

BAB 2

LANDASAN TEORI

2.1. Landasan Teori

Dalam landasan teori akan dibahas tentang teori-teori yang berhubungan dengan sistem yang akan dibangun.

2.1.1. Lalu lintas

Lalu lintas merupakan ruang gerak kendaraan dan orang di ruang lalu lintas jalan, sedang yang dimaksud dengan ruang lalu lintas jalan adalah prasarana yang diperuntukan bagi gerak pindah kendaraan, orang, dan/atau barang yang berupa jalan dan fasilitas pendukung.

Dalam mengatur lalu lintas supaya aman, lancar, tertib dan teratur pemerintah memberlakukan adanya peraturan-peraturan dalam berlalu lintas dengan adanya manajemen lalu lintas dan/atau menggunakan rambu-rambu sebagai larangan, peringatan, perintah atau petunjuk bagi pengguna jalan[11]

2.1.2. Rambu Lalu Lintas

Rambu lalu lintas adalah bagian perlengkapan jalan yang berupa lambang, huruf, angka, kalimat, dan/atau panduan yang berfungsi sebagai peringatan, larangan, perintah, atau petunjuk bagi pengguna jalan[12].

Macam-macam rambu lalu lintas dilihat dari kelompok-kelompoknya:

a. Rambu peringatan

Rambu peringatan mempunyai ciri, warna dasar kuning, warna garis tepi hitam, warna lambang/huruf/angka hitam

b. Rambu petunjuk

Memiliki warna dasar biru, hijau atau coklat, warna dan huruf/angka putih

c. Rambu larangan

Memiliki ciri warna dasar putih, garis tepi merah, lambang hitam, warna huruf dan angka hitam, dan warna kata-kata merah

d. Rambu perintah

Warna dasar biru, warna garis tepi putih, warna lambang putih, warna huruf dan angka putih

e. Rambu sementara

Warna dasar jingga, warna garis tepi hitam, warna lambang dan/atau tulisan hitam

2.1.3. Pelanggaran Lalu Lintas

Pengertian dari pelanggaran lalu lintas diterbitkan oleh Direktorat Jenderal Pembinaan Badan Peradilan Umum Departemen Kehakiman edisi pertama tahun 1993 yang berbunyi:

“Pelanggaran lalu lintas adalah setiap pelanggaran yang dilakukan oleh pemakai jalan baik terhadap rambu-rambu lalu lintas maupun dalam cara mengemudi jalan, orang yang menggunakan kendaraan bermotor maupun pejalan kaki. “

UU No. 22 tahun 2009 membagi dua tindak pidana pelanggaran lalu lintas yaitu:

1. Tindak pidana pelanggaran lalu lintas, yang terdiri dari beberapa jenis pelanggaran, yaitu:
 - a. Pelanggaran terhadap alat pemberi isyarat lalu lintas.
 - b. Pelanggaran terhadap marka jalan.
 - c. Pelanggaran terhadap rambu - rambu lalu lintas.
 - d. Pelanggaran terhadap kecepatan maksimum dan minimum.
 - e. Pelanggaran terhadap peringatan bunyi.
 - f. Pelanggaran terhadap persyaratan administratif pengemudi dan kendaraan.
2. Tindak pelanggaran angkutan jalan, terdiri dari beberapa jenis pelanggaran yaitu:

- a. Pelanggaran terhadap persyaratan teknis dan layak jalan kendaraan.
- b. Pelanggaran terhadap uatan kendaraan (Undang – Undang Nomor 22 Tahun 2009 tentang Lalu Lintas dan Angkutan Jalan BAB VII tentang kendaraan).

Berdasarkan pemaparan di atas, disimpulkan bahwa pelanggaran lalu lintas merupakan perbuatan yang bertentangan dengan apa yang secara tegas tertulis dalam UU No. 22 tahun 2009 ssebagai pelanggaran[13].

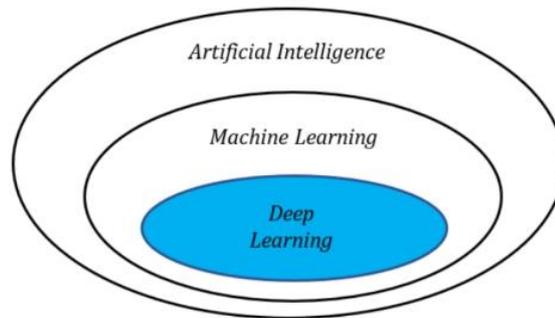
2.1.4. Pelanggaran Dilarang Belok

Pelanggaran Menurut Undang-Undang Nomor 22 Tahun 2009 Tentang Lalu Lintas Dan Angkutan Jalan Pasal 112 ayat 3 berbunyi “Pada persimpangan Jalan yang dilengkapi Alat Pemberi Isyarat Lalu Lintas, Pengemudi Kendaraan dilarang langsung berbelok kiri, kecuali ditentukan lain oleh Rambu Lalu Lintas atau Alat Pemberi Isyarat Lalu Lintas”

2.1.5. Deep Learning

Deep Learning adalah bagian dari jaringan saraf tiruan (JST). Dipublikasikan oleh Rina Dechter pada tahun 1986 dan menurut Google trends algoritma ini terus berkembang dan mulai luas dikenal pada tahun 2014[14]. *Deep Learning* adalah bagian dari jaringan saraf tiruan (JST). Dipublikasikan oleh Rina Dechter pada tahun 1986 dan menurut Google trends algoritma ini terus berkembang dan mulai luas dikenal pada tahun 2014 [15]. *Deep learning* adalah sebuah metode pembelajaran terhadap data yang bertujuan untuk membuat representasi (abstraksi) data secara bertingkat menggunakan sejumlah lapisan pengolahan data. Hal penting dari *deep learning* adalah representasi data tersebut tidak dibuat secara

eksplisit oleh manusia tetapi dihasilkan oleh sebuah algoritma pembelajaran [16].



Gambar 2.1 Deep learning adalah sebagai cabang ilmu dari Artificial Intelligence dan Machine Learning

2.1.6. Convolutional Neural Network (CNN)

Convolutional Neural Network merupakan salah satu algoritma deep learning yang dikembangkan dari Multilayer Perceptron (MLP), dirancang untuk mengolah data dua dimensi. Algoritma ini digunakan untuk klasifikasi data yang sudah diberi label menggunakan metode supervised learning dengan cara kerjanya yang memerlukan data yang dilatih dan memiliki variable yang sudah ditargetkan, tujuan dari metode ini adalah pengelompokan suatu data menuju data yang sudah ada [17].

Jaringan konvolusi adalah jaringan khusus yang dimiliki CNN, citra masukan diolah pada lapisan ini berdasarkan filter yang sudah ditentukan. Hasil dari setiap lapisan ini berupa sebuah pola dari beberapa bagian citra yang nanti akan diklasifikasikan. Di bawah ini adalah cara kerja CNN [18]:

1. Citra input

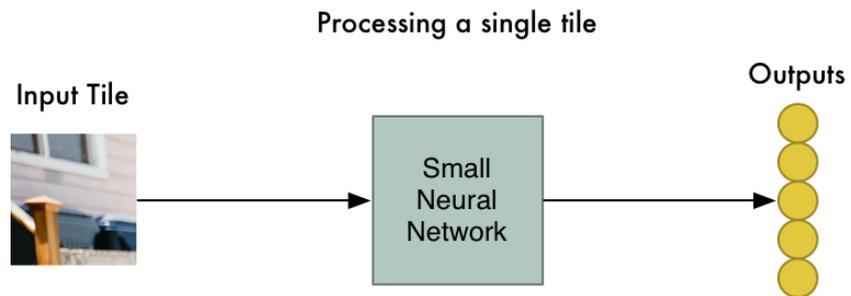


2. Membagi citra menjadi citra yang lebih kecil



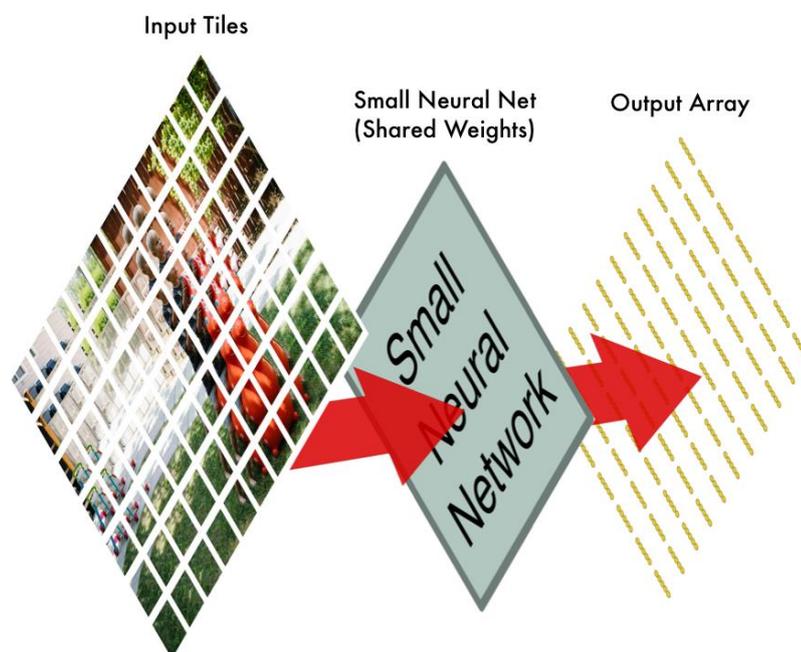
Dari citra masukan yang dimasukkan sebelumnya, hasil dari konvolusinya adalah berupa 77 citra yang lebih kecil dengan konvolusi yang sama.

3. Memasukkan small neural network ke setiap gambar yang telah di bagi.



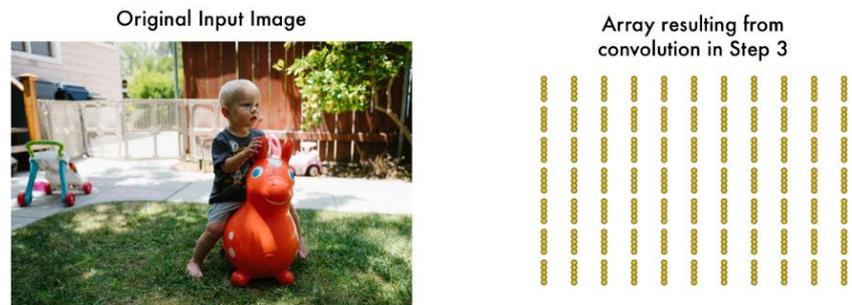
Proses ini diulang sebanyak 77 kali pada setiap citra kecil yang sudah dibagi di proses sebelumnya. Setiap citra kecil akan memiliki bobot neural network yang sama dengan citra aslinya atau disebut sebagai weights sharing. Jika ada yang tampak menarik dalam setiap citranya, maka bagian itu akan ditandai sebagai object of interest.

4. Menyimpan hasil dari setiap citra kecil ke array baru



Menyimpan hasil dari pemrosesan setiap bagian citra ke dalam grid dengan pengaturan yang sama dengan citra aslinya.

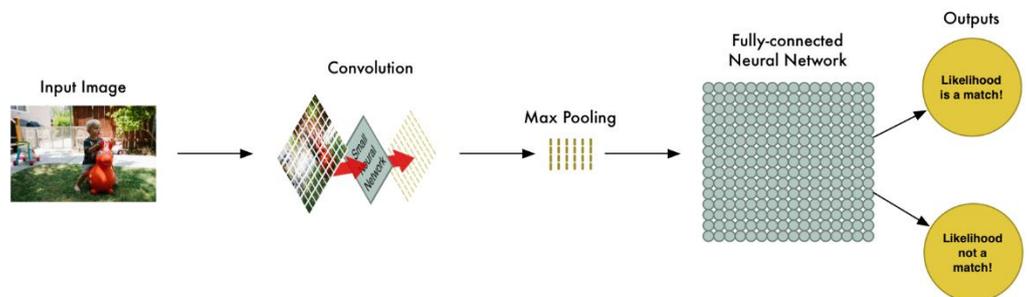
5. Downsampling



hasil dari langkah keempat adalah array yang sudah memetakan bagian mana saja dari citra aslinya yang paling menarik. Tetapi array tersebut masih cukup besar.

Untuk mengurangi ukuran array, dilakukan downsampling menggunakan algoritma max pooling. Array yang ada akan dilihat dengan ukuran kernel 2x2 dan akan tetap menyimpan ukuran piksel paling besar di setiap kernel.

6. Membuat prediksi



Setelah merubah citra yang berukuran besar menjadi array kecil, berikutnya adalah menggunakan array kecil sebagai masukan untuk neural network. Masukan ini adalah proses terakhir untuk menentukan citra tersebut cocok atau tidak. Langkah ini diberi nama fully connected network untuk membedakan dengan langkah konvolusi.

2.1.7. Object Detection

Object Detection atau deteksi objek dalam pengolahan citra suatu proses yang digunakan untuk menentukan keberadaan objek tertentu dalam suatu citra digital. Proses deteksi tersebut dapat dilakukan dengan berbagai macam metode yang umumnya melakukan pembacaan fitur-fitur dari seluruh objek pada citra maupun video. Deteksi objek bertujuan untuk memisahkan objek atau *foreground* dari citra layar atau *background*, akan dibuat sebuah model latar pada sebuah citra bergerak berdasarkan nilai pixel yang masuk seiring waktu[19].

2.1.8. Tracking Objek

Tracking objek pada citra bergerak merupakan sebuah cara untuk mengikuti elemen citra yang sudah dilokalisasi menjadi objek yang bergerak seiring waktu secara otomatis dalam citra yang bergerak seiring waktu secara otomatis dalam citra bergerak. *Tracking* memiliki peran penting dalam pemrosesan video karena dapat menjadi satu patikan untuk mengekstrak beberapa perkiraan properti dari objek bergerak dalam video[19].

2.1.9. Citra

Secara harafiah, citra adalah gambar pada bidang dua dimensi. Ditinjau dari sudut pandang matematis, citra merupakan fungsi menerus dari intensitas cahaya pada bidang dwimarta. Sumber cahaya yang menerangi objek, kemudian objek memantulkan kembali sebagian dari berkas cahaya tersebut. Pantulan cahaya tersebut ditangkap oleh alat-alat topik, misalnya pada mata manusia, kamera, dan sebagainya, sehingga bayangan objek yang disebut citra tersebut terekam[20].

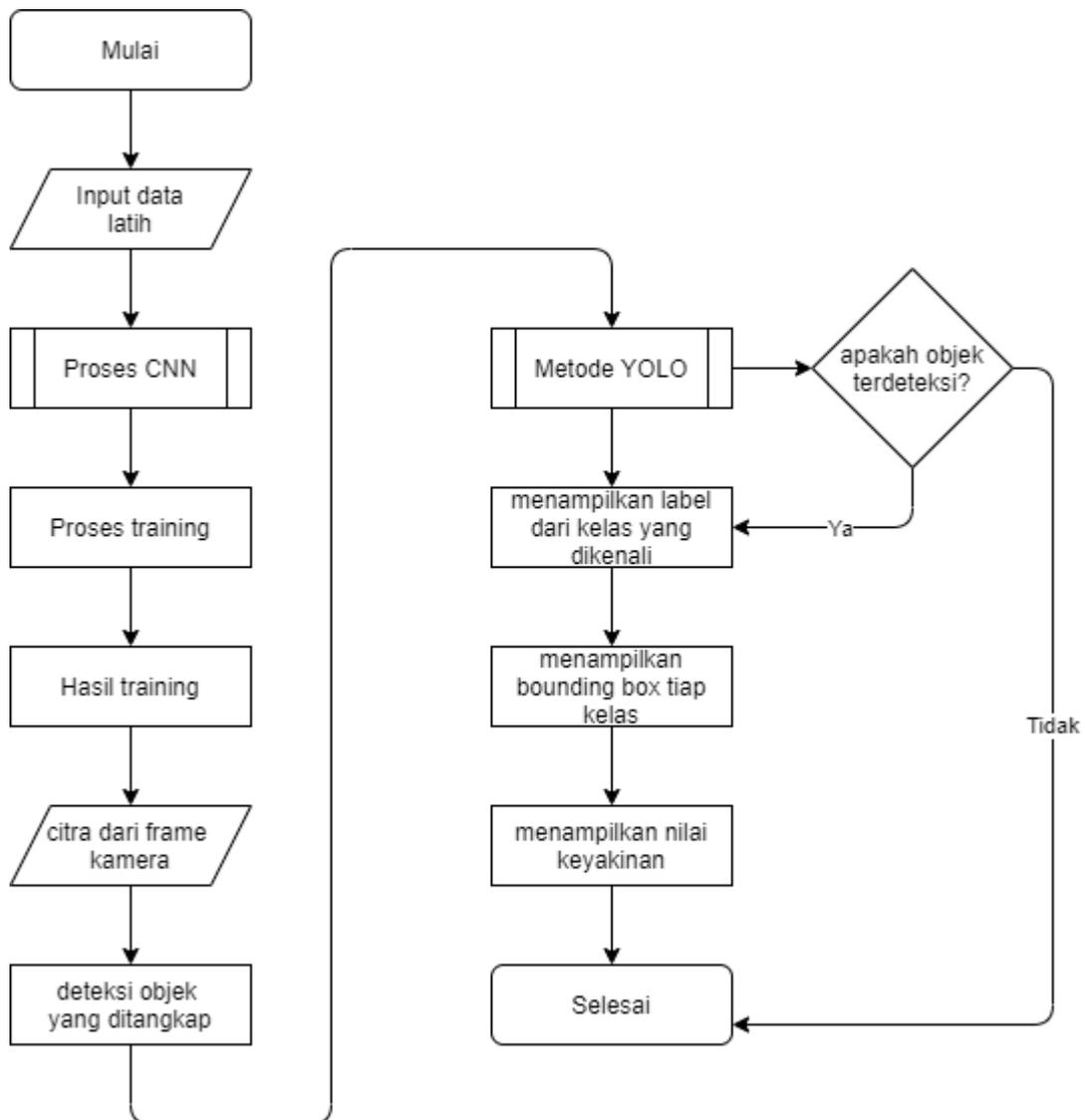
Citra sebagai keluaran dari suatu sistem perkaman data dapat bersifat[MUR92]:

- a. Optik berupa foto,
- b. Analog berupa sinyal video seperti gambar pada monitor televisi

c. Digital yang dapat langsung disimpan pada suatu pita magnetik

2.1.10. Proses *You Only Look Once (YOLO)*

Model pertama dari YOLO dijelaskan pertama kali oleh Joseph Redmon, pada tahun 2015 jurnal yang berjudul *You Only Look Once: Unified, Real-Time Object Detection*. Tercatat bahwa Ross Girshick, pengembang dari R-CNN, merupakan penulis dan contributor dalam jurnal tersebut. Pendekatannya melibatkan jaringan saraf tunggal yang dilatih end-to-end menggunakan gambar sebagai input dan memprediksi bounding box dan label kelas untuk setiap bounding box. Teknik ini menawarkan akurasi prediksi yang lebih rendah, meskipun beroperasi pada 45 fps hingga 155 fps dengan optimasi kecepatan dari versi model ini. Berikut flowchar dari pendeteksian objek menggunakan algoritma YOLO:



1. Input citra latih

Tahap pertama adalah memasukkan citra yang akan dilatih

2. Proses CNN

Pada bagian ini citra yang sudah dimasukkan akan dilakukan proses CNN yang meliputi pemecahan citra, *weights sharing*, pembentukan hasil array, *downsampling*.

3. Proses training

Disini adalah tahap pelatihan citra untuk dapat mendeteksi objek yang ditargetkan.

4. Hasil Training

Keluaran dari proses training adalah *weight* yang dapat digunakan dalam sistem untuk mendeteksi objek.

5. Citra dari frame kamera

Citra yang terdapat di dalam kamera akan digunakan sebagai data uji.

6. Deteksi objek yang ditangkap

Disini adalah tahap dimana sistem berusaha mengenali objek yang dideteksi berdasarkan hasil training yang didapatkan.

7. Metode YOLO

Data masukan yang ada akan di proses dalam metode YOLO ini yang meliputi *resize*, pembagian grid, penentuan letak *bounding box*, dan menghitung nilai keyakinan.

8. Menampilkan label dari kelas yang dikenali

Jika objek dapat dikenali dan sudah diketahui kelasnya maka berikutnya adalah menampilkan label berdasarkan kelas yang dikenali.

9. Menampilkan *bounding box* setiap kelas

Setelah mendapatkan label dari kelas yang terdeteksi proses berikutnya adalah menampilkan *bounding box* di objek yang terdeteksi.

10. Menampilkan nilai keyakinan

Tahap terakhir adalah menampilkan nilai keyakinan dari objek yang terdeteksi berdasarkan proses YOLO yang sudah dilakukan.

2.1.11. Algoritma YOLO

You Only Look Once (YOLO) adalah sebuah algoritma yang dikembangkan untuk mendeteksi sebuah objek secara *real-time*. Sistem pendeteksian yang dilakukan adalah dengan menggunakan *repurpose classifier* atau *localizer* untuk melakukan deteksi. Sebuah model diterapkan di beberapa lokasi dan skala pada citra. Daerah dengan citra yang diberi nilai atau *score* paling tinggi akan dianggap sebagai sebuah pendeteksian[21].

Yolo menggunakan pendekatan jaringan saraf tiruan untuk memprediksi atau mendeteksi objek pada sebuah citra. Jaringan ini membagi beberapa wilayah dan memprediksi setiap kotak pembatas dan probabilitas untuk setiap wilayah. Kotak-kotak pembatas atau *grid* ini kemudian dibandingkan dengan setiap probabilitas yang diprediksi. Yolo memiliki beberapa kelebihan dibandingkan dengan sistem yang berorientasi pada *classifier*.

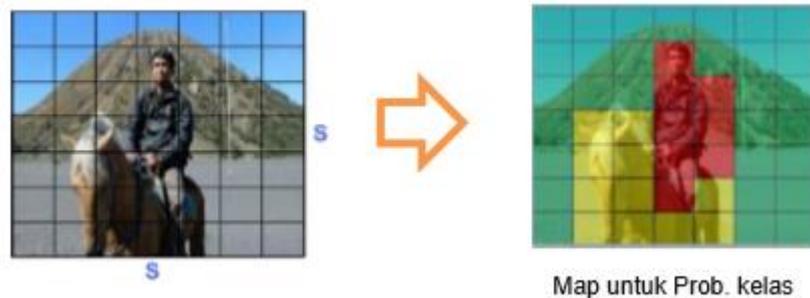
Klasifikasi secara umum adalah proses untuk mengidentifikasi label dari data yang diuji, sedangkan pada Yolo, klasifikasinya dengan *localization*, yaitu terdapat tambahan pemberian lokasi objek dalam bentuk bounding box (b_x, b_y, b_h, b_w). tahap algoritma yolo dilihat pada langkah-langkah seperti berikut[1].

a. Langkah pertama

Baca Data citra dengan ukuran sembarang

b. Langkah kedua

Ubah ukuran citra menjadi 448 x 448. Lalu buat grid pada citra dengan ukuran $S \times S$ *grids*.



Gambar 2.2 Grid pada citra

Jika $S=7$, maka tiap *grid cell* ukurannya 64×64 . Sehingga terdapat sebanyak 49 *grid cell*. Dan misalkan terdapat 2 kelas ($nC=2$) pada citra, yaitu “Manusia” ($c1$), “Kuda” ($c2$), dan “Background”.

c. Langkah Ketiga

Tiap *grid cell*, misal $nB=2$, terdapat B yang berisikan 5 nilai, yaitu lokasi koordinat x diasumsikan berdasarkan garis (x_{br}), koordinat y berdasar kolom (y_{kol}), ukuran dan nilai *confidence* (x,y,w,h,cf) terhadap nB box yang ada, yaitu $B1$ dan $B2$.

$Cf = 0$ (selalu di set =0, jika dalam *grid cell* adalah *background*). *Confidence* (cf) = $P(\text{object}) * IoU$. Yang mana nB menyatakan banyaknya *bbox*, $B1$ untuk *bbox1*, $B2$ untuk *bbox2*, dan *bbox* menyatakan *bounding box*. Persamaan 2 merupakan *intersection Over Union* (IoU).

$$IoU = \frac{\text{Irisan}}{\text{Gabungan}}$$

The diagram illustrates the Intersection Over Union (IoU) formula. It shows two overlapping bounding boxes: a green one labeled 'truth' and a red one labeled 'pred.'. The intersection of the two boxes is shaded yellow and labeled 'Irisan'. The union of the two boxes is shaded blue and labeled 'Gabungan'. The formula is presented as $IoU = \frac{\text{Irisan}}{\text{Gabungan}}$.

Gambar 2.3 Rumus IoU

d. Langkah keempat

Jika 2 bbox mengacu pada kelas yang sama, maka hasil ukuran tensornya adalah $(S \times S \times (nB \times 5 + nC)) = (7 \times 7 \times (2 \times 5 + 2)) = (7 \times 7 \times 12)$ tensor.

e. Langkah ke lima

Jika 2 bbox mengacu pada kelas yang berbeda atau karena terdapat *overlapping object*, maka hasil ukuran tensornya adalah $(S \times S \times (nB \times 5 + 2 \times nC)) = (7 \times 7 \times (2 \times 5 + 2 \times 2)) = (7 \times 7 \times 14)$ tensor

f. Langkah keenam

Tiap kelas pada matriks bb, lakukan $\text{set_ekor} = 0$, jika skor $< \text{thresh1}(0.02)$, kemudian urutkan secara *descending*.

g. Langkah ketujuh

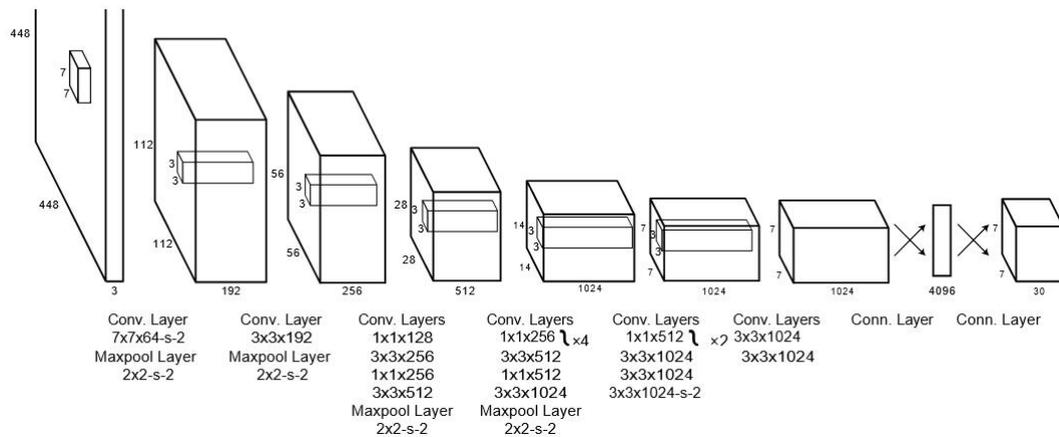
Lakukan *Non-Maximum Suppression* (NMS).

h. Langkah kedelapan

Plot *bounding box* berdasarkan hasil NMS

2.1.11.2. Arsitektur YOLO

Jaringan arsitektur YOLO terinspirasi dari GoogLeNet sebagai klasifikasi citra. Memiliki 24 lapisan konvolusi yang diikuti dengan dua lapisan yang berhubungan penuh. YOLO menggunakan lapisan pengurangan berukuran 1×1 , kemudian diikuti dengan lapisan konvolusi berukuran 3×3 [1]. Berikut adalah gambar dari arsitektur YOLO:



Gambar 2.2.4 Arsitektur YOLO

2.1.11.3. YOLO v3

YOLOv3 adalah versi yang diperbarui dari versi sebelumnya yaitu YOLOv1 dan YOLOv3. Pembaruan pertama adalah *multi-label classification*, menggunakan *logistic classifier* untuk mengkalkulasi kemungkinan dari suatu objek dari label yang sudah dibuat.

Pembaruan kedua adalah penggunaan dari prediksi kotak pembatas (*bounding box*). Berhubungan dengan skor objektivitas kotak pembatas jangkar yang tumpang tindih dengan objek aslinya. Mengabaikan jangkar lainnya yang tumpang tindih dengan objek aslinya dari *threshold* yang sudah ditentukan.

Pembaruan ketiga adalah membuat penggunaan prediksi menggunakan konsep fitur jaringan piramida. YOLOv3 memprediksi 3 kotak dengan skala yang berbeda, lalu melakukan ekstraksi fitur dari setiap skala tersebut[22]. Hasil prediksi dari jaringan merupakan 3D tensor yang mengkodekan kotak pembatas, nilai objektivitas dan prediksi kelas.

2.1.12. Pengujian *Confusion Matrix*

Confusion Matrix adalah suatu metode yang digunakan dalam menghitung akurasi pada konsep *data mining confusion matrix*

sebagaimana digambarkan pada tabel dibawah yang menyatakan jumlah data uji yang benar '*positif*' diklasifikasian dan jumlah data uji yang salah '*negatif*' diklasifikasikan[23]

		Prediksi	
		Positif / 1	Negatif / 0
Aktual	Positif / 1	TP	FN
	Negatif / 0	FP	TN
	TOTAL	TP + FP	FN + TN

Dari tabel diatas dapat dijelaskan sebagai berikut:

1. *True Positive* (TP) merupakan jumlah *record* dari kelas positif yang benar diklasifikasikan sebagai sebagai positif.
2. *False Negative* (FN) merupakan jumlah *record* dari kelas positif yang salah diklasifikasikan sebagai kelas negatif.
3. *False Positive* (FP) merupakan jumlah *record* dari kelas negatif yang salah diklasifikasikan sebagai kelas positif.
4. *True Negative* (TN) merupakan jumlah *record* dari kelas negatif yang benar diklasifikasikan sebagai kelas negatif.

Berikut adalah rumus perhitungan akurasi:

$$Akurasi = \frac{TP+TN}{TP+TN+FP+FN} \times 100\%$$

2.1.13. Model Waterfall

Model air terjun (*waterfall*) sering disebut sekuensial linier atau alur hidup klasik. Model waterfall menyediakan pendekatan alur hidup perangkat lunak secara sekuensial atau terurut dimulai dari Tahap analisis kebutuhan dalam proses pengumpulan data dan kebutuhan, tahap desain pembuatan program perangkat lunak termasuk struktur data dan arsitektur

perangkat lunak, pengkodean desain harus ditransletkan kedalam perngkat lunak, tahap pengujian dan pemeliharaan[24].

2.1.14. Bahasa Python

Bahasa pemograman python adalah bahasa pemograman yang dibuat oleh Guido van Rossum dari Amsterdam, Belanda. Pada awalnya motivasi pembuatan bahasa pemograman ini adalah bentuk bahasa skrip tingkat tinggi pada sistem operasi terdistribusi Amoeba. Bahasa pemograman ini menjadi umum digunakan untuk kalangan engineer seluruh dunia dalam pembuatan perangkat lunaknya. Python banyak digunakan untuk membuat berbagai macam program, seperti program CLI, program GUI (desktop), Aplikasi Mobile, Web, IoT, Game, Program, untuk Hacking, dan sebagainya[25].

Berikut adalah kelebihan dari bahasa pemrograman Python:

1. Python merupakan bahasa pemrograman ke-3 paling populer di dunia.
2. Python relative lebih mudah dipelajari dan digunakan dibandingkan bahasa pemrograman lain. Sintaks sederhana, mudah dibaca, dan diingat karena filosofi Python sendiri menekankan pada aspek kemudahan dibaca (*readability*). Kode python mudah ditulis dan mudah dibaca, sehingga lebih mudah diperbaiki jika ada kesalahan.
3. Python merupakan bahasa multifungsi. Dengan Python, dapat dengan mudah mengembangkan sebuah produk, baik itu web, *software*, aplikasi web, maupun game, robotika, data mining, sampai dengan kecerdasan bauta. Bahasa ini juga dapat membuat aplikasi berbasis desktop maupun berbasis mobile.
4. Penulisan kode lebih efisien dibandingkan bahas lain seperti C, C++, ataupun Java. Untuk melakukan dengan lima baris kode dengan bahasa lain, dengan Python bisa jadi hanya diperlukan satu baris kode saja. Hal ini menyebabkan pembuatan program dalam Pyhton menjadi lebih ringkas dan lebih cepat dibandingkan bahasa lain.

5. Python kaya akan dukungan *library* standar. Tersedia banyak modul dan ekstensi program yang sudah siap digunakan untuk membuat program sesuai kebutuhan.

Adapun kekurangan dari bahasa pemrograman Python sebagai berikut:

1. Beberapa penugasan terdapat di luar jangkauan kemampuan Python. Seperti bahasa pemrograman dinamis lainnya, Python tidak secepat atau efisien sebagai statis, tidak seperti bahasa pemrograman C.
2. Python cukup buruk untuk pengembangan platform mobile (Andorid/IOS).
3. Python merupakan interpreter sehingga bukan perangkat bantu terbaik untuk pengantar kinerja kritis. Jadi, Python bukan pilihan yang baik untuk tugas – tugas intensif memori.
4. Hamper mustahil untuk membuat game 3D grafis tinggi menggunakan Python.