

Deteksi Bola dan Gawang dengan Metode YOLO Menggunakan Kamera Omnidirectional pada Robot KRSBI-B

Farhan Fadhillah Sanubari, Riky Dwi Puriyanto
Program Studi Teknik Elektro, Universitas Ahmad Dahlan, Indonesia

INFORMASI ARTIKEL

Riwayat Artikel:

Dikirimkan 11 Oktober 2022
Direvisi 16 November 2022
Diterima 16 Desember 2022

Kata Kunci:

Robot;
KRSBI-B;
Deep Learning;
You Only Look Once;
YOLO

Penulis Korespondensi:

Riky Dwi Puriyanto,
Universitas Ahmad Dahlan,
Yogyakarta, Indonesia.
Surel: rikydp@ee.uad.ac.id

ABSTRACT / ABSTRAK

This research is a form of development of object detection capabilities on wheeled soccer robots using an omnidirectional camera with the You Only Look Once (YOLO) method, where the results show that the robot can detect more than one object, namely the ball and the goal on the green field. This study uses the KRSBI-Wheeled UAD robot using an omnidirectional camera as a tool to carry out the detection process and then uses OpenCV 4.0, Deep Learning, and a laptop as a place to create a detection model, as well as balls and goals as objects to be detected. The results obtained from this study are that the two types of YOLO models tested, namely YOLOv3 and YOLOv3-Tiny, can detect ball and goal objects in two different types of frame sizes, namely 320×320 and 416×416, which can be seen from the performance of the YOLOv3 model which has an mAP value of 76%. On the 320×320 frame and an mAP value of 87.5% in the 416×416 frame, then the YOLOv3-Tiny model has an mAP value of 68.1% in the 320×320 frame, and an mAP value of 75.5% in the 416×416 frame where the YOLOv3 model can detect both object class is much more stable compared to YOLOv3-Tiny.

Penelitian ini merupakan bentuk pengembangan dari kemampuan deteksi objek pada robot sepak bola beroda dengan menggunakan kamera omnidirectional dengan metode *You Only Look Once* (YOLO) dimana hasil penelitian menunjukkan bahwa robot dapat mendeteksi lebih dari satu objek yaitu bola dan gawang di atas lapangan hijau. Penelitian ini menggunakan robot KRSBI-Beroda UAD dengan memakai kamera omnidirectional sebagai alat untuk melakukan proses pendeteksian lalu menggunakan OpenCV 4.0, *Deep Learning*, dan laptop sebagai tempat membuat model pendeteksian, serta bola dan gawang sebagai objek yang akan dideteksi. Hasil yang didapatkan dari penelitian ini yaitu kedua jenis model YOLO yang diuji yaitu YOLOv3 dan YOLOv3-Tiny dapat mendeteksi objek bola dan gawang pada dua jenis ukuran frame yang berbeda yaitu 320×320 dan 416×416 yang dapat dilihat dari performa pada model YOLOv3 memiliki nilai mAP sebesar 76% pada frame 320×320 dan serta nilai mAP sebesar 87,5% pada frame 416×416 lalu pada model YOLOv3-Tiny memiliki nilai mAP sebesar 68,1% pada frame 320×320 serta nilai mAP sebesar 75,5% pada frame 416×416 yang dimana model YOLOv3 dapat mendeteksi kedua kelas objek jauh lebih stabil dibandingkan dengan model YOLOv3-Tiny.

This work is licensed under a [Creative Commons Attribution-Share Alike 4.0](https://creativecommons.org/licenses/by-sa/4.0/)



Sitasi Dokumen ini:

F. F. Sanubari and R. D. Puriyanto, "Deteksi Bola dan Gawang Dengan Metode YOLO Menggunakan Kamera Omnidirectional Pada Robot KRSBI-B," *Buletin Ilmiah Sarjana Teknik Elektro*, vol. 4, no. 2, pp. 76-85, 2022, DOI: [10.12928/biste.v4i2.6712](https://doi.org/10.12928/biste.v4i2.6712).

1. PENDAHULUAN

Image processing pada sistem *computer vision* saat ini sedang berkembang sangat pesat dan mulai diterapkan di berbagai bidang. Hal ini dapat dilihat dari berbagai macam riset yang berfokus pada bidang tersebut. Riset-riset tersebut memunculkan beberapa algoritma yang dapat digunakan dalam melakukan *image processing*. Salah satunya adalah *Deep Learning*. *Deep learning* merupakan salah satu bagian dari *Machine learning* yang memungkinkan algoritma sistem mampu untuk belajar dan berkembang sendirinya hanya dengan data yang disediakan dan pengalaman yang dialami tanpa peran manusia yang signifikan [1][2].

Dari waktu ke waktu metode ini juga dikembangkan sehingga kemampuannya jauh lebih baik dari sebelumnya. Seiring dengan berkembangnya *deep learning*, muncul lah berbagai model baru pada ranah pengenalan dan pendeteksian objek pada *computer vision*, contohnya seperti *convolutional Neural Network* (CNN) terus *Region based Convolutional Network* (R-CNN) [3][4], *Spatial Pyramid Pooling Network* (SPP-Net), *faster R-CNN* [5][6][7][8][9] dan *You Only Look Once* (YOLO) [10][11].

YOLO merupakan algoritma pendeteksian objek yang diciptakan oleh Joseph Redmond di tahun 2015. Algoritma ini dapat mendeteksi secara *real time* berdasarkan CNN. Algoritma ini akan membagi citra ke dalam grid berukuran $s \times s$ yang kemudian pada tiap grid akan memprediksi *bounding box* serta peta kelas masing-masing grid. Apabila pada satu grid terprediksi objek, maka pada grid tersebut akan diprediksi *bounding box* yang mengelilingi objek tersebut. Nilai *confidence* akan dihitung pada masing-masing *bounding box* yang kemudian akan diseleksi berdasarkan nilai yang didapat [12][13].

Pada YOLOv1, dengan 24 lapisan konvolusi (*convolutional layer*) yang diikuti oleh 2 lapisan yang terhubung penuh (*fully connected layer*), sistem sudah dapat mendeteksi objek namun beberapa objek kecil tidak dapat terdeteksi oleh sistem [14][15][16]. Lalu untuk meningkatkan kemampuan deteksinya, maka dibuatlah YOLOv2 yang merupakan hasil peningkatan dari YOLOv1 dengan meningkatkan *batch normalization*, menaikkan resolusi klasifikasi, serta menggunakan 19 *convolution layer* dengan 5 *max pooling layer* [17][18][19]. YOLOv3 merupakan peningkatan dari YOLOv2 dengan menggunakan 53 *convolutional layer* dengan *filters* 3×3 dan 1×1 sehingga menghasilkan kemampuan deteksi yang jauh lebih akurat dibandingkan dengan versi sebelumnya [20][21][22]. Namun pada YOLOv3 memerlukan device yang canggih agar mendapatkan kecepatan frame yang sesuai. Berdasarkan masalah itu, dibuatlah YOLOv3-Tiny yang merupakan algoritma sederhana dari YOLOv3. Algoritma ini bekerja dengan mengurangi *depth* pada *convolutional layer* yang menyebabkan kecepatan deteksi pada algoritma ini jauh lebih cepat dibandingkan dengan YOLOv3 namun akurasi deteksi pada algoritma ini akan berkurang. Metode ini sudah banyak diaplikasikan untuk tujuan tertentu, contohnya seperti deteksi penggunaan masker [23][24], deteksi bagasi yang terbengkalai [25], deteksi jenis-jenis spesies ikan [26], deteksi kendaraan [27][28], serta masih banyak lagi pengaplikasian metode ini.

Penggunaan metode YOLO pada robot kontes sepak bola beroda masih sangat minim. Hal ini disebabkan oleh kurang tajamnya tangkapan citra pada kamera *omnidirectional*. Contohnya jika objek bola sudah sedikit menjauhi robot, dibandingkan kamera biasa, pada kamera *omnidirectional* akan menampilkan objek bola tersebut semakin mengecil sehingga sedikit susah untuk dideteksi. Metode YOLO juga sering menggunakan laptop atau komputer dengan spesifikasi yang cukup canggih sehingga membuat metode ini jarang digunakan pada kontes robot sepak bola beroda Indonesia.

Pada penelitian sebelumnya, metode yang sudah pernah dilakukan untuk mendeteksi bola dan gawang menggunakan kamera *omnidirectional* adalah pendeteksian menggunakan metode *radial search line* [29][30], [31]. Selain itu juga metode yang bisa digunakan juga untuk mendeteksi bola dan gawang pada kamera *omnidirectional* adalah dengan menggunakan metode *color filtering HSV* dan menggunakan metode *scan lines* [32]. Pada deteksi bola dan gawang, hal yang menjadi permasalahan deteksi adalah ketika objek atau robot bergerak. Sistem deteksi yang digunakan diwajibkan untuk dapat selalu mendeteksi objek-objek tersebut.

Penelitian ini bertujuan untuk mengembangkan pendeteksian objek pada kamera *omnidirectional* yang digunakan di dalam robot KRSBI-B untuk bisa mendeteksi bola dan gawang lalu mengamati hasil pendeteksian dan membandingkan kemampuan deteksi antara model YOLOv3 dan YOLOv3-Tiny.

2. METODE

Pada sistem akan menggunakan *deep learning* dengan metode YOLO (*You Only Look Once*) menggunakan kamera *omnidirectional* pada robot kontes robot sepak bola beroda (KRSBI-B). Metode YOLO yang digunakan adalah YOLOv3 dan YOLOv3-Tiny yang kemudian akan dibandingkan hasil dari deteksi kedua metode tersebut.

2.1. Perancangan Perangkat Keras dan Software

Perancangan pada robot KRSBI-B terdapat dua bagian perancangan, yang pertama yaitu bagian fisik robot, dan yang kedua *image processing* yang fokus pada pendeteksian objek bola dan gawang pada kamera

omnidirectional menggunakan algoritma YOLO. maka dari itu penelitian ini akan menggunakan *omnidirectional camera system* sebagai teknik pengambilan data citra digital.

2.1.1. Perancangan Perangkat Keras

Perancangan pada perangkat keras terbagi atau dua bagian yaitu bagian fisik robot dan bagian vision robot. Robot yang digunakan untuk bahan penelitian ini adalah robot KRSBI-B Universitas Ahmad Dahlan. Berikut ini adalah ukuran yang digunakan:

- a) Ukuran Maksimum Robot: 50×50 cm
- b) Tinggi Maksimum Robot: 70 cm

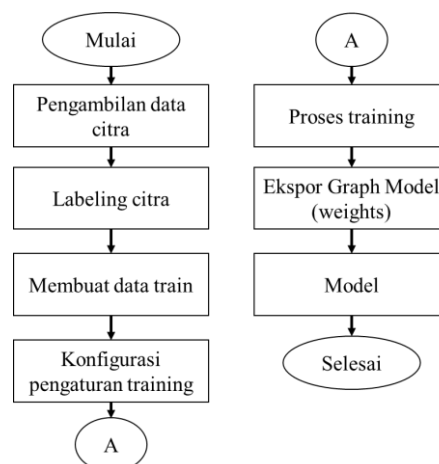
Ukuran tersebut mengikuti dari peraturan yang ada pada kontes robot sepak bola beroda disertai dengan kamera omnidirectional yang terdiri dari kamera *Logitech*, akrilik tabung dan lensa cembung. Pemilihan lensa akan mempengaruhi proses pendeteksian pada penelitian ini. Semakin bagus kamera dan lensa yang digunakan, maka hasil tangkapan citra akan jauh lebih baik dan lebih jernih. Maka dari itu pemilihan lensa dan kamera yang digunakan merupakan aspek yang sangat penting pada penelitian ini. Bentuk fisik untuk robot yang digunakan dalam penelitian ditunjukkan pada [Gambar 1](#).



Gambar 1. Bentuk fisik robot

2.1.2. Rencana Implementasi Software

Untuk mendeteksi objek yang berupa bola dan gawang, maka laptop akan membaca inputan kamera Logitech yang menggunakan usb 2.0, kemudian hasil inputan tersebut akan diproses menggunakan library YOLO. Sebelum melakukan implementasi model pada software, peneliti terlebih dahulu melakukan pelatihan gambar dengan menggunakan struktur atau konfigurasi dari *library*. Adapun proses latihan yang diimplementasikan dengan diagram alir seperti pada [Gambar 2](#).



Gambar 2. Diagram alir proses pelatihan

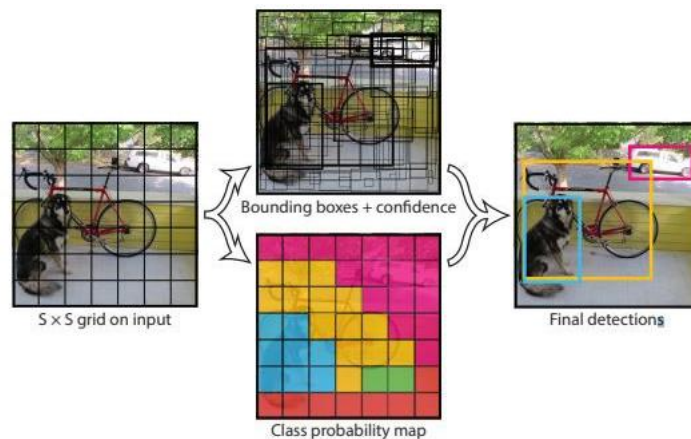
Proses pelatihan gambar dimulai dari pengambilan gambar yang akan dijadikan data citra, kemudian dilakukan pelabelan pada masing-masing gambar yang telah diambil sebelumnya, kemudian nilai pelabelan gambar yang disimpan dalam file .txt tersebut akan dipersiapkan menjadi data *train*, setelah dilakukan konfigurasi pengaturan training, lalu dilakukan proses training sesuai dengan pengaturan sebelumnya. Setelah

selesai akan didapatkan *graph model* atau dataset pendeteksian baru dengan format file *.weights*. Setelah itu dilakukan proses *modelling* dengan program utama dari YOLO. Pada penelitian ini, pembagian *dataset* yang digunakan adalah 7000 *dataset training* dan 1000 *dataset validation*.

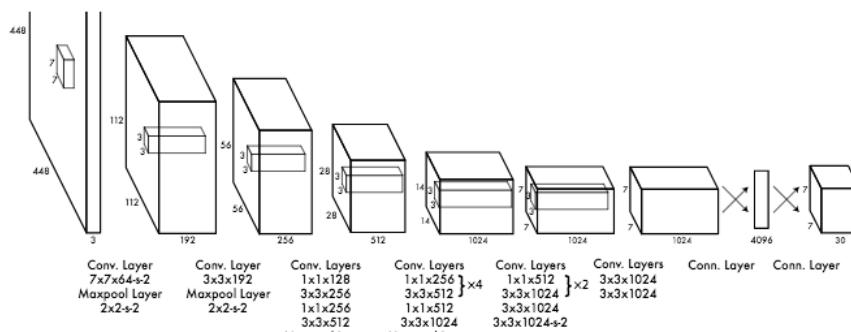
2.2. Algoritma YOLO

YOLO menerapkan jaringan saraf tunggal ke gambar penuh. Jaringan ini membagi gambar menjadi daerah dan memprediksi kotak pembatas dan probabilitas untuk setiap wilayah. Kotak pembatas ini diberi bobot oleh probabilitas yang diprediksi [23]. Metode ini menerapkan jaringan saraf tunggal ke gambar penuh. Jaringan ini membagi gambar menjadi daerah dan memprediksi kotak pembatas dan probabilitas untuk setiap wilayah. Kotak pembatas ini diberi bobot oleh probabilitas yang diprediksi.

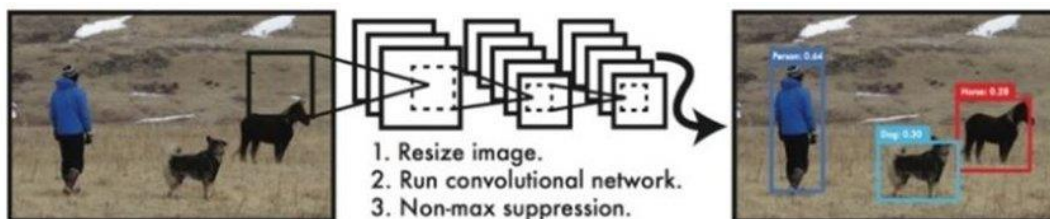
Jaringan deteksi YOLO memiliki 24 lapisan konvolusi (convolutional layer) yang diikuti oleh 2 lapisan yang terhubung penuh (fully connected layer). Beberapa lapisan konvolusi menggunakan lapisan reduksi 1x1 sebagai alternatif dalam mengurangi kedalaman feature maps yang diikuti oleh 3x3 lapisan konvolusional (convolutional layer). Seperti yang ditunjukkan pada Gambar 3, Gambar 4, dan Gambar 5.



Gambar 3. Proses pengolahan data pada algoritma YOLO



Gambar 4. Arsitektur YOLO



Gambar 5. Sistem deteksi YOLO

Pada penelitian kali ini akan digunakan 2 jenis algoritma YOLO yang berbeda untuk mendeteksi objek bola dan gawang, yaitu YOLOv3 dan YOLOv3-Tiny.

1. YOLOv3

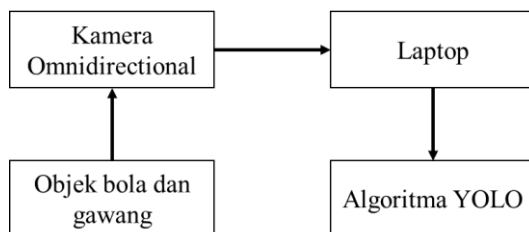
YOLOv3 adalah implementasi *deep learning* dengan menghubungkan input (citra asli) dengan output. Algoritma ini menggunakan Darknet-53 sebagai bagian terpenting untuk melakukan proses ekstraksi pada fitur. Pada arsitektur YOLOv3 ini *convolutional layer* yang digunakan adalah *convolutional 1x1* dan *convolutional layer 3x3* untuk melakukan ekstraksi pada fitur.

2. YOLOv3-Tiny

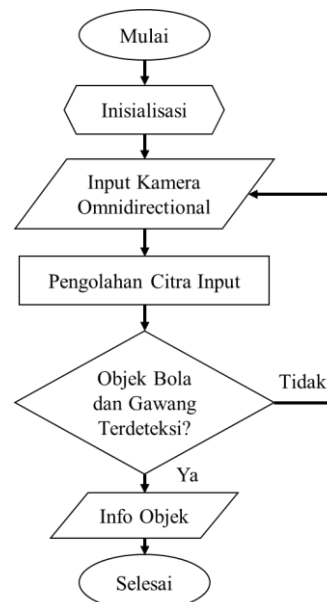
YOLOv3-Tiny merupakan algoritma sederhana dari YOLOv3. Algoritma ini bekerja dengan mengurangi *depth* pada *convolutional layer* yang menyebabkan kecepatan deteksi pada algoritma ini jauh lebih cepat dibandingkan dengan YOLOv3 namun akurasi deteksi pada algoritma ini akan berkurang.

2.3. Sistem Pengujian

Pada penelitian ini akan dilakukan pengujian pendeteksian dengan membandingkan antara kecepatan deteksi menggunakan 2 jenis metode YOLO. Pada pengujian ini akan digunakan objek bola berwarna orange dan gawang berwarna putih untuk dideteksi di atas lapangan berwarna hijau. Metode yang akan digunakan pada penelitian ini yaitu metode deep learning dengan menggunakan algoritma YOLOv3 dan YOLOv3-Tiny. Penelitian ini akan memperlihatkan perbandingan kemampuan deteksi YOLOv3 dan YOLOv3-Tiny. Diagram blok serta diagram alir dari sistem pengujian ini ditunjukkan pada [Gambar 6](#) dan [Gambar 7](#).



Gambar 6. Diagram Blok Sistem



Gambar 7. Flowchart Sistem

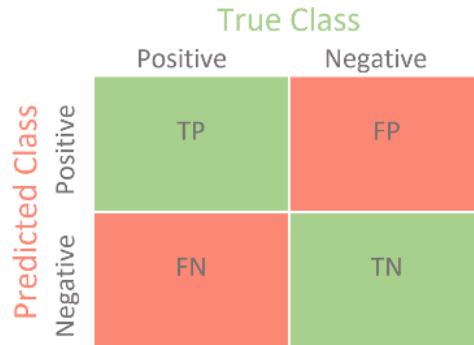
Pada proses *image processing* gambar input dari kamera omnidirectional yang di tangkap secara *real time* akan ditransformasikan menjadi gambar lain dengan mengacu pada *dataset* yang sudah didapatkan pada proses training sebelumnya, sehingga citra output akan berisi informasi berupa nama objek yaitu bola dan gawang.

3. HASIL DAN PEMBAHASAN

Pada penelitian ini akan dipaparkan hasil dari pengujian performa dan deteksi pada model YOLOv3 dan YOLOv3-Tiny.

3.1. Pengujian Performa

Pada pengujian kali ini akan menggunakan persamaan confusion matrix. Confusion matrix adalah metode pengukuran performa untuk masalah klasifikasi machine learning yang ditunjukkan pada [Gambar 8](#) dengan TP adalah *True Positive*, yaitu jumlah data yang bernilai positif dan diprediksi dengan benar oleh sistem sebagai positif. FP adalah *False Positive*, yaitu jumlah data yang bernilai negatif tetapi diprediksi sebagai positif oleh sistem. TN adalah *True Negative*, yaitu jumlah data yang bernilai negatif dan diprediksi dengan benar oleh sistem sebagai negatif. FN adalah *False Negative*, yaitu jumlah data yang bernilai positif tetapi diprediksi sebagai negatif oleh sistem.



Gambar 8. Confusion matrix

Berdasarkan nilai TP, FP, TN, dan FN, dapat diperoleh persamaan pengujian performa sebagai berikut.

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \quad Classification\ Error = 1 - Accuracy$$

$$Precision = \frac{TP}{TP + FP} \quad Recall = \frac{TP}{TP + FN}$$

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

$$Average\ Precision = \frac{1}{11} \times \sum Recall_i Precision(Recall_i)$$

Accuracy adalah tingkat akurasi model pendeteksian saat mendeteksi objek yang diteliti. Error Classification adalah nilai error yang didapatkan dari proses klasifikasi objek. Precision adalah nilai presisi yang didapatkan dari perbandingan nilai positif yang diprediksi dengan benar oleh sistem. Recall adalah rasio dari total keseluruhan data positif yang diklasifikasikan dengan benar kemudian dibagi dengan total sampel positif. F1 Score bertujuan untuk menghitung kombinasi dari presisi dan recall. Average Precision adalah perhitungan yang didapatkan dari perhitungan. Recall dan Precision yang selanjutnya dihitung menggunakan grafik Interpolated Precision.

3.2. Hasil Pengujian Deteksi Objek

Pada penelitian kali ini, proses pengujian deteksi akan dilakukan dengan menggunakan dataset validasi yang berjumlah 1000 dataset dengan pembagian seperti yang dapat dilihat pada [Tabel 1](#).

Tabel 1. Pembagian *dataset* validasi

| Objek | Jumlah <i>Dataset</i> |
|-----------------|-----------------------|
| Bola | 150 |
| Gawang | 150 |
| Bola dan Gawang | 700 |

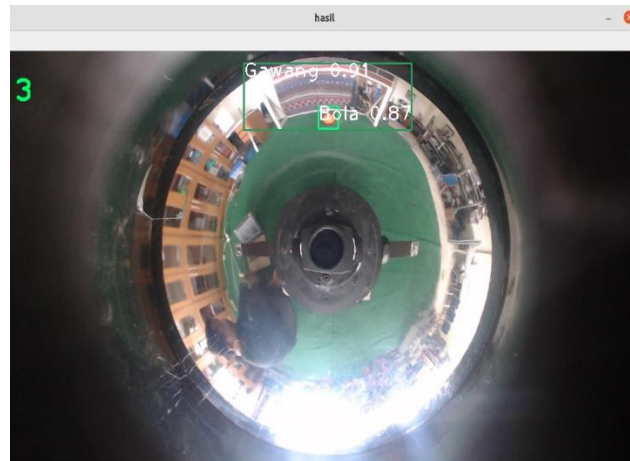
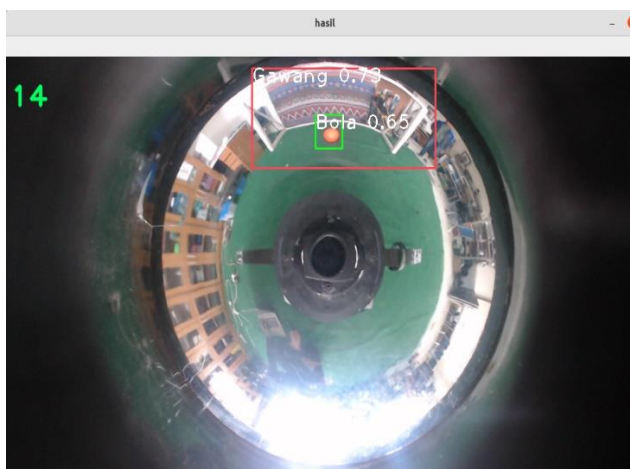
Dengan menggunakan IoU threshold sebesar 0,5 akan didapatkan hasil pengujian berikut dengan menggunakan 2 ukuran frame yang berbeda, yaitu 320×320 dan 416×416. Hasil deteksi objek dapat dilihat pada [Tabel 2](#) dan [Tabel 3](#) serta pada [Gambar 9](#), [Gambar 10](#), [Gambar 11](#), dan [Gambar 12](#).

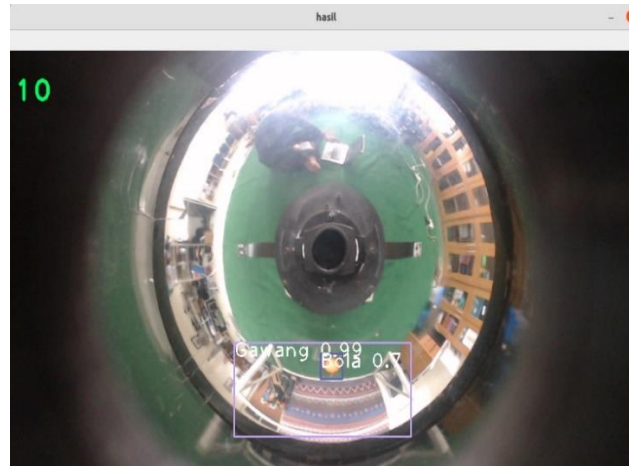
Tabel 2. Hasil pengujian deteksi pada frame 320×320

| Confusion Matrix | YOLOv3 | | YOLOv3-Tiny | |
|------------------|--------|--------|-------------|--------|
| | Bola | Gawang | Bola | Gawang |
| TP | 728 | 609 | 648 | 536 |
| FP | 35 | 160 | 81 | 76 |
| TN | 150 | 150 | 150 | 150 |
| FN | 87 | 81 | 121 | 238 |

Tabel 3. Hasil pengujian deteksi pada frame 416×416

| <i>Confusion Matrix</i> | YOLOv3 | | YOLOv3-Tiny | |
|-------------------------|--------|--------|-------------|--------|
| | Bola | Gawang | Bola | Gawang |
| TP | 786 | 778 | 720 | 600 |
| FP | 4 | 56 | 50 | 60 |
| TN | 150 | 150 | 150 | 150 |
| FN | 60 | 16 | 80 | 190 |

**Gambar 9.** Pengujian YOLOv3 pada frame 320×320**Gambar 10.** Pengujian YOLOv3 pada frame 416×416**Gambar 11.** Pengujian YOLOv3-Tiny pada frame 320×320



Gambar 12. Pengujian YOLOv3-Tiny pada frame 416×416

3.3. Hasil Pengujian Kinerja

Setelah mendapatkan hasil pengujian deteksi dari kedua ukuran frame, selanjutnya adalah mengalkulasi data yang sudah ada dengan persamaan sebelumnya sehingga didapatkan hasil pengujian performa seperti yang dapat dilihat pada Tabel 4 dan Tabel 5. Kedua hasil pengujian tersebut didapatkan dari hasil kalkulasi data pengujian deteksi yang dilihat dari besarnya nilai *confident score* yang dibatasi dengan nilai *IoU Threshold* yang sudah ditentukan sebelum dilakukan proses pengujian deteksi objek.

Tabel 4. Hasil pengujian performa pada frame 320×320

| Load Model | YOLOv3 | YOLOv3-Tiny |
|----------------------|--------|-------------|
| Accuracy | 81,8% | 74,2% |
| Error Classification | 18,1% | 25,8% |
| Precision | 87,3% | 88,2% |
| Recall | 88,7% | 77,1% |
| F1-score | 88,2% | 82,1% |
| mAP@0,5 | 76% | 68,1% |

Tabel 5. Hasil pengujian performa pada frame 416×416

| Load Model | YOLOv3 | YOLOv3-Tiny |
|----------------------|--------|-------------|
| Accuracy | 93,2% | 81% |
| Error Classification | 6,8% | 19% |
| Precision | 96,3% | 92,2% |
| Recall | 95,5% | 83% |
| F1-score | 95,8% | 87,1% |
| mAP@0,5 | 87,5% | 75,5% |

4. KESIMPULAN

Pada penelitian ini memberikan hasil pendeteksian menggunakan model YOLOv3 dan YOLOv3-Tiny pada robot KRSBI-B menggunakan kamera *omnidirectional*. *Dataset* yang digunakan adalah 7000 untuk training dan 1000 untuk validasi dan didapatkan hasil pengujian performa pada kedua jenis model menggunakan 2 jenis ukuran frame didapatkan hasil mAP pada setiap percobaan YOLOv3 untuk kedua frame jauh lebih besar dibandingkan dengan YOLOv3-Tiny. Kedua model pada pengujian deteksi langsung dapat mendeteksi objek bola dan gawang dengan baik. Penelitian ke depannya diharapkan dapat menghubungkan sistem deteksi metode YOLO atau metode lain yang termasuk dalam deep learning dengan program penggerak robot agar dapat diimplementasikan dalam strategi pergerakan robot.

UCAPAN TERIMA KASIH

Terima kasih untuk semua pihak-pihak yang telah membantu peneliti dalam menjalankan proses penelitian ini. Peneliti berharap agar penelitian ini dapat berguna bagi banyak orang dan bisa lebih dikembangkan lagi.

REFERENSI

- [1] N. O'Mahony *et al.*, "Deep Learning vs. Traditional Computer Vision," In *Science and Information Conference*, pp. 128-144, 2019, https://doi.org/10.1007/978-3-030-17795-9_10.

- [2] B. Liu, Y. Zhang, D. He and Y. Li, "Identification of Apple Leaf Diseases Based on Deep Convolutional Neural Networks," *Symmetry*, vol. 10, no. 1, p. 11, 2017, <https://doi.org/10.3390/sym10010011>.
- [3] Z. He and L. Zhang, "Multi-adversarial faster-rcnn for unrestricted object detection," In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 6668-6677, 2019, <https://doi.org/10.48550/arXiv.1907.10343>.
- [4] B. Cheng *et al.*, "Revisiting RCNN: On Awakening the Classification Power of Faster RCNN," In *Proceedings of the European conference on computer vision (ECCV)*, pp. 453-468, 2018, <https://doi.org/10.48550/arXiv.1803.06799>.
- [5] C. Eggert, S. Brehm, A. Winschel, D. Zecha and R. Lienhart, "A Closer Look: Small Object Detection in Faster R-CNN," *2017 IEEE International Conference on Multimedia and Expo (ICME)*, pp. 421-426, 2017, <https://doi.org/10.1109/ICME.2017.8019550>.
- [6] H. Jiang and E. Learned-Miller, "Face Detection with the Faster R-CNN," *2017 12th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2017)*, pp. 650-657, 2017, <https://doi.org/10.1109/FG.2017.82>.
- [7] M. N. Chaudhari, M. Deshmukh, G. Ramrakhiani and R. Parvatikar, "Face Detection Using Viola Jones Algorithm and Neural Networks," *2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA)*, pp. 1-6, 2018, <https://doi.org/10.1109/ICCUBEA.2018.8697768>.
- [8] J. Li *et al.*, "Facial Expression Recognition with Faster R-CNN," *Procedia Computer Science*, vol. 107, pp. 135-140, 2017, <https://doi.org/10.1016/j.procs.2017.03.069>.
- [9] P. R. Sihombing and A. M. Arsani, "Comparison of Machine Learning Methods in Classifying Poverty in Indonesia in 2018," *Jurnal Teknik Informatika (JUTIF)*, vol. 2, no. 1, pp. 51-56, 2021, <https://doi.org/10.20884/1.jutif.2021.2.1.52>.
- [10] J. Du, "Understanding of Object Detection Based on CNN Family and YOLO," In *Journal of Physics: Conference Series*, vol. 1004, no. 1, p. 012029, 2018, <https://doi.org/10.1088/1742-6596/1004/1/012029>.
- [11] C. Liu, Y. Tao, J. Liang, K. Li and Y. Chen, "Object Detection Based on YOLO Network," *2018 IEEE 4th Information Technology and Mechatronics Engineering Conference (ITOEC)*, pp. 799-803, 2018, <https://doi.org/10.1109/ITOEC.2018.8740604>.
- [12] X. Han, J. Chang and K. Wang, "Real-Time Object Detection Based on YOLO-v2 for Tiny Vehicle Object," *Procedia Computer Science*, vol. 183, pp. 61-72, 2021, <https://doi.org/10.1016/j.procs.2021.02.031>.
- [13] Q. Wu and Y. Zhou, "Real-Time Object Detection Based on Unmanned Aerial Vehicle," *2019 IEEE 8th Data Driven Control and Learning Systems Conference (DDCLS)*, pp. 574-579, 2019, <https://doi.org/10.1109/DDCLS.2019.8908984>.
- [14] C. -C. Wang, H. Samani and C. -Y. Yang, "Object Detection with Deep Learning for Underwater Environment," *2019 4th International Conference on Information Technology Research (ICITR)*, pp. 1-6, 2019, <https://doi.org/10.1109/ICITR49409.2019.9407797>.
- [15] W. Fang, L. Wang and P. Ren, "Tinier-YOLO: A Real-Time Object Detection Method for Constrained Environments," in *IEEE Access*, vol. 8, pp. 1935-1944, 2020, <https://doi.org/10.1109/ACCESS.2019.2961959>.
- [16] S. Khalili and A. Shakiba, "A Face Detection Method via Ensemble of Four Versions of YOLOs," *2022 International Conference on Machine Vision and Image Processing (MVIP)*, pp. 1-4, 2022, <https://doi.org/10.1109/MVIP53647.2022.9738779>.
- [17] J. Redmon and A. Farhadi, "YOLO9000: better, faster, stronger," In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7263-7271, 2017, <https://doi.org/10.48550/arXiv.1612.08242>.
- [18] J. Sang *et al.*, "An Improved YOLOv2 for Vehicle Detection," *Sensors*, vol. 18, no. 12, p. 4272, 2018, <https://doi.org/10.3390/s18124272>.
- [19] J. Zhang, M. Huang, X. Jin and X. Li, "A Real-Time Chinese Traffic Sign Detection Algorithm Based on Modified YOLOv2," *Algorithms*, vol. 10, no. 4, p. 127, 2017, <https://doi.org/10.3390/a10040127>.
- [20] J. Redmon and A. Farhadi, "Yolov3: An Incremental Improvement," *arXiv preprint*, vol. 1804, p. 02767, 2018, <https://doi.org/10.48550/arXiv.1804.02767>.
- [21] L. Zhao and S. Li, "Object Detection Algorithm Based on Improved YOLOv3," *Electronics*, vol. 9, no. 3, p. 537, 2020, <https://doi.org/10.3390/electronics9030537>.
- [22] A. Mujahid *et al.*, "Real-Time Hand Gesture Recognition based on deep learning YOLOv3 model," *Applied Sciences*, vol. 11, no. 9, p. 4164, 2021, <https://doi.org/10.3390/app11094164>.
- [23] R. Liu and Z. Ren, "Application of Yolo on Mask Detection Task," *2021 IEEE 13th International Conference on Computer Research and Development (ICCRD)*, pp. 130-136, 2021, <https://doi.org/10.1109/ICCRD51685.2021.9386366>.
- [24] X. Jiang, T. Gao, Z. Zhu and Y. Zhao, "Real-Time Face Mask Detection Method Based on YOLOv3," *Electronics*, vol. 10, no. 7, p. 837, 2021, <https://doi.org/10.3390/electronics10070837>.
- [25] T. Santad, P. Silapasupphakornwong, W. Choensawat and K. Sookhanaphibarn, "Application of YOLO Deep Learning Model for Real Time Abandoned Baggage Detection," In *2018 IEEE 7th Global Conference on Consumer Electronics (GCCE)*, pp. 157-158, 2018, <https://doi.org/10.1109/GCCE.2018.8574819>.
- [26] K. Priyankan and T. G. I. Fernando, "Mobile Application to Identify Fish Species Using YOLO and Convolutional Neural Networks," In *Proceedings of International Conference on Sustainable Expert Systems*, pp. 303-317, 2021, https://doi.org/10.1007/978-981-33-4355-9_24.
- [27] J. Zhu *et al.*, "MME-YOLO: Multi-Sensor Multi-Level Enhanced YOLO for Robust Vehicle Detection in Traffic Surveillance," *Sensors*, vol. 21, no. 1, p. 27, 2020, <https://doi.org/10.3390/s21010027>.
- [28] B. Benjdira, T. Khurshed, A. Koubaa, A. Ammar and K. Ouni, "Car Detection Using Unmanned Aerial Vehicles:

- Comparison Between Faster R-CNN and YOLOv3,” In *2019 1st International Conference on Unmanned Vehicle Systems-Oman (UVS)*, pp. 1-6, 2019, <https://doi.org/10.1109/UVS.2019.8658300>.
- [29] W. J. Chaplin, H. M. Cegla, C. A. Watson, G. R. Davies and W. H. Ball, “Filtering Solar-Like Oscillations for Exoplanet Detection in Radial Velocity Observations,” *The Astronomical Journal*, vol. 157, no. 4, p. 163, 2019, <https://doi.org/10.3847/1538-3881/ab0c01>.
- [30] G. Buldgen *et al.*, “CORALIE Radial Velocity Search for Companions Around Evolved Stars (CASCADES),” *Astronomy & Astrophysics*, vol. 657, 2022, <https://doi.org/10.1051/0004-6361/202040079>.
- [31] R. B. Hamarsudi, I. K. Wibowo and M. M. Bachtiar, “Radial Search Lines Method for Estimating Soccer Robot Position Using an Omnidirectional Camera,” In *2020 International Electronics Symposium (IES)*, pp. 271-276, 2020, <https://doi.org/10.1109/IES50839.2020.9231901>.
- [32] G. C. Felbinger, P. Götttsch, P. Loth, L. Peters and F. Wege, “Designing Convolutional Neural Networks Using a Genetic Approach for Ball Detection,” In *Robot World Cup*, pp. 150-161, 2018, https://doi.org/10.1007/978-3-030-27544-0_12.

BIOGRAFI PENULIS



Farhan Fadhillah Sanubari merupakan alumni Program Studi Teknik Elektro, Fakultas Teknologi Industri Universitas Ahmad Dahlan, Yogyakarta.



Riky Dwi Puriyanto merupakan dosen tetap di Program Studi Teknik Elektro, Fakultas Teknologi Industri, Universitas Ahmad Dahlan, Yogyakarta.