

DETECTION OF VEHICLE TYPE AND LICENSE PLATE WITH CONVOLUTIONAL NEURAL NETWORK MODEL YOLOV7

Suhartono¹, Satria Gunawan Zain², Andi Ardilla³

^{1,2,3}Informatics and Computer Engineering, Faculty of Engineering, Universitas Negeri Makassar, Indonesia
Email: ¹suhartono@unm.ac.id, ²satria.gunawan.zain@unm.ac.id, ³andiardilla1@gmail.com

(Article received: September 09, 2024; Revision: October 17, 2024; published: April 22, 2024)

Abstract

This research was conducted in response to issues related to the efficiency and effectiveness of vehicle type and license plate detection. The increasingly congested traffic conditions and the expanding use of motor vehicles have posed challenges in traffic monitoring and regulation. Therefore, there is a need to develop a solution that can save time and resources while providing more comprehensive information in vehicle monitoring. This research implements the Convolutional Neural Network (CNN) algorithm with the latest YOLOv7 model from YOLO to detect vehicle types and vehicle number plates simultaneously to make it more efficient and effective, save time and resources, and provide more complete information. The research method used is Research and Development (R&D) with an experimental approach. The stages include image acquisition, labeling, dataset sharing, YOLOv7 model training, testing, prediction results, and conversion to text using Optical Character Recognition (OCR). The research results show that the ResNet34 model architecture achieves a total accuracy of 89.7% for 3x3 convolution layers and 88.6% for 5x5 convolution layers. The YOLOv5 architecture performs well on 3x3 convolution layers with an overall accuracy of 71.9%, and 58.3% for 5x5 convolution layers. However, the YOLOv7 and Mobilenet architectures tend to have lower accuracy, namely the Mobilenet architecture with a 3x3 convolution layer with a total accuracy of 63.4%, and 73.4% for the 5x5 convolution layer. Computing speed is also considered, with YOLOv5 and YOLOv7 having higher speeds than ResNet34 and Mobilenet. Tests were carried out in various lighting conditions, resulting in accurate detection of vehicle types and vehicle number plates of 90% in the morning, 85% in the afternoon and 77% at night. Overall, the system succeeded in recognizing objects with an accuracy of 84% from a total of 720 data tested, but the accuracy of converting vehicle number plates using OCR reached 22%. The results of this research demonstrate the performance and effectiveness of the YOLOv7 algorithm in detecting vehicle types and vehicle number plates, as well as providing insight into accuracy in various lighting conditions and OCR conversion.

Keywords: Convolutional Neural Network, Optical Character Recognition, Testing Accuracy, Vehicle Detection, YOLOv7.

DETEKSI JENIS KENDARAAN DAN PLAT NOMOR DENGAN CONVOLUTIONAL NEURAL NETWORK MODEL YOLOV7

Abstrak

Penelitian ini dilakukan sebagai respons terhadap permasalahan yang berkaitan dengan efisiensi dan efektivitas dalam deteksi jenis kendaraan dan pelat nomor kendaraan. Kondisi lalu lintas yang semakin padat dan perluasan pemakaian kendaraan bermotor memunculkan tantangan dalam pengawasan dan pengaturan lalu lintas. Oleh karena itu, perlu dikembangkan solusi yang mampu menghemat waktu dan sumber daya, serta memberikan informasi yang lebih lengkap dalam pemantauan kendaraan. Penelitian ini mengimplementasikan algoritma Convolutional Neural Network (CNN) dengan model YOLOv7 keluaran terbaru dari YOLO untuk mendeteksi jenis kendaraan dan pelat nomor kendaraan secara bersamaan agar lebih efisien dan efektif, dapat menghemat waktu dan sumber daya, serta memberikan informasi yang lebih lengkap. Metode penelitian yang digunakan adalah Research and Development (R&D) dengan pendekatan eksperimental. Tahapannya meliputi akuisisi citra, pelabelan, pembagian dataset, pelatihan model YOLOv7, pengujian, hasil prediksi, dan konversi ke teks menggunakan Optical Character Recognition (OCR). Hasil penelitian menunjukkan bahwa arsitektur model ResNet34 mencapai akurasi total keseluruhan 89,7% untuk layer konvolusi 3x3 dan 88,6% untuk layer konvolusi 5x5. Arsitektur YOLOv5 memiliki performa baik pada layer konvolusi 3x3 dengan akurasi total keseluruhan 71,9%, dan 58,3% untuk layer konvolusi 5x5. Namun, arsitektur YOLOv7 dan Mobilenet cenderung memiliki akurasi lebih rendah, yaitu pada arsitektur Mobilenet layer konvolusi 3x3 dengan akurasi total keseluruhan 63,4%,

dan 73,4% untuk layer konvolusi 5x5. Kecepatan komputasi juga diperhatikan, dengan YOLOv5 dan YOLOv7 memiliki kecepatan lebih tinggi dibanding ResNet34 dan Mobilenet. Pengujian dilakukan pada berbagai kondisi pencahayaan, menghasilkan akurasi deteksi jenis kendaraan dan pelat nomor kendaraan sebesar 90% pada pagi hari, 85% pada siang hari, dan 77% pada malam hari. Secara keseluruhan, sistem berhasil mengenali objek dengan akurasi 84% dari total 720 data yang diuji namun, akurasi konversi plat nomor kendaraan menggunakan OCR mencapai 22%. Hasil penelitian ini menunjukkan performa dan efektivitas algoritma YOLOv7 dalam mendeteksi jenis kendaraan dan pelat nomor kendaraan, serta memberikan wawasan tentang akurasi pada berbagai kondisi pencahayaan dan konversi OCR.

Kata kunci: Akurasi Pegujian, Convolutional Neural Network, Deteksi Kendaraan, Optical Character Recognition, YOLOv7.

1. PENDAHULUAN

Kendaraan memiliki peran penting dalam mobilitas manusia dan barang-barang, khususnya di era globalisasi. Di Indonesia, pertumbuhan jumlah kendaraan mengalami peningkatan signifikan, terutama kendaraan pribadi seperti mobil dan sepeda motor [1]. Data dari Badan Pusat Statistik (BPS) mencatat pertumbuhan rata-rata tahunan sebesar 7,6% dari tahun 2018 hingga 2020. Peningkatan jumlah kendaraan ini berdampak pada kemacetan lalu lintas dan pelanggaran aturan yang meningkatkan risiko kecelakaan [2]. Beberapa perilaku berkendara yang tidak patuh terhadap aturan seperti penggunaan helm, melanggar rambu-rambu, dan parkir sembarangan menjadi penyebab utama kecelakaan [2][3].

Pemerintah Indonesia berusaha mengatasi masalah ini melalui teknologi tilang elektronik (E-tilang) berdasarkan Peraturan Mahkamah Agung No. 12 Tahun 2016 [4]. Teknologi ini memanfaatkan CCTV dan foto di lapangan untuk meningkatkan efisiensi penegakan hukum lalu lintas [5]. Kemajuan dalam teknologi kecerdasan buatan, terutama dalam pendeteksian objek, juga berperan penting dalam penanganan masalah lalu lintas [6][7]. Metode seperti *Convolutional Neural Network* (CNN) dan model *You Only Look Once* (YOLO) telah terbukti efektif dalam pendeteksian objek secara *real-time* [8].

Beberapa penelitian sebelumnya telah berhasil mengembangkan sistem deteksi objek terkait kendaraan. Contohnya, penelitian berjudul "Implementasi Deteksi *Real Time* Klasifikasi Jenis Kendaraan Di Indonesia Menggunakan Metode YOLOv5" mencapai tingkat akurasi sebesar 90% dalam mengenali jenis kendaraan secara *real-time* [9]. Penelitian lain yang berjudul "Implementasi Algoritma YOLO dan *Tesseract* OCR (*Optical Character Recognition*) Pada Sistem Deteksi Plat Nomor Otomatis" menggunakan Model YOLOv3 dan mencapai akurasi 100% dalam mendeteksi pelat nomor otomatis. Hasil OCR menggunakan *library Tesseract* juga mencapai tingkat akurasi sebesar 92,32% dalam mengenali seluruh karakter pelat mobil dan motor [10]. Selanjutnya penelitian dari jurnal internasional dengan judul "Implementation of number plate detection system for vehicle registration using IOT and recognition using CNN" dengan hasil

akurasi pengenalan karakter 98,5% [11]. Selain itu, penelitian Suhartono dengan judul "Sistem *Object Recognition* Plat Nomor Kendaraan Untuk Sistem Parkir Bandara" menggunakan algoritma CNN model YOLOv5, berhasil mendeteksi plat nomor kendaraan dan mengkonversinya ke teks dalam bentuk excel. Percobaan pertama mencapai akurasi sebesar 83% dalam deteksi plat dan 97,89% dalam hasil konversi teks. Pada percobaan kedua, akurasi deteksi plat meningkat menjadi 98%, dan akurasi konversi teks mencapai 99,49% [12]. Kemudian *peer-reviewed journal* dengan judul "Automatic number plate recognition (ANPR) in smart cities: A systematic review on technological advancements and application cases" dengan akurasi real-time dari OCR pada level 97,3% [13]. Berikutnya penelitian dengan judul "Towards Automatic License Plate Recognition in Challenging Conditions" dengan akurasi *recognition* sebanyak 79,28% [14]. Terakhir penelitian dengan judul "License plate detection using YOLO v4" capaian tingkat akurasi deteksi sebanyak 89% [15].

Dengan demikian, penelitian-penelitian ini menunjukkan kemajuan yang signifikan dalam pengembangan sistem deteksi kendaraan, yang dapat digunakan dalam berbagai konteks seperti pengawasan lalu lintas dan manajemen parkir. Namun masih ada ruang untuk meningkatkan akurasi dan kecepatan dalam mendeteksi jenis kendaraan dan pelat nomor secara bersamaan. Oleh karena itu, penelitian ini bertujuan mengimplementasikan algoritma CNN dengan model YOLOv7 untuk mendeteksi jenis kendaraan dan pelat nomor dengan akurasi tinggi. Berdasarkan pemahaman tersebut, penelitian ini akan menjawab dua pertanyaan utama: (1) Bagaimana implementasi algoritma model YOLOv7 dalam mendeteksi jenis kendaraan dan pelat nomornya? (2) Bagaimana hasil pengujian algoritma model YOLOv7 untuk pengambilan citra jenis kendaraan dan pelat nomor kendaraan?.

Diharapkan hasil penelitian ini akan membawa kontribusi penting dalam peningkatan pengawasan lalu lintas dengan memanfaatkan kecerdasan buatan. Dengan kemampuan yang lebih baik dalam mengklasifikasikan jenis kendaraan dan membaca pelat nomor, diharapkan akan mampu mengurangi

pelanggaran tata tertib berkendara dan meningkatkan keamanan jalan raya.

2. METODE PENELITIAN

Penelitian ini menggunakan metode penelitian eksperimen untuk menginvestigasi pengaruh suatu variabel atau lebih terhadap variabel lain. Metode ini bersifat *validation* atau menguji, yaitu menguji pengaruh satu variabel atau lebih terhadap variabel lainnya. Desain penelitian eksperimental adalah metode yang digunakan untuk mengidentifikasi apakah adanya hubungan kausal antara variabel bebas (perlakuan) dan variabel terikat (hasil atau perubahan) [16]. Namun, dalam kasus ini, penelitian menggunakan desain pre-eksperimental, yang merupakan variasi sederhana dari desain eksperimental. Dalam desain penelitian eksperimental, ada dua kelompok utama variabel, yaitu:

1. Variabel Bebas (Perlakuan)

Ini adalah variabel yang dimanipulasi oleh peneliti untuk melihat apakah perubahan dalam variabel ini akan memiliki dampak pada variabel terikat. Dalam konteks ini, variabel bebas dapat merujuk pada metode atau teknik yang digunakan, seperti penerapan algoritma pengenalan pelat nomor kendaraan [17].

2. Variabel Terikat (Hasil)

Ini adalah variabel yang diamati atau diukur untuk melihat apakah ada perubahan sebagai akibat dari penerapan variabel bebas. Dalam penelitian ini, variabel terikat adalah hasil dari deteksi dan pengenalan pelat nomor kendaraan menggunakan metode yang telah diterapkan [18].

Desain penelitian pre-eksperimental ini memiliki beberapa karakteristik:

1. One-Shot Case Study Research Design

Dalam desain ini, hanya ada satu kelompok variabel terikat yang diamati setelah diberikan perlakuan. Ini berarti tidak ada kelompok kontrol atau perbandingan yang digunakan. Peneliti mengamati hasil setelah penerapan variabel bebas tanpa membandingkannya dengan kelompok lain.

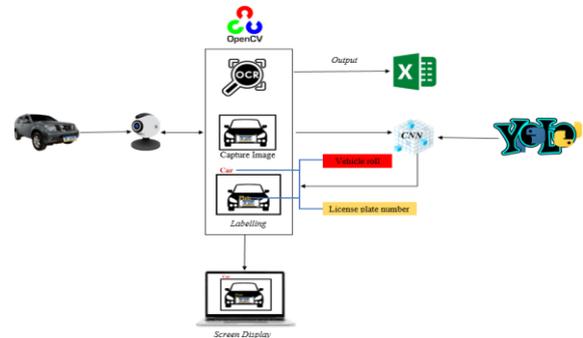
2. Posttest Only

Dalam desain ini, penelitian dilakukan setelah penerapan variabel bebas (perlakuan) kepada subjek atau objek yang diamati. Ini berarti bahwa pengukuran variabel terikat dilakukan setelah perlakuan dilakukan, dan tidak ada pengukuran sebelumnya untuk dibandingkan.

Desain pre-eksperimental seperti ini biasanya digunakan untuk mengamati dampak perubahan yang mungkin diakibatkan oleh suatu perlakuan, namun tidak memberikan dasar yang kuat untuk mengklaim adanya hubungan kausal.

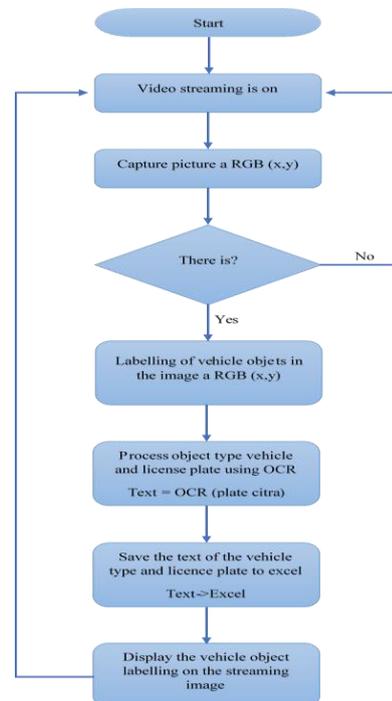
Dalam konteks ini, penelitian menggunakan desain pre-eksperimental untuk mengamati dampak penerapan algoritma deteksi dan pengenalan pelat nomor kendaraan terhadap hasil deteksi dan pengenalan. Meskipun desain pre-eksperimental

lebih sederhana daripada desain eksperimental yang lebih kompleks, ia tetap dapat memberikan wawasan awal tentang potensi efektivitas metode yang digunakan dalam penelitian ini.



Gambar 1. Arsitektur Sistem

Gambar 1 merupakan Arsitektur Sistem yang dimana OpenCV bekerja dalam melakukan *capturing image*, hasil *capture* kemudian menjadi masukan terhadap algoritma CNN model YOLO. Keluaran dari CNN menghasilkan data berupa gambar yang selanjutnya diproses dengan pengenalan karakter gambar hasil *capture* tadi yakni objek kendaraan dan pelat nomor. Objek tersebut di proses menggunakan OCR untuk mengkonversi citra gambar ke *text*. Proses yang dilakukan sistem secara keseluruhan dapat digambarkan dalam bentuk *flowchart* seperti pada Gambar 2.



Gambar 2. Flowchart Proses Kerja Sistem

Gambar 2 menggambarkan suatu alur kerja yang dimulai dengan pengawasan video (video streaming on). Dalam alur ini, langkah pertama adalah menangkap gambar dalam format RGB pada koordinat (x, y) yang ditentukan. Setelah gambar ditangkap, tahap berikutnya adalah memeriksa

apakah gambar tersebut mengandung objek kendaraan atau tidak (*There is?*). Jika hasil pemeriksaan menunjukkan bahwa tidak ada objek kendaraan yang terdeteksi, alur kembali ke tahap "Video streaming is on" untuk melanjutkan pemantauan. Namun, jika terdapat objek kendaraan dalam gambar, maka proses akan melanjutkan ke langkah berikutnya.

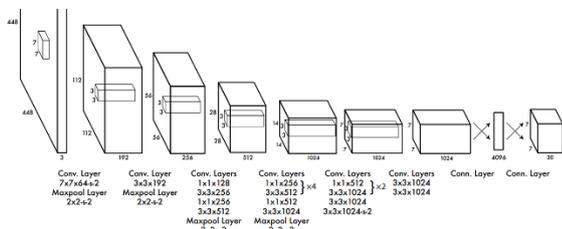
Pada langkah selanjutnya, terjadi pelabelan objek kendaraan dalam gambar dengan jenis kendaraan yang sesuai. Selain itu, gambar plat nomor kendaraan juga diproses menggunakan teknologi *Optical Character Recognition* (OCR) untuk mengekstrak teks dari plat nomor. Setelah informasi kendaraan dan plat nomor berhasil diekstrak, data tersebut kemudian disimpan atau ditampilkan dalam format *Excel*.

Terakhir, hasil pelabelan objek kendaraan ditampilkan kembali pada gambar streaming yang sedang dipantau. Dengan demikian, alur kerja ini mencakup pemantauan video yang berkelanjutan, pengenalan objek kendaraan, ekstraksi teks dari plat nomor, penyimpanan data, dan penampilan data kembali pada gambar streaming.

3. Arsitektur YOLOV7

Fungsi aktivasi memberikan sifat nonlinear pada model, dan karakteristiknya memiliki dampak langsung terhadap kinerja jaringan saraf. Dalam YOLOv7, fungsi aktivasi awalnya adalah SiLU, yang merupakan gabungan linier berbobot dari fungsi Sigmoid, yang telah terbukti lebih efektif dalam konteks model yang dibahas [19].

Penelitian ini bertujuan untuk mengkaji algoritma deteksi masker wajah. Proses pendeteksian objek dalam penelitian ini menggunakan YOLOv7, yang dipilih karena keberhasilannya sebagai proyek *open source* yang dikembangkan oleh *University of Washington* untuk memprediksi bounding boxes dan mengklasifikasikan objek pada citra. Model YOLO diilhami oleh GoogleNet untuk tugas klasifikasi citra dan menggabungkan 24 lapisan konvolusi yang diikuti oleh 2 lapisan yang terkoneksi. Berikut ini dapat terlihat arsitektur YOLO dalam Gambar 2.



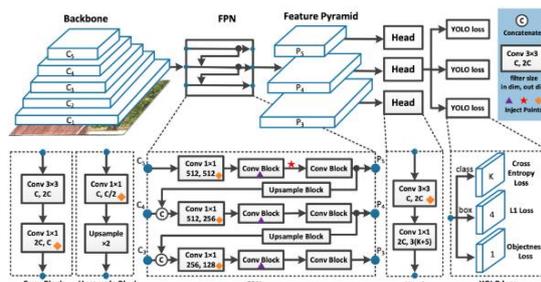
Gambar 3. Arsitektur Metode YOLO

Gambar 3 merupakan arsitektur dari algoritma YOLO menggunakan *Convolutional Neural Network* yang memiliki 24 *convolutional layers* dan diikuti oleh 2 *fully connected layers*. *Convolutional layer* berfungsi untuk mengekstraksi fitur dari input gambar, sedangkan *fully connected layer* berperan dalam memprediksi probabilitas output dan

koordinat. YOLO algoritma telah diklaim mampu beroperasi pada tingkat 45 frame per detik (fps) saat menggunakan kartu grafis Titan X. Oleh karena itu, metode YOLO sangat sesuai untuk aplikasi real-time, seperti analisis video. Seiring dengan permintaan akan algoritma deteksi objek yang lebih baik, YOLO telah berkembang menjadi versi-versi yang lebih baru, seperti YOLO v2 dan yang terbaru, YOLO v3. Perbedaan antara setiap versi YOLO terletak pada peningkatan akurasi dan kecepatan deteksi [10]. Namun, diperlukan sumber daya komputasi yang tinggi dalam metode Deep Learning, yang mengharuskan penggunaan perangkat keras berkinerja tinggi. Oleh karena itu, dalam penelitian ini, akan digunakan *Google Colab* sebagai alternatif untuk mengatasi kendala tersebut.

Google Colab merupakan sebuah lingkungan pengembangan terpadu (IDE) untuk bahasa pemrograman Python, di mana komputasi dan pemrosesan dilakukan oleh server milik Google yang menggunakan perangkat keras berkinerja tinggi [20]. Dari segi perangkat lunak, *Google Colab* telah menyediakan sebagian besar pustaka yang diperlukan. Untuk penelitian ini, kami membutuhkan pustaka seperti Keras, TensorFlow, NumPy, Pandas, serta berbagai pendukung lainnya, termasuk Modul Matplotlib untuk visualisasi data. Selain itu, *Google Colab* juga menyediakan berbagai versi, termasuk TensorFlow versi 1.x dan 2.x, serta berbagai versi Python dari 2.x hingga 3.x. Dari segi perangkat keras, *Google Colab* menyediakan fasilitas berupa media penyimpanan terintegrasi dengan *Google Drive*, prosesor yang mencakup CPU, GPU, dan TPU, serta alokasi RAM yang cukup. Dengan keandalan servernya, hampir seluruh proses pemrosesan berjalan dengan lancar menggunakan *Google Colab* selama koneksi internet tetap stabil [20].

YOLO (*You Only Look Once*) v7 adalah model terbaru dalam keluarga YOLO. Model YOLO adalah detektor objek satu tahap yang mengambil bingkai gambar, mengolahnya melalui tulang punggung, menggabungkan fitur-fitur ini di leher, dan kemudian mengirimkannya ke kepala jaringan. YOLO bertugas memprediksi lokasi dan kelas objek di mana kotak pembatas harus digambar. Setelah itu, YOLO melakukan post-processing menggunakan teknik non-maximum suppression (NMS) untuk menghasilkan prediksi akhirnya.



Gambar 4. Struktur jaringan YOLO yang dijelaskan dalam PP-YOLO

Gambar 4 terdapat berbagai komponen kunci dalam arsitektur jaringan YOLO. Pertama, terdapat "Backbone C1-C5," yang kemungkinan besar merujuk kepada lapisan-lapisan konvolusi dalam tahap ekstraksi fitur yang merupakan bagian terdalam dari jaringan. Selanjutnya, terdapat "FPN" (*Feature Pyramid Network*) yang digunakan untuk membangun piramida fitur dari lapisan-lapisan sebelumnya (C1-C5) untuk mendapatkan representasi fitur dalam berbagai skala. Selain itu, ada "Feature Pyramid" yang menghasilkan P3, P4, dan P5, yang mewakili piramida fitur. Lalu, terdapat "conv 3x3" dan "conv 1x1" dengan berbagai keluaran (256, 128, 512, 256, 512, 512), yang merujuk kepada lapisan-lapisan konvolusi dengan filter 3x3 dan 1x1 yang digunakan untuk transformasi fitur. "Conv block" mungkin merujuk kepada blok konvolusi yang kompleks. Terakhir, ada "conv 3x3 C, 2C," yang merupakan lapisan konvolusi dengan filter 3x3 dan keluaran ganda dari jumlah saluran (C) yang digunakan untuk pengolahan fitur lebih lanjut. Semua elemen ini bekerja bersama-sama dalam arsitektur YOLO untuk tugas deteksi objek yang akurat dan efisien.

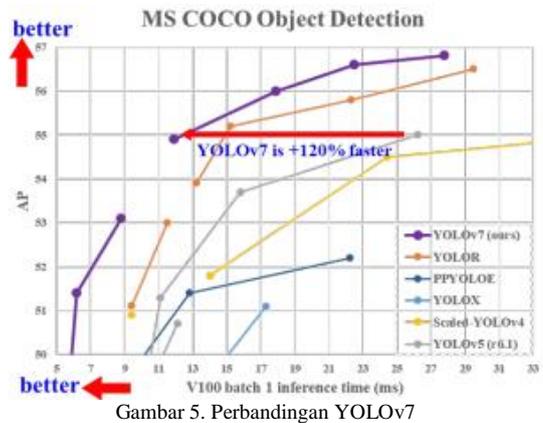
YOLO v7 yang dikembangkan oleh WongKinYiu dan Alexey Bochkovskiy (AlexeyAB) merupakan kelanjutan dari evolusi YOLO yang dimulai oleh Joseph Redmon. AlexeyAB mengambil alih pengembangan YOLO setelah Joseph Redmon meninggalkan industri Computer Vision (CV) karena pertimbangan etika. AlexeyAB awalnya mempertahankan versi fork YOLOv3 sebelum akhirnya merilis YOLOv4, yang merupakan peningkatan dari model sebelumnya.

WongKinYiu memulai perjalanannya dalam Computer Vision (CV) dengan mengembangkan jaringan Cross Stage Partial. Hal ini membantu perbaikan efisiensi dalam membangun tulang punggung YOLOv4 dan YOLOv5. Kemudian, WongKinYiu terus berkontribusi pada keluarga penelitian YOLO dengan pengembangan Scaled-YOLOv4. Model ini memperkenalkan pendekatan penskalaan yang efisien pada CSP YOLOv4 untuk meningkatkan kinerja deteksi objek di peta COCO yang canggih.

Para penulis YOLOv7 berusaha untuk mencapai tingkat ketelitian yang lebih tinggi dalam deteksi objek dengan merancang arsitektur jaringan yang mampu memprediksi kotak pembatas dengan lebih akurat dibandingkan dengan model-model sebelumnya pada tingkat kecepatan inferensi yang setara.

Gambar 5 memberikan perbandingan arsitektur model YOLOv5, YOLOX, dan YOLOv7. Dalam perbandingan tersebut, YOLOv7 dinilai sebagai yang paling unggul di sudut kiri atas karena kinerjanya yang lebih cepat dan akurat daripada kedua jaringan rekan-rekannya. Hasil evaluasi menunjukkan bahwa YOLOv7 memiliki peningkatan dalam Average Precision (AP) minimum hingga nilai tertentu, yang

berkisar antara 25,00% hingga 60,00%. Ini menandakan bahwa YOLOv7 memiliki kemampuan deteksi objek yang lebih baik dengan tingkat akurasi yang lebih tinggi dibandingkan dengan YOLOv5 dan YOLOX, khususnya dalam konteks AP minimum yang telah disebutkan.



Gambar 5. Perbandingan YOLOv7

3. HASIL DAN PEMBAHASAN

Hasil penelitian ini berhasil mengimplementasikan algoritma CNN model YOLOv7 untuk melakukan deteksi jenis kendaraan dan plat nomor kendaraan pada citra atau gambar, berikut adalah hasil yang dicapai.

3.1. Hasil Implementasi Deteksi Jenis Kendaraan dan Plat Nomor Kendaraan

3.1.1. Deteksi Jenis Kendaraan

Algoritma CNN dengan model YOLOv7 mampu mendeteksi jenis kendaraan dengan akurasi sebesar 84%, kecepatan pemrosesan citra yang berhasil dicapai adalah 18 *frame* per detik, serta jenis kendaraan yang berhasil dideteksi antara lain bus, mobil, motor dan truk.

3.1.2. Deteksi Plat Nomor Kendaraan

Algoritma CNN dengan model YOLOv7 berhasil mendeteksi plat nomor kendaraan dengan akurasi sebesar 84,1%. Faktor-faktor seperti posisi dan sudut pandang kendaraan pada citra, kondisi cahaya serta ukuran dan jenis font pada plat nomor kendaraan mempengaruhi akurasi deteksi.

3.2. Hasil pengumpulan Kebutuhan

3.2.1. Kebutuhan Perangkat Keras (*Hardware*)

Kebutuhan perangkat keras yang digunakan dalam pembuatan sistem ini meliputi, laptop AMD Ryzen7 dengan kemampuan komputasi tinggi (2.9 GHz hingga 4.2 GHz), RAM 8GB dan storage 512 GB SSD. Dan webcam xiaomi 1080 sebagai input sensor dengan resolusi 1080p dan fitur *night vision* untuk deteksi pada malam hari.

3.2.2. Kebutuhan Perangkat Lunak (*Software*)

Kebutuhan perangkat lunak yang digunakan dalam pembuatan sistem ini meliputi, roboflow sebagai pengatur dataset gambar dan labeling image untuk pelatihan model. Google colaboratory untuk akses ke sumber daya komputasi GPU dan lingkungan pengembangan python (*Pytorch*) untuk melatih model YOLOv7, anaconda terminal untuk menginstal dan menjalankan program python objek deteksi YOLOv7 dan OpenCV sebagai pustaka python untuk memproses gambar dan video dalam proses deteksi objek.

3.3. Hasil Rancangan Sistem Deteksi Jenis Dan Plat Nomor Kendaraan

3.3.1. Membangun *Prototyping*

Setelah memenuhi kebutuhan, langkah berikutnya adalah membangun prototipe GUI dengan kemampuan mendeteksi jenis kendaraan dan plat nomornya melalui OCR. Perangkat keras terdiri dari laptop sebagai monitor dan webcam di depan kendaraan. Citra diambil dari webcam, diolah oleh sistem, dan hasilnya ditampilkan di GUI laptop. Evaluasi prototipe dilakukan untuk mengukur efektivitasnya dan menentukan perlu tidaknya perubahan sebelum mengembangkan versi final. Evaluasi melibatkan pengujian fungsionalitas untuk memastikan sistem berjalan baik sesuai kebutuhan. Hasil evaluasi digunakan untuk memperbaiki prototipe sebelum menciptakan versi final sistem.

3.3.2. Pengkodean Sistem

Sistem ini dibangun dengan menggunakan bahasa pemrograman Python dan lingkungan pengembangan *Visual Studio Code*. Python dipilih karena kemudahan pemahaman dan ketersediaan pustaka untuk pengembangan. *Visual Studio Code* dipilih karena dukungan yang baik untuk Python. Pada tahap pengambilan citra, pustaka OpenCV digunakan untuk membuka dan mengambil gambar dari webcam. Citra yang diambil kemudian diproses menggunakan model deteksi objek YOLOv7 yang telah dilatih sebelumnya. Model ini memungkinkan deteksi kendaraan. Hasilnya adalah sistem yang dapat mengambil gambar melalui webcam, mendeteksi kendaraan menggunakan model YOLOv7, dan menampilkan hasilnya.

3.3.3. Pengumpulan Dataset

Pada tahap ini terkumpul 7400 dataset berupa jenis kendaraan yakni mobil, motor, bus dan truk beserta plat nomor yang dikumpulkan dari periode bulan Januari hingga Februari 2023. Citra yang terkumpul dibagi dalam 3 folder yaitu data *training* set 70%, data *validation* set 20% dan data *testing* set 10% yang dilakukan pelabelan menggunakan roboflow.

3.3.4. *Labelling Image*

Pembuatan label pada objek gambar, dikenal sebagai *labelling image*, adalah tahap krusial dalam mengembangkan model komputer untuk pengenalan objek. Ini melibatkan penggunaan perangkat lunak pelabelan seperti roboflow dan YOLOv7 untuk menandai objek dalam gambar dengan kotak pembatas dan memberikan nama objek. Terdapat beberapa format anotasi gambar seperti Pascal, *CreateML*, dan YOLO. Pemilihan format anotasi penting karena memengaruhi kualitas model. Dalam penelitian ini, dipilih format YOLO sesuai dengan model YOLOv7. YOLO adalah metode deteksi objek menggunakan *deep learning* dan *convolutional neural network* (CNN) tanpa proposal. Dengan ini, model memahami objek dengan lebih akurat.

Hasil pelabelan berupa file txt yang berisi info kotak pembatas. File ini menjadi input untuk pelatihan model atau analisis lebih lanjut.

3.3.5. *Training Dataset*

Setelah melakukan pelabelan gambar, langkah selanjutnya adalah melatih dataset untuk menghasilkan model pengenalan objek. Tujuannya adalah agar sistem dapat mengenali objek yang telah diberi label, seperti motor, bus, mobil, dan truk [21][22]. Proses pelatihan dilakukan menggunakan Google Colaboratory dengan arsitektur model YOLOv7. Langkah pertama adalah menginstal dependensi yang dibutuhkan. Ini melibatkan mengkloning repositori YOLOv7 dari GitHub, pindah ke direktori yang tepat, dan memastikan semua dependensi terinstal dengan menggunakan perintah yang diberikan pada arsitektur YOLOv7. Kemudian, dengan menggunakan *package* Python bernama "roboflow", data gambar yang sudah diolah sebelumnya diakses. Paket ini memungkinkan akses ke dataset atau gambar yang disimpan di roboflow setelah melalui proses pelabelan dan transformasi. Langkah selanjutnya adalah menjalankan program training. Dalam tahap ini, perintah training untuk YOLOv7 dijalankan. Ini melibatkan navigasi ke direktori YOLOv7, dan menggunakan perintah khusus untuk menjalankan skrip pelatihan dengan parameter seperti ukuran batch, jumlah epoch, konfigurasi dataset, bobot awal, dan perangkat GPU yang akan digunakan. Proses pelatihan kemudian dimulai sesuai dengan konfigurasi yang telah ditentukan. Dengan langkah-langkah ini, dataset dilatih menggunakan arsitektur YOLOv7 untuk menghasilkan model yang dapat mengenali objek-objek yang telah diberi label dengan akurasi yang tinggi. Berikut merupakan kode yang digunakan pada tahap training di *Google Colaboratory*.

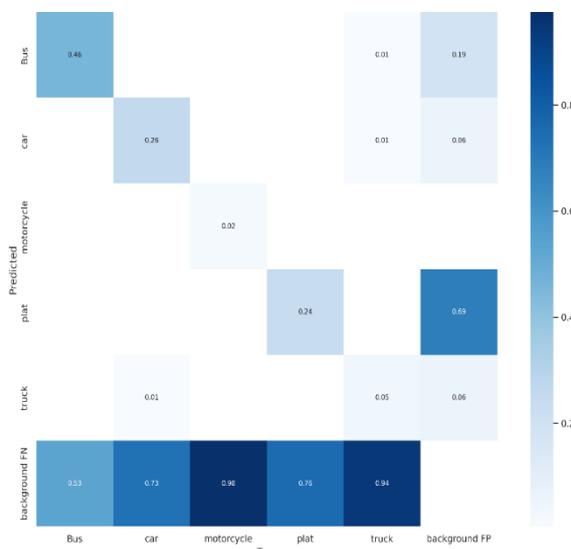
Gambar 6 menyajikan kode perintah yang digunakan untuk memulai pelatihan jaringan YOLOv7 dalam lingkungan Google Colab. Pertama, kode mengarahkan direktori kerja saat ini ke `~/content/yolov7'` dan mengunduh model awal

YOLOv7 yang diperlukan. Kemudian, setelah mengonfigurasi berbagai parameter seperti ukuran batch, jumlah epoch, lokasi data konfigurasi (data.yaml), berat awal model, dan perangkat yang digunakan, pelatihan dimulai dengan menjalankan skrip 'train.py' menggunakan perintah Python. Kode ini mengarahkan proses pelatihan jaringan YOLOv7, yang bergantung pada data yang telah dipersiapkan, dan bertujuan untuk menghasilkan model yang disesuaikan dengan tugas atau proyek yang sedang dikerjakan. Pastikan untuk menggantikan parameter dan konfigurasi sesuai dengan kebutuhan dan tujuan pelatihan.

```
# download COCO starting checkpoint
%cd /content/yolov7
!wget
https://github.com/WongKinYiu/yolov7/releases/download/v0.1/yolov7_training.pt

# run this cell to begin training
%cd /content/yolov7
!python train.py --batch 16 --epochs 16 --data
{dataset.location}/data.yaml --weights 'yolov7_training.pt' --
device 0
```

Gambar 6 Kode training model YOLOV7



Gambar 7. Confusion Matrix YOLOv7 Layer 3x3

Gambar 7 menampilkan hasil *confusion matrix* yakni output dari proses training menggunakan layer 3x3. *Confusion matrix* adalah alat evaluasi penting dalam pengenalan objek seperti YOLOv7. Dalam hal ini, menggunakan matriks 3x3 yang menghitung empat nilai : *True Positive (TP)*, *True Negative (TN)*, *False Positive (FP)*, *False Negative (FN)*.

TP : Jumlah objek yang benar dikenali oleh model.

TN : Jumlah area tanpa objek yang benar dikenali oleh model.

FP : Jumlah area yang salah dikenali sebagai objek model.

FN : Jumlah objek yang seharusnya dikenali tetapi tidak oleh model.

Dari matriks ini, dapat dihitung beberapa metrik evaluasi:

$$Presisi = \frac{TP}{TP+FP} \tag{1}$$

Presisi menunjukkan sejauh mana objek yang dikenali oleh model benar [23].

$$Recall = \frac{TP}{TP+FN} \tag{2}$$

Recall menunjukkan sejauh mana model mendeteksi objek yang sebenarnya ada [23].

$$Akurasi = \frac{TP}{TP+TN+FP+FN} \tag{3}$$

Akurasi menunjukkan sejauh mana model mengenali dan mengklasifikasikan dengan benar. Contoh nilai untuk kategori bus:

Presisi : 0.69

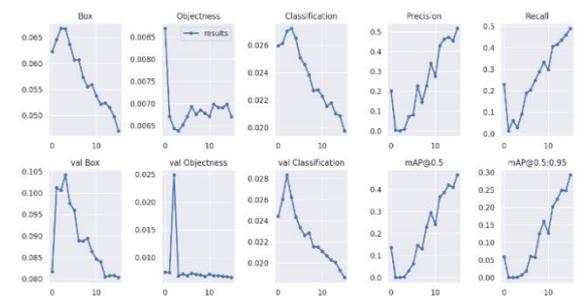
Recall : 0.46

Akurasi : 0.58

Untuk akurasi total keseluruhan, dapat dihitung dengan rumus sebagai berikut.

$$Akurasi\ Total = \frac{TP+TN}{N} \times 100\% \tag{4}$$

Dalam kasus ini didapatkan akurasi total keseluruhan sebesar 17.1%. Dengan memahami nilai-nilai ini, dapat dievaluasi performa model dalam mengenali objek-objek yang diberi label pada gambar atau citra.

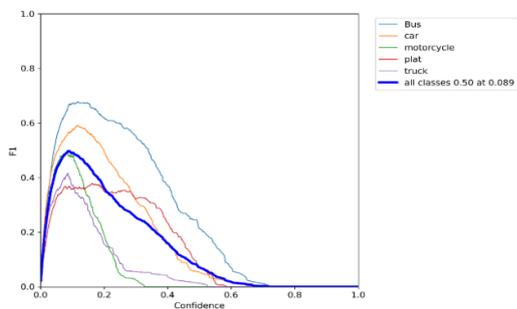


Gambar 8. Hasil Result

Gambar 8 merupakan hasil result training YOLOV7 layer 3x3 yang terdapat beberapa *output* yakni Box dengan nilai maksimal 0,065, *Objectness* dengan nilai maksimal 0,0085, *Classification* dengan nilai maksimal 0,026, *precision* 0,5 *recall* 0,5, val Box dengan nilai 0,105, *val objectness* 0,025, *val Classification* 0,028, *mAP@0.5* dengan nilai 0,4, *mAP@0,5:0,95* dengan nilai 0,30.

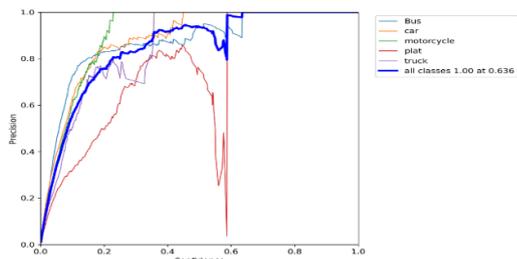
File *result* adalah plot hasil dari training model YOLOv7 yang menunjukkan perubahan performa selama *training*. Pada sumbu x, terdapat jumlah epoch/iterasi *training*, dan pada sumbu y, terdapat metrik evaluasi seperti *Train/Box Loss*, yaitu ukuran

ketepatan model memprediksi *bounding box* pada data pelatihan. *Metrics/Precision*, yaitu tingkat akurasi model dalam mengenali objek yang benar dari data pelatihan. *Metrics/Recall*, jumlah objek yang dikenali oleh model dari total objek yang seharusnya dikenali dari data pelatihan. *Val/Loss*, adalah evaluasi kemampuan model memprediksi objek data validasi yang belum pernah dilihat sebelumnya. *Val/Obj Loss*, ialah akurasi model dalam memprediksi keberadaan objek data validasi. *Val/Cls Loss*, ialah akurasi model dalam memprediksi jenis objek data validasi. *Metrics/mAP_0.5*, rata-rata *average precision* pada IoU threshold 0.95 data validasi, mengukur kualitas pengenalan objek dengan variasi IoU. Grafik ini mempresentasikan perubahan kinerja model YOLOv7 seiring berjalannya *training* berdasarkan metrik-metrik tersebut.



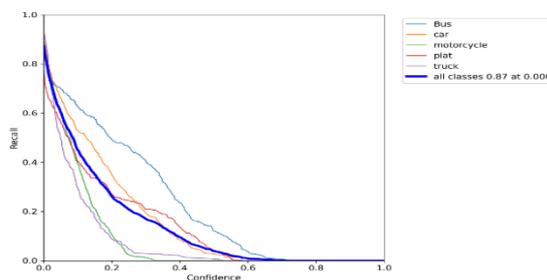
Gambar 9. Grafik F1_curve

Gambar 9 menampilkan output dari hasil training YOLOv7 layer 3x3 dengan nilai kelas Bus diatas 0,6, kelas car diatas 0,4, kelas motorcycle diatas 0,4, kelas plat diatas 0,2, kelas truck diatas 0,2. dan semua kelas dengan rata-rata 0.5. F1_curve dalam YOLOv7 adalah grafik yang mengilustrasikan bagaimana F1-score berubah selama pelatihan model. F1-score mengukur keseimbangan antara akurasi dan kelengkapan deteksi, menghindari kesalahan positif dan negatif. Grafik ini membantu memahami perkembangan performa YOLOv7 seiring waktu, terkait keseimbangan presisi dan kelengkapan. Grafik F1_curve menampilkan perubahan nilai F1-score untuk semua kelas objek selama pelatihan. Nilai “*all class*” 0.50 pada threshold 0.089 menunjukkan kemampuan model secara umum dalam mendeteksi objek dengan akurasi. Berbagai kelas (bus, mobil, sepeda motor, plat, truk) diidentifikasi dengan warna yang berbeda. Pemandanya membantu memvisualisasikan performa deteksi untuk masing-masing kelas objek.



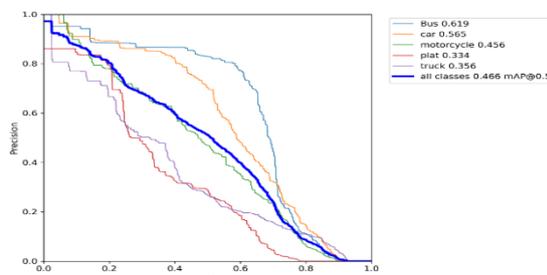
Gambar 10. Grafik P_curve

Gambar 10 P_curve merupakan grafik yang menunjukkan presisi dari berbagai ambang batas deteksi untuk setiap kelas objek yang dihasilkan oleh model YOLOv7. Dalam grafik ini, “*all class*” merepresentasikan presisi keseluruhan dari semua kelas objek yang dideteksi oleh model. Nilai “*all class*” sebesar 1.00 pada *threshold* 0.636 menunjukkan bahwa model memiliki presisi sempurna dalam mengenali dan mendeteksi objek untuk semua kelas. P_curve membantu memahami bagaimana presisi model YOLOv7 berubah terhadap variasi ambang batas deteksi, memungkinkan penyesuaian ambang batas yang optimal sesuai kebutuhan aplikasi deteksi objek.



Gambar 11. Grafik R_curve

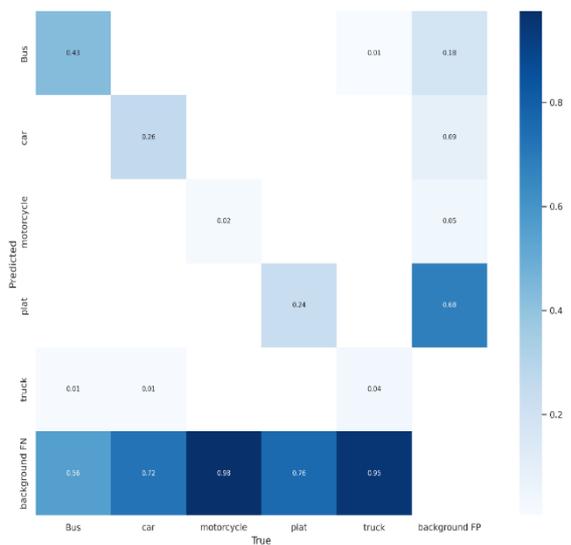
Gambar 11 merupakan Grafik R_curve yang mencerminkan perubahan tingkat *recall* seiring variasi ambang batas deteksi pada model YOLOv7. Grafik ini membantu mengevaluasi sejauh mana model mampu menemukan objek dengan benar. Pada R_curve, “*all class*” mencerminkan recall keseluruhan yang dihitung dari seluruh kelas objek yang berhasil dideteksi oleh hasil YOLOv7. Pada nilai “*all class*” 0.87 dan *threshold* 0.000, model mampu mendeteksi sebagian besar objek dari semua kelas dengan tingkat recall yang tinggi. R_curve pada YOLOv7 adalah grafik yang menggambarkan perubahan nilai *recall* terhadap berbagai nilai *threshold* deteksi untuk setiap kelas objek yang dihasilkan oleh model. Grafik ini membantu pemahaman tentang kemampuan model dalam mengenali objek yang seharusnya ada dalam gambar.



Gambar 12. Grafik PR_curve

Gambar 12 yakni PR_curve dalam YOLOv7 adalah grafik yang menggambarkan hubungan antara presisi (*precision*) dan *recall* pada model deteksi objek. Grafik ini menunjukkan bagaimana keseimbangan antara presisi dan recall berubah dengan perubahan *threshold* deteksi. PR_curve

membantu mengevaluasi performa model dalam hal presisi dan *recall* serta memahami *trade-off* antara keduanya. Dalam grafik *PR_curve*, sumbu x adalah nilai *recall*, sedangkan sumbu y adalah nilai presisi. Setiap titik pada kurva mewakili ambang batas deteksi tertentu. *PR_curve* membantu mengukur sejauh mana model mampu mengenali dan mendeteksi objek dengan akurat. Nilai “*all classes*” mencerminkan presisi rata-rata dari semua kelas objek, yaitu 0.466 *mAP@0.5*. Semakin tinggi nilai presisi, semakin akurat model dalam mengidentifikasi objek yang benar.



Gambar 13. *Confusion Matrix YOLOv7 Layer 5x5*

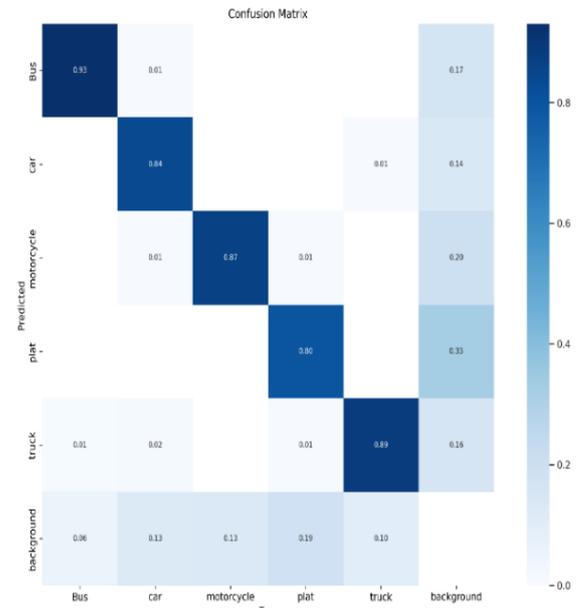
Gambar 13 adalah hasil *confusion matrix* dari YOLOv7 menggunakan *layer 5x5* menunjukkan evaluasi kinerja model dalam mendeteksi berbagai kelas objek seperti bus, car, motorcycle, plat dan truk. Setiap metrik dievaluasi untuk setiap kelas objek berbeda, dan hasilnya adalah bus memiliki nilai presisi sekitar 0.69, *recall* 0.43 dan akurasi sekitar 0.56. Untuk car memiliki nilai presisi sekitar 0.74, *recall* 0.26 dan akurasi sekitar 0.40. Truck memiliki nilai presisi sekitar 0.66, *recall* 0.04 dan akurasi sekitar 0.03, untuk motorcycle memiliki nilai presisi sekitar 0.28, *recall* 0.02 dan akurasi sekitar 0.22. Dan untuk plat sendiri memiliki nilai presisi sekitar 0.26, *recall* 0.24 dan akurasi sekitar 0.16. Selanjutnya, akurasi total keseluruhan model YOLOv7 dengan *layer 5x5* dihitung dengan membagi prediksi benar semua kelas dengan total sampel dan dikalikan 100%. Dalam kasus ini, akurasi total keseluruhan adalah sekitar 16.5%.

3.4. Perbandingan 3 Arsitektur

Arsitektur yang dibandingkan dalam penelitian ini adalah YOLOv5, *Mobilenet*, dan ResNet34. Tujuan dari perbandingan arsitektur YOLOv5, *Mobilenet*, dan ResNet34 dalam penelitian ini adalah untuk memahami perbedaan dalam efisiensi komputasi dan kinerja relatif antara ketiga arsitektur

tersebut dalam konteks deteksi dan klasifikasi objek. Adapun hasil dari ketiga arsitektur tersebut sebagai berikut.

3.4.1. Arsitektur YOLOv5



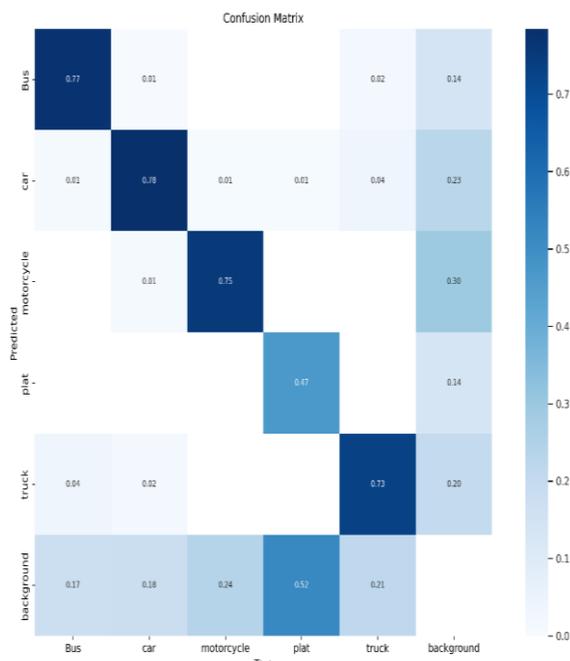
Gambar 14. *Confusion Matrix YOLOv5 Layer 3x3*

Gambar 14, disajikan visualisasi dari *confusion matrix* dengan *layer 3x3*, merupakan tabel yang menggambarkan hasil prediksi model terhadap kelas objek sebenarnya. Setiap sel dalam matriks ini memberikan informasi tentang seberapa baik model mengklasifikasikan objek dalam kategori benar atau salah. Berdasarkan nilai-nilai dalam *confusion matrix* untuk setiap kelas, dapat dihitung nilai presisi, *recall*, dan akurasi menggunakan rumus yang diberikan. Sebagai contoh, untuk kelas “bus,” didapatkan nilai presisi sekitar 0.83, *recall* sekitar 0.93, dan akurasi sekitar 0.94. Hasil perhitungan ini juga diterapkan untuk kelas “car,” “truck,” “motorcycle,” dan “plat.”

Dengan melihat hasil *confusion matrix*, dapat dilihat bahwa model YOLOv5 dengan *layer 3x3* memiliki performa yang beragam untuk setiap kelas objek. Model ini lebih akurat dalam mengenali “bus” dan “motorcycle,” sedangkan akurasi sedikit lebih rendah untuk “car,” “plat,” dan “truck.” Penilaian ini penting untuk evaluasi kemampuan model dalam mengenali objek-objek tertentu, dengan akurasi sebagai metrik utama yang mengukur sejauh mana model mengenali objek secara benar. Selain itu, nilai akurasi total keseluruhan dari *confusion matrix* dapat dihitung untuk mengukur performa keseluruhan model, dengan hasil sekitar 71.9%.

Gambar 15, evaluasi menggunakan model YOLOv5 dengan *layer 5x5* menghasilkan nilai *True Positive (TP)* untuk setiap kelas objek. Nilai TP mencerminkan jumlah objek yang berhasil dikenali secara akurat oleh model. Contohnya, kelas “bus” memiliki TP sebesar 0.77, menunjukkan model

mampu mengenali objek bus dengan tingkat akurasi yang tinggi. Hal serupa terjadi pada kelas “car” dengan TP sebesar 0.78, dan pada kelas “motorcycle” dengan TP sebesar 0.75. Namun, kelas “plat” memiliki TP lebih rendah, yaitu 0.47, menunjukkan model memerlukan peningkatan dalam mengenali objek plat nomor. Evaluasi juga menunjukkan nilai TP untuk kelas “truck” sebesar 0.73. Secara keseluruhan, evaluasi ini menggambarkan bahwa model YOLOv5 dengan layer 5x5 memiliki kemampuan yang baik dalam mengenali objek-objek dalam gambar, terutama pada kelas “bus,” “car,” “motorcycle,” dan “truck.” Meskipun ada variasi dalam akurasi setiap kelas objek, hasil evaluasi ini memberikan wawasan tentang kinerja model dalam tugas deteksi objek. *Confusion matrix* dan hasil evaluasi ini dapat membantu menganalisis sejauh mana model mengenali objek dengan benar dan memberikan akurasi total keseluruhan sekitar 58.3%.



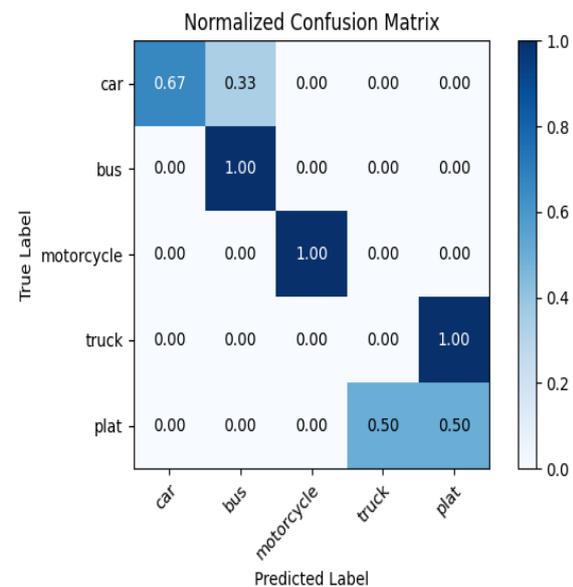
Gambar 15. *Confusion Matrix* YOLOv5 Layer 5x5

3.4.2. Arsitektur Mobilenet

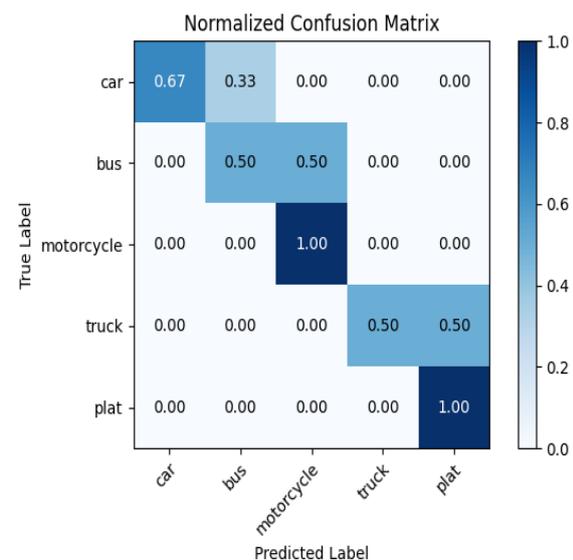
Mobilenet adalah arsitektur jaringan saraf untuk pengolahan gambar di perangkat mobile dengan sumber daya terbatas [24]. Ini menggunakan *depthwise separable convolution* untuk efisiensi. Modul Roboflow diimpor dan dataset diakses. Model dasar diatur agar tidak dapat dilatih. *Mobilenet* dibangun menggunakan API *sequential* dari *TensorFlow*, memungkinkan deteksi objek seperti mobil, bus, truk, sepeda motor, dan plat dengan efisien pada perangkat terbatas.

Gambar 16 *Confusion matrix* pada arsitektur *Mobilenet* layer 3x3 memberikan gambaran tentang performa model dalam mengklasifikasikan objek pada masing-masing kelas. Setiap baris dan kolom merepresentasikan kelas sebenarnya dan kelas yang

diprediksi. Diagonal utama menunjukkan jumlah prediksi yang benar (*true positive*), sedangkan sel di luar diagonal utama menunjukkan prediksi yang salah (*false positive* atau *false negative*). Evaluasi model pada kelas “bus” menunjukkan akurasi sekitar 88% dengan recall sebesar 0.75. Pada kelas “car”, akurasi mencapai 90% dengan *recall* 1. Pada kelas “plat”, akurasi sekitar 25% dengan *recall* 0.33. Kelas “motorcycle” memiliki akurasi 100% dengan *recall* 1. Namun, kelas “truck” memiliki akurasi 50% dengan *recall* 0. Model ini memiliki akurasi total keseluruhan sekitar 63.4% dalam mengenali objek dari kelima kelas.



Gambar 16. *Confusion Matrix* Mobilenet Layer 3x3



Gambar 17. *Confusion Matrix* Mobilenet Layer 5x5

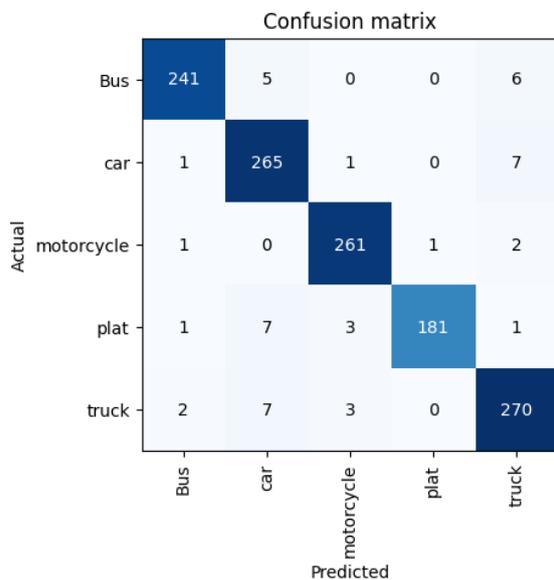
Gambar 17 yakni *Convusion matrix* yang merupakan output arsitektur *Mobilenet* dengan layer 5x5, hasil dari *confusion matrix* menggambarkan evaluasi performa model dalam mengklasifikasikan berbagai kelas objek. Nilai presisi, *recall*, dan akurasi

dihitung dengan rumus yang telah dijelaskan. Berdasarkan hasil evaluasi, diperoleh:

Pada kelas “car”, akurasi sekitar 91% dengan recall 1, kelas “bus”, akurasi sekitar 78% dengan recall 0.60. Pada kelas “truck”, akurasi sekitar 75% dengan recall 1, kelas “motorcycle”, akurasi sekitar 83% dengan recall 1, dan kelas “plat”, akurasi sekitar 100% dengan recall 0.66. Akurasi total keseluruhan dari arsitektur *Mobilenet* layer 5x5 adalah sekitar 73.4%. Ini mewakili sejauh mana model berhasil mengklasifikasikan objek-objek dari berbagai kelas secara keseluruhan

3.4.3. Arsitektur ResNet34

ResNet34 adalah jenis jaringan saraf untuk pengolahan gambar dengan 34 lapisan, yang menggunakan blok residu untuk lebih baik dalam mengenali objek dalam gambar [25]. Keunggulan utamanya adalah kemampuan mengatasi masalah pada jaringan dalam kasus kedalaman yang dalam. ResNet34 memiliki struktur dengan lapisan konvolusi, pengelompokan lapisan untuk efisiensi, dan lapisan global pooling untuk menggabungkan informasi. Dengan fitur-fitur ini, mampu mengenali objek dengan akurat dan telah berhasil dalam berbagai kompetisi.



Gambar 18. Confusion Matrix ResNet34 Layer 3x3

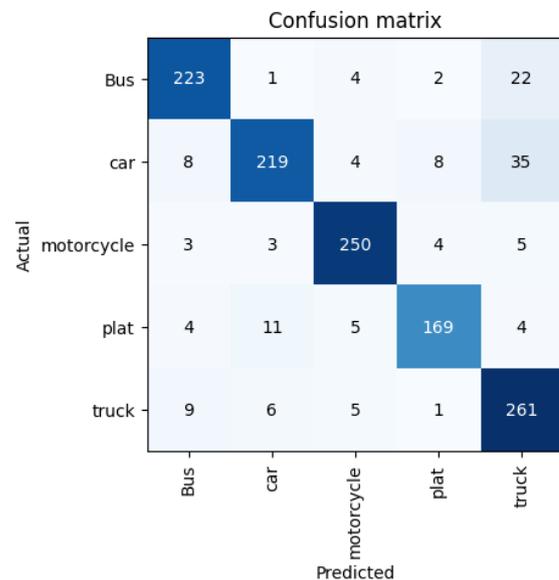
Pada Gambar 18, *Confusion Matrix* menunjukkan hasil evaluasi performa arsitektur ResNet34 dengan layer 3x3 dalam mengenali berbagai kelas objek. Matriks ini memberikan gambaran tentang sejauh mana model berhasil mengklasifikasikan objek-objek pada setiap kelas. Nilai-nilai seperti *Presiasi*, *Recall*, dan Akurasi dihitung untuk setiap kelas berdasarkan matriks ini.

Contohnya, untuk kelas “bus”, terdapat *True Positive* (TP) sebanyak 241 dan *False Positive* (FP) sebanyak 11. Dengan menghitung rumus presisi ($Presiasi = TP / (TP + FP)$) dan *recall* ($Recall =$

$TP / (TP + FN)$) nilai presisi adalah 0.95 dan nilai *recall* adalah 0.97. Selanjutnya, Akurasi dihitung dengan memasukkan TP, TN, FP, dan FN ke rumus akurasi ($Akurasi = (TP + TN) / (TP + FP + FN + TN)$) yang menghasilkan nilai akurasi sebesar 0.98.

Hasil evaluasi yang serupa juga dilakukan untuk kelas objek lainnya seperti “car”, “truck”, “motorcycle”, dan “plat”. Setiap kelas memiliki nilai-nilai *Presiasi*, *Recall*, dan Akurasi yang mencerminkan kemampuan model dalam mengenali objek pada kelas tersebut.

Untuk akurasi total keseluruhan (*Overall Total Accuracy*), nilai-nilai prediksi benar dari semua kelas dibagi dengan jumlah total sampel, kemudian dikonversi menjadi persentase. Pada pengujian ini, akurasi total keseluruhan mencapai 96.2%, yang mengindikasikan tingkat keberhasilan model dalam mengenali berbagai jenis objek.



Gambar 19. Confusion Matrix Resnet34 Layer 5x5

Gambar 19 merupakan pengujian arsitektur ResNet34 dengan layer 5x5, *confusion matrix* digunakan untuk mengukur kinerja sistem. Nilai-nilai seperti presisi, recall, dan akurasi dihitung untuk masing-masing kelas objek seperti bus, mobil, truk, sepeda motor, dan plat. Presisi mengukur sejauh mana hasil positif benar-benar positif, *recall* mengukur sejauh mana hasil positif terdeteksi, dan akurasi mengukur tingkat keseluruhan kebenaran hasil prediksi.

Contoh hasil pengujian menunjukkan bahwa untuk kelas bus, presisi mencapai 0.88, *recall* 0.90, dan akurasi 0.95. Ini berarti sistem cenderung memberikan hasil yang tepat untuk objek kelas bus. Begitu juga untuk kelas lainnya seperti mobil, truk, sepeda motor, dan plat nomor, nilai presisi, *recall*, dan akurasi dihitung untuk mengukur kinerja sistem dalam mengenali jenis-jenis objek tersebut.

Akhirnya, nilai akurasi total keseluruhan dari semua kelas dihitung untuk memberikan gambaran tentang kinerja sistem secara keseluruhan. Pada

contoh pengujian ini, akurasi total keseluruhan mencapai 88.6%, menunjukkan tingkat keberhasilan sistem dalam mengenali berbagai jenis objek dalam gambar.

3.5. Hasil Pengujian Sistem

Pengujian sistem dilakukan dengan metode terencana dan sistematis, melibatkan pengujian perangkat lunak dan prototipe secara menyeluruh. Hasil pengujian digunakan untuk perbaikan dan evaluasi. Jika hasil pengujian tidak sesuai dengan kebutuhan sistem, dilakukan perbaikan dan evaluasi ulang. Tujuannya adalah mencapai performa dan kinerja yang diharapkan. Pendekatan sistematis ini bertujuan menghasilkan sistem yang lebih handal dan stabil.

Tabel 1. Hasil Pengujian Perangkat Keras

Perangkat Keras	Parameter	Keterangan
Webcam Xiaomi 1080p	Kualitas Grafik	Baik
	Video	
	USB Port	Baik
Laptop Ryzen 7	AMD Power Port	Baik
	Indikator Power	Baik
	Kipas Pendingin	Baik
	USB port	Baik

Tabel 1 adalah hasil pengujian perangkat keras untuk pendeteksian jenis dan plat nomor kendaraan dengan YOLOv7 melibatkan laptop AMD Ryzen 7 dan webcam Xiaomi 1080p. Pengujian ini mencakup evaluasi terhadap waktu eksekusi deteksi objek dan akurasi deteksi pada kedua perangkat keras tersebut. Evaluasi dilakukan dengan dataset deteksi kendaraan, memungkinkan penilaian terhadap kinerja perangkat keras dalam mendeteksi objek kendaraan pada gambar.

Tabel 2. Hasil Pengujian Perangkat Lunak

Perangkat Lunak	Parameter	Keterangan
Sistem Operasi Windows 11	Tampilan <i>enviroment</i> OS Windows ketika dijalankan	Baik
	Tampilan <i>Dekstop</i> Kontrol	Baik
Visual Code	Studio	Baik
	Secure Shell (SSH) Connection	Baik

Tabel 2 merupakan pengujian pada perangkat lunak yang digunakan yakni operasi sistem windows 11 menggunakan anaconda terminal, google colaboryat dan OpenCV.

Hasil Pengujian Keseluruhan, dalam tahapan ini peneliti menggunakan pendekatan blackbox yang melibatkan pengujian keseluruhan sistem. *Blackbox testing* adalah metode di lapangan untuk menilai kinerja sistem. Pendekatan ini menguji sistem secara menyeluruh tanpa mengungkapkan detail internalnya. Setelah pengujian selesai, hasilnya disajikan dalam tabel untuk analisis dan pemahaman yang lebih mudah.

3.6. Pengujian Akurasi Deteksi Jenis dan Plat Nomor Kendaraan

Pengujian akurasi pendeteksian jenis dan plat kendaraan bertujuan untuk mengevaluasi kinerja sistem deteksi. Tujuannya adalah memastikan apakah sistem pendeteksian kendaraan dan plat nomor bekerja dengan baik atau perlu perbaikan. Pengujian melibatkan perbandingan hasil deteksi sistem dengan data kendaraan sebenarnya. Dengan melakukan pengujian secara teratur, diharapkan sistem dapat tetap akurat dan dapat diandalkan sesuai kebutuhan pengguna. Dalam penelitian ini, dilakukan berbagai skenario pengujian untuk memastikan performa sistem.

Tabel 3. Pengamatan Objek Deteksi

Jenis Kendaraan	Jumlah Objek	Keberhasilan Mendeteksi	Rata-Rata Akurasi
Bus	30	27	90%
Mobil	30	30	100%
Motor	30	30	100%
Truk	30	22	73%
Plat	120	109	90%
Akurasi total			90%
Bus	30	27	90%
Mobil	30	28	93%
Motor	30	29	96%
Truk	30	25	83%
Plat	120	95	79%
Akurasi total			85%
Bus	30	20	66%
Mobil	30	29	96%
Motor	30	27	90%
Truk	30	11	36%
Plat	120	99	82%
Akurasi total			77%
Akurasi total keseluruhan			84%

Tabel 3 adalah pengamatan objek deteksi. Pada pengamatan ini, dilakukan pengisian tabel pengamatan deteksi objek berdasarkan kondisi pagi, siang dan malam. Rata-rata akurasi dihitung dengan rumus:

$$\eta = \frac{TP+TN}{N} \times 100\% \quad (5)$$

Hasil pengamatan menunjukkan bahwa kondisi pagi hari, jenis kendaraan mobil dan motor memiliki akurasi 100%, bus dan plat memiliki akurasi 90%, sementara truk memiliki akurasi 73%. Akurasi total pendeteksian objek pada pagi hari adalah 90%. Pada kondisi siang hari, jenis kendaraan motor memiliki akurasi 96%, mobil 93%, bus 90%, dan truk 83%. Akurasi pendeteksian plat rendah, yaitu 79%. Akurasi total pendeteksian objek pada siang hari adalah 85%. Sementara pada kondisi malam hari, jenis kendaraan mobil memiliki akurasi 96% dan motor 90%, sementara itu plat 82% dan bus 66%. Akurasi pendeteksian truk rendah, yaitu 36%. Akurasi total pendeteksian objek pada malam hari adalah 77%.

Dari ketiga kondisi tersebut, dihitung akurasi total keseluruhan dengan rumus:

$$Akurasi\ Total = \frac{TP+TN}{N} \times 100\% \quad (6)$$

$$Akurasi\ Keseluruhan = \frac{608}{720} \times 100\%$$

$$Akurasi\ Keseluruhan = 84\%$$

Dengan demikian, hasil pengamatan menunjukkan bahwa secara keseluruhan, sistem berhasil mendeteksi objek dengan akurasi sebesar 84%.

Tabel 4. Konversi Plat Nomor DD 1005 QE

Percobaan konversi	Hasil Prediksi	Kesalahan	Keterangan	Presentase Keberhasilan
1	Ivu 1005 Qe	3	Gagal	62%
2	dd 1005 QC	1	Gagal	87%
3	dd 1005 Qel	1	Gagal	87%
4	DD 7005 Qe	1	Gagal	87%
5	dD QE 1005	0	Berhasil	100%
6	dd 1005 QE	0	Berhasil	100%
7	DD 7005 QE	1	Gagal	87%
8	Dd 1005 QE	0	Berhasil	100%
9	Dd 1005 QE	0	Berhasil	100%
10	dd 1005 QE	0	Berhasil	100%

Tabel 4 menampilkan hasil konversi plat nomor "DD 1005 QE" yang terdiri dari 8 karakter dalam 10 percobaan konversi. Dari keseluruhan percobaan, terdapat 5 percobaan yang tidak berhasil mengenali karakter pada plat nomor kendaraan. Pada percobaan pertama, terjadi 3 kesalahan dalam mengenali karakter, termasuk kesalahan dalam jumlah karakter yang seharusnya "DD" tetapi terdeteksi sebagai "Ivu". Akibatnya, presentase keberhasilan pada percobaan ini adalah 62%. Pada percobaan kedua, satu karakter "E" salah terdeteksi sebagai "C",

sehingga presentase keberhasilannya 87%. Pada percobaan ketiga, satu kesalahan terjadi karena adanya karakter "I" yang lebih, menghasilkan presentase keberhasilan 87%. Pada percobaan ketujuh, karakter angka "1" salah terbaca sebagai angka "7", memberikan presentase keberhasilan 87%. Lima percobaan lainnya berhasil mengenali karakter dengan sempurna pada percobaan ke-5, ke-6, ke-8, hingga ke-10, masing-masing memiliki presentase keberhasilan 100%.

Tabel 5. Konversi Plat Nomor

Plat Nomor	Jumlah Karakter Plat	Jumlah Totah Kesalahan Karakter	Jumlah Kegagalan Percobaan Konversi	Nilai Rata-Rata Presentase Keberhasilan
DD 1005 QE	8	7	5	91%
DD 1521 LW	8	14	10	82,1%
DD 2943 OY	8	31	9	61,1%
DD 8615 HES	9	19	8	76,1%
DP 1027 BI	8	9	7	88,5%

Tabel 5 menggambarkan hasil konversi plat nomor beserta kegagalan percobaan dan rata-rata presentase keberhasilannya. Misalnya, plat nomor DD 1005 QE dengan 8 karakter mengalami 5 kegagalan dan 5 keberhasilan konversi, menghasilkan rata-rata presentase keberhasilan sekitar 91%. Plat nomor DD 1521 LW dengan 8 karakter mengalami 10 kegagalan, sehingga rata-rata keberhasilannya adalah 82.1%.

Hal serupa terjadi pada plat nomor lain seperti DD 2943 OY (8 karakter, 9 kegagalan, rata-rata keberhasilan 61.1%), DD 8615 HES (9 karakter, 8 kegagalan, rata-rata keberhasilan 76.1%), dan DD 1027 BI (8 karakter, 7 kegagalan, rata-rata keberhasilan 88.5%). Kegagalan umumnya terjadi karena kesalahan dalam mengenali karakter yang tidak sesuai dengan karakter asli pada plat nomor kendaraan.

Untuk menghitung akurasi keseluruhan, digunakan rumus:

$$Overall\ Accuracy = \frac{Total\ kegagalan\ percobaan}{Total\ percobaan} \times 100\% \quad (7)$$

$$Overall\ Accuracy = (1 - (\frac{39}{50})) \times 100\%$$

$$Overall\ Accuracy = (1 - (0.78)) \times 100\%$$

$$Overall\ Accuracy = 0.22 \times 100\%$$

$$Overall\ Accuracy = 22\%$$

Tabel 6. Pengujian Layer Konvolusi Terhadap Empat Arsitektur

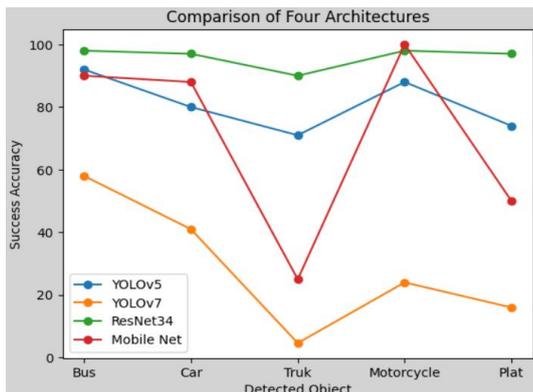
Arsitektur Model	Layer	Ukuran Jendela	Akurasi (%)
YOLOv5	CNN	3x3	71,9%
	CNN	5x5	58,3%
YOLOv7	CNN	3x3	17,1%
	CNN	5x5	16,5%
ResNet34	CNN	3x3	89,7%
	CNN	5x5	88,6%
Mobilenet	CNN	3x3	63,4%
	CNN	5x5	73,4%

Tabel 6 memuat akurasi berdasarkan ukuran jendela pada berbagai arsitektur model. ResNet34 mencapai akurasi tertinggi dengan layer 3x3 sekitar 89% dan layer 5x5 sekitar 88.6%. Mobilenet memiliki akurasi 63.4% pada layer 3x3 dan 73.4% pada layer 5x5. YOLOv5 memiliki akurasi 71.9% pada layer 3x3 dan 58.3% pada layer 5x5. Di antara ketiganya, YOLOv7 memiliki akurasi paling rendah, yaitu 17.1% pada layer 3x3 dan 16.5% pada layer 5x5.

Tabel 7. Hasil Tingkat Akurasi Dari Empat Model Arsitektur

Model Arsitektur	Bus (%)	Car (%)	Truck (%)	Motorcycle (%)	Plat (%)
YOLOv5	92%	80%	71%	88%	74%
YOLOv7	58%	41%	4.63%	24%	16%
ResNet34	98%	97%	90%	98%	97%
Mobilenet	90%	88%	25%	100%	50%

Tabel 7 membandingkan empat arsitektur dalam pengenalan lima kelas objek: bus, car, truck, motorcycle, dan plat. Arsitektur ResNet34 menonjol dengan akurasi tertinggi untuk semua kelas: bus 98%, car 97%, truck 90%, motorcycle 98%, dan plat 97%. ResNet34 juga memiliki akurasi tertinggi untuk empat kelas objek ini. Mobilenet, meskipun bukan yang tertinggi, memiliki akurasi yang signifikan: bus 90%, car 88%, truck 25%, motorcycle 100%, dan plat 50%. Ini berhasil mengenali objek truck dengan akurasi 100%. YOLOv5 menunjukkan performa bagus dengan akurasi: bus 92%, car 80%, truck 71%, motorcycle 88%, dan plat 74%. Pencapaiannya dalam mendeteksi bus mencapai 92%, melampaui arsitektur Mobilenet. Namun, arsitektur YOLOv7 menunjukkan akurasi yang rendah dibanding yang lain: bus 58%, car 41%, truck 4.63%, motorcycle 24%, dan plat 16%.



Gambar 20. Arsitektur Dalam Deteksi Objek

Tabel 9. Nilai Rata-Rata Intensitas Cahaya Berdasarkan Frame Pada Video

Bukti Video	Jumlah Frame	Keberhasilan Pendeteksian		Kegagalan Pendeteksian		Nilai Rata-Rata Intensitas Cahaya
		Jenis Kendaraan	Plat Kendaraan	Jenis Kendaraan	Plat Kendaraan	
	140	51	0	89	140	12.048942029734345
	322	299	244	23	78	7.588182850415981
	197	119	141	78	56	13.06435218967381
	746	215	275	531	417	89.77252503630726

Tabel 9 menampilkan nilai rata-rata intensitas cahaya berdasarkan jumlah frame pada video tiap kendaraan. Pada video pertama yaitu bus dengan jumlah 140 frame, hanya berhasil mengenali jenis kendaraan sebanyak 51 frame dan gagal dalam mengenali plat kendaraan dengan nilai rata-rata intensitas cahaya 12.04. Video kedua yaitu mobil

Gambar 20 menunjukkan grafik dari Tabel 7 yang membandingkan performa berbagai arsitektur dalam mengenali objek. Hasil perbandingan, ResNet34 mampu mengenali objek sekitar 90% hingga 98%, Mobilenet dapat mengenali objek dengan akurasi berkisar 25% hingga 100%, YOLOv5 mencapai akurasi deteksi objek sekitar 71% hingga 92% dan YOLOv7 memiliki kemampuan mengenali objek dengan akurasi sekitar 4.63% hingga 58%. Grafik ini memberikan gambaran perbandingan performa relatif antara arsitektur-arsitektur tersebut dalam tugas deteksi objek.

Tabel 8. Perbandingan Kecepatan Komputasi Dari Empat Model Arsitektur

Model Arsitektur	Rata-rata kecepatan komputasi (ms)
YOLOv5	225.023
YOLOv7	217.969
ResNet34	11122.244
Mobilenet	82686.43

Tabel 8 menampilkan nilai rata-rata kecepatan komputasi dalam satuan milisecond pada tiap model arsitektur. Dapat dilihat dari tabel bahwa YOLOv5 memiliki rata-rata kecepatan komputasi sebesar 225.023 milisecond, pada YOLOv7 rata-rata kecepatan komputasi sebesar 217.969 milisecond, sedangkan pada ResNet34 rata-rata kecepatan komputasi sebesar 11122.244 milisecond, dan Mobilenet rata-rata kecepatan komputasi sebesar 82686.43 milisecond. Dari tabel tersebut, YOLO merupakan model arsitektur yang memiliki kecepatan komputasi yang sangat cepat dibandingkan dengan ResNet34 dan mobilenet.

dengan jumlah 322 frame, berhasil mengenali jenis kendaraan sebanyak 299 frame dan 244 frame yang berhasil dalam mengenali plat kendaraan dengan nilai rata-rata intensitas cahaya 7.58. Video ketiga yaitu motor dengan jumlah 197 frame, berhasil mengenali jenis kendaraan sebanyak 119 frame dan 141 frame yang berhasil dalam mengenali plat kendaraan

dengan nilai rata-rata intensitas cahaya 13.06. Video terakhir yaitu truk dengan jumlah 746 *frame*, berhasil mengenali jenis kendaraan sebanyak 215 *frame* dan 275 *frame* yang berhasil dalam mengenali plat kendaraan dengan nilai rata-rata intensitas cahaya 89.77.

4. DISKUSI

Pada penelitian ini, digunakan algoritma *Convolutional Neural Network* (CNN) dengan model *You Only Look Once* (YOLO) untuk deteksi jenis dan plat nomor kendaraan pada gambar. YOLO merupakan metode *real-time* deteksi objek yang menggabungkan kecepatan dan ketepatan dengan konsep konvolusi dan pemrosesan spasial. CNN adalah arsitektur jaringan yang efektif untuk proses gambar. Dalam implementasi, model CNN YOLOv7 dilatih dengan 7.400 dataset gambar kendaraan berlabel. Pelabelan melibatkan arsitektur jaringan, fungsi *loss*, *optimizer*, dan *augmentasi* data untuk performa yang lebih baik. Kelas objek terdiri dari motor, mobil, truk, bus, dan plat nomor. Setelah pelabelan, data dipindahkan ke platform roboflow untuk pelatihan di Google Colab. Proses melibatkan setting arsitektur YOLO7, fungsi *loss*, *optimizer*, dan *augmentasi* data seperti *flipping*, *zooming*, dan perubahan kecerahan. Hasil evaluasi menggunakan metrik akurasi dari *confusion matrix* dan F1-score setelah pengujian pada dataset "*testing*" di roboflow. Performa model dibandingkan dengan beberapa arsitektur seperti YOLOv5, YOLOv7, ResNet34, dan *Mobilenet*. Dalam pengujian lapangan, akurasi deteksi berfluktuasi berdasarkan waktu dan kondisi. Dalam skenario *confusion matrix*, ResNet34 memiliki akurasi terbaik, YOLOv7 memiliki akurasi terendah. Namun, YOLOv5 memiliki akurasi tinggi untuk beberapa kelas objek, sedangkan YOLOv7 lebih rendah. Arsitektur ResNet34 memiliki performa yang baik untuk semua kelas objek. *Mobilenet* memiliki performa sedang. Kecepatan komputasi juga diperbandingkan, dengan YOLO lebih cepat. Dalam kesimpulan, ResNet34 memiliki performa terbaik dalam mendeteksi objek, sementara YOLOv7 memiliki performa rendah. Namun, YOLOv7 menunjukkan peningkatan signifikan dibandingkan dengan penelitian sebelumnya berjudul "Deteksi Dan Klasifikasi Kendaraan Berbasis Algoritma *You Only Look Once* (YOLO)," yang menggunakan YOLOv3 dengan nilai mAP hanya sekitar 0,57 atau 0,57%, sedangkan YOLOv7 mencapai nilai mAP rata-rata sekitar 0,95 atau 95% [26]. Analisis lebih lanjut diperlukan untuk memahami faktor-faktor yang mempengaruhi performa model, serta pembahasan kelebihan dan kelemahan serta upaya pengembangan di masa depan.

5. KESIMPULAN

Pengimplementasian algoritma model YOLOv7 melalui tahapan akuisisi citra, pelabelan, pembagian

dataset, pelatihan model YOLOv7, pengujian, hasil prediksi dan konversi ke teks menggunakan OCR. Hasil pengujian sistem deteksi jenis kendaraan dan plat kendaraan pada kondisi pagi hari didapatkan akurasi sebesar 90%, pada kondisi siang hari didapatkan akurasi sebesar 85% dan kondisi malam hari didapatkan akurasi sebesar 77%. Dari ketiga kondisi tersebut didapatkan akurasi sebesar 84% dalam melakukan pengenalan objek deteksi jenis kendaraan dan plat kendaraan dari jumlah data yang diuji secara langsung 720 dan yang berhasil terdeteksi 608. Sedangkan hasil pengujian sistem deteksi pada OCR yang mengkonversi plat nomor kendaraan didapatkan nilai akurasi sebesar 22%. Berdasarkan akurasi tersebut pada penelitian ini secara keseluruhan telah menyajikan sistem yang kuat dan efektif untuk pengenalan jenis dan plat nomor kendaraan beserta konversi ke *text*. Kebaruan dari penelitian ini dibandingkan dengan penelitian-penelitian sebelumnya yakni pada pengenalan 4 jenis kelas kendaraan yaitu mobil, motor, bus, truk dan 1 kelas plat nomor, bukan hanya itu penerapan pengenalan karakter dilakukan menggunakan algoritma terbaru YOLOv7 *object detection* lalu membandingkannya dengan arsitektur model CNN yang lain.

Adapun saran kepada peneliti selanjutnya ialah lebih memfokuskan pada peningkatan akurasi deteksi objek pada kondisi pencahayaan yang rendah seperti pada skenario pengamatan malam hari, mengoptimalkan proses konversi data OCR untuk meningkatkan tingkat keberhasilan deteksi plat nomor kendaraan, juga melakukan evaluasi dan penyesuaian terhadap sudut kamera.

DAFTAR PUSTAKA

- [1] S. T. Informatika, "Implementasi Deteksi Real Time Klasifikasi Jenis Kendaraan Di Indonesia Menggunakan Metode YOLOV5," *Jurnal Pendidikan Tambusai*, vol. 6, pp. 13971–13982, 2022.
- [2] R. Yudhistira, A. Pratama, H. Koesyanto, and I. Artikel, "Kejadian Kecelakaan Pada Pengemudi Ojek Online," *HIGEIA (Journal of Public Health ...)*, vol. 4, no. Special 1, pp. 13–24, 2020.
- [3] R. Nurfauziah and H. Krisnani, "Perilaku Pelanggaran Lalu Lintas Oleh Remaja Ditinjau Dari Perspektif Konstruksi Sosial," *Jurnal Kolaborasi Resolusi Konflik*, vol. 3, no. 1, p. 75, 2021,
- [4] M. Riska, "Efektivitas Penerapan E-Government Melalui Sistem E-Tilang Pada Satuan Polisi Lalu Lintas (Satlantas) Polresta Pekanbaru," *J Chem Inf Model*, vol. 6, no. 9, pp. 1689–1699, 2019.
- [5] A. Fadli, A. R. Razak, and M. Tahir, "Kinerja Polisi Lalu Lintas Dalam Sosialisasi E-Tilang Di Kota Makassar," *Kajian Ilmiah*

- Mahasiswa Administrasi Publik (KIMAP)*, vol. 2, no. 2, pp. 742–755, 2021.
- [6] A. Haryono, S. Bismantoko, G. M. Putra, and T. Widodo, “Accuracy in Object Detection based on Image Processing at the Implementation of Motorbike Parking on the Street,” *Proceedings of the 2019 2nd International Conference on Applied Engineering, ICAE 2019*, pp. 0–4, 2019,
- [7] A. Bacchar, M. Ali, and B. Ayed, “Automatic Number Plate Recognition System based on Deep Learning Ousseme Kriaa 1 PUBLICATION 4 CITATIONS SEE PROFILE,” no. April, 2020.
- [8] A. F. Fandisyah, N. Iriawan, and W. S. Winahju, “Deteksi Kapal di Laut Indonesia Menggunakan YOLOv3,” *Jurnal Sains dan Seni ITS*, vol. 10, no. 1, 2021,
- [9] R. Akbar, Weriana, R. A. Siroj, and M. W. Afgani, “Experimental Research Dalam Metodologi Pendidikan,” *Jurnal Ilmiah Wahana Pendidikan, Januari*, vol. 2023, no. 2, pp. 465–474, 2023.
- [10] I. H. Al amin and A. Aprilino, “Implementasi Algoritma Yolo Dan Tesseract Ocr Pada Sistem Deteksi Plat Nomor Otomatis,” *Jurnal Teknoinfo*, vol. 16, no. 1, p. 54, 2022,
- [11] M. A. Jawale, P. William, A. B. Pawar, and N. Marriwala, “Implementation of number plate detection system for vehicle registration using IOT and recognition using CNN,” *Measurement: Sensors*, vol. 27, Jun. 2023,
- [12] S. G. Z. & S. Suhartono, “Sistem Objek Recognition Plat Nomor Kendaraan Untuk Sistem Parkir Bandara,” *Jessi*, vol. 03, no. November, pp. 127–134, 2022.
- [13] J. Tang, L. Wan, J. Schooling, P. Zhao, J. Chen, and S. Wei, “Automatic number plate recognition (ANPR) in smart cities: A systematic review on technological advancements and application cases,” *Cities*, vol. 129, Oct. 2022,
- [14] F. Sultan, K. Khan, Y. A. Shah, M. Shahzad, U. Khan, and Z. Mahmood, “Towards Automatic License Plate Recognition in Challenging Conditions,” *Applied Sciences (Switzerland)*, vol. 13, no. 6, Mar. 2023,
- [15] R. Rathi, A. Sharma, N. Baghel, P. Channe, S. Barve, and S. Jain, “License plate detection using YOLO v4,” *Int J Health Sci (Qassim)*, pp. 9456–9462, May 2022,
- [16] D. Setiawan, D. W. Udjiyanto, and A. Syari’udin, “Analisis faktor-faktor yang mempengaruhi tingkat kemiskinan di dua zona wilayah indonesia (WIB & WITA) tahun 2014-2020 menggunakan model ekonometrika data panel dinamis dengan pendekatan Generalized Method of Moment Arellano-Bond,” *Jurnal Ekonomi dan Bisnis*, vol. 11, no. 1, pp. 361–367, 2022.
- [17] N. Purwanto, “Variabel Dalam Penelitian Pendidikan,” *Jurnal Teknodik*, vol. 6115, pp. 196–215, 2019,
- [18] Ningsih, “Hubungan Media Pembelajaran dengan Peningkatan Siswa Pada Mata Pelajaran PAI di SMP Iptek Sengkol Tangerang Selatan,” *Tarbawai: Jurnal Pendidikan Agama Islam*, vol. 6, no. 01, pp. 77–92, 2021.
- [19] S. Elfwing, E. Uchibe, and K. Doya, “Sigmoid-weighted linear units for neural network function approximation in reinforcement learning,” *Neural Networks*, vol. 107, pp. 3–11, Nov. 2018,
- [20] R. T. Handayanto and H. Herlawati, “Prediksi Kelas Jamak dengan Deep Learning Berbasis Graphics Processing Units,” *Jurnal Kajian Ilmiah*, vol. 20, no. 1, pp. 67–76, Jan. 2020,
- [21] N. D. Miranda, L. Novamizanti, S. Rizal, F. T. Elektro, and U. Telkom, “Convolutional Neural Network Pada Klasifikasi Sidik Jari Menggunakan Resnet-50 Classification of Fingerprint Pattern Using Convolutional Neural Network in Clahe Image,” *Jurnal Teknik Informatika (JUTIF)*, vol. 1, no. 2, pp. 61–68, 2020.
- [22] K. Khairunnas, E. M. Yuniarno, and A. Zaini, “Pembuatan Modul Deteksi Objek Manusia Menggunakan Metode YOLO untuk Mobile Robot,” *Jurnal Teknik ITS*, vol. 10, no. 1, 2021,
- [23] Y. A. Usen and C. Hayat, “Design and Build Vehicle Plate Detection System Using You Only Look Once Method Based on Android,” *Jurnal Teknik Informatika (Jutif)*, vol. 4, no. 4, pp. 807–818, 2023,
- [24] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, “YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors,” *Cornell University*, pp. 1–15, 2022.
- [25] J. P. N. L. Chairisni, “Sistem Pendeteksian Dan Pengenalan Ekspresi Wajah Dengan Algoritma Yolo Dan Convolutional Neural Network,” *Universitas Tarumanagara*, p. 255, 2020.
- [26] A. Amwin, “Deteksi Dan Klasifikasi Kendaraan Berbasis Algoritma You Only Look Once (Yolo),” 2021. Accessed: Oct. 11, 2023. [Online].