

Deteksi Bola dan Lingkungannya Dengan Menggunakan Algoritma YOLO (You Only Look Once)

Arif Prayoga¹, M. Jasa Afroni², Anang Habibi³

Universitas Islam Malang^{1,2,3}
arifagoy329@gmail.com¹, jasaafрони@unisma.ac.id², ananghabibi@unisma.ac.id³

Abstract

In robotics, there are categories of robots that have the ability to play soccer, one of which is a wheeled soccer robot. This robot is designed to be able to detect objects in the match environment, especially ball, goal and robot objects. The development of an object detection system, including balls, goals, and robots, with a high level of accuracy is a major requirement so that the robot can identify objects precisely to carry out further tasks. This research uses Ultralytics YOLO (You Only Look Once) as an object detection system. YOLO is a deep-learning implementation that can be adopted for this purpose. YOLOv8 was used as the network model in this study, and the training process involved the Pytorch framework. During the network training stage, a mean average precision (mAP) of 99% was obtained. At the network testing stage, the system successfully detected objects by labeling "ball", "goal", and "robot" along with the bounding box on the generated image frames. The detection system developed in this research produces excellent network performance values, with an accuracy value of 98.4%, precision of 99.4%, recall of 98.9%, and F1-score of 99.2%. The network performance was evaluated using 500 image frames as a test set during the network training stage.

Keywords— robots, Deep-learning, Object Detection, YOLO.

Abstraksi

Dalam bidang robotika, terdapat kategori robot yang memiliki kemampuan untuk bermain sepakbola, salah satunya adalah robot sepakbola beroda. Robot ini dirancang untuk dapat mendeteksi objek di lingkungan pertandingan, khususnya objek bola, gawang dan robot. Pengembangan sistem deteksi objek, termasuk bola, gawang, dan robot, dengan tingkat keakuratan tinggi menjadi kebutuhan utama agar robot dapat mengidentifikasi objek secara tepat untuk menjalankan tugas-tugas selanjutnya. Penelitian ini menggunakan *Ultralytics YOLO (You Only Look Once)* sebagai sistem pendeteksi objek. YOLO merupakan implementasi *deep-learning* yang dapat diadopsi untuk tujuan ini. YOLOv8 digunakan sebagai model jaringan dalam penelitian ini, dan proses pelatihannya melibatkan *framework* Pytorch. Selama tahap pelatihan jaringan, diperoleh mean *Average Precision* (mAP) sebesar 99%. Pada tahap pengujian jaringan, sistem berhasil mendeteksi objek dengan memberikan label "bola", "gawang", dan "Robot" beserta *bounding box* pada frame gambar yang dihasilkan. Sistem deteksi yang dikembangkan dalam penelitian ini menghasilkan nilai performa jaringan yang sangat baik, dengan nilai *accuracy* sebesar 98,4%, *precision* sebesar 99,4%, *recall* sebesar 98,9%, dan *F1-score* sebesar 99,2%. Performa jaringan dievaluasi menggunakan 500 frame gambar sebagai *test set* selama tahap pelatihan jaringan.

Kata Kunci : robot, *deep-learning*, deteksi objek, YOLO.

I. PENDAHULUAN

Robot mengalami kemajuan pesat, kemajuan dalam riset perkembangan robot ini juga didukung oleh pemerintah Indonesia dengan diadakannya Kontes Robot Indonesia (KRI). KRI memungkinkan pelajar mengembangkan kreativitas dengan berbagai bidang yang diperlombakan, salah satu dari divisi KRI adalah Kontes Robot Sepak Bola Indonesia Beroda (KRSBI-B) yang telah digelar sejak 2017, KRSBI-B menjadi ajang pencarian tim

terbaik di tingkat regional dan nasional dalam pengembangan robot sepakbola beroda.

Perkembangan robot selaras dengan perkembangan kecerdasan buatan (Artificial Intelligence), di mana dalam perkembangannya AI membantu dan meniru kegiatan sehari-hari manusia, seperti misalnya melihat dan mendeteksi objek dengan menggunakan *computer vision*. Dalam penerapannya, *computer vision* memerlukan *sensing device* yang merupakan perangkat penginderaan computer dan dibantu dengan algoritma pada *interpreting device* merupakan

program yang diterapkan pada komputer untuk memproses dan menafsirkan citra [1].

Algoritma YOLO (*You Only Look Once*) memungkinkan sistem komputer memahami dan menganalisis gambar secara *real-time* dengan kecepatan tinggi. YOLO memiliki keunggulan dalam akurasi dan kecepatan dibandingkan dengan algoritma deteksi objek lainnya.

Penelitian dengan judul "Aplikasi Penghitung Jarak dan Jumlah Orang Berbasis YOLO Sebagai Protokol Kesehatan Covid-19". mencapai tingkat keakuratan pembacaan sebesar 90,04% berdasarkan perbandingan jumlah data percobaan yang berhasil dan jumlah data pengamatan pada setiap citra[2]. Penelitian YOLO lain berhasil menghitung jumlah mobil, sepeda motor, bus, dan truk yang melintas. Hasil pengujian terbaiknya adalah pada siang hari dengan tingkat akurasi sebesar 83%, 93%, dan 94%. Pengujian pada malam hari dengan kamera memiliki akurasi terendah sebesar 68%, 77%, dan 78% [3].

Penelitian terbaru dalam teknik pengolahan citra menunjukkan bahwa YOLO memiliki tingkat akurasi lebih baik, khususnya dalam deteksi objek *real-time*. Dengan ini Algoritma YOLO dipilih penulis untuk mengembangkan system deteksi yang dapat diimplementasikan pada robot sepakbola untuk mendeteksi bola, gawang, dan robot di sekitarnya. tersebut tetapi juga mempersiapkan partisipasi dalam kontes robot di masa depan[4].

II. METODE PENELITIAN

Untuk memverifikasi keefektifan pendekatan yang diusulkan, sebuah lingkungan eksperimental yang diatur menggunakan Windows 11 sebagai sistem operasi dan PyTorch sebagai *Deep Learning Framework*. YOLOV8 digunakan sebagai model jaringan dasar. Konfigurasi lingkungan percobaan diuraikan pada Tabel 1.

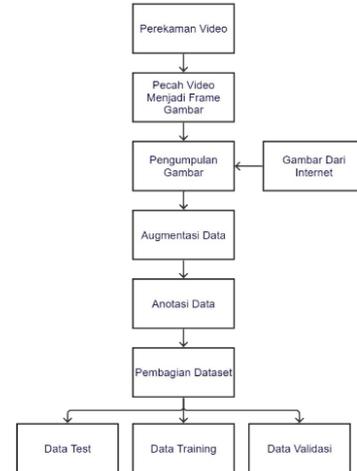
Tabel 1. Konfigurasi Lingkungan Penelitian

Konfigurasi	Versi
Sitem Operasi	Windows 11
CPU	Intel i7 12500H
GPU	RTX 3050ti
RAM	16GB
Pyhon	Python 3.10
Deep Leraning Framework	Pytorch 2.1.2
CUDA	12.1

2.1 RANCANGAN SISTEM

A. RANCANGAN DATA

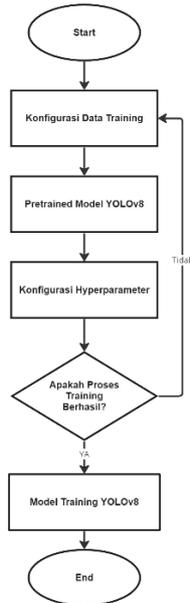
Proses pendeteksian dan pengklasifikasian objek berbasis citra digital memerlukan sejumlah besar data untuk dilatih, sehingga mendapatkan bobot optimal dalam pendeteksian dan pengklasifikasian objek. Selain itu, diperlukan pula sejumlah data lain untuk proses pengujian untuk menguji keakuratan dari metode. Perancangan data pada penelitian ini dapat dilihat pada diagram pada **Gambar 1** berikut :



Gambar 1. Blok diagram Rancangan Data

Diagram blok rancangan data pada Gambar 1 menjelaskan langkah-langkah dalam pengolahan data. Pertama, video direkam menggunakan webcam untuk mendapatkan citra bola, gawang, dan robot, dan hasilnya disimpan dalam format file video. Selanjutnya, file video dibagi menjadi beberapa frame gambar dan diberikan tambahan gambar dari sumber internet yang kemudian dilakukan proses augmentasi data. Pada tahap anotasi data, setiap citra diberi keterangan seperti koordinat, tingkat Confidence *bounding box*, dan jenis kerusakan jalan. Keterangan ini diperlukan untuk pelatihan dan pengujian model. Selanjutnya, data set dibagi menjadi training set dan test set dengan tujuan mencegah overfitting dan mengevaluasi model. Overfitting dapat terjadi jika model memiliki performa baik pada data pelatihan namun buruk pada data yang belum pernah dilihat sebelumnya (test dataset). Training set, yang merupakan bagian terbesar dari dataset, digunakan untuk proses pelatihan. Setelah pelatihan selesai, test set digunakan untuk mengevaluasi kinerja model, dengan ukuran yang umumnya adalah 66.37% / 33.3%, 75% / 25%, dan 90% / 10% untuk pembagian training set dan test set.

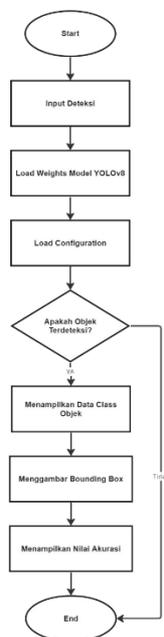
B. RANCANGAN SISTEM TRAINING



Gambar 2. Flowchart Sistem Training

Rancangan sistem training dilakukan dengan melakukan konfigurasi data training yang berisi dataset dan informasi *class* objek. Training pada custom dataset membutuhkan Pretrained model yolo v8 yang didapat dari website ultralytics, tahap selanjutnya yaitu dilakukan konfigurasi *Hyperparameter* YOLOV8, yang dilakukan dengan menyesuaikan nilai *hyperparameter* diantaranya adalah Epochs, *Batch* dan *Optimizer*. Setelah proses training berhasil, akan didapatkan file weights berekstensi ".pt".

C. RANCANGAN DETEKSI OBJEK



Gambar 3. Flowchart Deteksi Objek

Gambar 3 mengilustrasikan proses deteksi objek yang berperan penting dalam menilai tingkat keakuratan yang dihasilkan dalam penelitian ini. Input deteksi diperoleh dari data test, video pengujian, atau bahkan deteksi secara *real-time* melalui webcam. Selanjutnya, ditambahkan model YOLOv8, yang merupakan output dari proses pelatihan algoritma YOLOv8 dan ditambahkan juga file konfigurasi dataset yang memuat informasi kelas dan path dataset. Pada tahap berikutnya, algoritma deteksi objek dari YOLOv8 digunakan untuk mencocokkan gambar dari data uji dan data pelatihan. Selanjutnya, dilakukan percabangan untuk memeriksa apakah objek bola, gawang, dan robot terdeteksi. Jika tidak terdeteksi, program akan berhenti, tetapi jika terdeteksi, proses akan melanjutkan ke tahap berikutnya. Ketika objek bola, gawang, dan robot terdeteksi, label dari kelas objek, kotak pembatas, dan nilai keakuratan akan ditampilkan.

III. HASIL DAN PEMBAHASAN

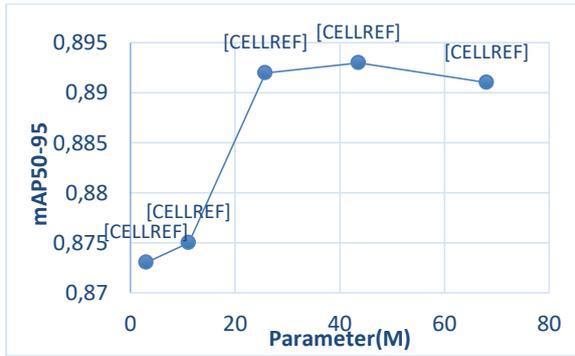
3.1 Pengujian Model Yolo

Pengujian model YOLOV8 dikombinasikan dengan beberapa *hyperparameter* dengan nilai yang telah ditentukan seperti Epoch sebesar 300, Image 640, *Batch* 8, dan *Optimizer* yang digunakan adalah SGD (*Stochastic Gradient Descent*). Model YOLOV8 yang digunakan dalam proses pengujian yaitu YOLOV8n, YOLOV8s, YOLOV8m, YOLOV8l dan YOLOV8x. Pengujian model ini bertujuan untuk membandingkan hasil terbaik dari kelima model.

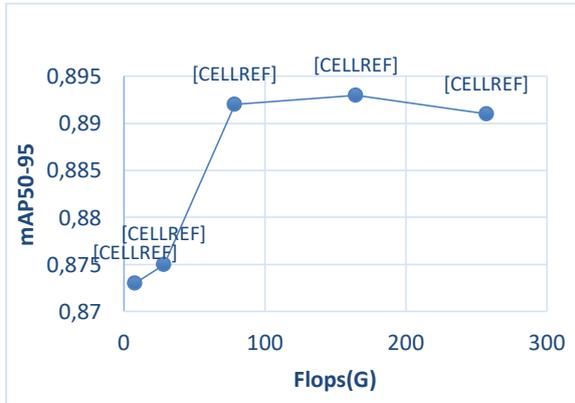
Tabel 2. Kinerja Model Yolo

Model	mAP ₅₀	mAP ₅₀₋₉₅	Param (G)	Flops (G)	Inf (ms)
YOLOV8n	0.995	0.873	3,01	8.1	5.5
YOLOV8s	0.994	0.875	11,1	28.4	12.6
YOLOV8m	0.994	0.892	25,8	78.7	24.9
YOLOV8l	0.995	0.893	43,6	164.8	44.7
YOLOV8x	0.995	0.891	68,1	257.4	65.4

Dalam implementasi ke robot soccer dibutuhkan tingkat akurasi deteksi yang tinggi dengan beban kinerja CPU serendah mungkin agar tidak membebankan sistem deteksi bola pada robot. Dapat dilihat pada Gambar 4 dan Gambar 5. yang menunjukkan grafik performa tiap model YOLOv8 pada pengujian deteksi deteksi bola, gawang dan robot terhadap tingkat akurasi.



Gambar 4. Grafik mAP dan Parameter



Gambar 5. Perbandingan mAP Flops(G)

Dari perbandingan pada Gambar 4 dan Gambar 5 dapat dilihat bahwa YOLOV8m memiliki beban kinerja CPU yang cukup rendah dengan tetap mempertahankan akurasi deteksi yang cukup tinggi sehingga model YOLOV8m dipilih sebagai model utama dalam penelitian ini. Model YOLOV8m selanjutnya akan dilakukan pengujian pada Hyperparameter dan pengaruh perubahan dataset untuk mengetahui parameter terbaik untuk meningkatkan akurasi deteksi.

3.2 PENGUJIAN HYPERPARAMETER

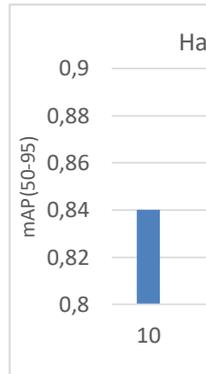
Pengujian *Hyperparameter* bertujuan untuk mengetahui konfigurasi terbaik untuk mendeteksi objek dengan tingkat akurasi yang maksimal dengan menggunakan model YOLOV8m.

A. EPOCHS

Epoch merupakan suatu parameter yang digunakan untuk menentukan jumlah berapa kali sistem melakukan pengulangan proses pelatihan (training). Epoch merupakan kondisi yang melalui pelatihan yang telah di update dalam satu siklus penuh. Pengujian ini bertujuan untuk mengetahui pengaruh nilai epoch terhadap mAP yang dihasilkan. Berikut merupakan pengujian dari nilai Epoch dapat dilihat pada **Tabel 1** Hasil pengujian nilai Epoch dibawah ini:

Tabel 3. Pengujian Epoch

Model	Epochs	mAP ₅₀	mAP ₅₀₋₉₅	Inf(ms)
YOLOV8m	10	0.984	0.84	22.1
	20	0.991	0.865	21.9
	50	0.994	0.882	23.2
	100	0.992	0.878	22.3
	200	0.995	0.891	21.8
	300	0.994	0.892	22



Gambar 6. Hasil Pengujian Epoch

B. BATCHS

Pengujian *batch size* mengacu pada besarnya ukuran *batch* (jumlah dari pembelajaran mesin atau pelatihan mesin yang digunakan dalam 1 iterasi). Pengujian nilai *batch* juga berpengaruh terhadap hasil yang diperoleh oleh sistem. Penelitian ini menggunakan dataset sebanyak 1000 frame atau gambar, pengujian *batch* yang diuji yaitu dengan ukuran 4 *batch*, 8 *batch* dan 16 *batch*. Maksud dari 4 *batch*, 8 *batch* dan 16 *batch* yaitu, untuk pengujian dengan 4 *Batch* dari dataset sebanyak 1000 gambar tersebut diambil kelipatan 4 frame hingga dataset terakhir. Sampel pertama yaitu gambar ke-1 hingga ke-4 diambil kemudian dilatih atau di training. Setelah itu dilanjutkan pada sampel kedua data ke-5 hingga data ke-8 kemudian di training, begitu seterusnya. Sama halnya dengan pengujian 8 dan 16 *batch*, dilakukan sampai data terakhir. Berikut merupakan pengujian variasi nilai *Batch* yang dilakukan pada penelitian ini dapat dilihat pada **Tabel 4** Hasil pengujian nilai *batch* dibawah ini

Tabel 4. Pengujian Batch

Model	Epochs	Batch	mAP ₅₀	mAP ₅₀₋₉₅	Inf(ms)
YOLOV8m	300	4	0.994	0.891	22.1
		8	0.986	0.891	21.8
		16	0.994	0.892	22.4

Hasil pengujian terbaik yang diperoleh dari pengujian nilai *batch* yaitu pada percobaan ketiga pada saat menggunakan 16 *Batch* diperoleh nilai mAP_{0.5-9.5} sebesar 0.892. Berdasarkan hasil yang diperoleh dapat disimpulkan bahwa nilai mAP mengalami perubahan yang cukup kecil ketika nilai *Batch* divariasikan, hal ini juga dipengaruhi oleh banyaknya data yang digunakan.

C. OPTIMIZER

Optimasi merupakan salah satu metode yang digunakan dalam training jaringan syaraf tiruan. Stochastic Gradient Descent (SGD) dan Adaptive moment estimation (Adam). *Optimizer* Stochastic Gradient Descent (SGD) merupakan suatu teknik pembelajaran suatu sistem yang melakukan

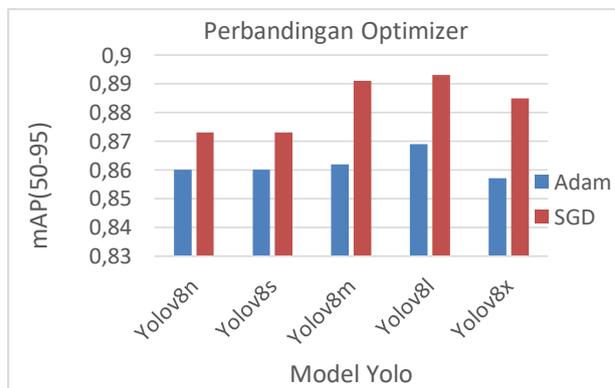
update untuk setiap satu data. Berikut merupakan hasil dari pengujian *optimizer* pada pengujian ini pada Tabel 5 Hasil pengujian *optimizer*.

1000	0.987	0.983	0.986	0.986	0.995	0.992	0.876
2000	0.994	0.985	0.982	0.995	0.994	0.99	0.882
5000	0.993	0.988	0.989	0.995	0.995	0.993	0.9

Tabel 5. Pengujian *Optimizer*

Model	Optimizer	mAP ₅₀	mAP ₅₀₋₉₅	Inf(ms)
YOLOV8n	Adam	0.986	0.86	6.2
	SGD	0.995	0.873	5.9
YOLOV8s	Adam	0.994	0.86	10.7
	SGD	0.995	0.873	11.3
YOLOV8m	Adam	0.992	0.862	22.8
	SGD	0.994	0.891	22.3
YOLOV8l	Adam	0.993	0.869	35.1
	SGD	0.995	0.893	35.4
YOLOV8x	Adam	0.99	0.857	56.3
	SGD	0.994	0.885	56.1

dapat disimpulkan bahwa proses training yang dilakukan dengan menggunakan *optimizer* SGD lebih baik dari pada menggunakan *optimizer* Adam. Hal ini dapat dilihat pada grafik perbandingan *optimizer* Adam dan SGD pada Gambar 7.



Gambar 7. Hasil Pengujian Epoch

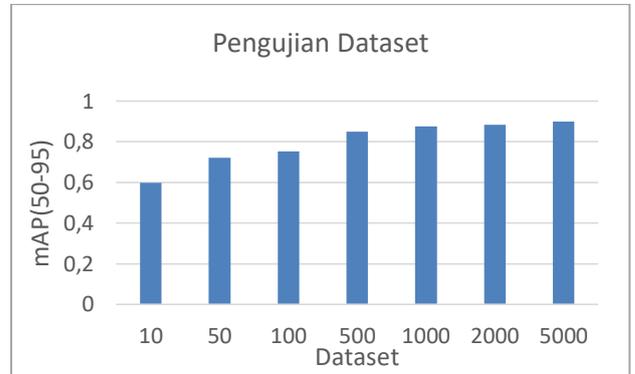
3.3 PENGUJIAN DATASET

Pengujian dataset dilakukan untuk mengetahui pengaruh banyaknya dataset dengan tingkat akurasi deteksi. Dengan model YOLOv8m dan *hyperparameter* dengan nilai epoch 300, *batch* 16 dan *optimizer* SGD, dataset yang telah di-training kemudian dilakukan pengujian yang hasilnya dapat dilihat pada Tabel 6.

Tabel 6. Pengujian Dataset

Dataset	P	R	AP			mAP ₅₀	mAP ₅₀₋₉₅
			Bola	Gawang	Robot		
10	0.849	0.748	0.853	0.831	0.844	0.843	0.597
50	0.935	0.849	0.899	0.925	0.957	0.927	0.721
100	0.931	0.877	0.922	0.927	0.972	0.94	0.753
500	0.988	0.978	0.982	0.994	0.992	0.99	0.848

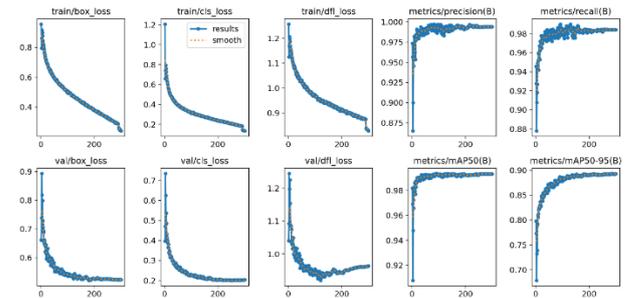
Dari percobaan tersebut dapat dilihat bahwa semakin banyak jumlah dataset untuk proses training berpengaruh terhadap peningkatan akurasi. Hal ini dapat dilihat pada grafik pada Gambar 8.



Gambar 8. Hasil Pengujian Dataset

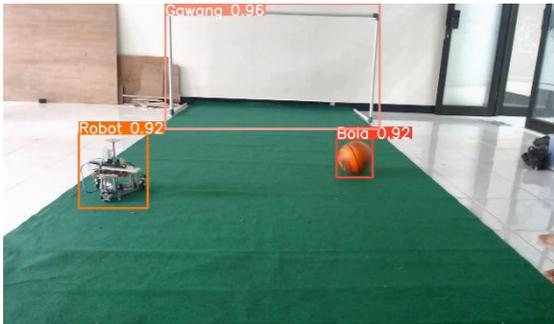
3.5 PENGUJIAN PENDETEKSIAN

Dari pengujian model, *Hyperparameter* dan dataset dilakukan training dan validasi dengan menggunakan model YOLOv8m, konfigurasi parameter yaitu 300 Epoch, 8 *batch* dan *Optimizer* SGD, dengan menggunakan dataset sejumlah 5000 menghasilkan Grafik Training dan validasi seperti pada Gambar 9.

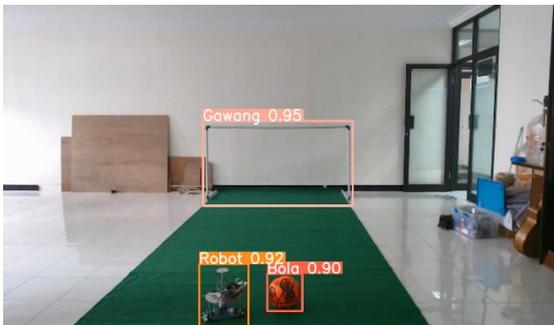


Gambar 9. Grafik Training dan Validasi

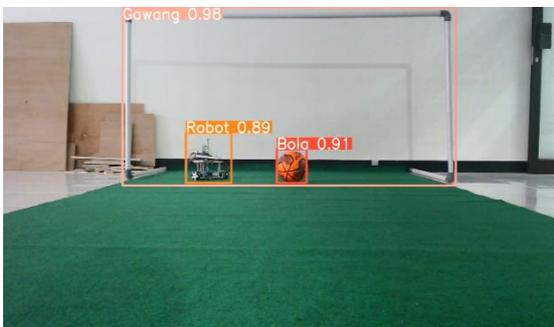
Hasil Training tersebut digunakan untuk pendeteksian objek bola, gawang dan robot dengan menggunakan beberapa frame dari video pengujian yang contohnya dapat dilihat pada Gambar 10 sampai Gambar 12.



Gambar 10. Hasil Pengujian Realtime

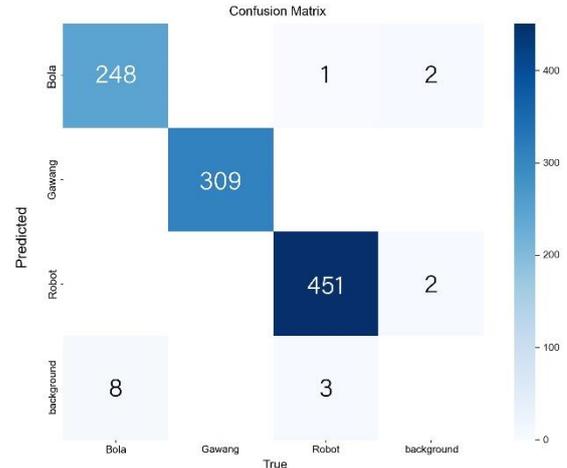


Gambar 11. Hasil Pengujian Real Time



Gambar 12. Hasil Pengujian Real Time

Frame video hasil pengujian yang didapat dengan jumlah 500 frame gambar menghasilkan citra objek yang terdapat jumlah seluruh *class* adalah 1020 citra objek bola, gawang dan robot dengan pembagian *class* bola sejumlah 256, objek gawang 309 citra dan objek dan robot sejumlah 455. Dengan menggunakan *IoU threshold* sebesar 0,5 dilakukan pengujian pendeteksian yang didapatkan hasil pengujian seperti pada *confusion matrix* Gambar 12.



Gambar 12. Confusion matrix tiap class

Dari *confusion matrix* tiap *class* pada Gambar 12 dapat dilihat bahwa kelas bola terdeteksi sebanyak 248 dengan 8 kelas bola tidak terdeteksi yang dianggap sebagai *background*, 1 objek bola terdeteksi sebagai kelas robot dan 2 objek bukan bola yang terdeteksi sebagai bola. Kelas objek gawang terdeteksi seluruhnya dengan jumlah 309. Pada kelas objek robot terdapat 451 objek yang terdeteksi benar, 3 objek tidak terdeteksi yang dianggap sebagai *background* dan 2 objek yang bukan robot terdeteksi sebagai robot. Dari data tersebut didapatkan *confusion matrix* pada seluruh kelas objek yang dapat dilihat pada Tabel 7.

Tabel 7. Confusion matrix tiap class

		TRUE	
		Positive	Negative
Predict	Positive	1008	5
	Negative	11	0

Dari Tabel 7 diatas didapati nilai *True Positive* (TP) sejumlah 1008, *True Negative*(TN) 0, *False Positive* (FP) 11 dan *False Negative* (FN) sejumlah 5. Dari data *confusion matrix* seluruh kelas objek bola, gawang dan robot ini, berikutnya data tersebut akan dilakukan pengukuran hasil kinerja YOLOv8 dengan menghitung akurasi, kalkulasi error, presisi, *recall*, *f1-score* dan mAP. dari hasil deteksi jaringan. Berikut Persamaan 1 sampai Persamaan 6 untuk mencari nilai akurasi, presisi, *recall*, *F1-score*, dan mAP.

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \quad (1)$$

$$Accuracy = \frac{1008 + 0}{1008 + 11 + 5 + 0} = \frac{1008}{1024} = 0,984$$

$$Calculation Error = 1 - Accuracy \quad (2)$$

$$Calculation Error = 1 - 0,984375 = 0,0156$$

$$Precision = \frac{TP}{TP + FP} \quad (3)$$

$$Precision = \frac{1008}{1008 + 11} = \frac{1008}{1019} = 0,995$$

$$Recall = \frac{TP}{TP + FN} \quad (4)$$

$$Recall = \frac{1008}{1008 + 5} = \frac{1008}{1013} = 0,989$$

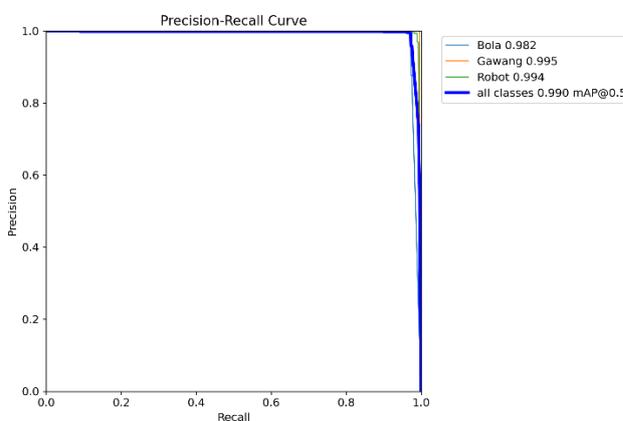
$$F1 - Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (5)$$

$$F1 - Score = 2 \times \frac{0,995 \times 0,989}{0,995 + 0,989} = 0,992$$

Presisi rata-rata (AP) adalah cara untuk meringkas kurva perolehan presisi menjadi satu nilai yang mewakili rata-rata semua presisi. AP dihitung berdasarkan Persamaan 6 berikut.

$$AP = \frac{1}{11} \times \sum_{k=0}^{k=n-1} [Recall(k) - Recall(k+1)] \times Precision \quad (6)$$

Dengan menggunakan loop yang melewati semua presisi / recall, perbedaan antara recall saat ini dan recall berikutnya dihitung dan kemudian dikalikan dengan presisi saat ini. Dan didapatkan grafik pre dan recal seperti pada Gambar 13.



Gambar 13. Grafik kurva Precision Recall

Dapat dilihat hasil dari AP dari class bola adalah 0.982, AP dari class gawang adalah 0.989 dan AP dari class robot adalah 0.995. dan didapat mean average precision (mAP) dari semua class adalah 0.993.

Hasil Pengujian

Dengan hasil dari perhitungan akurasi, presisi, recall, F1-score, dan mAP. Hasil dari perhitungan tersebut dapat dilihat pada akhir dapat dilihat pada Tabel 8.

Tabel 8. Hasil Pengujian

Pengujian	Hasil
Accuracy	98.4%
Error Calculation	1.4%
Precision	99,5%
Recall	98,9%
F1-score	99,2%
mAP@0,5	99,3%

IV. KESIMPULAN

Berdasarkan pengambilan data yang telah dilakukan pada penelitian yang berjudul "Deteksi Bola Dan Lingkungan Pada Robot Soccer Unisma Dengan Menggunakan Algoritma Yolo (You Only Look Once)" ini dapat ditarik kesimpulan sebagai berikut.

1. Penelitian ini menguji kemampuan algoritma YOLO, khususnya YOLOV8, dalam membedakan bola dan lingkungannya. Hasilnya menunjukkan bahwa YOLOV8 mampu mendeteksi dan membedakan objek bola gawang dan robot dengan baik. Algoritma ini bekerja dengan membagi objek yang terdeteksi ke dalam kelas-kelas yang berbeda, termasuk bola, gawang, dan robot.
2. Pengujian menunjukkan bahwa model YOLOV8m memberikan keseimbangan terbaik antara kinerja dan akurasi, dengan nilai mAP mencapai 89,2%. Ini berarti model YOLOV8m mampu mendeteksi objek dengan cukup akurat tanpa membebani kinerja sistem secara berlebihan. Konfigurasi parameter terbaik untuk mencapai akurasi tertinggi adalah dengan menggunakan Epoch 300, Batch size 16, dan Optimizer SGD. Konfigurasi ini menghasilkan nilai mAP yang signifikan, yaitu 98,4%. Hal ini menunjukkan bahwa pemilihan parameter yang tepat dapat meningkatkan akurasi pendeteksian secara signifikan. Penelitian ini juga menunjukkan bahwa dataset yang cukup besar, yaitu sebanyak 5000 gambar, mampu menghasilkan tingkat akurasi yang tinggi, yaitu 98,4%. Ini menunjukkan bahwa semakin banyak data yang digunakan untuk melatih model, semakin baik pula akurasi pendeteksian yang dihasilkan. Secara keseluruhan, hasil pengujian menunjukkan bahwa algoritma

YOLO mampu mendeteksi objek dengan tingkat akurasi yang tinggi.

DAFTAR PUSTAKA

- [1] A. F. Joseph Redmon, Santosh Divvala, Ross Girshick, "*Efficient convnet-based object detection for unmanned aerial vehicles by selective tile processing*," Proc. IEEE Conf. Comput. Vis. pattern Recognit., 2016, doi: 10.1145/3243394.3243692.
- [2] R. A. Fatekha, B. S. B. Dewantara, and H. Oktavianto, "Sistem Deteksi Bola pada Robot Kiper Pemain Sepakbola Beroda," J. Integr., vol. 13, no. 2, pp. 127–134, 2021, doi: 10.30871/ji.v13i2.3133.
- [3] F. Indaryanto, A. Nugroho, and A. F. Suni, "Aplikasi Penghitungan Jarak dan Jumlah Orang Berbasis YOLO Sebagai Protokol Kesehatan Covid-19," Edu Komputika J., vol. 8, no. 1, pp. 31–38, 2021, doi: 10.15294/edukomputika.v8i1.47837.
- [4] F. Rofii, G. Priyandoko, M. I. Fanani, and A. Suraji, "*Vehicle Counting Accuracy Improvement By Identity Sequences Detection Based on Yolov4 Deep Neural Networks*," Teknik, vol. 42, no. 2, pp. 169–177, 2021, doi: 10.14710/teknik.v42i2.37019.
- [5] F. F. Sanubari and R. D. Purianto, "*Deteksi Bola dan Gawang dengan Metode YOLO Menggunakan Kamera Omnidirectional pada Robot KRSBI-B*," Bul. Ilm. Sarj. Tek. Elektro, vol. 4, no. 2, pp. 76–85, 2022, doi: 10.12928/biste.v4i2.6712.
- [6] W. Zhang, "*A Fruit Ripeness Detection Method using Adapted Deep Learning-based Approach*," International Journal of Advanced Computer Science and Applications, vol. 14, (9), 2023, doi: 10.14569/IJACSA.2023.01409121.
- [7] M. Hussain, "*YOLO-v1 to YOLO-v8, the Rise of YOLO and Its Complementary Nature toward Digital Manufacturing and Industrial Defect Detection*," Machines, vol. 11, no.7, p. 677, 2023, doi: 10.3390/machines11070677
- [8] C. Dewi, A. P. Shun Chen, and H. Juli Christanto, "*YOLOv7 for Face Mask Identification Based on Deep Learning*," 2023 15th Int. Conf. Comput. Autom. Eng. ICCAE 2023, no. June, pp. 193–197, 2023, doi: 10.1109/ICCAE56788.2023.10111427.
- [9] D. Kumar et al., "*Object Detection in Adverse Object Detection in Adverse Weather for Autonomous Driving Through Data Merging and YOLOv8*," 2023, doi: 10.20944/preprints202309.0050.v1.
- [10] Hidayatullah, P. et al "*DeepSperm: A robust and real-time bull sperm-cell detection in densely populated semen video*," 2021, Computer Methods and Programs in Biomedicine, 209, p. 106302. doi:10.1016/j.cmpb.2021.106302.