

Deteksi Objek di Lapangan pada Robot Sepakbola Beroda Menggunakan Metode YOLOV5

Fadhli Dzil Ikram¹, Agus Khumaidi², M. Basuki Rahmat³, Joko Endrasmono⁴, Mat Syai'in⁵, Dimas Pristovani Riananda⁶

e-mail: fdzil@student.ppns.ac.id, aguskhumaidi@ppns.ac.id, mbasuki.rahmat@ppns.ac.id, endrasmono@ppns.ac.id, matt.syaiin@ppns.ac.id, dimaspristovani@ppns.ac.id

^{1,2,3,4,5,6}Jurusan Teknik Kelistrikan Kapal, Politeknik Negeri Perkapalan Negeri Surabaya, Institut Teknologi Sepuluh Nopember, Jl. Teknik Kimia, Keputih, Kec. Sukolilo, Surabaya, Jawa Timur 60111

Informasi Artikel

Riwayat Artikel

Diterima 31 Mei 2024

Direvisi 21 Juli 2024

Diterbitkan 31 Juli 2024

Kata kunci:

Deteksi objek
robot sepak bola beroda
YOLOv5

ABSTRAK

Penelitian ini bertujuan untuk mendeteksi objek pada robot sepak bola beroda menggunakan metode YOLOv5, yang dipilih karena kecepatannya yang lebih tinggi dibandingkan versi sebelumnya. Dataset berjumlah 4555 gambar digunakan untuk melatih dan memvalidasi model. Parameter pelatihan meliputi ukuran gambar 416x416 piksel, batch size 16, dan jumlah epoch 1000. Hasil menunjukkan presisi 0,919 dan recall tertinggi 0,97 pada tingkat kepercayaan 0,9. Kurva F1 mencapai rata-rata puncak 0,89 pada keyakinan 0,467, dengan presisi tertinggi 1,00 pada keyakinan 0,925. Sistem deteksi objek secara keseluruhan mencapai akurasi 95%. Penelitian ini membuktikan bahwa YOLOv5 efektif untuk deteksi objek pada robot sepak bola beroda, dengan performa yang memadai.

ABSTRACT

This research aims to detect objects on wheeled soccer robots using the YOLOv5 method, which was chosen for its higher speed and accuracy compared to previous versions. A dataset of 4555 images was used to train and validate the model. Training parameters included an image size of 416x416 pixels, a batch size of 16, and 1000 epochs. The results showed a precision of 0.919 and the highest recall of 0.97 at a confidence level of 0.9. The F1 curve reached an average peak of 0.89 at a confidence level of 0.467, with the highest precision being 1.00 at a confidence level of 0.925. The overall object detection system achieved an accuracy of 95%. This research proves that YOLOv5 is effective for object detection on wheeled soccer robots, providing adequate performance.

Penulis Korespondensi:

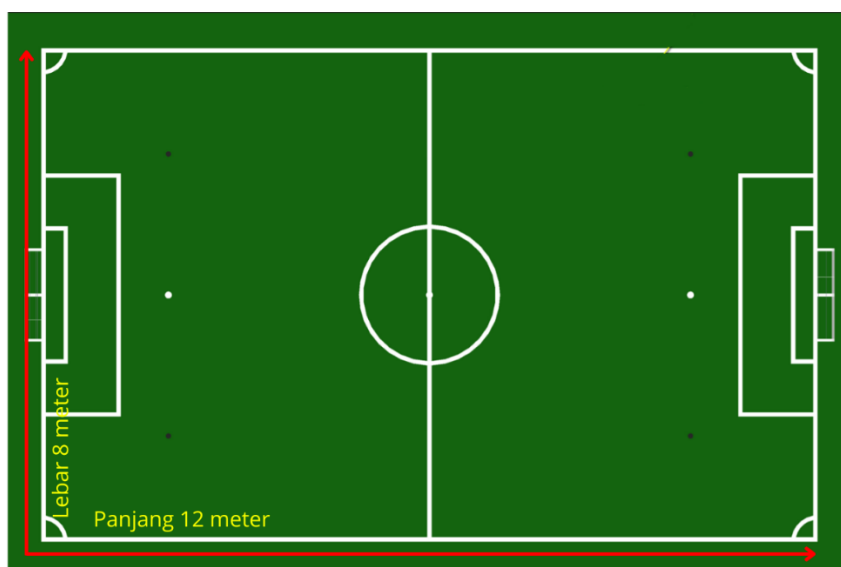
Fadhli Dzil Ikram,
Teknik Otomasi,
Politeknik Perkapalan Negeri Surabaya,
Jl. Teknik Kimia, Keputih, Kec. Sukolilo, Surabaya, Jawa Timur 60111
Email: fdzil@student.ppns.ac.id
Nomor HP/WA aktif: +62 856-0600-6443



1. PENDAHULUAN

Kontes Robot Sepak Bola Indonesia (KRSBI) kategori beroda merupakan salah satu cabang kompetisi dalam Kontes Robot Indonesia (KRI) yang diadakan setiap tahun oleh Kementerian Riset, Teknologi, dan Pendidikan Tinggi. Kompetisi KRSBI kategori beroda mengadopsi aturan dari RoboCup *Middle Size League* (MSL), sebuah turnamen robot sepak bola di tingkat internasional, dengan beberapa penyesuaian seperti ukuran lapangan dan jumlah robot yang berpartisipasi. Dalam pertandingan ini, dua tim bersaing untuk menguasai bola dan mencetak gol sebanyak mungkin guna meraih kemenangan.

Kontes Robot Sepak Bola Indonesia Beroda (KRSBI) tingkat Nasional tahun 2023 dilaksanakan secara tatap muka. Pertandingan KRSBI beroda memiliki konsep yang mirip dengan sepak bola pada umumnya, yaitu mencetak gol sebanyak mungkin dan menjaga gawang agar tidak kebobolan. Namun, dalam KRSBI beroda hanya terdapat tiga pemain, yakni dua penyerang dan satu robot penjaga gawang. Pada setiap pertandingan, dua tim saling bersaing untuk menguasai bola dan mencetak gol sebanyak-banyaknya ke gawang lawan dalam waktu yang telah ditentukan. Oleh karena itu, robot penjaga gawang harus memiliki kemampuan untuk mencari bola, menghalau bola, menggiring bola, menendang bola, serta dapat dikendalikan secara remote dan otonom, ukuran lapangan sepak bola dapat dilihat pada gambar 1.



Gambar 1. Contoh arena lapangan robot sepak bola beroda

Penelitian sebelumnya yang dilakukan oleh Dwi Yuniarto dengan judul "Implementasi Jaringan Syaraf Tiruan untuk Meningkatkan Akurasi Tendangan Robot Sepak Bola ke Gawang" [1] membahas mengenai peningkatan akurasi tendangan bola ke gawang. Namun, dalam penelitian tersebut, hanya letak gawang yang diperhitungkan, sedangkan pendeteksian keberadaan robot penjaga gawang belum dilakukan, sehingga tidak memungkinkan untuk menentukan area kosong pada gawang lawan.

2. METODE PENELITIAN

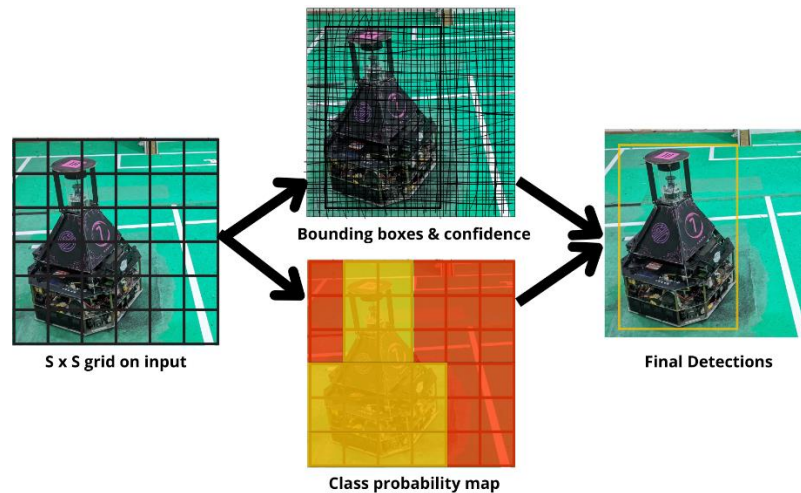
2.1 You Only Look Once (YOLO)

You Only Look Once (YOLO) adalah algoritma yang dikembangkan dari algoritma sebelumnya, yaitu *Convolutional Neural Network* (CNN). Algoritma ini dirancang khusus untuk pengenalan objek secara *real-time*, dimana YOLO mampu memproses gambar dengan kecepatan 45 frame per *second* (FPS). Deteksi dilakukan dengan memanfaatkan *classifier* atau *localizer* yang diadaptasi. Model diterapkan pada gambar di berbagai lokasi dan skala, dan area dengan skor tertinggi dianggap sebagai objek yang terdeteksi [2].

Pendekatan YOLO berbeda signifikan dari metode sebelumnya dengan menerapkan jaringan saraf tunggal pada seluruh gambar. Jaringan ini membagi gambar menjadi beberapa wilayah, kemudian memprediksi kotak



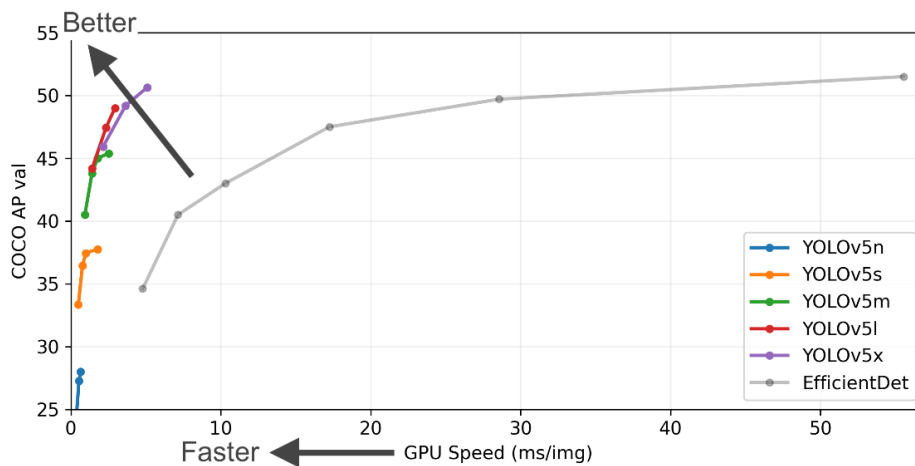
pembatas dan probabilitas di setiap wilayah. Gambar dibagi menjadi grid NxN, kemudian memprediksi *bounding box* dan tingkat kepercayaan untuk setiap *bounding box* serta probabilitas kelas secara bersamaan. Sistem deteksi objek dengan YOLO dapat dilihat pada gambar 2 [3].



Gambar 2. Deteksi Objek Yolo

Arsitektur jaringan YOLO, seperti yang ditunjukkan pada Gambar 2, terinspirasi dari model *GoogleNet* untuk klasifikasi gambar. Jaringan YOLO memiliki 24 lapisan konvolusi yang diikuti oleh 2 lapisan *fully connected*. Berbeda dengan modul awal *GoogleNet*, YOLO menggunakan lapisan reduksi 1x1 yang diikuti oleh lapisan konvolusional 3x3. Pelatihan lapisan konvolusional dilakukan menggunakan klasifikasi *ImageNet* dengan resolusi setengah (224x224 input gambar), kemudian resolusinya digandakan sehingga mampu mendeteksi objek.

Versi YOLO yang digunakan dalam penelitian ini adalah YOLOv5, karena memiliki sistem deteksi yang cepat dan akurasi yang lebih tinggi dibandingkan dengan versi YOLO sebelumnya dan metode deteksi objek lainnya. YOLOv5 juga dikenal mampu beroperasi dengan baik pada CPU maupun GPU, sehingga sangat cocok untuk aplikasi deteksi objek secara *real-time* [4].



Gambar 3. kecepatan GPU mengukur waktu inferensi rata-rata per gambar

Kecepatan GPU dalam mengukur waktu inferensi rata-rata per gambar dapat dilihat pada Gambar 3. YOLOv5 menggunakan jenis detektor *One Stage Detector*, yang mampu mengenali objek tanpa memerlukan langkah awal. Sebaliknya, *Two Stage Detector* melakukan tahap awal dengan mendeteksi area penting terlebih dahulu, kemudian mengklasifikasikannya untuk memastikan apakah ada objek di area tersebut. Keunggulan *One Stage Detector* adalah kecepatannya dalam membuat prediksi objek, sehingga sangat efektif untuk digunakan secara *real-time*.



2.2 Kamera Webcam C922

Dalam proses pengolahan citra, diperlukan sensor untuk menangkap atau mensegmentasi citra di sekitarnya yang kemudian akan diproses di PC. Pada penelitian tugas akhir ini, peneliti menggunakan dua kamera: kamera omnidirectional 360 untuk navigasi dan segmentasi citra di sekitar robot, serta kamera *webcam Logitech* tipe C922, seperti yang ditunjukkan pada Gambar 4, untuk menangkap citra di depan robot.



Gambar 4. Webcam logitech C922

Berikut adalah beberapa alasan penggunaan *webcam Logitech C922*. Video HD tampak jelas, jernih, dan mulus (1028p/30fps). Fitur HD ini sangat penting karena dalam pengolahan citra, resolusi menentukan ukuran objek yang terdeteksi. *Autofocus* cepat (hingga maksimal 7 cm dari lensa kamera) [5]. Dengan fitur autofocus, pengolahan citra menjadi lebih mudah karena kamera dapat fokus pada objek seperti robot lawan, rekan tim, dan bola. Dilengkapi dengan koreksi cahaya HD, C922 mengoptimalkan kondisi pencahayaan untuk menghasilkan gambar yang lebih terang dan berkualitas lebih baik. Fitur koreksi cahaya ini memudahkan proses pengolahan citra.

2.2 Python

Python adalah bahasa pemrograman interpretatif serbaguna dengan filosofi desain yang menekankan keterbacaan kode. *Python* dianggap sebagai bahasa yang menggabungkan kapabilitas dan kemampuan dengan sintaksis yang jelas, serta dilengkapi dengan pustaka standar yang besar dan komprehensif. Selain itu, *Python* didukung oleh komunitas besar, sehingga masalah yang muncul dapat lebih mudah diatasi melalui diskusi dengan pengguna *Python* lainnya [6].

Python menggunakan metode pemrosesan interpretatif, di mana kode program dieksekusi baris per baris secara langsung tanpa memerlukan proses kompilasi. Hal ini mirip dengan bahasa pemrograman PHP dan *JavaScript*. Saat ini, kode *Python* banyak digunakan untuk pengembangan perangkat lunak dan memiliki keunggulan karena dapat dijalankan pada berbagai sistem operasi. Berikut ini adalah platform sistem operasi yang mendukung bahasa pemrograman *Python*.

2.3 PyTorch

PyTorch adalah salah satu pustaka dalam bahasa pemrograman *Python* yang digunakan untuk komputasi *Deep Learning*. *PyTorch* menekankan fleksibilitas dan memungkinkan model *Deep Learning* diekspresikan dengan sintaks *Python*. Pendekatan dan kemudahan penggunaannya telah dimanfaatkan oleh komunitas peneliti sejak awal, dan dalam beberapa tahun setelah pustaka ini dirilis, *PyTorch* telah berkembang menjadi alat yang sangat menonjol untuk pengolahan *Deep Learning* yang digunakan secara luas.

PyTorch menyediakan struktur data inti yang disebut Tensor, yaitu array multidimensi yang mirip dengan array yang disediakan oleh pustaka NumPy. Dengan dasar ini, banyak fitur dibangun untuk mempermudah pembuatan dan menjalankan proyek, serta merancang dan melatih arsitektur baru Jaringan Syaraf Tiruan (*Neural Network*). Tensor dapat mempercepat operasi matematika (dengan asumsi perangkat keras dan perangkat lunak yang digunakan merupakan teknologi terbaru) dan *PyTorch* memiliki alat untuk melakukan pelatihan terdistribusi, proses pekerja (*worker process*) untuk memanfaatkan beberapa sumber daya pada satu mesin, serta pustaka tambahan untuk fungsi *Deep Learning* umum [7].



PyTorch yang digunakan bersama dengan pustaka *torchvision* memiliki kemampuan lebih dalam melakukan komputasi *Deep Learning* pada objek digital seperti gambar (2D) dan video (3D). Salah satu metode yang sering digunakan adalah *Convolutional Neural Network (CNN)*, yang memanfaatkan *Cortical Neurons* dalam analisis citra. *Torchvision* sangat handal untuk pemodelan menggunakan konsep CNN, karena dukungannya terhadap berbagai arsitektur *Convolutional Neural Network (CNN)*.

2.3 Compute Unified Device Architecture (CUDA)

Nvidia mengembangkan platform komputasi paralel dan antarmuka pemrograman aplikasi (API) yang dikenal sebagai CUDA (*Compute Unified Device Architecture*). CUDA memungkinkan teknik yang disebut GPGPU, yang memungkinkan insinyur dan pengembang perangkat lunak memanfaatkan unit pemrosesan grafis (GPU) yang mendukung CUDA untuk pemrosesan tujuan umum (*General-Purpose Computing on Graphics Processing Units*). Untuk eksekusi kernel komputasi, CUDA adalah platform yang menyediakan lapisan perangkat lunak yang menawarkan akses langsung ke set instruksi virtual GPU dan elemen komputasi paralel.

3. HASIL DAN PEMBAHASAN

3.1 Pemrosesan Awal

Penelitian ini memanfaatkan dataset berjumlah 4555 gambar yang terbagi menjadi tiga kelas, yakni Bola, Robot, dan Gawang. Semua gambar dalam dataset telah dianotasi, sehingga tidak perlu melakukan anotasi ulang. Contoh gambar yang digunakan dalam penelitian ini dapat dilihat pada Gambar 5 di bawah ini:



Gambar 5. Contoh gambar dataset

Dalam penelitian ini, peneliti menggunakan dataset kombinasi yang sebagian besar terdiri dari gambar bola, robot, dan gawang yang diambil secara langsung (data primer). Selain itu, beberapa gambar diunduh dari internet (data sekunder) untuk sebagian besar gambar bola, robot, dan gawang. Dataset yang dikumpulkan oleh peneliti berjumlah 4555 gambar yang terdiri dari 3644 data gambar pelatihan dan 911 data gambar validasi dapat dilihat pada table 1.

Tabel 1. Dataset

Class	Train	Validation
Bola	2000	500
Robot	1344	211
Gawang	300	200
Total data	3644 (80%)	911 (20%)

Gambar dataset pertama-tama melewati proses pra-pemrosesan, yang mencakup tahap pemisahan dataset dan penyesuaian ukuran gambar. Pemisahan dataset dilakukan untuk membagi data menjadi data latih, data validasi, dan data uji. Setiap gambar diberi label dengan nama gambar, yang dilakukan dengan menetapkan kelas nama dan kotak pembatas untuk setiap objek dalam gambar. Selain itu, penyesuaian ukuran gambar dilakukan untuk mengubah ukuran gambar menjadi 416x416 piksel. Tujuan dari penyesuaian ukuran ini adalah untuk standarisasi ukuran gambar dan penyesuaian dengan kebutuhan model.



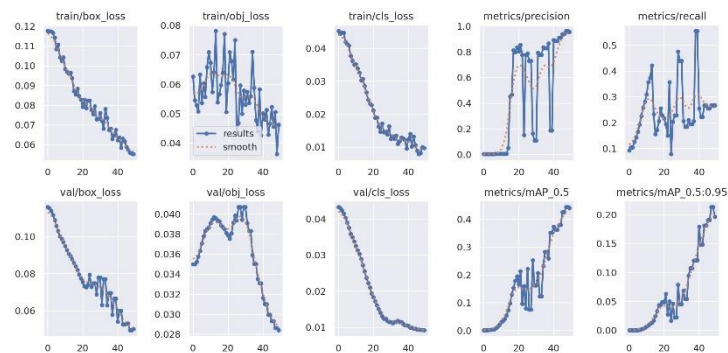
3.2 Pemrosesan Awal

Langkah berikutnya adalah melatih model menggunakan Google Colab. Berikut adalah tabel parameter pelatihan yang digunakan dalam penelitian ini dapat dilihat pada table 2 :

Tabel 2. *Training Parameters*

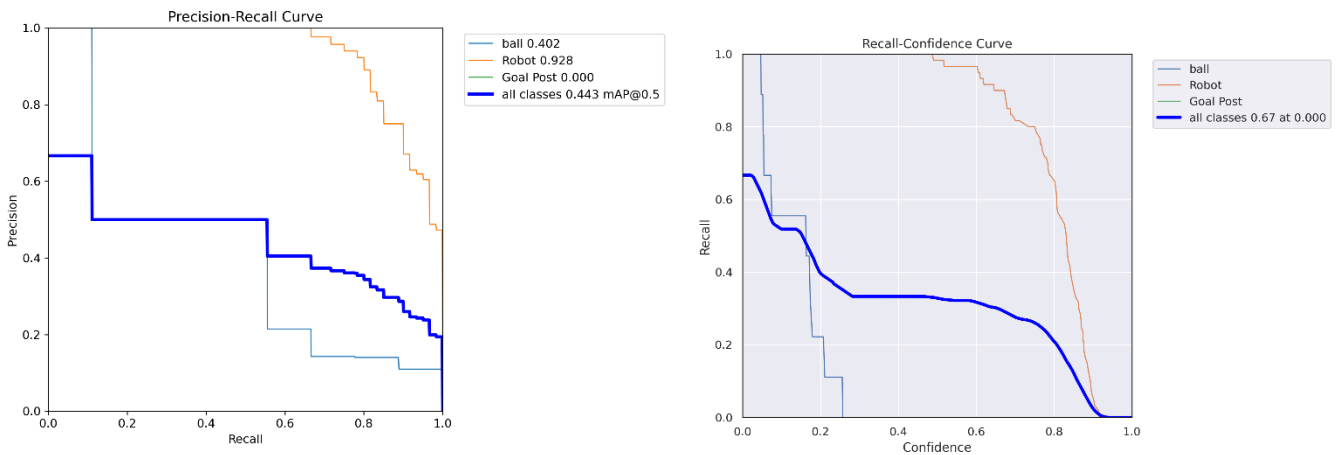
Parameter	Nilai
Image Size	416x416
Batch Size	16
Epochs	500
Lr0 & lrf	0.01
Momentum	0.937
Weight_decay	0.0005

Pelatihan mencapai kondisi optimal ketika iterasi melebihi 250 pada grafik kerugian pelatihan (train/cls_loss) dan kerugian validasi (val/cls_loss). Berikut adalah hasil kinerja pelatihan secara menyeluruh dalam penelitian ini:



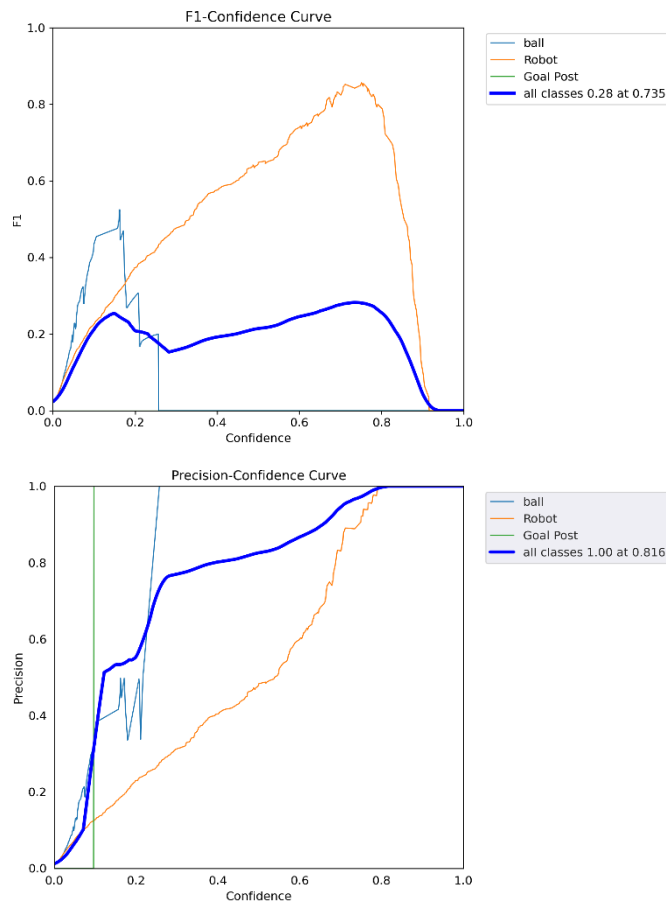
Gambar 6. Hasil Evaluasi Data Pelatihan

Hasil pelatihan sistem deteksi kebakaran menunjukkan presisi sebesar 0,919 relatif terhadap nilai recall. Rata-rata nilai recall tertinggi adalah 0,97 pada tingkat kepercayaan 0,9.



Gambar 7. Recall Value





Gambar 8. Precision Confidence Curve

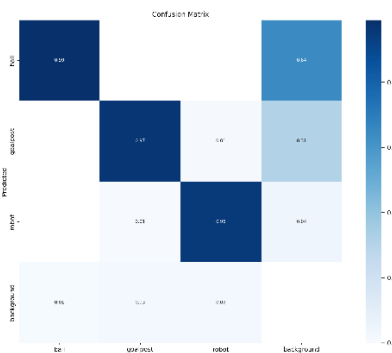
Pada Gambar 7 di atas, kurva nilai F1 dan presisi terhadap nilai keyakinan menunjukkan bahwa nilai F1 mencapai puncak rata-rata sebesar 0,89 pada nilai keyakinan sebesar 0,467. Sementara itu, nilai presisi memiliki rata-rata sebesar 1,00 pada nilai keyakinan 0,925.

Untuk menghitung akurasi dari matriks konfusi, kita dapat menggunakan persamaan berikut dengan menggunakan variabel True Positive (TP), True Negative (TN), False Positive (FP), dan False Negative (FN):

$$\text{Accuracy} = \frac{TP+TN}{TP+FP+FN+TN}$$

$$\text{Accuracy} = \frac{0.93+4.78}{0.93+4.78+0.22+0.07}$$

$$\text{Accuracy} = 0.95\%$$



Gambar 9. Confusion Matrix



Oleh karena itu, dalam penelitian ini, sistem deteksi bola, tiang gawang, robot memperoleh nilai akurasi yang moderat, yaitu 0,95 atau 95%. Deteksi jenis kendaraan menggunakan metode YOLOv5s berjalan dengan baik dan menghasilkan akurasi yang memadai. Berikut ini adalah gambar deteksi untuk setiap kelas:



Gambar 10. Deteksi Objek

4. KESIMPULAN

Penelitian ini bertujuan untuk mengembangkan sistem deteksi objek pada robot sepak bola beroda menggunakan metode YOLOv5. Dengan dataset yang terdiri dari 4555 gambar yang mencakup bola, robot, dan gawang, model dilatih menggunakan parameter-parameter tertentu, seperti ukuran gambar 416x416 piksel, batch size 16, dan jumlah epoch 1000. Hasil pelatihan menunjukkan presisi sebesar 0,919 dengan recall tertinggi mencapai 0,97 pada tingkat kepercayaan 0,9. Kurva F1 mencapai puncak rata-rata 0,89 pada keyakinan 0,467, dan presisi tertinggi adalah 1,00 pada keyakinan 0,925. Akurasi keseluruhan sistem mencapai 95%. Dengan hasil ini, terbukti bahwa metode YOLOv5 efektif untuk digunakan dalam deteksi objek pada robot sepak bola beroda, memberikan performa yang memadai.

DAFTAR PUSTAKA

- [1] TAUFIQ, R., Hayaty, N., & Ritha, N. (2023). IMPLEMENTASI METODE YOU ONLY LOOK ONCE (YOLO) PADA PENGENALAN OBJEK RIMPANG.
- [2] Yuniarto, D. (2020). Implementasi Jaringan Syaraf Tiruan untuk Meningkatkan Akurasi Tendangan Robot Sepak Bola ke Gawang.
- [3] FAIZ, R. (2023). IDENTIFIKASI WARNA LINTASAN BUOY PADA AUTONOMOUS SUBMARINE SURFACE VEHICLE (ASSV) DI EVENT KONTES KAPAL CEPAT TAK BERAWAK NASIONAL (KKCTBN) MENGGUNAKAN METODE YOU ONLY LOOK ONCE (YOLO).
- [4] J. Redmon, S. Divvala, R. Girshick, dan A. F. (2016). "Anda hanya melihat sekali: Terpadu, deteksi objek real-time,." dalam Proceedings of the IEEE conference on computer vision and pattern recognition, 779–788.
- [5] Bochkovskiy, A., Wang, C.-Y., & Liao, H.-Y. M. (2020). YOLOv4: Optimal Speed and Accuracy of Object Detection. <http://arxiv.org/abs/2004.10934>
- [6] Syahrudin, A. N., & Kurniawan, T. (2018). Input dan Output pada Bahasa Pemrograman Python. Jurnal Dasar Pemrograman Python STMIK, June 2018, 1–7.
- [7] M. Shell. (2002) IEEEtran homepage on CTAN. [Online]. Available: <http://www.ctan.org/tex-archive/macros/latex/contrib/supported/IEEEtran/>
- [8] FLEXChip Signal Processor (MC68175/D), Motorola, 1996.
- [9] "PDCA12-70 data sheet," Opto Speed SA, Mezzovico, Switzerland.
- [10] A. Karnik, "Performance of TCP congestion control with rate feedback:TCP/ABR and rate adaptive TCP/IP," M. Eng. thesis, Indian Institute of Science, Bangalore, India, Jan. 1999.

