

Metode Deteksi Obyek Berbasis Computer Vision dan Deep Learning untuk Sistem Inspeksi Cacat pada Permukaan Printed Circuit Board (PCB)

1st Dhimas Prasetyantono Adji

Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia

dhimaspras@student.telkomuniversity.ac.id

2nd Fityanul Akhyar

Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia

fityanul@telkomuniversity.ac.id

3rd Ledy Novamizanti

Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia

ledyaldn@telkomuniversity.ac.id

Abstrak — Sistem inspeksi cacat permukaan Printed Circuit Board (PCB) memainkan peran penting dalam memastikan kualitas produk elektronik. Dalam studi ini, kami mengusulkan metode pendeteksian objek berbasis computer vision dan deep learning untuk meningkatkan efisiensi dan akurasi proses pemeriksaan PCB. Salah satu poin utama dari metode ini adalah subsistem GUI yang dimaksudkan untuk memfasilitasi interaksi pengguna dan visualisasi hasil pemeriksaan. GUI dikembangkan menggunakan kerangka perangkat lunak yang intuitif dan interaktif. Antarmuka pengguna memiliki fungsi seperti memilih gambar papan sirkuit, mengatur parameter deteksi, dan melihat hasil pemeriksaan. Pengguna dapat melihat visualisasi yang jelas dari lokasi dan jenis cacat yang terdeteksi pada PCB. Selain itu, subsistem GUI memungkinkan pengguna memvalidasi dan menyimpan hasil pemeriksaan untuk analisis lebih lanjut. Studi ini memiliki potensi untuk menyederhanakan proses inspeksi di industri elektronik dengan mengurangi waktu dan biaya yang diperlukan untuk mendeteksi dan memperbaiki cacat PCB. Subsistem GUI yang dikembangkan dapat diintegrasikan ke dalam sistem inspeksi yang ada dan dapat digunakan dengan mudah dan efisien oleh inspektur atau operator pabrik.

Kata kunci— Printed Circuit Board (PCB), computer vision, deep learning, subsistem GUI

I. PENDAHULUAN

Dalam industri elektronik, kualitas dan keandalan produk merupakan faktor penting yang mempengaruhi reputasi perusahaan dan kepuasan pelanggan. Papan sirkuit tercetak (PCB) adalah komponen penting dalam perangkat elektronik, dan kegagalan apa pun di sirkuit dapat menyebabkan kegagalan seluruh sistem. Oleh karena itu, mengembangkan metode inspeksi yang efektif untuk mendeteksi dan mengklasifikasikan cacat PCB sangatlah penting[1].

Dalam beberapa dekade terakhir, teknologi computer vision dan deep learning telah menjadi sorotan dalam bidang pengenalan pola dan deteksi objek[2]. Metode berbasis computer vision memungkinkan analisis visual otomatis pada citra atau video dengan menggunakan algoritma dan model yang dapat memahami dan mengekstraksi informasi yang bermanfaat. Sedangkan deep

learning, sebagai cabang dari machine learning, telah terbukti sangat sukses dalam pengenalan pola kompleks dan pengklasifikasian data [3].

Banyak penelitian mengenai inspeksi kecacatan pada sebuah PCB yang telah dilakukan oleh peneliti sebelumnya. Pada tahun 2019, Runwie Ding dkk mengusulkan metode TDD-net: Tiny Defect Detection Network untuk deteksi PCB. Hasil penelitian ini mendapatkan nilai Mean Average Precision sebesar 98,9% namun, pada hubungan peta fitur dari beberapa tingkat dan manfaat informasi struktural pada tingkat deteksi masih rendah[4].

Penelitian yang dilakukan oleh Muhammad Azka dkk. pada tahun 2022 sistem dapat mendeteksi enam jenis cacat pada PCB menggunakan algoritma ResNest-101 sebagai arsitektur metode deteksi objek dan dihasilkan nilai akurasi yaitu nilai mean Average Precision (mAP) sebesar 99,2% serta nilai Average Recall (AR) sebesar 99,5%[5].

Nur Kurniasari dan Judi Prajetno Sugiono pada tahun 2021 telah melakukan penelitian mengenai cacat PCB menggunakan metode Convolutional Neural Network dan arsitektur Yolov5 secara bersamaan memprediksi beberapa kotak pembatas serta probabilitas kelas. Berdasarkan fleksibilitas yang digunakan berjenis RIGID PCB, berfokus kepada enam jenis kerusakan yaitu sirkuit terbuka, jalur pendek, gigitan tikus, spur, lubang jarum, dan bola solder. Penelitian ini menghasilkan nilai akurasi klasifikasi tertinggi sebesar 80,7% dengan rata-rata deteksi kotak pembatas sebesar 80% dan kecepatan klasifikasi deteksi jalur terputus 1,89 it/s menggunakan nilai threshold 0,267 dan 0,283[6].

Pada penelitian ini, fokus kami adalah pengembangan subsistem GUI yang terintegrasi dengan metode deteksi obyek berbasis computer vision dan deep learning untuk sistem inspeksi cacat pada permukaan PCB. Subsistem GUI ini bertujuan untuk meningkatkan interaksi pengguna dengan sistem inspeksi dan memfasilitasi pemahaman yang lebih baik tentang hasil inspeksi. Pengguna dapat melihat visualisasi yang jelas tentang lokasi dan jenis cacat yang terdeteksi pada PCB, mengatur parameter deteksi, serta memvalidasi dan menyimpan hasil inspeksi.

II. KAJIAN TEORI

A. Graphical User Interface (GUI)

Graphical user interface pertama kali dikembangkan oleh Xerox Pako Alto pada tahun 1970 akhir. GUI merupakan Suatu antarmuka (user interface) yang memungkinkan pengguna dapat berinteraksi dengan sebuah perangkat[7], dimana pengguna melalui objek-objek grafis seperti tombol (button), checkbox, dan menu sehingga memungkinkan pengguna pemula sekalipun dengan mudah menavigasi berbagai fungsi komputer.

B. OpenCV

OpenCV (Open-Source Computer Vision Library) adalah sebuah perangkat lunak yang dirancang untuk melakukan pengolahan secara real time pada citra digital. Saat ini OpenCV telah mendapatkan dukungan dari perusahaan Willow Garage dan Itseez. Perangkat lunak ini disediakan secara gratis yang menjadikan hal ini memungkinkan pengguna menggunakannya secara bebas[8]. Tujuannya untuk menyediakan alat yang kuat untuk pengolahan citra secara instan. Adapun terdapat fitur yang berguna, termasuk beberapa fitur utama sebagai berikut:

1. Pengolahan Citra

OpenCV menyediakan berbagai fungsi untuk pengolahan citra, termasuk operasi dasar seperti pemotongan, pemutaran, dan perubahan ukuran gambar. Selain itu, fitur-fitur lanjutan seperti peningkatan kecerahan/kontras, penajaman gambar, penghapusan noise, dan penghilangan latar belakang juga dapat dilakukan dengan menggunakan OpenCV.

2. Sampling gambar dan transformasi

OpenCV memiliki kemampuan untuk melakukan transformasi geometri pada citra. Ini termasuk rotasi, pergeseran, dan skalasi citra. OpenCV juga mendukung transformasi perspektif yang lebih kompleks, seperti pengaturan ulang perspektif pada citra untuk mengoreksi distorsi.

3. Segmentasi Citra.

Fitur segmentasi citra pada OpenCV memungkinkan pemisahan objek dari latar belakangnya. Metode seperti thresholding, pengklasifikasi piksel (pixel classification), pemrosesan morfologi, dan pemisahan warna dapat digunakan untuk segmentasi citra dalam OpenCV.

C. Tkinter

Tkinter merupakan pustaka grafis secara langsung didalam python yang dapat memberikan kemudahan bagi programmer dalam pembuatannya. Tkinter menyediakan widget seperti *button*, *scrollbar*, *listbox*, *checkboxbutton* dan lain sebagainya. Widget di definisikan mengkapsulasikan detail implementasi sehingga mempermudah pemrograman GUI.

D. PyTorch

PyTorch merupakan sebuah kerangka kerja Machine Learning yang bersifat sumber terbuka, yang mendasarkan fungsionalitasnya pada bahasa pemrograman Python. Daya tarik dan penerimaan terhadap PyTorch terus mengalami

peningkatan seiring waktu berjalan. Keunikan PyTorch terletak pada kemampuannya untuk mempermudah proses pembuatan model jaringan saraf tiruan (neural network) bagi para pengembang. Dengan menyediakan alat yang intuitif, PyTorch memungkinkan para praktisi Machine Learning untuk merancang dan mengembangkan model-model tersebut dengan lebih mudah[9].

III. METODE

A. Pengambilan Citra

Seperti pada Tabel 1, pengambilan Citra ini menggunakan dataset publik yang di rilis oleh Peking University. Dataset memiliki 6 subclass cacat sebagai klasifikasi, yaitu *missing hole*, *open circuit*, *short*, *spur*, *spurious copper*, dan *mouse bite* dengan total 10.668 gambar yang di sesuaikan dengan ukuran 600 x 600. Dataset menyediakan file anotasi bounding box dalam format XML.

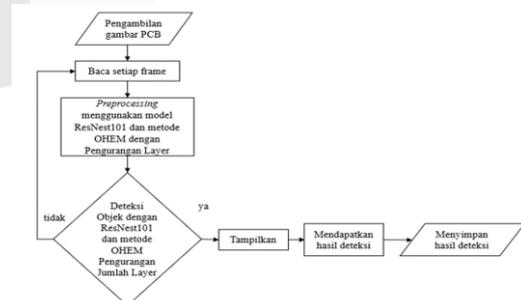
TABEL 1
Dataset dengan 6 class cacat

Label	Jumlah Gambar	Total Gambar Dataset	Format Gambar
Missing Hole	1832	10668	JPG
Open Circuit	1740		
Short	1732		
Mouse Bite	1852		
Spur	1752		
Spurious Copper	1760		

B. Perancangan Sistem

Proses perancangan sistem ini diawali dengan identifikasi dan analisa kebutuhan sistem yang akan dirancang. Ada beberapa hal penting untuk ada didalam sistem yang ingin dibangun yaitu, sistem dapat mendeteksi objek kecacatan PCB yang berada pada sebuah gambar, dan output dari sistem adalah menyimpan hasil deteksi berupa gambar kedalam file direktori yang sudah ditentukan.

Proses tahapan sebuah sistem yang akan dirancang dimulai dengan melakukan pelatihan model dengan file-file gambar sebagai input yang menghasilkan output model yang telah dilatih. File GUI ketika berjalan, akan memunculkan sebuah tampilan layar sebagai kamera dan button capture. Langkah selanjutnya adalah pembuatan flowchart yang dapat dilihat pada Gambar 1.

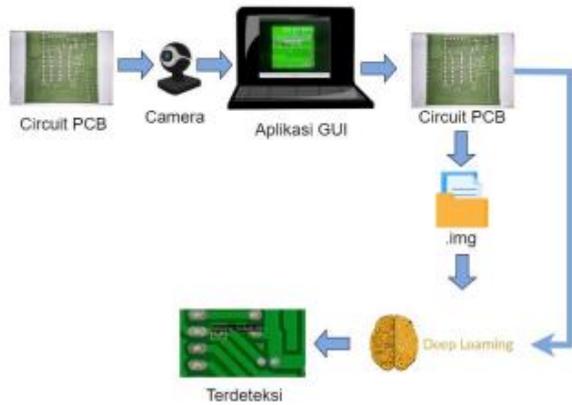


GAMBAR 1
Flowchart cara kerja GUI

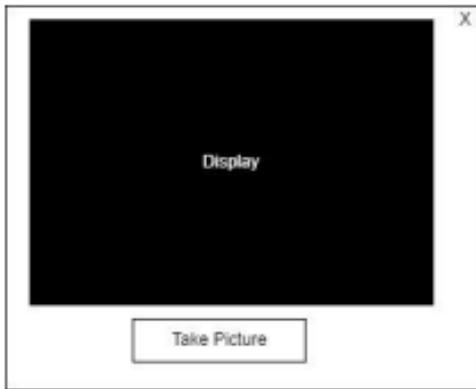
C. Implementasi

Setelah mengetahui cara kerja sistem, selanjutnya adalah implementasi sistem menggunakan bahasa pemrograman Python untuk membuat tampilan visual Graphical User

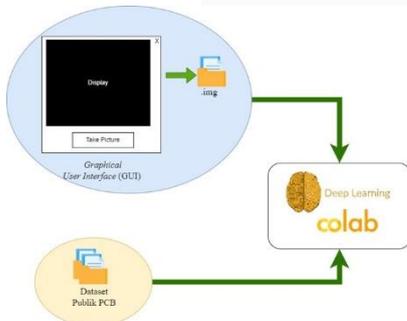
Interface (GUI) dengan ukuran layar 640x840 pixel serta menambahkan interaksi button *Capture* dan *DisplayBox*. Seperti pada Gambar 3.



GAMBAR 2
Cara Kerja PCB secara umum



GAMBAR 3
Ilustrasi Aplikasi GUI



GAMBAR 4
Ilustrasi Deep Learning

Pada Gambar 2 merupakan cara kerja deteksi PCB secara umum. Sistem menggunakan aplikasi GUI untuk mengambil gambar circuit PCB secara langsung dibantu oleh kamera. Kemudian gambar akan diproses dan menghasilkan deteksi cacat PCB. Selanjutnya gambar akan di simpan pada perangkat. Pada deep learning, menggunakan Google Colab Pro sebagai framework. Sistem ini menggunakan PyTorch yang merupakan kerangka kerja dari deep learning untuk membangun sebuah model.

Sistem selanjutnya adalah implementasi model deep learning seperti pada Gambar 4. User mengunggah gambar yang telah diambil oleh GUI atau dari dataset publik kedalam Google Colab Pro. Kemudian, sistem melakukan

proses pengolahan citra deteksi objek menggunakan model pretrained yaitu ResNest-101 yang ditambah metode OHEM. Selanjutnya, sistem akan mendeteksi kerusakan PCB secara akurat dan menghasilkan nilai akurasi yang diinginkan.

D. Pengujian

Pengujian GUI dilakukan pada setiap fitur yang diterapkan dengan menggunakan gambar sirkuit PCB yang mengalami kerusakan. Pengujian ini menggunakan spesifikasi perangkat Nvidia Core dengan memory 4 GB serta kamera eksternal Inbex 720p dengan skenario yang sesuai pada Tabel 2.

TABEL 2
Pengujian GUI

Fitur	Skenario Pengujian	Hasil Pengujian	Status
Display	Kamera menampilkan gambar PCB dan mendeteksi secara real-time	Kamera berhasil berfungsi mendeteksi secara real-time	Berhasil
Button Capture	Button dapat menyimpan gambar hasil deteksi objek.	Button Capture berfungsi menyimpan hasil deteksi	Berhasil

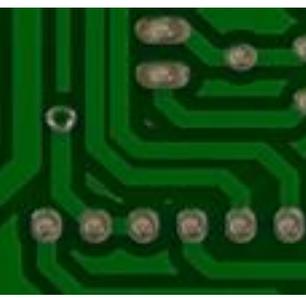
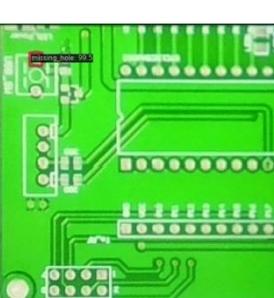
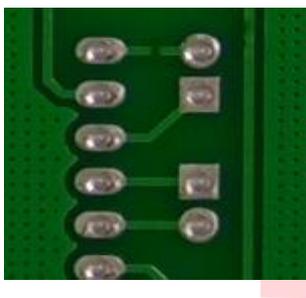
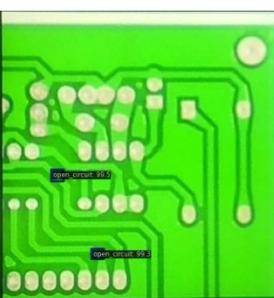
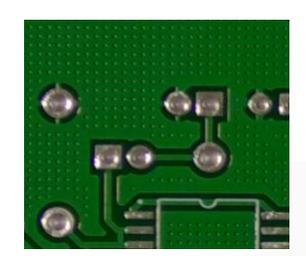
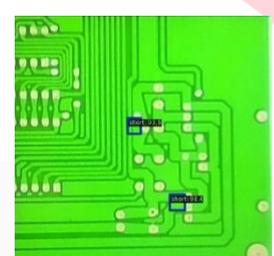
IV. HASIL DAN PEMBAHASAN

A. Hasil Pengujian Gambar Terdeteksi

Pengujian untuk sistem deteksi cacat PCB yang telah diimplementasikan dapat berjalan dengan baik adalah dengan Pengujian deteksi pada circuit PCB. Pengujian dilakukan secara langsung dengan pengambilan gambar circuit PCB menggunakan kamera dan mendeteksi objek secara real-time.

TABEL 3
Pengujian Gambar PCB diterima

Class	Object	Result
Mouse Bite		

Missing Hole		
Open Circuit		
Short		

Pada Tabel 2 merupakan hasil pengujian pengambilan circuit PCB menggunakan GUI dengan camera. Hasil pengujian gambar telah berhasil terdeteksi secara akurat sesuai dengan klasifikasi kerusakan PCB.

B. Drafting Pengerjaan GUI

Setelah mengetahui ilustrasi Graphical User Interface yang akan digunakan, selanjutnya yaitu melakukan pengerjaan dengan menggunakan Visual Studio Code dan Anaconda3 sebagai base dan environment yang sudah disediakan, karena Anaconda3 bertujuan menyederhanakan manajemen paket yang dikelola.

Dengan bantuan Anaconda3 ini, dapat membantu pengerjaan dalam pembuatan Graphical User Interface nantinya. Melalui ilustrasi GUI tersebut dapat dilihat apa yang diperlukan sehingga akan memudahkan pengguna dalam pengoperasian.

Pada tahap ini, terfokus dalam merancang antarmuka yang dapat memvisualisasikan proses deteksi objek secara efektif. Rancangan ini menggunakan pilihan mode deteksi pada gambar, yang dimana antarmuka akan memudahkan dalam melakukan uji deteksi objeknya.

```
class WebcamApp:
    def __init__(self, window):
        self.window = window
        self.window.title("Webcam App")
        self.video = cv2.VideoCapture(0)

        self.canvas = tk.Canvas(window, width=640, height=480)
        self.canvas.pack()

        self.btn_capture = tk.Button(window, text="Capture", command=self.capture_frame)
        self.btn_capture.pack(pady=10)

        device = 'cuda'
        config = 'D:\VIUGAS\ANIR\pcb-detection\deploy\pcb_config.py'
        checkpoint = 'D:\VIUGAS\ANIR\pcb-detection\deploy\pcb_checkpoint.pth'

        # load the model from a config file and a checkpoint file
        device = torch.device(device)
        self.model = init_detector(config, checkpoint, device=device)

        # init visualizer
        self.visualizer = VISUALIZERS.build(self.model.cfg.visualizer)
        # the dataset meta is loaded from the checkpoint and
        # then pass to the model in init_detector
        self.visualizer.dataset_meta = self.model.dataset_meta

        self.update()
```

GAMBAR 5 Implementasi GUI

Pada Gambar 5, perancangan GUI dimulai pemberian nama class "WebcamApp" dengan menambahkan Canvas menggunakan metode "tk.Canvas" dengan ukuran "width" 640px dan "height" 480px dan meletakkan button Capture menggunakan metode "tk.Button" yang diletakkan di tengah bawah pada Canvas.

Setelah mendefinisikan Canvas dan meletakkan button Capture, langkah selanjutnya berdasarkan Gambar 6, Didalam fungsi Update perlu diterapkan nilai ambang batas "Threshold" dengan nilai yang telah ditetapkan. Hal ini akan menghasilkan sesuai dengan letak dimana area deteksi pada setiap objek yang dideteksi. Objek yang dimaksud merupakan sebuah PCB dengan masing masing kelas jenis kecacatannya.

```
def update(self):
    score_thr = 0.5

    ret_val, frame = self.video.read()
    if ret_val:
        result = inference_detector(self.model, frame)

        frame_rgb = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB) # Convert BGR to
        # frame_rgb = mcv.inconvert(frame_rgb, 'rgb', 'rgb')
        self.visualizer.add_datasample(
            name='result',
            image=frame_rgb,
            data_sample=result,
            draw_gt=False,
            pred_score_thr=score_thr,
            show=False)

        frame_rgb = self.visualizer.get_image()
        # frame_rgb = mcv.inconvert(frame_rgb, 'rgb', 'rgb')
        self.photo = ImageTk.PhotoImage(image=Image.fromarray(frame_rgb))
        self.canvas.create_image(0, 0, image=self.photo, anchor=tk.NW)

    self.window.after(15, self.update)
```

GAMBAR 6 Fungsi Update frame

```
def capture_frame(self):
    path = 'D:\VIUGAS\ANIR\pcb-detection\deploy\captures'
    score_thr = 0.5
    ret_val, frame = self.video.read()
    if ret_val:
        result = inference_detector(self.model, frame)

        frame_rgb = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB) # convert BGR to RGB
        self.visualizer.add_datasample(
            name='result',
            image=frame_rgb,
            data_sample=result,
            draw_gt=False,
            pred_score_thr=score_thr,
            show=False)

        processed_frame_rgb = self.visualizer.get_image() # Processed frame from the visualizer
        processed_frame_rgb = cv2.cvtColor(processed_frame_rgb, cv2.COLOR_BGR2RGB) # convert rgb to BGR
        current_datetime = datetime.now() # get the current date and time
        date_time_str = current_datetime.strftime("%Y-%m-%d_%H-%M-%S")
        output_filename = f"{current_datetime.strftime('%Y-%m-%d_%H-%M-%S')}.jpg" # create the output filename with the current date and
        cv2.imwrite(os.path.join(path, output_filename), processed_frame_rgb)
        print("Frame captured")
```

GAMBAR 7 Fungsi Capture frame

Pada Gambar 7, Setelah kamera sudah dapat menampilkan gambar yang terdeteksi dengan memberikan area berbentuk kotak pada setiap jenis kecacatan PCB. Sistem akan menyimpan output gambar yang telah terdeteksi kedalam directory lokal komputer yang telah dibuat sebelumnya dengan metode "cv2.imwrite".

C. Analisis Hasil Pengujian

Analisis hasil pengujian dilakukan untuk mengetahui keakuratan deteksi objek pada Deteksi Obyek Berbasis Computer Vision Dan Deep Learning untuk Sistem Inspeksi Cacat Pada Permukaan Printed Circuit Board (PCB) berjalan sesuai dengan harapan. Dapat ditunjukkan pada hasil deteksi objek GUI berhasil memenuhi tujuan dengan baik, yaitu dapat mendeteksi objek secara akurat. Hasil yang sesuai juga ditunjukkan dengan melalui pengujian yang matang dengan GUI yang dapat langsung digunakan.

Pengujian GUI telah berjalan sesuai dengan harapan. Pengujian dilakukan dengan pengambilan gambar circuit PCB yang dicetak pada kertas, kemudian pengambilan gambar oleh kamera eksternal. Sistem akan memproses untuk memprediksi apakah mendeteksi objek atau tidak. Hal ini telah menunjukkan bahwa sistem telah berjalan dengan baik karena berhasil mendeteksi PCB dengan akurat.

V. KESIMPULAN

Tugas Akhir ini mengusulkan pengembangan Graphical User Interface (GUI) deteksi objek untuk mempermudah inspeksi cacat pada permukaan PCB. Penelitian ini berhasil merancang dan mengimplementasikan sebuah GUI yang intuitif dan mudah digunakan. GUI mampu menyediakan berbagai fitur yang memungkinkan operator untuk menginspeksi serta mengidentifikasi cacat pada PCB dengan cepat dan akurat. Pengujian GUI dilakukan menggunakan sejumlah sampel PCB dengan cacat yang diketahui sebelumnya. Hasil pengujian menunjukkan bahwa GUI ini efektif dalam mengidentifikasi cacat pada PCB dengan tingkat keberhasilan tinggi. Pengembangan GUI ini dapat dijadikan landasan untuk pengembangan sistem inspeksi cacat pada PCB yang lebih canggih dan otomatis di masa depan.

REFERENSI

- [1] K. Shringarpure *et al.*, "Sensitivity Analysis of a Circuit Model for Power Distribution Network in a Multilayered Printed Circuit Board," *IEEE Trans Electromagn Compat*, vol. 59, no. 6, pp. 1993–2001, 2017.
- [2] V. T. Nguyen and H. A. Bui, "A Real-time Defect Detection in Printed Circuit Boards Applying Deep Learning," *EUREKA, Physics and Engineering*, vol. 2022, no. 2, pp. 143–153, Mar. 2022.
- [3] J. Shen, N. Liu, and H. Sun, "Defect detection of printed circuit board based on lightweight deep convolution network," *IET Image Processing*, vol. 14, no. 15, pp. 3932–3940, 2020.
- [4] Runwei Ding, Linhui Dai, Guangpeng Li, and Hong Liu, "TDD-net: a tiny defect detection network for printed circuit boards," *CAAI Trans Intell Technol*, vol. 4, no. 2, pp. 110–116, 2019.
- [5] F. Akhyar *et al.*, "Observation of Attention Mechanism Baseline for PCB Surface Inspection System," in *2022 IEEE Asia Pacific Conference on Wireless and Mobile (APWiMob)*, 2022, pp. 1–6.
- [6] N. Kurniasari and J. P. Sugiono, "Deteksi Jalur Yang Terputus Pada Rangkaian Listrik Dalam Pcb Menggunakan Metode Convolutional Neural Network (Cnn)," *Jurnal Sistem Cerdas Dan Rekayasa (Jscr)*, vol. 3, pp. 2656–7504, 2021.
- [7] N. Huda, "Kenalan dengan Istilah GUI yang Memudahkan Para Developer." <https://www.dewaweb.com/blog/apa-itu-gui/> (accessed Jul. 21, 2023).
- [8] srimulia, "Mengenal OpenCV Dalam Python: Pengertian, Sejarah, Dukungan pada OS, Fitur-fitur." <https://idmetafora.com/news/read/1177/Mengenal-OpenCV-Dalam-Python-Pengertian-Sejarah-Dukungan-pada-OS-Fitur-fitur.html> (accessed Jul. 16, 2023).
- [9] R. Mulyawan, "PyTorch." <https://rifqimulyawan.com/kamus/pytorch/> (accessed Jul. 16, 2023).
- [10] N. D. Miranda, L. Novamizanti, and S. Rizal, "Convolutional Neural Network Pada Klasifikasi Sidik Jari Menggunakan Resnet-50," *J. Tek. Inform.*, vol. 1, no. 2, pp. 61–68, 2020.
- [11] P. D. Wananda, L. Novamizanti, R. D. Atmaja, Sistem Deteksi Cacat Kayu dengan Metode Deteksi Tepi SUSAN dan Ekstraksi Ciri Statistik, *ELKOMIKA: Jurnal Teknik Energi Elektrik, Teknik Telekomunikasi & Elektronika*, Vol 6, No 1, 2018.
- [12] H. M. Lathifah, L. Novamizanti, and S. Rizal, "Fast and Accurate Fish Classification from Underwater Video using You only Look Once," in *IOP Conference Series: Materials Science and Engineering*, IOP Publishing Ltd, Dec. 2020.
- [13] F. Akhyar, L. Novamizanti, T. Putra, E. N. Furqon, M. C. Chang & C. Y. Lin, Lightning YOLOv4 for a surface defect detection system for sawn lumber, In *2022 IEEE 5th International Conference on Multimedia Information Processing and Retrieval (MIPR)*, pp. 184–189, 2022.