

BAB 2

LANDASAN TEORI

Pada bab ini akan dibahas tentang teori-teori dasar dan referensi pustaka yang berkaitan dengan pembangunan sistem pengenalan objek menggunakan algoritma dan sensor ultrasonik pada kompetisi Robogames kategori TableTop Challenge.

1.1 TableTop Nav

TableTop Nav atau TableTop Challenge merupakan salah satu kategori dalam kompetisi Robogames yang diadakan 1 tahun sekali di Amerika Serikat [1]. Dalam kategori TableTop Nav robot diharuskan mendeteksi objek kemudian mendorongnya ke ujung meja. Pada kategori ini mempunyai 3 fase yaitu:

Tahap I: robot bergerak dari satu ujung meja ke ujung lainnya dan kembali.

Tahap II: robot mendorong balok hingga jatuh dari tepi meja.

Tahap III: Membuat robot mendorong balok ke dalam kotak Sepatu (gawang) yang dipasang di ujung meja.

Robot perlu menyelesaikan 3 fase tantangan tersebut untuk menang dan menjadi yang tercepat. Berikut merupakan spesifikasi objek, arena dan robot yang ditentukan [2]:

- a) Kisaran Berat Robot: maksimum 5 lbs
- b) Dimensi Robot: Tidak Ditentukan
- c) Spesifikasi Arena: 2,5 x 8 kaki
- d) blok (objek) umumnya berukuran 2 persegi

1.2 Deteksi Objek

Deteksi objek adalah kemampuan sebuah sistem untuk mengidentifikasi objek tertentu dalam gambar atau video. Namun, proses deteksi ini tidak selalu memberikan hasil yang akurat, karena ada kemungkinan sistem tidak mampu mengenali objek yang diinginkan. Hal ini sering kali disebabkan oleh penggunaan

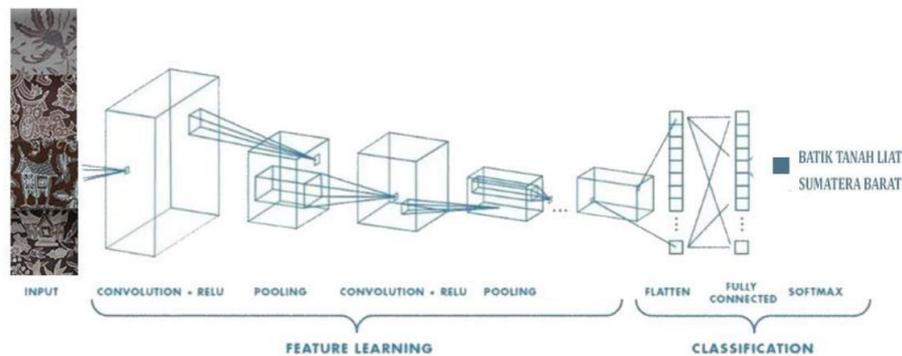
dataset yang kurang optimal. Jika dataset yang digunakan tidak cukup beragam atau representatif, sistem mungkin kesulitan untuk mendeteksi objek dengan benar, terutama dalam situasi yang kompleks atau saat objek tersebut tampil dalam kondisi yang berbeda dari data pelatihan. Oleh karena itu, kualitas dan keragaman dataset sangat penting dalam meningkatkan akurasi dan keandalan sistem deteksi objek [14].

1.3 Roboflow

Roboflow adalah platform web yang berfungsi untuk manajemen dataset, terutama dalam pengembangan computer vision. Ini menyediakan kerangka kerja bagi pengembang untuk mengumpulkan data, melakukan prapemrosesan, dan melatih model secara lebih efektif. Dengan Roboflow, pengguna dapat berbagi dataset, melakukan anotasi objek dengan bounding box, serta menerapkan prapemrosesan seperti mengubah gambar menjadi grayscale dan melakukan augmentasi pada dataset [15].

1.4 Convolutional Neural Network

Convolutional Neural Network (CNN) adalah algoritma deep learning yang dirancang untuk memproses data gambar. Algoritma ini menentukan bobot dan bias pada berbagai elemen dalam gambar untuk membedakan objek satu dengan yang lain. CNN merupakan salah satu arsitektur neural network yang sering digunakan untuk data gambar dan terdiri dari dua lapisan utama: feature learning dan classification layer seperti yang dapat dilihat pada Gambar 2.1 [16].



Gambar 2. 1. Arsitektur CNN

Pada Gambar 2. 1 menunjukkan arsitektur CNN terdiri dari dua bagian utama: feature learning dan classification. Pada bagian feature learning, terdapat beberapa lapisan seperti convolution layer dan pooling layer yang memproses gambar input menjadi feature maps, yaitu representasi angka-angka dari gambar tersebut. Pada bagian classification, lapisan-lapisan fully connected menerima feature maps ini, dan dengan bantuan lapisan aktivasi seperti ReLU dan softmax, menghasilkan output berupa klasifikasi gambar ke dalam kategori tertentu. Bagian lapisan arsitektur CNN dapat dilihat pada Tabel 2. 1.

Tabel 2. 1. Lapisan arsitektur CNN

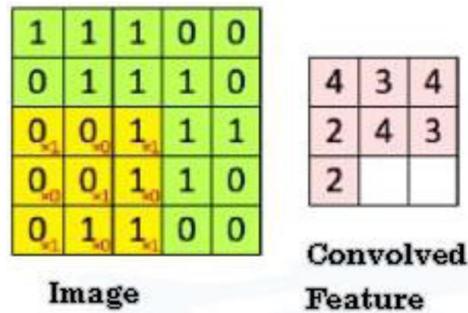
Algoritma Convolutional Neural Network	
Feature Learning	<ul style="list-style-type: none"> a. Input Layer b. Convolution Layer c. Activation Layer d. Pooling Layer
Classification	<ul style="list-style-type: none"> e. Fully Connected Layer f. Output Layer

a. Input Layer

Lapisan ini berguna untuk menampung pixel value dari citra yang diinputkan. Citra yang telah diinputkan memiliki 3 channel warna RGB (Red, Green, Blue).

b. Convolution Layer

Convolutional layer terdiri dari neuron yang tersusun sedemikian rupa sehingga membentuk sebuah filter dengan panjang dan tinggi (pixels). Proses konvolusi menggunakan kernel dan stride, proses konvolusi ini adalah proses kombinasi antara dua buah matriks yang berbeda untuk menghasilkan suatu nilai matriks yang baru. Dalam pengolahan citra, konvolusi berarti mengaplikasikan sebuah kernel (kotak kuning) pada citra di semua offset yang memungkinkan seperti ilustrasi pada Gambar 2.2.

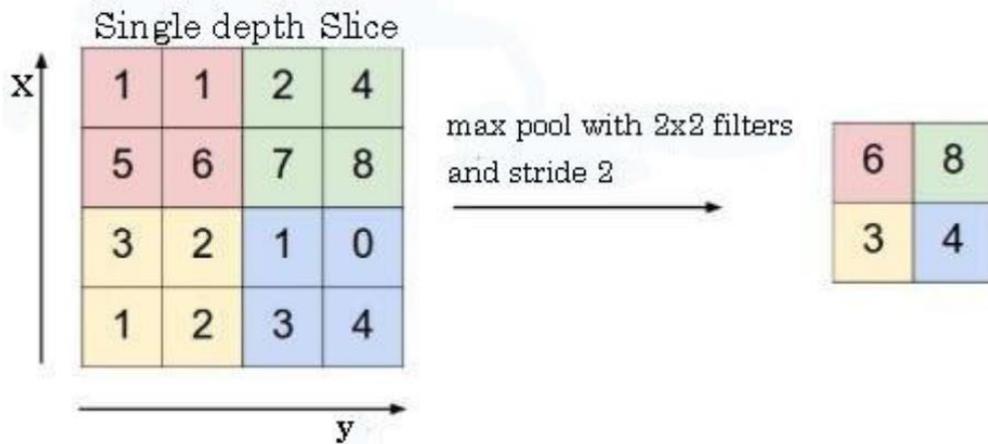


Gambar 2. 2. Ilustrasi operasi konvolusi

Kotak hijau secara keseluruhan adalah gambar yang akan dilakukan konvolusi. Kernel bergerak dari sudut kiri atas ke kanan bawah. Sehingga hasil konvolusi dari gambar 6 dapat dilihat dari gambar di sebelah kanan. Tujuannya dilakukan konvolusi pada data gambar yaitu untuk mengekstraksi fitur dari gambar input.

c. Activation layer

Pada lapisan ini, nilai-nilai pada feature map diubah sesuai dengan fungsi aktivasi yang digunakan, yaitu fungsi ReLU (Rectified Linear Unit). Fungsi ReLU pada dasarnya menetapkan batas nilai dari 0 hingga tak terhingga dan saat ini menjadi salah satu fungsi aktivasi yang populer. Selanjutnya, pada pooling layer, dilakukan pengurangan ukuran setiap tumpukan feature map. Pooling biasanya dilakukan dengan filter berukuran 2x2, yang diterapkan dengan langkah dua pada setiap bagian input. Contoh operasi max-pooling dapat dilihat pada Gambar 2.3.

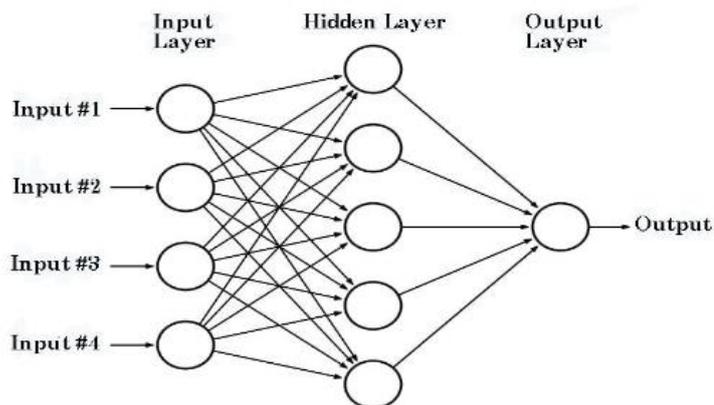


Gambar 2. 3. Max pooling

Kotak-kotak berwarna merah, hijau, kuning, dan biru di sisi kiri mewakili kelompok kotak yang akan dipilih nilai maksimumnya. Hasil dari proses pemilihan ini dapat dilihat pada kumpulan kotak di sisi kanan.

e. Fully connected layer

Lapisan Fully-connected, yang juga dikenal sebagai dense layer, adalah lapisan di mana setiap neuron dari lapisan sebelumnya terhubung secara penuh dengan semua neuron di lapisan berikutnya, serupa dengan struktur pada jaringan syaraf tiruan.



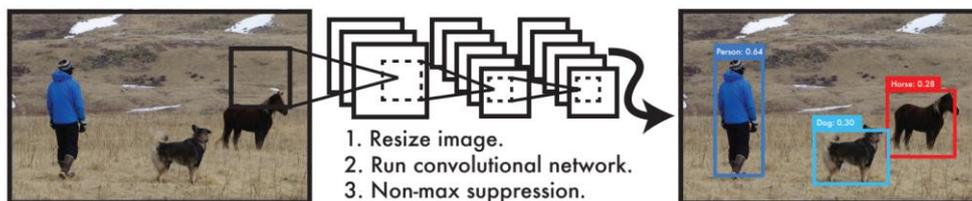
Gambar 2. 4. Fully connected layer

f. Output Layer

Setelah itu dilanjutkan pada proses klasifikasi, yang mana dengan bantuan aktivasi softmax akan diklasifikasi input sesuai dengan target kategorinya.

1.5 YOLO (You Only Look Once)

YOLO atau *You Only Look Once* adalah pendekatan baru untuk deteksi objek. Berbeda dengan sistem lainnya yang menggunakan kembali pengklasifikasi untuk melakukan deteksi, YOLO memperlakukan masalah ini sebagai regresi dengan menghasilkan *bounding boxes* dan probabilitas kelas secara langsung dari gambar dalam satu evaluasi menggunakan jaringan syaraf tunggal. Ilustrasi proses sistem deteksi YOLO dapat dilihat pada Gambar 2.5.



Gambar 2. 5. Sistem deteksi YOLO

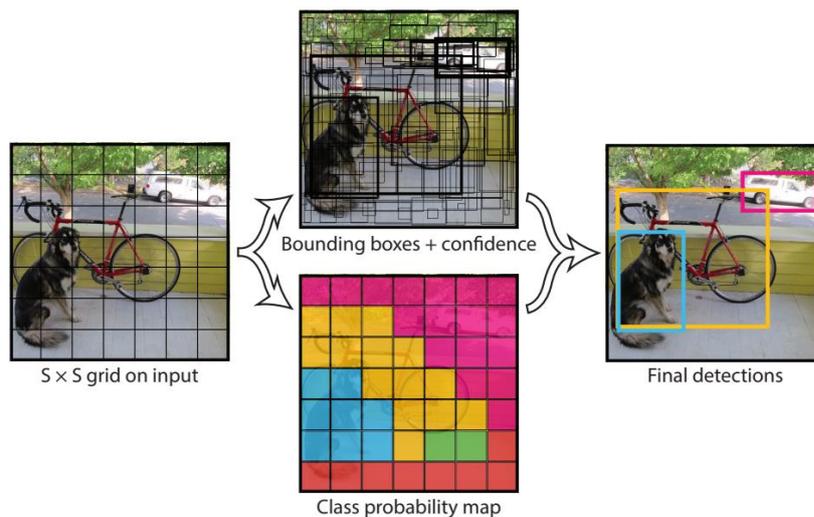
Pada tahap awal, sistem YOLO akan mengubah ukuran gambar masukan menjadi 448 x 448, kemudian menjalankan jaringan konvolusi tunggal pada gambar, lalu sistem akan menerapkan treshold pada deteksi yang dihasilkan berdasarkan *model's confidence* [17].

YOLO menyatukan komponen deteksi objek yang terpisah ke dalam satu jaringan saraf. Pada jaringan YOLO, menggunakan fitur dari keseluruhan gambar untuk memprediksi semua kotak pembatas dan juga memprediksi semua kotak pembatas di semua kelas pada gambar secara bersamaan. YOLO membagi gambar input menjadi grid berukuran $S \times S$. Jika pusat suatu objek berada di dalam sebuah sel grid, maka sel tersebut bertanggung jawab untuk mendeteksi objek tersebut. Setiap sel grid memprediksi B bounding boxes dan confidence scores. Confidence score mencerminkan seberapa yakin model bahwa bounding boxes tersebut mengandung objek serta seberapa akurat objek itu terprediksi. Biasanya, confidence didefinisikan sebagai $Pr(Object) * IOU_{predtruth}$, dan jika tidak ada objek di dalam sel tersebut, nilai confidence akan menjadi nol. Setiap sel grid juga memprediksi probabilitas kelas bersyarat C atau $Pr(Classi|Object)$. Dalam hal ini, hanya satu set probabilitas kelas yang diprediksi per sel grid, tanpa memperhitungkan

jumlah bounding boxes B . Saat pengujian, probabilitas kelas bersyarat dikalikan dengan confidence score dari bounding boxes individual [17].

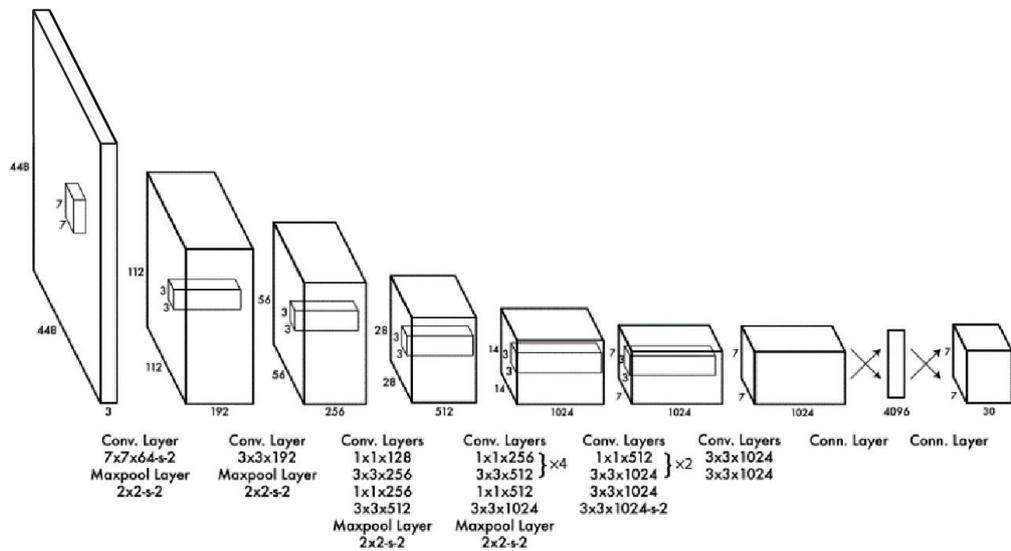
$$Pr(Class_i|Object) * Pr(Object) * IOU_{pred}^{truth} = Pr(Class_i) * IOU_{pred}^{truth} \quad (1)$$

Pada persamaan (1) adalah skor *confidence* untuk setiap kotak, yang khusus terkait dengan kelas tertentu. Skor ini merepresentasikan probabilitas bahwa kelas tersebut muncul di dalam kotak serta seberapa baik kotak yang diprediksi sesuai dengan objek yang sebenarnya. Gambar 2.6, memberikan ilustrasi mengenai prediksi *bounding boxes* B , yang dikodekan sebagai *tensor* dengan dimensi $S \times S \times (B * 5 + C)$. Skor ini tidak hanya mengukur kehadiran objek dari kelas tertentu, tetapi juga kualitas kesesuaian antara bounding box yang dihasilkan dengan posisi sebenarnya dari objek tersebut [17].



Gambar 2. 6. Model memprediksi *bounding boxes*

YOLO diimplementasikan sebagai convolutional neural network (CNN). Arsitektur ini terinspirasi oleh model GoogleNet yang digunakan untuk klasifikasi gambar. Jaringan YOLO terdiri dari 24 lapisan konvolusi yang diikuti oleh 2 fully connected layers. Dalam arsitekturnya, lapisan reduksi 1×1 digunakan diikuti oleh lapisan konvolusi 3×3 untuk menangkap fitur-fitur penting dari gambar. Prediksi yang dihasilkan dari jaringan ini berupa tensor berukuran $7 \times 7 \times 30$ [17]. Untuk melihat detail lebih lanjut tentang arsitektur jaringan YOLO, dapat merujuk pada Gambar 2.7.

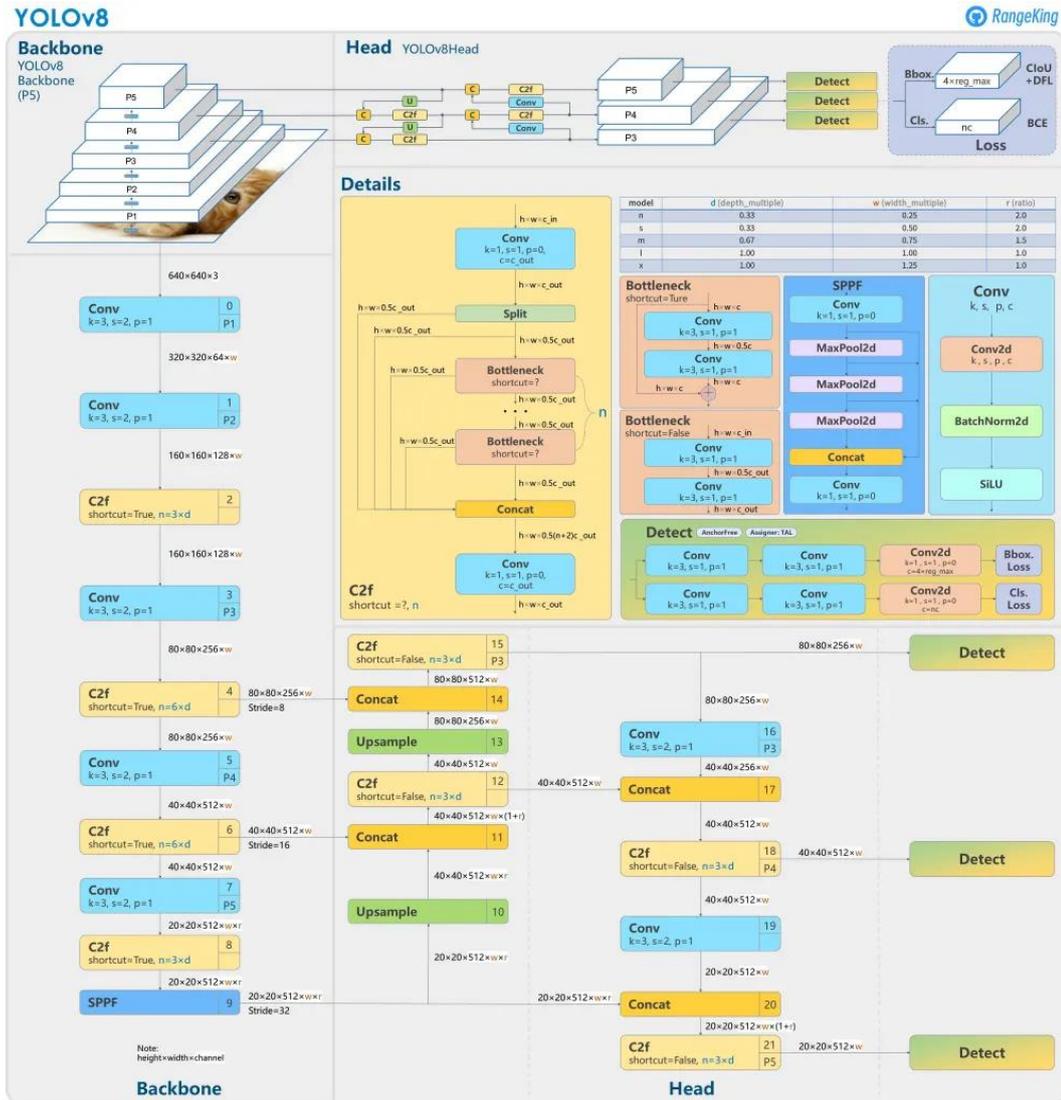


Gambar 2. 7. Arsitektur YOLO

1.6 YOLOv8 (You Only Look Once v8)

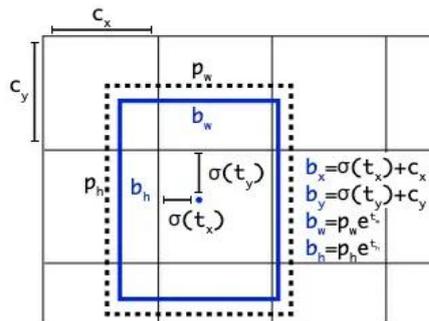
YOLOv8 adalah versi terbaru dari algoritme YOLO, yang dirilis pada tahun 2020. Versi ini menawarkan berbagai peningkatan dibandingkan versi sebelumnya, seperti jaringan tulang punggung yang lebih efisien, prediksi multi-skala, dan sistem jangkar yang baru. Arsitektur YOLOv8 terdiri dari tiga komponen utama: jaringan tulang punggung (backbone), leher (neck), dan kepala (head) [18].

Jaringan tulang punggung memanfaatkan Feature Pyramid Network (FPN) untuk mengekstraksi fitur penting dari gambar input. Leher menggunakan serangkaian Cross-Layer Connection (CLC) untuk lebih menyempurnakan fitur-fitur ini sebelum diteruskan ke kepala. Kepala kemudian menggunakan fitur yang telah disempurnakan tersebut untuk memprediksi bounding boxes, skor kelas objek, dan akurasi untuk setiap objek dalam gambar. Secara keseluruhan, YOLOv8 memiliki 105 lapisan [18]. Untuk gambaran lebih jelas tentang arsitekturnya, dapat dilihat pada Gambar 2.9.



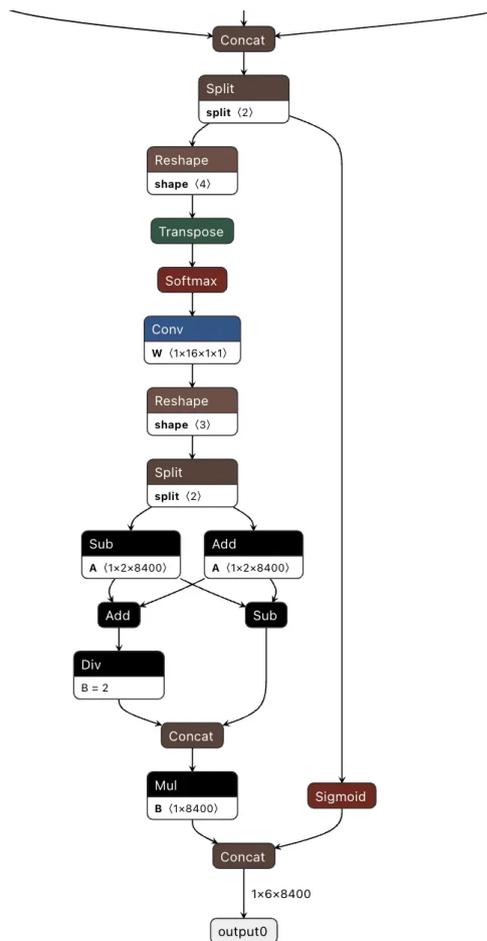
Gambar 2. 8. Arsitektur YOLOv8

Model YOLOv8 ini memiliki Anchor Free Detection. Ini berarti memprediksi langsung pusat objek bukan offset dari kotak jangkar (anchor box) [18]. Berikut merupakan visualisasi anchor box dalam YOLO yang dapat dilihat pada Gambar 2.9.



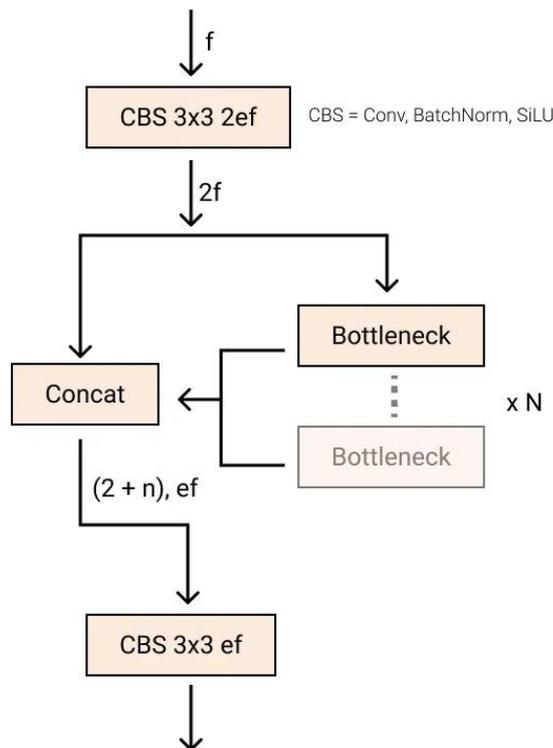
Gambar 2. 9. Anchor Free pada YOLO

Anchor Box merupakan komponen yang dikenal cukup kompleks dalam model YOLO sebelumnya, karena meskipun mereka bisa saja mencerminkan distribusi kotak tolok ukur yang diinginkan, mereka seringkali tidak sesuai dengan distribusi dataset yang spesifik [18].



Gambar 2. 10. Head pada YOLOv8

YOLOv8 memperkenalkan konvolusi baru dengan beberapa perubahan signifikan. Konvolusi pertama yang awalnya menggunakan ukuran 6x6 kini diganti dengan 3x3. Selain itu, blok penyusun utama diubah, di mana modul C2f digantikan oleh C3. Modul ini dijelaskan dalam gambar di bawah ini, di mana "f" mewakili jumlah fitur, "e" adalah tingkat ekspansi, dan CBS adalah blok yang terdiri dari Conv, BatchNorm, dan SiLU. Pada modul C2f, semua output dari Bottleneck (yang merupakan dua konvolusi 3x3 dengan koneksi residual) digabungkan. Namun, pada modul C3, hanya output dari Bottleneck terakhir yang digunakan [18]. Berikut merupakan gambaran dari konsep baru dari YOLOv8 dapat dilihat pada Gambar 2.11.



Gambar 2. 11. Konsep baru dari YOLOv8

1.7 OpenCV

Open Computer Vision (OpenCV) adalah library open source yang dirancang khusus untuk melakukan pengolahan citra, dengan tujuan memberikan komputer kemampuan pengolahan visual yang menyerupai penglihatan manusia. OpenCV menyediakan berbagai algoritma dasar untuk visi komputer dan juga

dilengkapi dengan modul untuk mendeteksi objek menggunakan metode computer vision. Dengan OpenCV, pengembang dapat dengan mudah menerapkan berbagai teknik pemrosesan citra dan analisis visual, menjadikannya alat yang sangat berguna dalam berbagai aplikasi komputer [19].

1.8 Bahasa Pemrograman Python

Python adalah bahasa pemrograman yang sangat populer dan kuat, yang dikenal karena fleksibilitasnya dalam mengembangkan aplikasi di berbagai domain, dengan keunggulan khusus dalam ilmu data dan analisis data. Desain Python menekankan pada keterbacaan dan kesederhanaan, sehingga mudah diakses oleh pemula, namun tetap cukup kuat untuk digunakan oleh pengembang berpengalaman. Dengan koleksi perpustakaan dan kerangka kerja yang luas, Python memudahkan pelaksanaan tugas-tugas manipulasi dan analisis data yang kompleks, yang menjadi semakin penting di era yang didorong oleh data seperti sekarang. Selain itu, Python didukung oleh komunitas besar dan aktif yang menyediakan berbagai sumber daya, tutorial, dan perpustakaan tambahan, yang semakin meningkatkan fungsionalitas dan kemudahan penggunaannya. Secara keseluruhan, kombinasi antara kekuatan Python, kemudahan belajar, dan dukungan dari komunitasnya menjadikannya pilihan utama bagi pengembang di berbagai bidang [20].

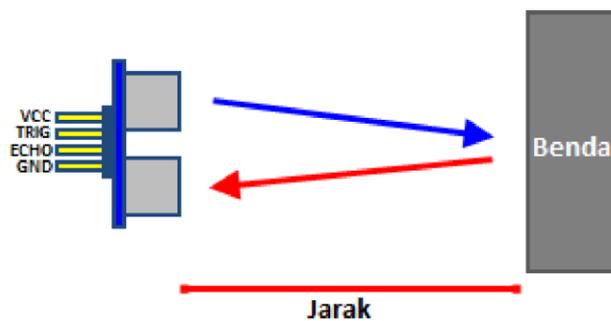
1.9 Sensor Jarak Ultrasonik

Untuk membantu robot dalam memvalidasi objek diperlukan sensor yang dapat mengukur jarak objek yang ada di depan robot, salah satunya adalah sensor ultrasonik. Berikut merupakan visualisasi dari sensor ultrasonik dapat dilihat pada Gambar 2.12.



Gambar 2. 12. Sensor ultrasonik HC-SR04

Sensor ultrasonik adalah perangkat yang berfungsi mengubah besaran fisik berupa bunyi menjadi sinyal listrik dan sebaliknya. Sensor ini bekerja berdasarkan prinsip pantulan gelombang suara, yang memungkinkan pengukuran jarak dan mendeteksi keberadaan suatu objek pada frekuensi tertentu. Dinamakan sensor ultrasonik karena menggunakan gelombang bunyi ultrasonik untuk melakukan pengukuran tersebut [21]. Adapun ilustrasi dari cara kerja sensor ultrasonik dapat dilihat pada Gambar 2.13.



Gambar 2. 13. Cara kerja sensor ultrasonik

Secara umum, perangkat ini memancarkan gelombang ultrasonik ke arah suatu area atau target tertentu. Ketika gelombang tersebut mengenai permukaan target, gelombang akan dipantulkan kembali. Sensor kemudian menangkap gelombang yang dipantulkan dan menghitung selisih waktu antara pengiriman gelombang dan penerimaan gelombang pantul. Selisih waktu ini digunakan untuk mengukur jarak atau mendeteksi keberadaan target.

1.10 Motor Servo

Motor servo digunakan untuk mengubah arah sensor ultrasonik dalam melakukan validasi. Berikut merupakan visualisasi motor servo yang dapat dilihat pada Gambar 2.14.



Gambar 2. 14. Motor servo

Motor servo adalah perangkat atau aktuator rotasi yang dilengkapi dengan sistem kontrol umpan balik tertutup (servo), yang memungkinkan pengaturan dan penentuan posisi sudut dari poros output motor dengan presisi. Motor servo terdiri dari motor DC, serangkaian gear, rangkaian kontrol, dan potensiometer. Gear yang terhubung ke poros motor DC berfungsi untuk memperlambat putaran poros sekaligus meningkatkan torsi motor servo. Potensiometer, yang resistansinya berubah saat motor berputar, berperan dalam menentukan batas posisi putaran poros motor servo [22].

Sistem kontrol loop tertutup pada motor servo sangat penting untuk mengatur gerakan dan posisi akhir dari poros motor. Sederhananya, posisi poros output akan dipantau untuk memastikan apakah sudah mencapai posisi yang diinginkan. Jika belum, sistem kontrol akan mengirimkan sinyal untuk mengoreksi dan mengarahkan poros ke posisi yang tepat [22]. Adapun spesifikasi motor servo yang digunakan yaitu:

- a. Dimensi: 23 x 29 x 12,2 mm b. Berat: 9 g (hanya motor)
- c. Kecepatan reaksi : 0,1 detik / 60 derajat (4,8 V tanpa beban)
- d. Stall torque (4,8V) : 1,6 kg/cm e. Suhu kerja : 0-55 C
- f. Dead band width : 10 μ s (mikro detik)

g. Tegangan kerja : 4,8 V

h. Material gear: nilon

i. Mode : Analog

j. Panjang kabel: 150 mm

1.11 Raspberry Pi

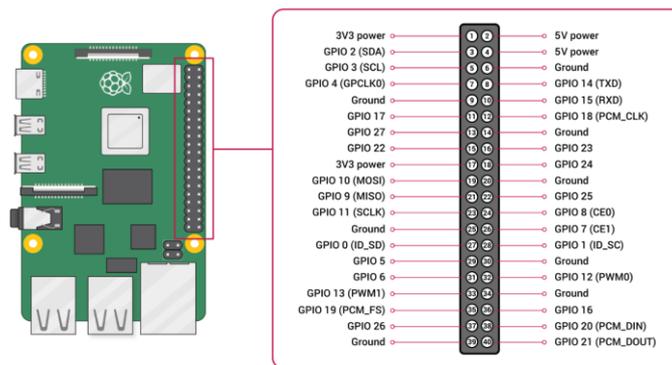
Raspberry Pi (RPi) adalah komputer kecil berbasis single board. Dalam pekerjaan yang diusulkan, perangkat ini digunakan untuk menghubungkan perangkat periferan dengan Raspberry Pi guna melacak produk-produk yang ada di sektor otomotif. Meskipun kinerja RPi relatif lebih lambat dibandingkan dengan komputer pribadi, perangkat ini tetap mampu menyediakan berbagai fitur atau kemampuan yang diharapkan dengan konsumsi daya yang jauh lebih rendah. Raspberry Pi memiliki aplikasi yang sangat luas di berbagai bidang, termasuk pendidikan, otomatisasi rumah, gaming, robotika, dan computer vision. Dalam makalah ini, diusulkan pendekatan baru untuk mengidentifikasi produk otomotif menggunakan Raspberry Pi versi 4. Pendekatan ini dirancang untuk membantu dalam melacak produk-produk otomotif di area penyusunan (kitting area) [23]. Berikut merupakan visualisasi Raspberry Pi yang dapat dilihat pada Gambar 2.15.



Gambar 2. 15. Raspberry Pi

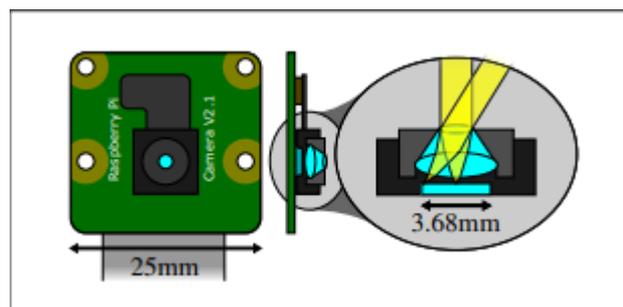
1.12 Raspberry Pi GPIO PIN

Pin GPIO pada Raspberry Pi berfungsi sebagai antarmuka fisik yang menghubungkan Raspberry Pi dengan perangkat lain. Pada dasarnya, pin GPIO dapat dianggap sebagai saklar yang dapat diaktifkan atau dimatikan (untuk input), atau sebagai perangkat yang bisa dinyalakan atau dimatikan (untuk output). Dari total 40 pin yang tersedia, 26 di antaranya adalah pin GPIO, sedangkan sisanya digunakan untuk pin daya atau ground. Untuk visualisasi lebih jelas tentang pin GPIO pada Raspberry Pi, dapat dilihat pada Gambar 2.16.



Gambar 2. 16. Raspberry Pi PIN GPIO [24]

1.13 Modul Camera Raspberry Pi



Gambar 2. 17. Modul camera Raspberry Pi

Komputer single-board Raspberry Pi dan modul kamera pendampingnya telah menjadi komponen utama dalam banyak proyek perangkat keras open-source. Kemampuan untuk mengintegrasikan komputer kecil dan terjangkau namun mumpuni ini memungkinkan tampilan dan pemrosesan gambar secara real-time,

menjadikannya dasar yang sangat baik untuk instrumen ilmiah yang memerlukan komponen pencitraan. Modul Kamera Raspberry Pi kini telah mencapai versi kedua, berupa papan breakout kecil (24×25 mm) yang menghubungkan sensor gambar Sony IMX219 [27] ke Raspberry Pi melalui kabel pita MIPI. Fitur ini didukung oleh kode real-time yang berjalan di firmware GPU1, yang memungkinkan tampilan video beresolusi tinggi dengan latensi rendah [25].