

Penerapanan Model Deteksi Objek Untuk Robot Menggunakan Model SSD Di Lingkungan Simulasi ROS

Ihham Rizqyakbar¹, Chandra Kusuma Dewa²

^{1,2}Universitas Islam Indonesia; Jl. Kaliurang No.Km. 14,5, Krawitan, Umbulmartani, Kec. Ngemplak, Kabupaten Sleman, Daerah Istimewa Yogyakarta 55584

^{1,2}Jurusan Informatika, FTI UII, Yogyakarta

e-mail: ¹19523066@students.uui.ac.id, ²chandra.kusuma@uui.ac.id

Abstrak

Penyelenggaraan Kontes Robotika Indonesia (KRI) merupakan bentuk inisiatif untuk mendorong inovasi di kalangan mahasiswa dan menyiapkan mereka untuk kompetisi internasional seperti RoboCup. Tantangan dalam KRI, seperti pendeteksian objek umumnya menggunakan metode segmentasi warna karena merupakan metode yang cukup efektif dalam pendeteksian objek dilapangan, akan tetapi metode ini memiliki kelemahan, khususnya jika warna pada objek sangat jauh berbeda dari yang sudah ditentukan pada robot. Hal ini menunjukkan perlunya teknik visi komputer yang lebih andal melalui deep learning. Studi ini mengusulkan penerapan model detektor satu tembakan (single shot detector) dalam sepak bola robotik menggunakan simulasi ROS Gazebo. Model seperti MobileNet V2 menawarkan kemampuan deteksi objek real-time yang penting untuk robotika otonom. Dengan fokus pada objek seperti bola dan gawang, penelitian ini bertujuan mengembangkan sistem robotik yang mampu mengidentifikasi dan mengukur jarak terhadap objek secara akurat berdasarkan karakteristik visual tanpa menggunakan sensor sebagai alat bantu lainnya.

Kata kunci— robotika, sistem operasi robot, deteksi objek, simulasi gazebo, pengukuran jarak

Abstract

The organization of the Indonesian Robotics Contest (KRI) is an initiative to encourage innovation among university students and prepare them for international competitions such as RoboCup. Challenges in KRI, such as object detection, generally use color segmentation methods because they are quite effective in detecting objects on the field, but this method has weaknesses, especially if the color of the object is very different from what has been determined for the robot. This indicates the need for more reliable computer vision techniques through deep learning. This study proposes the application of a single shot detector model in robotic soccer using ROS Gazebo simulation. Models like MobileNet V2 offer real-time object detection capabilities that are crucial for autonomous robotics. Focusing on objects such as balls and goals, this research aims to develop a robotic system capable of accurately identifying and measuring distances to objects based on visual characteristics without using other auxiliary sensors.

Keywords— robotics, robot operating system, object detection, gazebo simulation, distance measurement

1. PENDAHULUAN

Perkembangan industri otomasi telah melakukan revolusi industri dan mengubah perekonomian di kawasan Asia Tenggara secara cepat melalui teknologi yang disruptif seperti otomasi menggunakan robot. Sebagai negara yang akan melakukan transisi menuju negara maju pada tahun 2045 nanti, Indonesia sudah seharusnya memanfaatkan peluang otomasi untuk terus meningkatkan daya saing.

International Federation of Robotics telah menempatkan Indonesia di antara yang terendah dibandingkan negara lain di Asia Tenggara dalam hal penggunaan teknologi otomasi dan robotika pada tahun 2019. Dibandingkan negara tetangga seperti Malaysia dengan perbandingan tiga puluh empat robot per 10.000 karyawan di tahun yang sama [1]. Untuk mendorong pemanfaatan robotika perlu adanya inisiatif inovasi di sektor publik atau swasta serta peningkatan anggaran

pemerintah untuk penelitian dan pengembangan di industri robotika .

Salah satu upaya yang dilakukan pemerintah Indonesia untuk melakukan pengembangan robotika adalah dengan diadakannya beberapa kompetisi robotika, seperti Kontes Robot Indonesia (KRI), oleh Kementerian Pendidikan, Kebudayaan, Riset, dan Teknologi setiap tahunnya. KRI yang diikuti seluruh mahasiswa pada perguruan tinggi di wilayah Indonesia yang tercatat pada Pangkalan Data Perguruan Tinggi (PDDikti), ditujukan untuk mencari kandidat-kandidat yang akan mewakili Indonesia dalam perlombaan Robocup, perlombaan tingkat Internasional, serta menjadi ajang pengembangan dan rancang inovasi mahasiswa dalam pengembangan robot [2].

Perlombaan KRI memiliki tujuh divisi yang diperlombakan, salah satu divisi yang diperlombakan adalah Kontes Robot Indonesia Beroda (KRBI-B), yang merupakan perlombaan robot yang bermain sepak bola, seperti layaknya manusia, yang bergerak menggunakan roda dan melakukan permainan secara otomatis tanpa adanya campur tangan manusia dengan memanfaatkan berbagai sensor, seperti kamera, yang digunakan sebagai alat visualisasi robot dalam melihat lingkungan sekitarnya sehingga robot dapat mendapatkan informasi seperti mengetahui jarak bola terhadap posisi robot. Penggunaan sensor kamera juga menjadi jawaban untuk salah satu tantangan pada lomba KRI, yaitu untuk mendeteksi beberapa objek, seperti bola, gawang, dan obstacle yang berupa robot musuh maupun robot teman.

Metode segmentasi warna merupakan metode yang umumnya digunakan dalam melakukan deteksi objek pada perlombaan ini, akan tetapi terdapat beberapa kelemahan metode ini. Dalam penelitian yang dilakukan Agus Darmawan bahwa sistem pengenalan objek menggunakan segmentasi warna belum mampu melakukan pengenalan terhadap objek berdasarkan bentuk dari objek itu sendiri. Hal ini membuat objek lain yang memiliki warna yang sama dapat dikenali sebagai objek target, dan distorsi citra yang dihasilkan menggunakan kamera membuat bentuk dari citra tidak sesuai dengan bentuk aslinya [3].

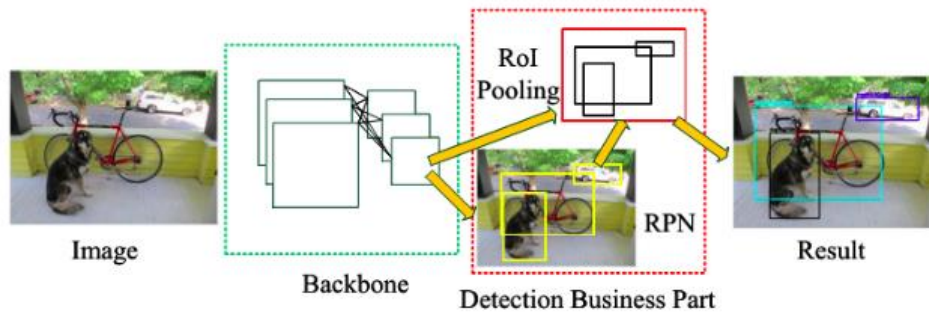
Penggunaan metode computer vision dengan deep learning dapat mempelajari sebuah objek dari bentuk objek tersebut dengan melakukan training data yang sudah dilabelkan. Dengan begitu objek lain yang tidak tersedia pada dataset di masa training tidak akan terdeteksi. Akan tetapi, permasalahan yang muncul saat menggunakan deep learning adalah waktu komputasi yang dibutuhkan cukup tinggi, sehingga dapat membuat delay dalam melakukan deteksi di waktu yang realtime.

Selain penggunaan segmentasi warna yang memiliki beberapa kekurangan yang sebelumnya telah disebutkan, terdapat sebuah metode lain yang cukup sering digunakan dalam melakukan pendeteksian objek, yaitu menggunakan metode machine learning.

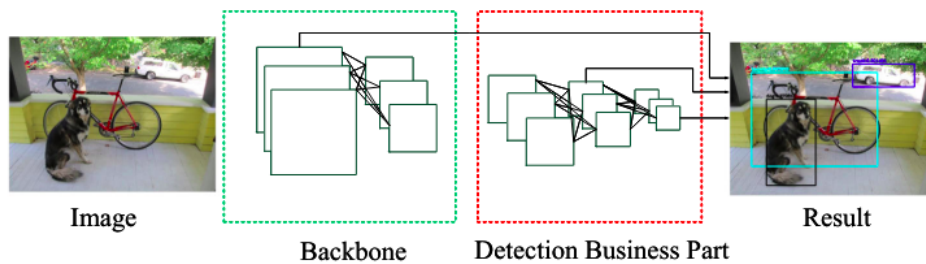
Machine learning adalah sebuah metode dimana komputer mencoba untuk secara otomatis mempelajari hubungan dan pola yang bermakna dari contoh-contoh dan pengamatan [4]. Kemajuan dalam machine learning telah memungkinkan peningkatan sistem cerdas dengan kapasitas kognitif yang menyerupai manusia yang meresap ke dalam kehidupan sehari-hari [5]. Kedua hal ini mendorong penerapan machine learning dalam otomasi, di mana kemampuan machine learning mempelajari pola data dan kemajuan teknologi membuat model-model ini memiliki kemampuan setara manusia.

Dalam menerapkan machine learning guna mendeteksi objek yang akan digunakan pada robot sepakbola, maka memerlukan sub-bidang dari machine learning itu sendiri, yaitu computer vision atau lebih dalamnya object detection model.

Berdasarkan arsitektur dan cara kerja, model object detection terbagi menjadi dua jenis, yaitu single-stage atau proposal-free model detector dan two-stage atau proposal-model. Pada metode two-stage, serangkaian kandidat objek akan di generate oleh Selective Search [6] atau Edge Boxes [7], kemudian lokasi objek tertentu dan label kategori yang sesuai ditentukan dengan menggunakan jaringan konvolusional [8]. Proses ini dapat dilihat pada Gambar 1

Gambar 1. Arsitektur *Two-Stage* Detection

Sebaliknya, metode one-stage menggunakan satu model *feed-forward* convolutional atau menggunakan model convolutional tereduksi bersamaan dengan extra multi-scale layers yang bertujuan untuk mendeteksi kelas objek dan lokasi dari objek tersebut seperti yang terlihat pada Gambar 2.

Gambar 2. Arsitektur *One-Stage* Detection

Perbedaan atas arsitektur inilah yang membedakan kedua metode tersebut. metode two-stage berfokus pada meningkatkan akurasi pada dataset evaluasi standar seperti PASCAL VOC dan MS COCO. Di lain sisi, metode one-stage berjalan lebih baik dalam hal kecepatan dan karena model ini memiliki parameter yang lebih sedikit, maka model one-stage memiliki size yang lebih kecil dibandingkan two-stage detector.

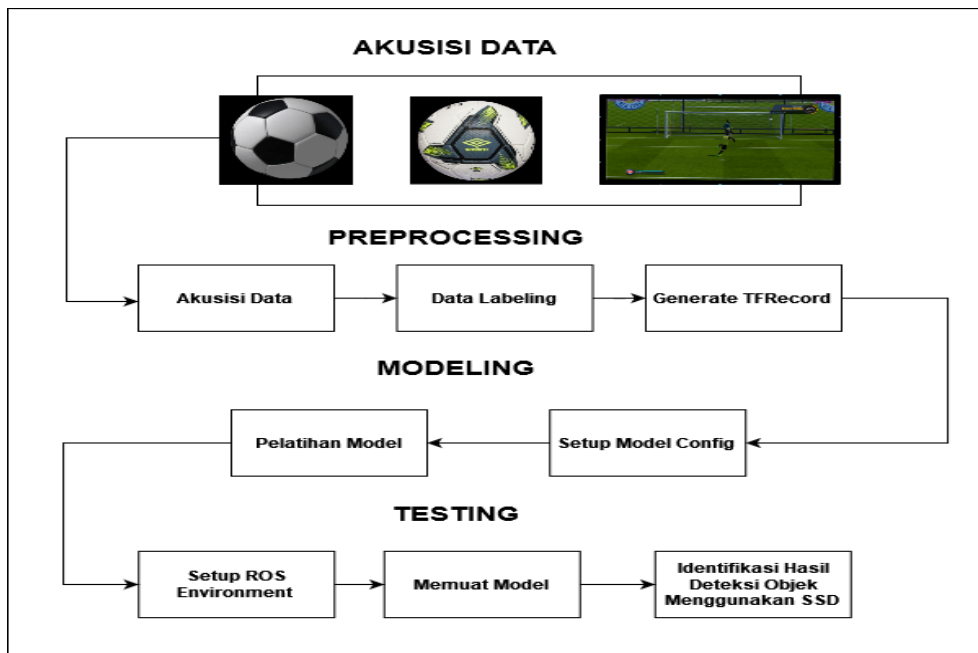
Single Shot Detection (SSD) merupakan sebuah arsitektur yang dapat melakukan deteksi objek secara realtime. Beberapa SSD model, seperti MobileNet V2 300* dapat melakukan deteksi secepat 19 milisecond [9]. Hal ini tentu saja dapat mengatasi masalah delay yang terjadi atas penggunaan deep learning.

Dengan dasar ini lah, penelitian ini diajukan, yaitu untuk melakukan penerapan atas model deteksi single shot detector pada robot sepak bola. Dalam penelitian ini lingkungan pengujian akan dilakukan pada sebuah aplikasi simulasi robotika ROS gazebo. Objek yang termasuk dalam penelitian ini termasuk berbagai variasi bola dan gawang. Tujuan dari penelitian ini adalah mengembangkan sistem robotika yang mampu melakukan deteksi objek yang telah ditentukan sebelumnya dengan mengenali bentuk dan ciri khas dari objek tersebut dan dapat mengambil informasi yang berguna, seperti menghitung jarak objek tersebut dengan posisi robot.

2. METODE PENELITIAN

Dalam penelitian ini, terdapat beberapa tahapan yang dilakukan guna mengidentifikasi keberhasilan model deep learning dalam melakukan deteksi bola dan gawang, tahapan ini dapat dilihat pada Gambar 3. Beberapa tahap itu yaitu akuisisi data, dimana

gambar objek, baik bola maupun gawang, diambil dan dikumpulkan menjadi sebuah dataset. Selanjutnya pada tahap preprocessing dilakukan data labeling, yaitu memberi keterangan objek pada data tersebut serta konversi dataset menjadi sebuah kesatuan format yang dapat diterima oleh Tensorflow Object Detection API, yaitu tfrecord. Setelah itu, dilanjutkan pada tahap modeling, dimana data pada tahap preprocessing digunakan untuk melatih model deep learning. Barulah di tahap pengujian dilakukan percobaan model yang telah dihasilkan di lingkungan simulasi gazebo menggunakan ROS.



Gambar 3. Alur Penelitian

2.1 Akuisisi Data

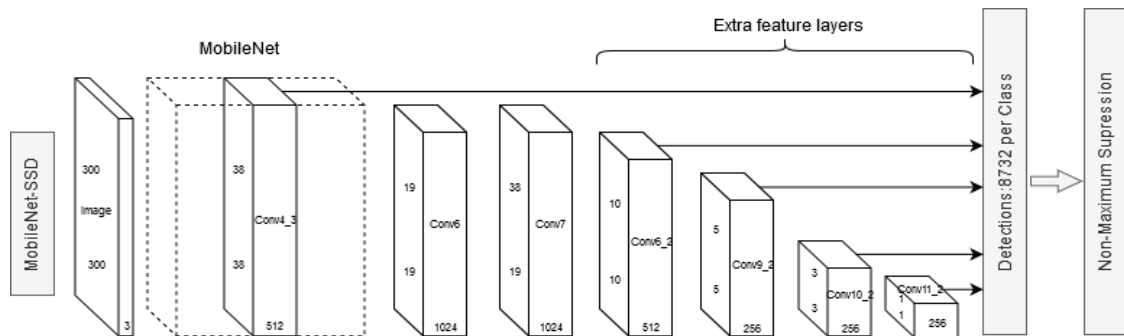
Pengakuisisian data menjadi tahap pertama dimana data didapatkan dari pihak ketiga atau *secondary dataset*. Beberapa dataset tersebut didapatkan melalui halaman web kaggle dan roboflow, dimana data bola didapatkan di kaggle dan data gawang didapatkan di roboflow. Total data yang diperoleh 35.895 data bola dan 268 data gawang.

2.2 Preprocessing

Setelah mendapatkan data yang dibutuhkan, maka tahap selanjutnya adalah preprocessing data. Dalam tahapan ini dilakukan data labeling dan konversi data menjadi tfrecord yang nantinya akan digunakan dalam melakukan training model. Proses data labeling dilakukan menggunakan aplikasi open source labelImg. Tujuan dari dilakukannya labeling adalah memberitahu komputer pada masa training atas kelas serta posisi objek pada sebuah gambar. Setelah dilakukan pelabelan pada gambar, maka berikutnya akan dilakukan pengerasian sebuah data menjadi sebuah format yang dikenali dan akan dimuat oleh model TensorFlow Object Detection, tfrecord.

2.3 Modelling

Setelah melalui tahap preprocessing, maka dilanjutkan pada tahap modeling, yang merupakan fase kunci dalam pembuatan custom object detection model menggunakan TensorFlow Object Detection API dengan arsitektur SSD MobileNet V2 300*. Tahap ini memiliki beberapa sub-tahapan didalamnya.



Gambar 4. Arsitektur SSD MobileNet V2

2. 3.1 Konfigurasi Model dan Parameter Tuning

Proses awal pada tahap modeling, yaitu dilakukannya konfigurasi model SSD MobileNet V2 300* serta melakukan tuning pada beberapa parameter kritis yang perlu diatur agar model dapat menghasilkan deteksi objek yang optimal. Beberapa parameter yang dioptimalkan antara lain (1) num_classes, parameter ini menentukan jumlah kelas objek yang akan diidentifikasi oleh model, (2) learning_rate, parameter ini menjadi faktor penting yang mempengaruhi kecepatan konvergensi model selama pelatihan, (3) batch_size, parameter ini menentukan jumlah sampel yang digunakan dalam satu iterasi pelatihan, (4) num_steps, mengindikasikan seberapa sering model melihat seluruh dataset. Pengaturan yang tepat dapat memastikan bahwa model telah melihat dataset dengan cukup baik untuk pembelajaran yang efektif, (5) max_number_of_boxes, parameter ini menentukan jumlah maksimum kotak pembatas objek yang dapat dihasilkan oleh model pada satu gambar.

2. 3.2 Model Fitting dan Model Evaluation

Pada tahapan ini dilakukan training model yang sudah dikonfigurasi dengan data yang ada. Proses ini sekaligus melakukan evaluasi model setiap 1000 steps-nya. Dalam mengetahui kinerja dari model detection object yang sedang dilakukan training pada setiap iterasinya, metrik mean Average classification loss, localization loss, dan total loss. Nilai ini bisa kita dapatkan dengan menggunakan persamaan cross-entropy loss seperti pada Persamaan 1.

$$H(y, \hat{y}) = - \sum_i^n (y_i \log(\hat{y}_i)) \quad (1)$$

Keterangan:

y: label benar

\hat{y} : label hasil prediksi

Sedangkan untuk mendapatkan nilai dari localization loss didapatkan menggunakan Persamaan 2.

$$L(y, \hat{y}) = \sum_i^n \sum_j^m SmoothL1(y_{i,j} - \hat{y}_{i,j}) \quad (2)$$

Keterangan:

y: label benar

\hat{y} : label hasil prediksi

Dalam mendapatkan nilai pada SmoothL1 digunakan fungsi Huber loss seperti pada Persamaan 3.

$$\text{SmoothL1}(x) = \begin{cases} 0.5 x^2 & \text{if } |x| < 1 \\ |x| - 0.5 & \text{otherwise} \end{cases} \quad (3)$$

Keterangan:

x: selisih dari label kebenaran dan prediksi

Setelah mendapatkan kedua nilai loss tersebut, maka kita dapat menghitung total loss dari model pendeteksi objek menggunakan Persamaan 4.

$$\text{Total Loss} = H(y, \hat{y}) + L(y, \hat{y}) \quad (4)$$

Untuk menguji kinerja model, maka dilakukan evaluasi model yang dilakukan bersamaan dengan masa training model pada tiap 2000 steps menggunakan data yang terpisah dengan data training yang digunakan. Berbeda dengan mengukur kinerja model pada masa training, pada masa evaluasi akan digunakan matrices Mean Average Precision (mAP) dan Recall dalam mengukur kinerja dari model yang telah dilakukan training. Dalam mengetahui nilai-nilai metrik tersebut, maka akan digunakan Tensorboard sebagai alat untuk memudahkan menganalisa kinerja model selama dilakukannya training dan evaluasi.

2.4 Testing

Setelah melakukan model telah selesai melakukan training dan mendapatkan kinerja yang baik berdasarkan beberapa metrik yang sudah ditetapkan, maka akan dilanjutkan pada tahap pengujian di lingkungan simulasi yang sudah ditetapkan. Pada tahap ini, lingkungan simulasi yang akan digunakan adalah lingkungan simulasi dari ROS gazebo. Dalam melaksanakan pengujian pada lingkungan ROS gazebo maka, beberapa langkah akan dilakukan, yaitu 1) Membangun lingkungan simulasi ros dan menjalankannya menggunakan roslaunch, 2) membuat script yang akan melakukan *load* model deteksi objek dan melakukan tugasnya, 3) membuat script untuk menghitung jarak antara objek dan robot.

3. HASIL DAN PEMBAHASAN

Dalam penelitian ini, data yang digunakan merupakan data sekunder yang berasal dari beberapa website, seperti kaggle dan roboflow. Total data yang diperoleh yaitu sebanyak 35.895 data bola dan 268 data gawang. Gambar yang akan digunakan untuk proses training pada tahap modeling berjumlah 1052 data atau sekitar ~87% dari total data yang akan digunakan dan sebanyak 152 data akan digunakan untuk evaluasi atau sebanyak ~13% data dari total data yang akan digunakan. Contoh dari gambar yang digunakan dapat dilihat pada Gambar 6.



Gambar 6. Gambar yang Memuat Objek Bola dan Gawang

Selanjutnya dilakukan preprocessing data untuk data yang sudah diperoleh dan dipisahkan menjadi data train dan data validation. Pada tahapan ini data dilakukan labeling, yaitu penandaan objek yang ada pada gambar. Proses melakukan labeling ini menggunakan aplikasi open source labelImg. Dalam melakukan labeling data, format anotasi yang digunakan adalah PascalVOC. PascalVOC merupakan sebuah standar anotasi dalam bidang computer vision, terkhususnya bidang object detection. Standar anotasi ini menghasilkan file xml dimana file tersebut terkandung berbagai informasi objek yang terlabelkan pada gambar, seperti nama file, path file, posisi bounding box (xmin, ymin, xmax, ymax), nama kelas objek yang berkaitan dengan gambar tersebut serta beberapa informasi lainnya [10]. PascalVOC merupakan format anotasi 1:1, dimana jumlah file xml yang ada akan merepresentasikan satu gambar yang ada. Selain itu perlu juga membuat suatu file berformat .pbtxt, yang merupakan suatu file untuk menyimpan metadata model mengenai nama kelas dari objek yang terdapat pada dataset yang ada. Contoh gambar yang sudah dilabelkan dapat dilihat pada Gambar 7, dimana terdapat bounding box berwarna hijau di sekitar objek yang ditandai

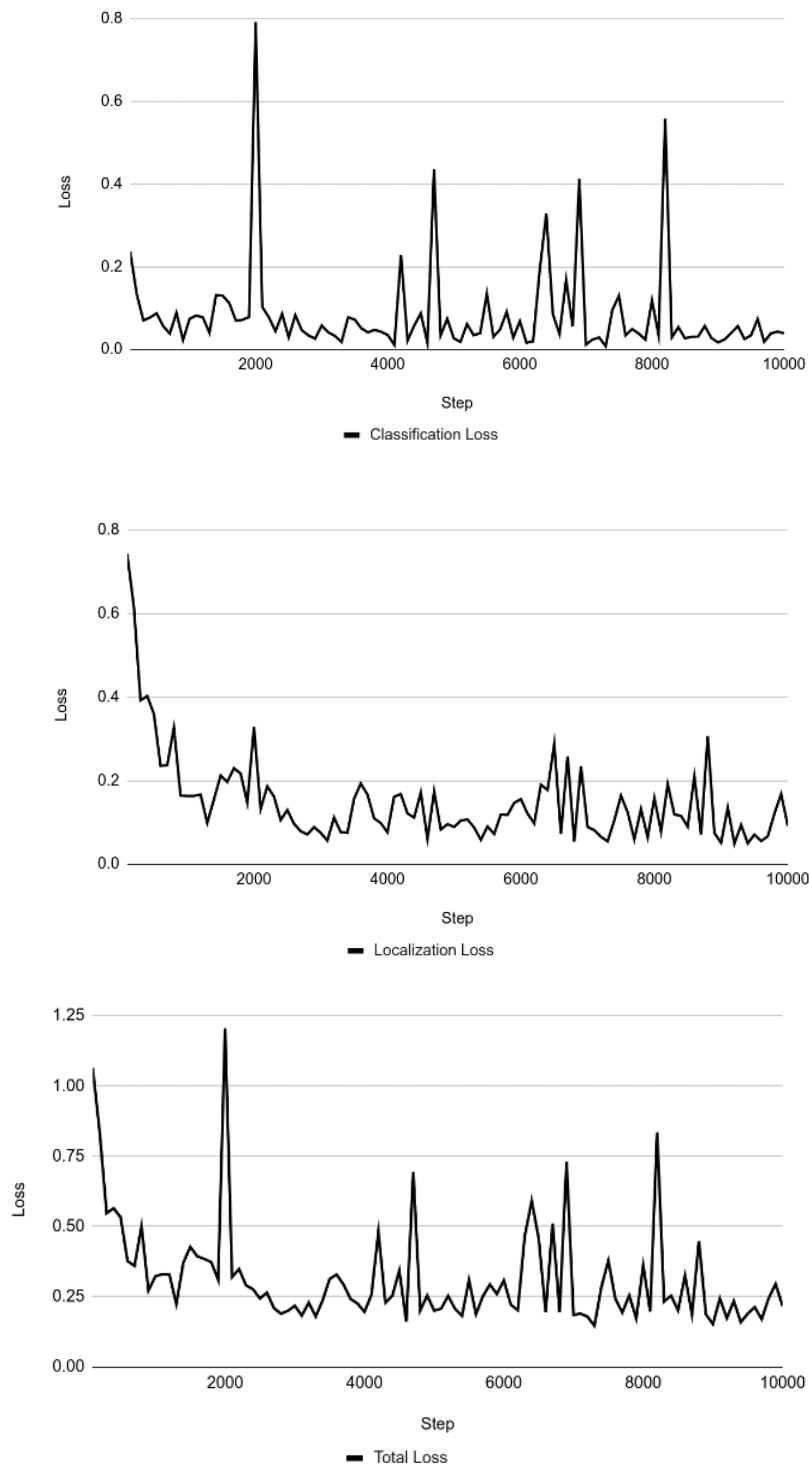


Gambar 7. Gambar yang Sudah Dilabelkan

Kemudian dilanjutkan dengan melakukan generate data bertipe tfrecord dari data yang sebelumnya sudah dilabelkan. Tujuan dilakukan konversi ini adalah model Tensorflow object detection menggunakan tfrecord dalam mengonsumsi data di masa training yang sudah di preprocess pada tahap sebelumnya.

3. 1 *Pelatihan dan Evaluasi Model*

Pada fase pelatihan data tfrecord yang sebelumnya telah di generate akan dikonsumsi oleh model selama n jumlah perulangan yang sudah ditentukan. Seperti yang sebelumnya sudah disebutkan, model Single Shot Detector (SSD) MobileNet V2 adalah model pendeteksi objek yang akan digunakan dalam penelitian ini. Pelatihan model dilakukan menggunakan laptop dengan CPU Intel I7, 16GB RAM, dan GPU Nvidia GTX 1650 TI dan berlangsung selama dua jam dengan rata-rata waktu pelatihan tiap 2000 steps-nya sekitar 23.5 menit. Pelatihan dilakukan. Berdasarkan hasil pelatihan model yang telah dilakukan dengan menggunakan 1052 data, didapatkan beberapa loss score sebesar 0.038 atau 3.8% classification loss, 0.091 atau 9.1% localization loss, dan mendapatkan total loss sebesar 0.215 atau 21.5% . Grafik loss tersebut dapat dilihat pada Gambar 8.

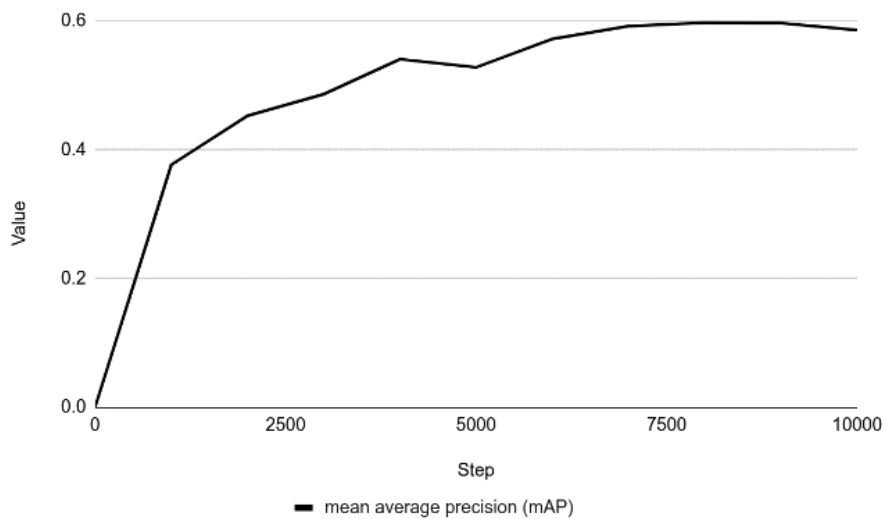


Gambar 8. Grafik Nilai Loss pada Masa Training

Di paling atas terdapat grafik classification loss, di tengah terdapat nilai dari localization loss dan grafik terbawah merupakan nilai dari total loss kedua metric tersebut.

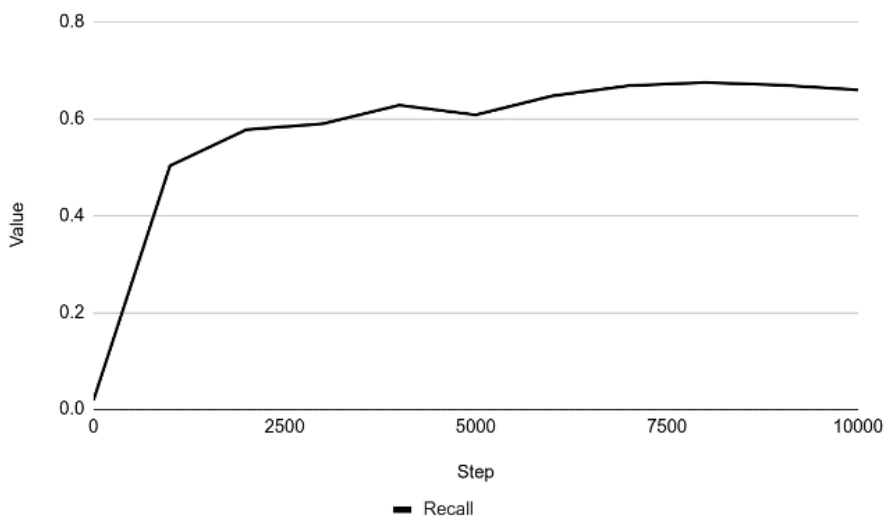
Data yang dihasilkan dalam tahap training menunjukkan bahwa model cukup akurat dalam melakukan klasifikasi objek yang ada pada suatu gambar, meskipun terdapat kesalahan yang sedikit lebih tinggi dalam menentukan lokasi bounding box pembatas dibandingkan dengan akurasi dari klasifikasi yang telah dilakukan. Secara keseluruhan model cukup baik dalam melakukan deteksi objek secara keseluruhan meskipun nilai dari total loss ini terlihat cukup tinggi jika dibandingkan dengan individual loss yang ada.

Pada tahap evaluasi didapatkan nilai mean average precision sebesar 58.6% seperti pada Gambar 9. Hal ini menunjukkan bahwa model memiliki tingkat akurasi yang cukup baik dalam mendeteksi dan mengklasifikasikan objek, walaupun masih ada ruang untuk perbaikan dalam peningkatan nilai mAP tersebut.



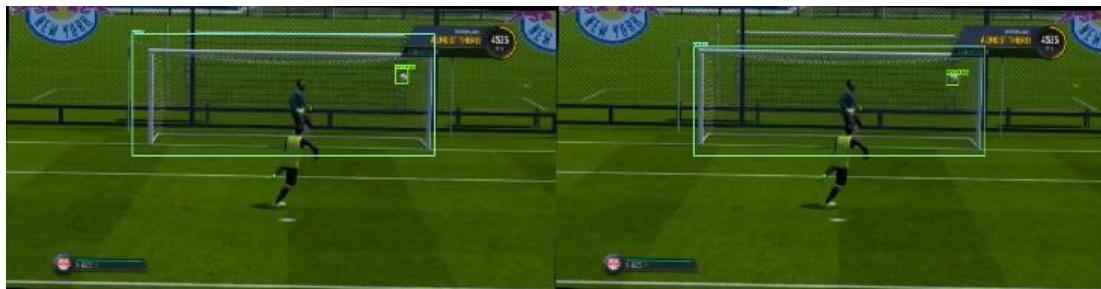
Gambar 9. Grafik Mean Average Precision (mAP)

Berikutnya, nilai recall yang didapatkan pada evaluasi sebesar 66%. Hal ini mengindikasikan bahwa model cukup baik dalam mendeteksi objek, tetapi masih melewati sebagian besar objek pada gambar.



Gambar 10. Grafik Recall

Berdasarkan hasil evaluasi yang telah dilakukan, dapat disimpulkan bahwa model yang dikembangkan menunjukkan kinerja yang cukup baik dalam mendeteksi dan mengklasifikasikan objek, dengan nilai mean average precision (mAP) sebesar 58.6%. Meskipun demikian, terdapat ruang untuk peningkatan lebih lanjut guna meningkatkan akurasi model. Selain itu, nilai recall sebesar 66% mengindikasikan bahwa model sudah cukup baik dalam mendeteksi objek, tetapi masih ada sejumlah objek yang terlewat dalam proses deteksi. Oleh karena itu, pengembangan lebih lanjut diperlukan untuk meningkatkan kemampuan model dalam mengenali objek secara lebih akurat dan komprehensif. Secara keseluruhan, hasil ini memberikan dasar yang kuat untuk optimisasi lebih lanjut dan menunjukkan potensi yang baik untuk aplikasi di berbagai bidang. Salah satu hasil evaluasi dapat dilihat pada Gambar 9.



Gambar 11. Data yang Dilakukan Label (Kiri), Hasil Prediksi Model (Kanan)

3. 2 Pengujian

Pengujian pada tahap ini akan dilakukan dengan menjalankan model deteksi objek sudah di train dan evaluasi pada tahap sebelumnya pada robot turtlebot di lingkungan simulasi gazebo ROS. Dalam melakukan pengujian ini maka beberapa hal harus disiapkan, pertama pengujian ini dilakukan menggunakan ROS 1 pada mesin Ubuntu Linux versi 20.04 dan telah terinstall beberapa aset model yang akan digunakan pada gazebo ROS. Model ini memiliki format SDF atau Simulation Description Format, yaitu format berbentuk XML yang menggambarkan objek dan lingkungan untuk simulasi robot, visualisasi, dan kontrol [11].

Salah satu model yang akan digunakan adalah turtlebot yang akan menjadi agen yang akan dipasang model deteksi objek untuk melihat sekitar lingkungannya. Model ini sebuah sensor kamera, bernama kinetic_camera yang nantinya akan digunakan dalam pengambilan gambar yang dilihat oleh turtlebot.

Gambar yang berasal pada kinetic_camera pada turtlebot akan diambil menggunakan subscriber yang tersedia pada ROS. Subscriber ini akan menggunakan key yang berasal dari Publisher yang dilakukan oleh kinetic_camera. Setelah pengambilan dari kamera didapatkan, maka akan dilakukan normalisasi data agar dapat diteruskan pada model pendeteksi objek.

Selain dari model turtlebot pada gazebo ROS, lingkungan simulasi yang akan digunakan perlu disiapkan. Lingkungan ini akan berupa sebuah file .world yang juga merupakan SDF file dan akan dijalankan menggunakan perintah roslaunch. file .world ini lah tempat berkumpulnya beberapa model SDF lainnya yang akan digunakan untuk keperluan pengujian simulasi. Dalam penelitian ini lingkungan pengujian akan mengambil lapangan pertandingan robot sepak bola dan terdapat beberapa model pendukung seperti bola dan gawang sesuai dengan ketentuan Robocup untuk middle size robot seperti pada Gambar 12.



Gambar 12. Lingkungan Simulasi pada Gazebo ROS

Setelah menyiapkan segala kebutuhan, maka pengujian akan dilakukan untuk memenuhi beberapa kebutuhan pada dalam robot sepak bola, yaitu mendeteksi dan mengenali objek di sekitarnya, di sini bola dan gawang, dan menghitung jarak robot ke objek. Dalam mengaproksimasi jarak antara posisi robot dan objek yang terdeteksi, maka akan digunakan persamaan berikut:

$$D = \tan(\theta_y) \cdot h_{camera} \quad (5)$$

Dimana cara menemukan θ_y , kita dapat menggunakan persamaan berikut:

$$\theta_y = \frac{(y_{center} - \frac{h_{image}}{2})}{\frac{h_{image}}{2}} \cdot \frac{FoV_{verikal}}{2}$$

FoV merupakan field of view mengacu pada area maksimum yang dapat ditangkap oleh kamera dalam gambar. Biasanya diukur dalam derajat dan dapat bervariasi tergantung pada lensa yang digunakan pada kamera. Pada kinetic_camera yang digunakan pada penelitian ini, FoV vertikal bernilai 60° dan Fov horizontal bernilai 70° .

Dalam percobaan ini akan dilakukan beberapa skenario uji coba menggunakan objek bola untuk menilai dari seberapa akurat aproksimasi perhitungan jarak dengan jarak asli pada peta dengan tujuh buah kategori, yaitu sangat dekat, dekat, cukup dekat, medium, cukup jauh, jauh, sangat jauh. Dari percobaan yang dilakukan didapatkan hasil yang dapat dilihat pada tabel 1

Table 1. Hasil Pengukuran pada Perhitungan Jarak

pose robot (x, y, x)	pose bola (x, y, x)	jarak estimasi (m)	jarak asli (m)	akurasi	keterangan
2,3, -0,5, 0,01	1,7, -0,7, 0,05	~0.65	0,57	56%	sangat dekat
2,6, -0,5, 0,01	1,7, -0,7, 0,05	~0.86	0,89	58.3%	dekat
2,8, -0,5, 0,01	1,7, -0,7, 0,05	~1.14	1,13	63%	cukup dekat
3,1, -0,5, 0,01	1,7, -0,7, 0,05	~1.42	1,35	53%	medium
3,3, -0,8, 0,01	1,7, -0,3, 0,05	~1.65	1,81	51%	cukup jauh
3,6, -0,9, 0,01	1,7, -0,3, 0,05	~1.92	2,1	50.3%	jauh
4,0, -1,0, 0,01	1,7, -0,3, 0,05	-	2,71	50%	sangat jauh

Berdasarkan percobaan yang telah dilakukan dengan melalui beberapa skenario yang telah dibuat dapat dilihat bahwa model memiliki rata-rata nilai akurasi dalam pendeteksian sebesar 54.5%. Nilai disini merupakan seberapa yakin model mendeteksi objek tersebut merupakan sebuah bola. Walaupun tergolong memiliki kinerja rata-rata, model masih dapat mendeteksi bola tersebut pada jarak 2,1meter walaupun nilai ini memiliki tingkat akurasi terendah dibanding skenario lainnya dan pada jarak 2.71meter dapat dilihat pada tabel 1 bahwa model sudah tidak dapat menjangkau objek bola karena sudah tidak mengenali fitur dari objek tersebut dari kejauhan.

4. KESIMPULAN

Berdasarkan dari hasil pengujian yang telah dilakukan pada gazebo ROS dalam menerapkan model objek deteksi pada robot menggunakan SSD MobileNet v2 telah dapat melakukan tugasnya, baik melakukan deteksi maupun menghitung jarak objek yang ada pada lingkungannya. Meskipun terdapat kekurangan pada nilai recall dan precision jika dibandingkan dengan model-model deteksi objek yang sebelumnya dilakukan, Hal ini masih dapat diterima mengingat lagi bahwa masih dapat mendeteksi objek hingga berjarak 2 meter dan memiliki ukuran ~150mb.

Selain dari itu, terdapat beberapa kekurangan lain dalam pengujian yang telah dilakukan, yaitu akurasi dalam melakukan deteksi yang tergolong masih di rata-rata. Hal ini ditunjukkan dengan data tabel dimana pada jarak 1.35 meter sampai dengan 2.1 meter akurasi menurun drastis menjadi 50.3 – 53%. Hal ini berkemungkinan besar dipengaruhi oleh dataset yang digunakan dalam training memiliki data latih sedikit untuk bola yang terletak jauh dari gambar atau terlihat kecil di gambar.

5. SARAN

Terdapat saran untuk penelitian selanjutnya, yaitu melakukan improvisasi model, baik dengan cara menambahkan layer RPN atau layer dari model lain ke model one-stage guna meningkatkan akurasi pada model deteksi objek. Selain itu, data yang digunakan pada masa training maupun validation disarankan menggunakan data yang memiliki objek tidak selalu di posisi tengah dan data yang digunakan lebih baik menggunakan data yang akan diuji pada lingkungan percobaan sehingga pengurangan akurasi pada lingkungan itu tidaklah signifikan.

DAFTAR PUSTAKA

- [1] Asian Robotics Review LLC, *"Automation & The Future of Southeast Asia,"* Asian Robotics Review, 2019.
- [2] Pusprenas, *"Kontes Robot Indonesia,"* 2024. [Online]. Available: <https://kontesrobotindonesia.id>.
- [3] A. Darmawan, *"Sistem Pembacaan Bola Menggunakan Omnidirectional Camera Untuk Pergerakan Robot Sepak Bola Beroda," dspace uii*, p. 73, 2018
- [4] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*, Philadelphia: Springer, 2006.

-
- [5] Y. R. Shrestha, . V. Krishna and G. von Krogh, "*Augmenting Organizational Decision-Making with Deep Learning Algorithms: Principles, Promises, and Challenges*," *sciencedirect*, Vol. 123, pp. 588-603, 2020.
 - [6] J. Ujilings, K. van de Sande, T. Gevers and A. Smeulders, "*Selective Search for Object Recognition*," *Springerlink*, Vol. 104, pp. 154-171, 2013.
 - [7] C. Lawrence and P. Dollar, "*Edge Boxes: Locating Object Proposals from Edges*," *Springerlink*, pp. 391-405, 2014.
 - [8] H. Zhang and X. Hong, "*Recent Progresses on Object Detection: A Brief Review*," *springer link*.
 - [9] Tensorflow, "*TensorFlow 2 Detection Model Zoo*," Github, 2021. [Online]. Available: https://github.com/tensorflow/models/blob/master/research/object_detection/g3doc/tf2_detection_zoo.md.
 - [10] M. Everingham, L. Gool, C. K. I. Williams , J. Winn and A. Zisserman, "*The Pascal Visual Object Classes (VOC) Challenge*," *International Journal of Computer Vision*, Vol. 88, pp. 303-338, 2010.
 - [11] "SDFormat," [Online]. Available: <http://sdformat.org/>. [Accessed 3 July 2024].