

---

## IMPLEMENTASI SISTEM OBJECT TRACKING UNTUK MENDETEKSI DUA OBJEK BERBASIS DEEP LEARNING

### Wahyu Muldayani

Fakultas Teknik, Program Studi Teknologi Rekayasa Elektronika  
Universitas Jember

Email: [wahyumuldayani.teknik@unej.ac.id](mailto:wahyumuldayani.teknik@unej.ac.id)

### Sumardi

Fakultas Teknik, Program Studi Teknik Elektro  
Universitas Jember

Email: [sumardi@unej.ac.id](mailto:sumardi@unej.ac.id)

### Arizal Mujibtamala N.I

Fakultas Teknik, Program Studi Teknologi Rekayasa Elektronika  
Universitas Jember

Email: [arizal.tamala@unej.ac.id](mailto:arizal.tamala@unej.ac.id)

### Muhammad Firza

Fakultas Teknik, Program Studi Teknik Elektro  
Universitas Jember

Email: [181910201077@mail.unej.ac.id](mailto:181910201077@mail.unej.ac.id)

### Taufan Firmansyah

Fakultas Teknik, Program Studi Teknik Elektro  
Universitas Jember

Email: [191910201039@mail.unej.ac.id](mailto:191910201039@mail.unej.ac.id)

### ABSTRAK

Kontes Robot Indonesia (KRI) adalah kompetisi penggambaran, perencanaan, dan pembuatan rekayasa dalam bidang robotika. Salah satu divisi yang dilombakan yaitu Kontes Robot Sepak Bola Beroda (KRSBI Beroda). Salah satu strategi pertandingan untuk memenangkan pertandingan yaitu saling umpan antar robot. Jadi robot diharuskan dapat melakukan identifikasi mana kawannya. Untuk melakukan tracking bola dan kawan dibutuhkan sebuah sistem pendeteksian objek. Pada penelitian ini, akan dikembangkan sistem tracking bola dan pendeteksian robot kawan dengan berbasis Deep Learning. Metode Deep Learning yang digunakan yaitu metode CNN (Convolutional Neural Network). Pada penelitian ini akan menggunakan kamera omnidirectional dan kamera *webcam* Logitech yang masing-masing akan digunakan untuk proses deteksi objek bola dan kawan. Pendeteksian objek yang dilakukan menggunakan algoritma YOLO yang arsitekturnya terdiri dari 24 layer kovolusi, 4 layer max pooling, dan 2 layer fully connected. Pendeteksian objek yang dilakukan menggunakan algoritma YOLO yang sebelumnya sudah di training menggunakan model YOLOv5s dengan jumlah dataset 1500 gambar bola dan 600 gambar kawan. Dari hasil training yolov5 dihasilkan pembacaan yang bagus dengan Mean Average Precision (mAP) mencapai 0.985, presisi sebesar 0.971 dan nilai recall mencapai 0.981. Dari hasil pengujian yang dilakukan sistem dapat mendeteksi bola mencapai jarak 700cm dan mendeteksi kawan mencapai jarak 900cm. Ketika intensitas cahaya terlalu rendah pendeteksian yang dilakukan tidak stabil. Robot JR EVO berhasil melakukan tracking bola maupun kawan.

**Kata kunci:** CNN, Deep Learning, Deteksi Objek, Kamera, Robot, YOLO

## ABSTRACT

*The Indonesian Robot Contest (KRI) is an engineered drawing, planning, and manufacturing in the field of robotics. One of the divisions being contested is the Wheeled Football Robot Contest (KRSBI Wheeled). One of the match strategies to win the match is mutual feedback between robots. So the robot is required to be able to provide assistance to its friends. To track the ball and friends, an object detection system is needed. In this research, a system for ball tracking and detection of friendly robots based on deep learning will be developed. The Deep Learning method used is the CNN (Convolutional Neural Network) method. In this study, an omnidirectional camera and a Logitech webcam camera will be used, each of which will be used for the ball and friend object detection process. Object detection is carried out using the YOLO algorithm whose architecture consists of 24 convolution layers, 4 max-pooling layers, and 2 fully connected layers. Object detection was carried out using the YOLO algorithm which had previously been trained using the YOLOv5s model with a total dataset of 1500 ball images and 600 friends images. From the YOLOv5 training results, good readings were obtained with a Mean Average Precision (mAP) reaching 0.985, a precision of 0.97, 1, and a recall value reaching 0.981. From the results of tests carried out, the system can detect the ball reaching a distance of 700cm detecting friends reaching a distance of 900cm. When the light intensity is too low the detection is unstable. The JR EVO robot has succeeded in tracking the ball and its friends.*

**Keywords:** CNN, Deep Learning, Object Detection, Camera, Robot, YOLO

## 1. PENDAHULUAN

Teknologi saat ini sedang dikembangkan dengan sangat pesat terutama pada bidang robotika. Meningkatnya perkembangan teknologi bidang robotika dapat dipacu dengan adanya berbagai kompetisi robotika yang selalu ada setiap tahunnya dengan jenis tema yang berbeda-beda. Salah satu kompetisi robotika khususnya di Indonesia yang setiap tahun rutin diadakan adalah Kontes Robot Indonesia (KRI) [Puspresnas]. Kontes Robot Sepak Bola Indonesia Beroda atau yang biasa disebut KRSBI Beroda adalah salah satu divisi yang diperlombakan pada Kontes Robot Indonesia (KRI). Pada divisi ini, robot harus dapat bermain sepak bola layaknya manusia dan bergerak secara autonomous tanpa bantuan kendali manusia. Robot diharuskan mencari dan melakukan identifikasi terhadap bola secara otomatis menggunakan kamera sebagai sensor pendeteksi bola. Oleh karena itu, untuk melakukan tujuan tersebut diperlukan sistem tracking bola yang dapat mengetahui posisi dari bola secara real time dimana setiap pergerakan bola akan dilacak oleh robot dan ketika bola dideteksi, maka robot akan menuju ke arah bola tersebut. Memenangkan sebuah pertandingan dalam lomba KRSBI Beroda tidak hanya diperlukan sistem tracking bola yang baik, namun dibutuhkan juga sebuah strategi yang baik. Salah satu strategi tersebut yaitu dengan sebuah kerja sama yang baik antar robot. Setiap robot harus bisa mendeteksi mana kawannya sehingga robot penyerang dapat melakukan sebuah umpan kepada robot penyerang yang lainnya yang nantinya berkesempatan untuk mencetak goal ke gawang lawan. Oleh karena itu, dibutuhkan juga pendeteksian robot kawan agar dapat menjalankan kerja sama yang baik antar robot penyerang.

Pada penelitian sebelumnya terdapat Perancangan *Object Tracking Robot* Berbasis Image Processing menggunakan Raspberry PI [Pradana]. Pada penelitian oleh Prianggodo pengolahan citra yang meliputi konversi ruang warna RGB ke HSV, color filtering, edge detector, dan circle hough transform. Pengujian yang dilakukan adalah pengaruh resolusi gambar pada kamera terhadap sistem gerak, jarak pandang robot, pengenalan warna, pengenalan bentuk bola, kecepatan waktu tracking dan intensitas cahaya minimum. Penelitian ini terdapat kekurangan yaitu intensitas cahaya saat proses pendeteksian sangat berpengaruh dimana harus tidak terlalu terang maupun gelap. Selain itu pendeteksian bola yang dilakukan berdasarkan warna bola bukan bentuk bola. Penelitian selanjutnya yaitu Pendeteksi Bola Pada Robot Penjaga Gawang Menggunakan Metode Hough Circle [Prianggodo]. Penelitian ini akan menggunakan metode Hough Circle dengan EmguCV untuk melakukan pengenalan warna dan bentuk bola. Penerapan metode ini dapat meminimalisir eror saat melakukan pendeteksian bola dan arah bola yang dating menuju penjaga gawang. Penelitian ini terdapat kekurangan yaitu ketika jarak bola diatas 600cm tidak dapat dideteksi.

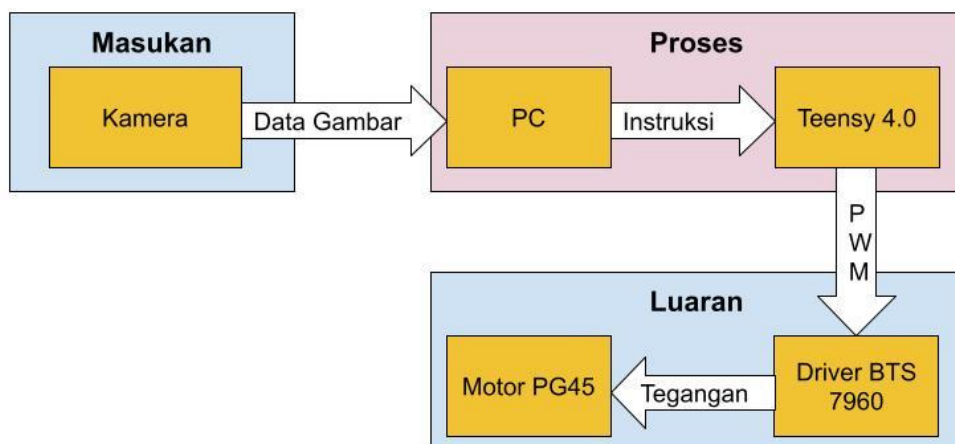
Pada penelitian kali ini akan dikembangkan sistem tracking bola dan pendeteksian robot kawan dengan berbasis Deep Learning. Deep learning atau pembelajaran mendalam adalah salah satu sub bidang yang spesifik pada Machine Learning dan merupakan cara baru dalam merepresentasikan data yang menekankan pada pembelajaran berbasis layer (lapisan) secara berturut – turut hingga bermakna [Pramestya]. Salah satu metode dari Deep Learning adalah CNN (Convolutional Neural Network). CNN adalah suatu teknik machine learning yang merupakan perluasan dari *MultiLayer Perceptron* (MLP) yang dibentuk untuk memproses data dua dimensi dalam bentuk gambar. Namun, di CNN setiap neuron diwakili dalam dua dimensi, sedangkan di MLP setiap neuron hanya memiliki satu dimensi [Eka]. Convolutional Neural Network terdiri dari lapisan output, lapisan input, dan lapisan tersembunyi. CNN berbeda dengan jaringan saraf biasa karena neuron di lapisannya disusun dalam tiga perspektif yaitu lebar, tinggi, dan kedalaman. Hal ini memungkinkan CNN untuk mengubah volume input dalam tiga dimensi menjadi volume output [Peemen]. CNN adalah arsitektur yang dapat dilatih dan terdiri dari beberapa lapisan. Setiap tahap terdiri dari tiga lapisan: lapisan konvolusi, lapisan fungsi aktivasi, dan lapisan penyatuan [Saha]. Salah satu algoritma dalam pendeteksian yaitu YOLO. Dengan menggunakan You Only Look Once atau YOLO, sistem hanya akan melihat sekali pada gambar untuk memprediksi objek apa itu dan dimana keberadaannya [Redmon]. Dengan adanya penelitian ini diharapkan dapat memperbaiki penelitian sebelumnya dan menjadi perkembangan teknologi yang dapat meningkatkan kinerja dari robot KRSBI Beroda Universitas Jember yaitu JR EVO.

## 2. METODOLOGI PENELITIAN

Pada bagian ini dijelaskan metode yang digunakan serta proses jalannya penelitian.

### 2.1 Desain Perancangan Elektronika

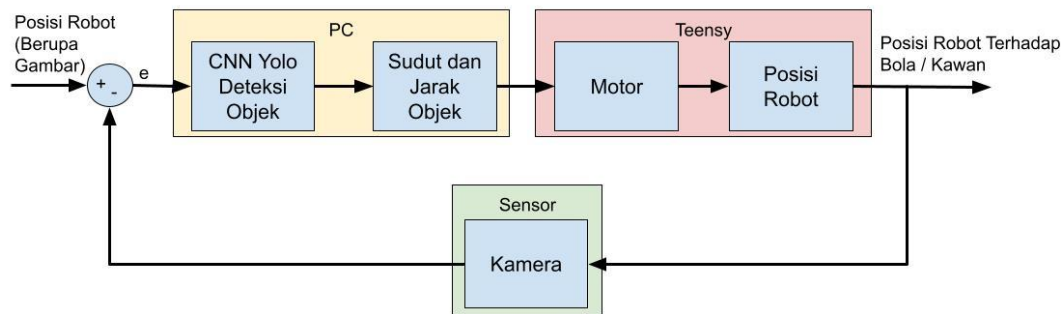
Perancangan ini merupakan perancangan elektrikal dan sistem kendali pada robot. Pada sistem elektrikal robot terdapat beberapa komponen yang digunakan antara lain Teensy 4.0, driver motor, motor DC, dan mini-PC. Pada Gambar 1 kamera akan mengirimkan data image pada mini pc lalu data akan diproses pada mini pc, kedua proses (image describing dan classification) ada pada mini pc. Setelah data selesai diproses lalu dikirimkan ke mikrokontroler untuk diteruskan kepada driver motor, dikarenakan driver motor memerlukan pin dengan fitur PWM, maka tidak bisa langsung dikontrol oleh mini pc dan harus melalui Teensy 4.0 terlebih dahulu.



Gambar 1. Diagram Blok Sistem Elektronika Kendali

## 2.2 Perancangan Sistem Kontrol Tracking

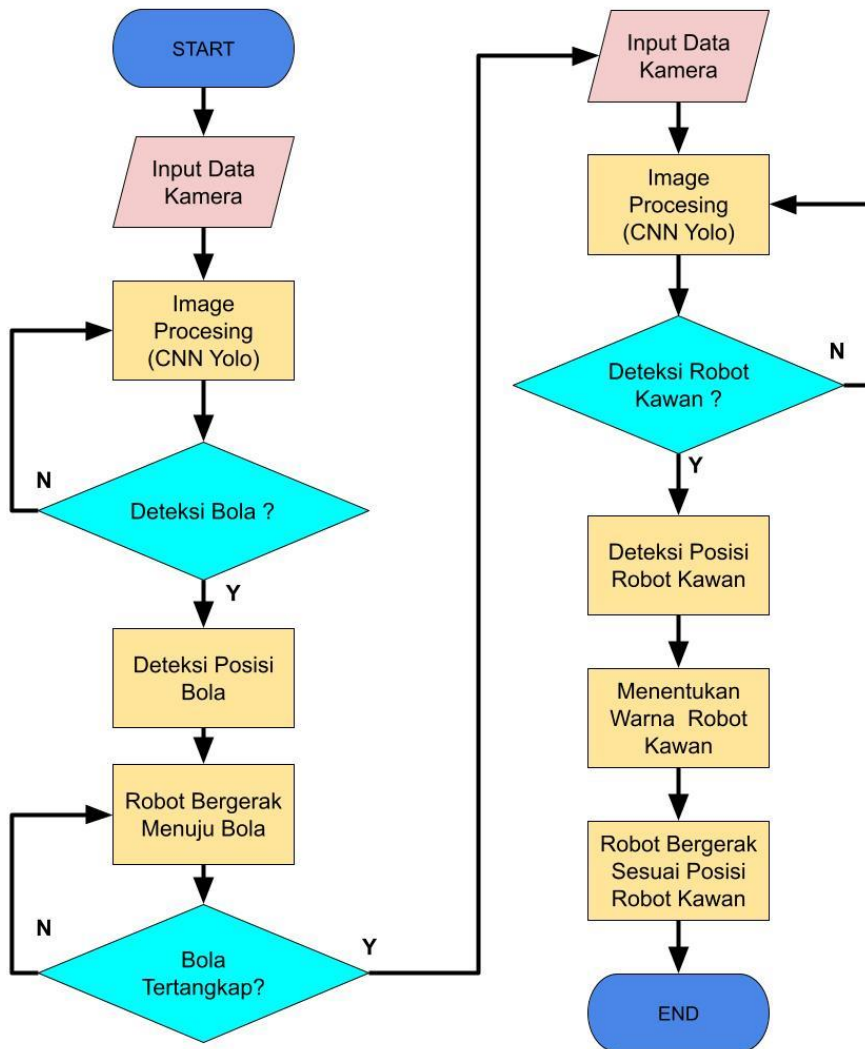
Pada bagian ini akan dijelaskan desain sistem kontrol pada robot untuk mengetahui respon robot ketika melakukan tracking terhadap bola maupun robot kawan. Gambar 2 merupakan desain sistem kontrol tracking dimana nilai set-point atau target yaitu jarak atau sudut objek. Nilai ini dapat dicapai dengan menggunakan kontrol robot dengan beberapa kondisi yaitu motor sudut, motornganan, motorngiri, dan motormanual. Dari kontrol robot tersebut nantinya akan menghasilkan nilai pwm yang akan dikirimkan ke driver motor yang akan diteruskan ke motor sehingga menghasilkan output pergerakan jarak atau sudut objek. Dalam proses tracking kamera memberikan feedback untuk mengambil gambar objek secara real time yang akan diproses oleh YOLO CNN yang nantinya akan menghasilkan nilai koordinat dari objek. Koordinat yang dihasilkan akan diolah untuk menjadi rumus jarak dan sudut. Untuk mencari jarak menggunakan rumus perbandingan antara jarak antar koordinat dengan jarak sebenarnya. Untuk mencari sudut menggunakan rumus trigonometri. Jika nilai jarak atau sudut sesuai dengan setpoint maka sistem tracking selesai.



Gambar 2. Diagram Blok Sistem Kontrol Traking

## 2.3 Flowchart Tracking Objek

Agar lebih mudah dalam mengetahui bagaimana sistem kerja tracking objek pada penelitian ini, maka diperlukan sebuah flowchart tracking objek. Alur kerja dari tracking objek dimulai dari input data gambar yang dilakukan dengan kamera. Input gambar akan melalui proses pengolahan citra untuk melakukan proses pendeteksian. Jika mendeteksi bola maka akan mendapatkan posisi bola jika tidak maka akan melakukan proses pendeteksian kembali. Setelah mendapatkan posisi bola robot bergerak menuju posisi bola. Jika bola sudah didapatkan maka akan melakukan proses input data kamera kembali jika tidak robot akan mendapatkan bola kembali. Input gambar akan melalui proses pengolahan citra lagi untuk melakukan pendeteksian robot kawan. Jika mendeteksi robot kawan maka akan mendapatkan posisi robot kawan jika tidak maka melakukan proses pendeteksian kembali. Setelah mengetahui posisi robot kawan lalu akan menentukan warna identitas kawan. Setelah itu robot akan bergerak menyesuaikan posisi dari robot kawan. Berikut flowchart tracking objek dapat dilihat pada Gambar 3.

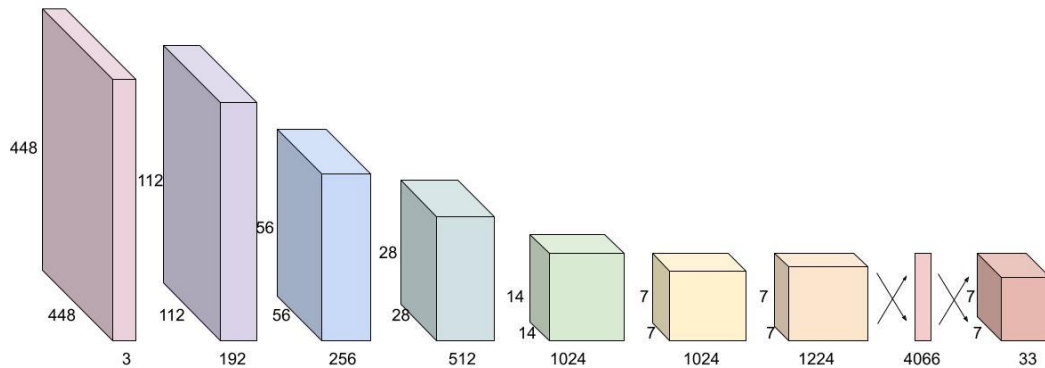


Gambar 3. Flowchart Sistem Treacking Objek

#### 2.4 Desain Arsitektur CNN

Kedalaman arsitektur mempengaruhi tingkat kemampuan jaringan. Arsitektur Convolutional Neural Network dalam penelitian ini dirancang dengan menggunakan arsitektur dari *You Only Look Once* atau yang biasa disebut YOLO. Arsitektur ini terdiri dari 24 layer konvolusi, 4 layer max pooling, dan 2 layer fully connected. Gambaran susunan arsitektur pada penelitian ini dapat dilihat pada Gambar 4.

Arsitektur penelitian ini akan menggunakan lapisan reduksi 1x1 untuk mengecilkan kedalaman dari map fitur serta diikuti lapisan konvolusi 3x3 dengan langkah (stride) 1. Lapisan konvolusi masing-masing memiliki lapisan maxpooling dengan filter 2x2 yang berfungsi untuk mengurangi resolusi fitur. Pada arsitektur YOLO, Sistem menerima gambar masukan dari bentuk (448,448,3). Ini adalah gambar dengan Panjang dan lebar 448 x 448 dan kedalaman 3, dan melalui proses jaringan konvolusi, output akan memiliki bentuk (7,7,30), di mana 7x7 adalah ukuran sel grid (S=7), 30 adalah angka kotak pembatas B dikalikan dengan jumlah kelas ditambah jumlah komponen dalam kotak B. (Bx5+C dengan B=2, C=20).



Gambar 4. Arsitektur CNN

### 2.5 Deteksi dengan YOLO

Proses mendeteksi objek, YOLO memiliki alur proses pendeteksian. Berikut diagram alir deteksi objek YOLO. Algoritma pemrograman dari pendeteksian real time yaitu pertama inialisasi library yang akan digunakan seperti library opencv, library numpy, dan lain-lain. Selanjutnya untuk library opencv memanggil kamera yang akan digunakan sebagai input dengan program cv2.VideoCapture. Cv2 Video Capture membantu dalam menangkap capture video frame per frame yang dihasilkan oleh kamera. Selain itu memasukkan input model YOLO CNN yang sebelumnya sudah di training menggunakan Google Colab. Setelah kedua input telah dipanggil maka langsung menuju main program yaitu proses pendeteksian. Jika sistem berhasil melakukan pendeteksian maka akan dilakukan perhitungan untuk mengetahui sudut dan jarak dari objek tersebut. Jika tidak berhasil mendeteksi objek maka akan kembali menuju proses pendeteksian. Setelah jarak dan sudut telah diketahui maka akan dikirim ke mikrokontroler melalui komunikasi serial. Untuk menampilkan hasil tangkapan kamera menggunakan cv2.imshow. Proses pendeteksian diawali dengan input data berupa gambar. YOLO akan membagi gambar masukan menjadi SxS petak (grid). Jika pusat objek ada di salah satu petak, maka sel petak tersebut bertanggung jawab untuk mendeteksi objek itu yang hasil yang akan didapatkan berupa kotak anchor. Setelah kotak anchor terbentuk selanjutnya melakukan perhitungan kemungkinan adanya objek. Selain itu setiap sel kotak memprediksi kotak pembatas dan nilai keyakinan untuk kotak tersebut. Setelah itu setiap sel menghitung kemungkinan probabilitas kelas kondisional. Nilai ini akan dikalikan dengan nilai keyakinan dari kotak pembatas. Nilai ini menghasilkan probabilitas kelas yang muncul di kotak dan seberapa baik kotak dalam memprediksi objek. IoU merupakan metrik pengukuran yang digunakan untuk menghitung ketepatan detektor dan pengklasifikasi objek pada set data tertentu. Ini terdiri dari dua metrik evaluasi.

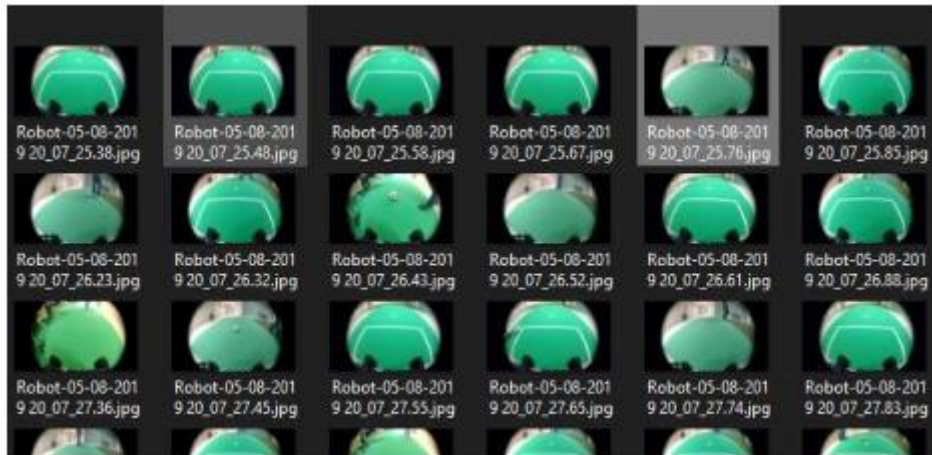
## 3. HASIL DAN PEMBAHASAN

Pada hasil dan pembahasan pada pengujian robot JR EVO yang menggunakan metode Convolutional Neural Network sebagai metode object tracking untuk mendeteksi bola dan robot kawan. Pengujian robot JR EVO dilakukan di Gedung Robotika Universitas Jember serta Laboratorium Elektronika dan Terapan Universitas Jember.

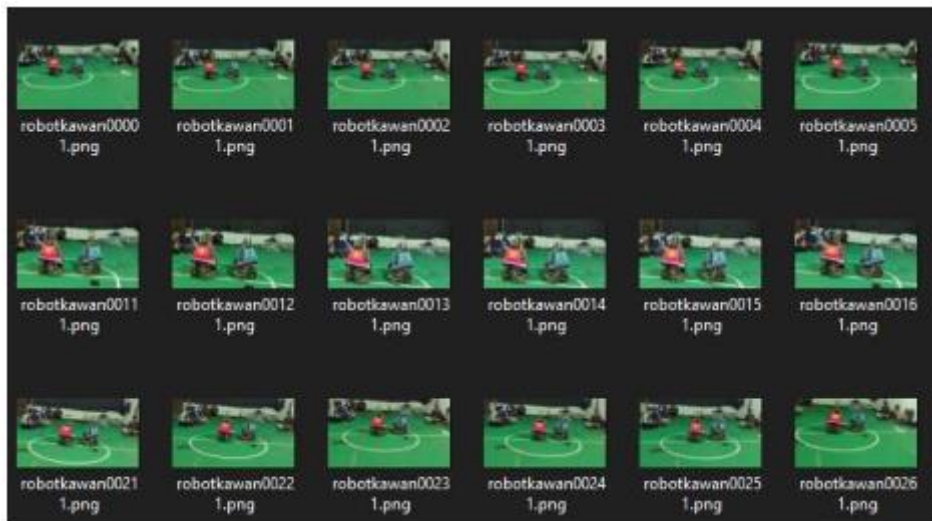
### 3.1 Akuisi dan Training Data

Akuisisi data merupakan merupakan tahap pengumpulan data objek yang pada penelitian ini berupa bola robot kawan. Data tersebut nantinya akan digunakan untuk data training dan data testing. Berikut diagram alir dari proses akuisisi dan preprocessing data. Pengumpulan dataset

dilakukan secara manual yaitu dengan cara mengambil video dari bola maupun robot kawan. Setelah dataset terkumpul, video akan diekstrak menjadi frame-frame gambar menggunakan software VLC dengan frame rate 25 fps. Dari hasil ekstrak video menghasilkan jumlah data gambar bola sebanyak 1500 gambar dan data gambar robot kawan 641 gambar ditunjukkan pada Gambar 5 dan Gambar 6.



Gambar 5. Data Training Bola



Gambar 6. Data Training Robot Kawan

Tahap selanjutnya yaitu dengan memberikan pelabelan dan *ground truth bounding box* pada citra dataset menggunakan LabelImg, yang akan disimpan dalam bentuk anotasi berformat txt. Setelah proses pelabelan, kumpulan data gambar dan hasil pelabelan yang berformat txt diupload di sebuah *tools* yaitu roboflow. Pada roboflow dilakukan tahap preprocessing yaitu proses resizing data gambar menjadi ukuran 416x416.

Proses training data menggunakan YOLOv5 yang akan dilatih menggunakan Google Colab, yang menyediakan akses gratis ke GPU yang andal dan tidak memerlukan konfigurasi. Proses training data diawali dengan menginstal environment dari YOLOv5. Environment tersebut berupa requirement yang dibutuhkan untuk menjalankan YOLOv5. Setelah itu melakukan persiapan dataset. Dataset yang akan digunakan untuk proses training adalah dataset yang di download dari

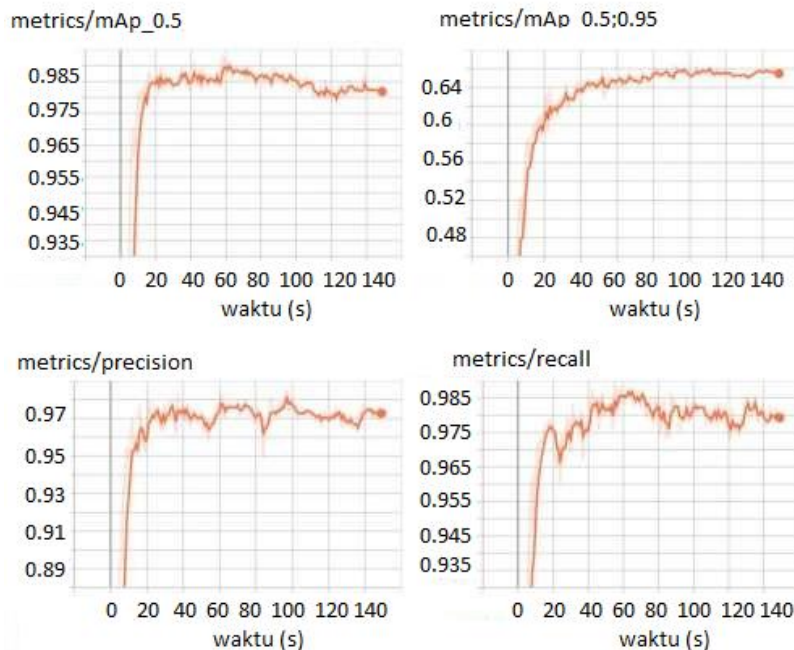
tools Roboflow. Proses selanjutnya yaitu melakukan konfigurasi dari argument model YOLOv5. Dari proses training nantinya akan muncul evaluasi performa dari YOLOv5. Proses terakhir yaitu proses inferensi data uji. Pada proses ini model akan melakukan pengujian pada data uji yang sebelumnya telah disiapkan di Roboflow. Proses ini dapat mengetahui apakah model dapat melakukan deteksi terhadap objek atau tidak. Konfigurasi argument dari model yang akan dilatih yaitu dengan menggunakan img 416, batch 16, epoch 16 dan weights yolov5s. Dari konfigurasi model tersebut terdapat beberapa argument diantaranya img, batch, epochs, dan weights. Img mendefinisikan ukuran gambar masukan. Ukuran gambar asli 1024 x 1024, lalu dikompres ke ukuran yang lebih kecil untuk membuat proses pelatihan lebih cepat. Ukuran gambar 416 x 416 adalah ukuran yang ideal untuk digunakan sebagai input gambar tanpa kehilangan banyak detail.

### 3.2 Hasil Training Data

Setelah proses training selesai akan diketahui bagaimana performa dari YOLOv5 dalam melakukan pendeteksian objek. Berikut hasil dari training data yang dilakukan dapat dilihat pada Tabel 1 hasil training data dan Gambar 5 grafik evaluasi performa.

**Tabel 1. Hasil Training Data**

No	Klasifikasi	Images	Label	P	R	mAP0.5	mAP0.5;0.95
1	Semua	426	498	0.972	0.984	0.985	0.658
2	Bola	426	249	0.945	0.959	0.968	0.454
3	Robot Biru	426	119	0.986	1	0.994	0.766
4	Robot Merah	426	130	0.985	0.992	0.994	0.755



**Gambar 7. Grafik Evaluasi Performa**

Dari Gambar 7, terdapat beberapa matrik nilai yaitu mean average precision (mAP), dengan nilai antara 0 dan 1. Semakin tinggi skor, model akan lebih akurat. Terdapat juga nilai presisi dimana memberi tahu seberapa tepat model dalam prediksi. Serta recal mengukur seberapa baik model menemukan semua nilai positif dalam kumpulan dataset. Dari data matrik dapat diketahui bahwa



nilai dari Mean Average Precision (mAP) mencapai 0.985. Terdapat juga nilai dari presisi sebesar 0.971 dan nilai recall mencapai 0.981. Ini membuktikan bahwa, dengan arsitektur YOLOv5, modelnya tidak hanya cepat, tetapi akurasi juga tinggi. Untuk masing masing nilai mAP, presisi, dan recall untuk setiap kelas dapat dilihat pada Gambar 8. Dari gambar tersebut, untuk kelas bola nilai mAP, presisi, dan recall masing masing yaitu 0.968, 0.945, dan 0.959. Untuk kelas robot biru nilai mAP, presisi, dan recall masing-masing yaitu 0.994, 0.986, dan 1. Untuk kelas robot merah nilai mAP, presisi, dan recall yaitu 0.994, 0.985, dan 0.992. Untuk YOLOv5 ini tidak ada untuk nilai akurasi. Hal ini dikarenakan fokus dari YOLOv5 yaitu seberapa yakin model dalam mendeteksi bahwa objek tersebut benar benar objek yang akan dideteksi.

### 3.3 Inferensi Data Uji

Dalam proses training data ini diperlukan proses inferensi yang digunakan untuk mengetahui apakah model dapat mendeteksi objek dengan baik.



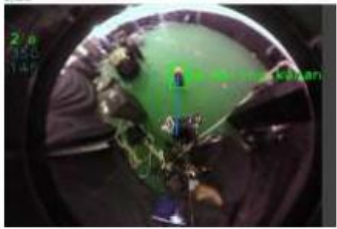
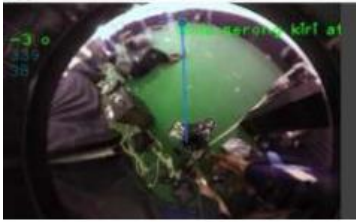
Gambar 8. Hasil Deteksi Uji

Seperti yang ditunjukkan pada Gambar 8 model memberikan hasil prediksi yang sangat mengesankan. Model akan memprediksi objek berdasarkan probabilitas untuk objek itu. Jadi tingkat akurasi pembacaan yang berbeda dipengaruhi probabilitas objek yang diprediksi oleh model.

### 3.4 Pengukuran Jarak Bola

Pada pengujian ini dilakukan dengan cara menguji sistem deteksi objek berupa bola yang telah dilatih sebelumnya menggunakan metode Convolutional Neural Network. Dari pengujian ini dapat diketahui seberapa kemampuan sistem dalam melakukan deteksi bola dengan berbagai variasi jarak.

**Tabel 2. Pengujian Jarak Bola**

Jarak	Hasil	Keterangan
100	 <p><b>Gambar 9. Jarak Bola 100</b></p>	Terdeteksi
700	 <p><b>Gambar 10. Jarak Bola 700</b></p>	Terdeteksi

Tabel 2 merupakan hasil pengujian jarak bola diatas dapat dikethui bahwa dari beberapa pengujian variasi jarak bola terhadap titik tengah kamera, jarak maksimum sistem dapat mendeteksi bola adalah 700cm atau 7 meter.

### 3.5 Pengukuran Jarak Kawan

Pada pengujian ini dilakukan dengan cara menguji sistem deteksi objek berupa robot kawan yang telah dilatih sebelumnya menggunakan metode Convolutional Neural Network. Dari pengujian ini dapat diketahui seberapa kemampuan sistem dalam melakukan deteksi robot kawan dengan berbagai variasi jarak.

**Tabel 3. Pengujian Jarak Kawan**

Jarak	Hasil	Keterangan
100	 <p><b>Gambar 11. Jarak Kawan 100</b></p>	Terdeteksi

900



Terdeteksi

**Gambar 12. Jarak Kawan 900**

Tabel 3 merupakan hasil pengujian jarak kawan diatas dapat diketahui bahwa dari beberapa pengujian variasi jarak robot kawan terhadap titik tengah kamera, jarak maksimum sistem dapat mendeteksi robot kawan adalah 900cm atau 9 meter.

### 3.6 Pengukuran Jarak Kawan

Pada pengujian ini dilakukan dengan cara menguji sistem deteksi objek berupa bola yang telah dilatih sebelumnya menggunakan metode Convolutional Neural Network. Dari pengujian ini dapat diketahui seberapa kemampuan sistem dalam melakukan deteksi bola dengan berbagai variasi tingkat intensitas cahaya.

**Tabel 4. Pengujian Intensitas Cahaya**

Intensitas	Hasil	Keterangan
8,75 lux		Terkadang terdeteksi terkadang tidak
120 lux		Terdeteksi jelas

**Gambar 13. Jarak Kawan 100**

**Gambar 14. Jarak Kawan 900**

### 3.7 Komunikasi Pada Mikrokontroller

Pada pengujian ini akan dilakukan penyesuaian data yang dikirim dari python di mini pc ke mikrokontroller. Pengujian ini bertujuan untuk menguji apakah data yang dikirim oleh python sesuai dengan data yang masuk pada mikrokontroller. Pengujian dilakukan dengan cara menghubungkan python dan mikrokontroller dengan protocol komunikasi serial. Data yang dikirim oleh python berupa kode yang nantinya akan digunakan menulis tulisan pada LCD 20x4. Kode

yang dikirim dari python yaitu nama kelas dari objek yang akan di deteksi. Ketika python mendeteksi bola maka pada LCD akan muncul kata “kita”. Ketika python mendeteksi robot berwarna magenta maka pada LCD akan muncul kata “aku”. Ketika python mendeteksi robot berwarna cyan maka pada LCD akan muncul kata “kamu”, diperlihatkan pada Gambar 15.



**Gambar 15. Tulisan Kita Tanda Deteksi Bola**

### **3.8 Pengujian Tracking Bola**

Pada pengujian ini, setelah komunikasi antara mini pc dan mikrokontroler berhasil maka robot JR EVO akan diuji coba dengan dijalankan secara langsung dengan skenario yang diawali dengan robot akan melakukan pendeteksian bola lalu akan melakukan tracking terhadap bola tersebut. Dari pengujian ini dapat diketahui apakah robot dapat melakukan pendeteksian dan tracking bola secara tepat atau tidak.

### **3.9 Pengujian Tracking Bola**

Pada pengujian ini, setelah komunikasi antara mini pc dan mikrokontroler berhasil maka robot JR EVO akan diuji coba dengan dijalankan secara langsung dengan skenario yang diawali dengan robot akan melakukan pendeteksian bola lalu akan melakukan tracking terhadap bola tersebut. Dari pengujian ini dapat diketahui apakah robot dapat melakukan pendeteksian dan tracking bola secara tepat atau tidak.

**Tabel 5. Pengujian Tracking Bola**

Percobaan	Jarak	Sudut	Keterangan
1	112	34	Berhasil Tracking Bola
2	134	-74	Berhasil Tracking Bola
3	101	-130	Berhasil Tracking Bola
4	180	151	Berhasil Tracking Bola
5	188	43	Berhasil Tracking Bola

Tabel 5 merupakan hasil pengujian tracking bola diatas dapat diketahui bahwa 5 pengujian yang dilakukan robot JR EVO mampu dalam melakukan deteksi bola dan melakukan tracking terhadap bola. Dalam pengujian yang dilakukan untuk melakukan tracking bola, selama bola masih dapat dideteksi oleh kamera maka robot dapat melakukan tracking bola.

### **3.10 Pengujian Tracking Kawan**

Pada pengujian ini, setelah komunikasi antara mini pc dan mikrokontroler berhasil maka robot JR EVO akan diuji coba dengan dijalankan secara langsung dengan skenario yang diawali

dengan robot akan melakukan pendeteksian terhadap robot kawan lalu akan melakukan tracking terhadap robot kawan tersebut. Dari pengujian ini dapat diketahui apakah robot dapat melakukan pendeteksian dan tracking robot kawan secara tepat atau tidak.

**Tabel 6. Pengujian Tracking Kawan**

Percobaan	Jarak	Sudut	Keterangan
1	150	-37	Berhasil Tracking Kawan
2	150	-41	Berhasil Tracking Kawan
3	150	45	Berhasil Tracking Kawan
4	150	20	Berhasil Tracking Kawan
5	150	43	Berhasil Tracking Kawan

Tabel 6 merupakan hasil pengujian tracking kawan diatas dapat diketahui bahwa 5 pengujian yang dilakukan robot JR EVO mampu dalam melakukan deteksi kawan dan melakukan tracking terhadap kawan. Dalam pengujian yang dilakukan untuk melakukan tracking kawan, selama robot kawan masih dapat ditangkap oleh kamera maka robot dapat melakukan tracking robot kawan.

#### 4. KESIMPULAN

Berdasarkan penelitian yang telah dilaksanakan, dan dari hasil pengujian yang telah dilakukan, maka dapat diambil kesimpulan yaitu dalam merancang sebuah sistem tracking obyek untuk mentracking bola dan robot kawan dengan baik diperlukan Convolutional Neural Network dengan menggunakan algoritma YOLO yang terdiri dari 24 layer konvolusi, 4 layer max pooling, dan 2 layer fully connected seperti pada penelitian ini. Diperlukan juga 1500 gambar training dan 215 gambar testing untuk membuat model dari pendeteksian objek menggunakan Yolov5. Data hasil pendeteksian nantinya akan dikirim ke mikrokontroler menggunakan komunikasi serial. Dari hasil training yolov5 dihasilkan pembacaan yang bagus dengan Mean Average Precision (mAP) mencapai 0.985, presisi sebesar 0.971 dan nilai recall mencapai 0.981. Performa pendeteksian menggunakan CNN dapat mendeteksi bola sejauh 700cm dan robot kawan sejauh 900cm. Namun ketika intensitas cahaya terlalu rendah pendeteksian bola dan robot kawan masih kurang stabil. Pengujian tracking bola dan kawan berhasil dilakukan.

#### DAFTAR PUSTAKA

- [1] Puspresnas. 2021. Pedoman KRI 2021. Edisi 1. Jakarta: Pusat Prestasi Nasional Kementerian Pendidikan dan Kebudayaan.
- [2] Pradana, A. W. dan D. Irmawati. 2020. *Pendeteksi warna dan bentuk bola pada robot penjaga gawang menggunakan emgucv*. Elinvo (Electronics, Informatics, and Vocational Education). 5(1):21–31.
- [3] Prianggodo, L. B. 2016. *Perancangan object tracking robot berbasis image processing menggunakan raspberry pi*. Laporan Tugas Akhir Teknik Elektronika.
- [4] Pramesty, R. H. 2018. *Deteksi dan Klasifikasi Kerusakan Jalan Aspal Menggunakan Metode YOLO berbasis Citra Digital*. Institut Teknologi Sepuluh Nopember.
- [5] Eka Putra, W. S. 2016. *Klasifikasi citra menggunakan convolutional neural network (cnn) pada caltech 101*. Jurnal Teknik ITS. 5(1).
- [6] Peemen, M., R. Shi, S. Lal, B. Juurlink, B. Mesman, dan H. Corporaal. 2016. *The neuro vector engine: flexibility to improve convolutional net efficiency for wearable vision*. Proceedings of the 2016 Design, Automation and Test in Europe Conference and Exhibition, DATE 2016. 1604–1609.

- [7] Saha, S. 2018. A Comprehensive Guide to Convolutional Neural Networks. <https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53>.
- [8] Redmon, J., S. Divvala, R. Girshick, dan A. Farhadi. 2016. *You only look once: unified, real-time object detection*. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition. 2016-Decem:779–788