth_multiple</i>)	w (<i>width_multiple</i>)	r (<i>ratio</i>)
YOLOv8n (<i>nano</i>)	0,33	0,25	2,00
YOLOv8s (<i>small</i>)	0,33	0,50	2,00
YOLOv8m (<i>medium</i>)	0,67	0,75	1,50
YOLOv8l (<i>large</i>)	1,00	1,00	1,00
YOLOv8x (<i>extra large</i>)	1,00	1,25	1,00

3.2.7. Evaluasi Performa Model

Tahapan evaluasi performa model mengevaluasi ketepatan dan ketepatan model Faster R-CNN dan YOLOv8 dalam menemukan cacat pada layout PCB. Untuk evaluasi ini, dua jenis dataset digunakan: dataset asli tanpa augmentasi dan hasil augmentasi. Ini memungkinkan untuk menilai dampak perubahan dataset terhadap efektivitas model.

Untuk Faster R-CNN, pengukuran Mean Average Precision (mAP) diperlukan untuk evaluasi pada ambang batas IoU 50% untuk dasar akurasi dan pada rentang 50%–95% untuk tingkat kesulitan yang lebih tinggi. Metrik lainnya termasuk kehilangan total, kehilangan klasifikasi, kehilangan regresi bounding box, kehilangan klasifikasi RPN, dan kehilangan RPN, yang menunjukkan area perbaikan yang mungkin terjadi selama proses pelatihan dan validasi.

Diki Fahrizal, 2024

ANALISIS KOMPARATIF ALGORITMA FASTER R-CNN DAN YOLOV8 UNTUK DETEKSI OBJEK CACAT PCB LAYOUT (IMPLEMENTASI GRAPHICAL USER INTERFACE)

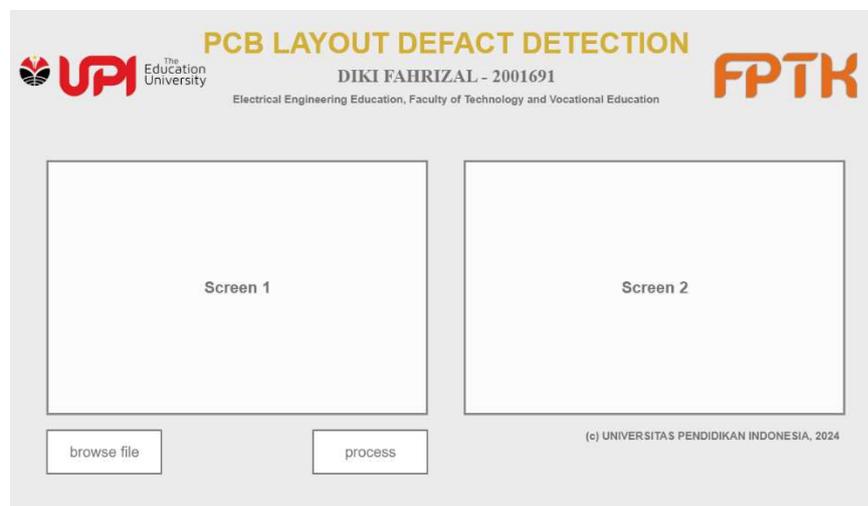
Universitas Pendidikan Indonesia | repository.upi.edu | perpustakaan.upi.edu

Meskipun demikian, YOLOv8 menggunakan matrik yang serupa untuk keseragaman evaluasi, seperti ketepatan dan recall, yang menentukan seberapa baik model menemukan kelas positif. Untuk menggabungkan precision dan recall dalam satu ukuran, skor F1 dihitung, dan confusion matrix memberikan gambaran visual tentang kinerja klasifikasi model. Penilaian ini juga mencakup penilaian umum antara precision dan recall untuk semua nilai ambang batas, yang menunjukkan efektivitas model dalam berbagai kondisi.

Kedua model tersebut dievaluasi untuk mengidentifikasi berbagai jenis cacat pada PCB. Hasilnya menunjukkan bahwa mereka dapat digunakan dalam sistem GUI.

3.2.8. Deployment

Setelah dilakukan analisis evaluasi performa, model terbaik dipilih untuk dilakukan deployment, yaitu proses mengintegrasikan model ke dalam penggunaan nyata. Model ini diimplementasikan dalam sebuah Graphical User Interface (GUI) yang dikembangkan menggunakan framework PyQt5, dipilih karena kemudahan penggunaannya, dukungan lintas platform, dan integrasi yang baik dengan Python.



Gambar 3. 7 Desain Graphical User Interface
(Sumber : Dokumen Pribadi)

Memiliki dua layar utama, GUI yang dirancang agar pengguna dengan mudah mengunggah gambar PCB dan melihat hasil analisis cacat secara visual. Layar 1 menampilkan gambar yang diunggah, dan Layar 2 menampilkan hasil analisis. Tombol "*browse file*" memungkinkan pengguna memilih gambar dan

Diki Fahrizal, 2024

ANALISIS KOMPARATIF ALGORITMA FASTER R-CNN DAN YOLOV8 UNTUK DETEKSI OBJEK CACAT PCB LAYOUT (IMPLEMENTASI GRAPHICAL USER INTERFACE)

Universitas Pendidikan Indonesia | repository.upi.edu | perpustakaan.upi.edu

tombol "*proses*" memungkinkan mereka memprosesnya. Tujuan desain GUI ini adalah untuk membuat pengoperasian lebih mudah dan dapat diakses di berbagai sistem operasi.

3.3. Lingkungan Komputasi

Lingkungan komputasi memainkan peran kunci dalam penelitian teknologi informasi, terutama dalam pengembangan dan pelatihan model pembelajaran mesin. Untuk mendukung penelitian ini, lingkungan komputasi meliputi perangkat keras dan perangkat lunak yang sesuai.

Perangkat keras untuk pengembangan atau spesifikasi laptop yang digunakan dalam penelitian ini yaitu sebagai berikut.

- CPU AMD Athlon™ 300U @ 3.3GHz
- RAM 8 GB
- SSD 256 GB
- GPU AMD Radeon™ Vega 3 Mobile Graphics

Perangkat keras yang digunakan untuk eksperimen serta selama proses pelatihan, validasi, dan testing model objek deteksi cacat pada PCB layout menggunakan spesifikasi perangkat keras dari Google Collaboratory, yaitu sebagai berikut.

- Prosesor Intel® Xeon® CPU @ 2.20 GHz
- RAM 16 GB
- GPU NVIDIA Tesla T4 (eksperimen) dan Tesla V100 (pelatihan, validasi, dan testing) dengan VRAM 16 GB

Selanjutnya adapun perangkat lunak (*software*) yang digunakan dalam penelitian ini, seperti pada **Tabel 3.4** di bawah ini.

Tabel 3. 4 Perangkat Lunak (Software) yang Digunakan

Kategori	Perangkat Lunak
Sistem Operasi	Microsoft Windows 11, sebagai operasi utama untuk pengembangan.
Alat Pengembangan dan IDE	Jupyter Notebook, untuk eksperimen interaktif dan dokumentasi <i>source code</i> . Google Colaboratory, untuk pengembangan dan

Kategori	Perangkat Lunak
	pelatihan model dengan akses ke GPU/TPU berbasis cloud.
	PyCharm , sebagai IDE untuk pengembangan GUI dengan menggunakan versi 2023.3.5.
Manajemen Data dan Kolaborasi	Google Drive , penyimpanan dataset gambar dan hasil model yang telah dilatih sebelum masuk ke repository lokal.
	GitHub , sebagai dokumentasi proyek penelitian yang dapat diakses oleh orang lain.
Bahasa Pemrograman dan Framework	Python, sebagai bahasa pemrograman utama dalam penelitian ini, menggunakan versi 3.10.12.
	PyTorch , sebagai framework dalam pengembangan algoritma Faster R-CNN dan YOLOv8.
	PyQt5 , sebagai framework dalam pengembangan GUI aplikasi
Tools Lainnya	Roboflow , sebagai platform anotasi, pengelolaan, dan augmentasi dataset penelitian.
	Netron , sebagai platform visualisasi arsitektur model machine learning untuk memudahkan analisis struktur model.