

SKRIPSI

**SISTEM DETEKSI RAMBU DAN LAMPU LALU LINTAS
UNTUK AUTONOMOUS CAR MENGGUNAKAN FASTER R-
CNN**

Disusun dan diajukan oleh:

KHAIRUL HIDAYAT

D121171501



DEPARTEMEN TEKNIK INFORMATIKA

FAKULTAS TEKNIK

UNIVERSITAS HASANUDDIN

MAKASSAR

2022

LEMBAR PENGESAHAN SKRIPSI
SISTEM DETEKSI RAMBU DAN LAMPU LALU LINTAS UNTUK
AUTONOMOUS CAR MENGGUNAKAN FASTER R CNN

Disusun dan diajukan oleh

KHAIRUL HIDAYAT

D121171501

Telah dipertahankan di hadapan Panitia Ujian yang dibentuk dalam rangka Penyelesaian Studi Program Sarjana Program Studi Teknik Informatika Fakultas Teknik Universitas Hasanuddin pada tanggal 13 Oktober 2022 dan dinyatakan telah memenuhi syarat kelulusan.

Menyetujui,

Pembimbing Utama,



Dr. Indrabayu, S.T., M.T., M.Bus.Sys.
Nip. 197507162002121004

Pembimbing Pendamping,



Anugrayani Bustamin, S.T., M.T.
Nip. 199012012018074001



Ketua Program Studi,

Dr. Indrabayu, S.T., M.T., M.Bus.Sys.
Nip. 19750716 200212 1 004

PERNYATAAN KEASLIAN

Yang bertanda tangan di bawah ini:

Nama : Khairul Hidayat
NIM : D121171501
Departemen : Teknik Informatika
Jenjang : S1

Menyatakan dengan ini karya tulisan saya berjudul:

SISTEM DETEKSI RAMBU DAN LAMPU LALU LINTAS UNTUK
AUTONOMOUS CAR MENGGUNAKAN FASTER R-CNN

Adalah karya tulisan saya sendiri dan bukan merupakan pengambilalihan tulisan orang lain bahwa skripsi yang saya tulis ini benar-benar merupakan hasil karya saya sendiri.

Apabila di kemudian hari terbukti atau dapat dibuktikan bahwa sebagian atau keseluruhan skripsi ini hasil karya orang lain, maka saya bersedia menerima sanksi atas perbuatan tersebut.

Makassar, 13 September 2022

Yang menyatakan,



Khairul Hidayat

ABSTRAK

Dalam sistem *autonomous car*, fitur dasar yang wajib dimiliki yaitu kemampuan untuk mendeteksi dan mengklasifikasi rambu dan lampu lalu lintas. Dengan mendeteksi rambu lalu lintas akan menyediakan berbagai informasi, dari batas kecepatan, peringatan, hingga informasi tentang kemana dan kapan harus berbelok. Dalam mendeteksi rambu dan lampu lalu lintas, penelitian ini menggunakan *Faster R-CNN (Region Based Convolutional Neural Networks)* sebagai sistem pendeteksi dan klasifikasi. Apabila sistem mendeteksi objek, maka *output* yang diberikan berupa hasil klasifikasi dan suara sesuai dengan jenis objek yang terklasifikasi. Analisis kinerja sistem menggunakan *Multi-Class Confusion Matrix* dengan ukuran *matrix* 12 x 12 sesuai dengan jumlah kelas dalam sistem. Terdapat dua jenis pengujian, yaitu pengujian 1 dengan 646 gambar dan pengujian 2 dilakukan secara *realtime* di jalan yang telah dikondisikan. Pengujian 1 menghasilkan akurasi sebesar 95.67%. Pengujian 2 dengan 4 skenario berdasarkan 4 kecepatan berbeda yaitu 10km/jam, 20km/jam, 30km/jam, dan 40km/jam menghasilkan akurasi berturut-turut yaitu 72.22%, 66.67%, 70.59% dan 68,75%. Selain itu, juga dilakukan pengujian jarak deteksi pada kecepatan 40km/jam dan menghasilkan jarak deteksi terjauh rambu yaitu 15-20 meter. Dengan jarak deteksi 15 meter, maka dapat memberi waktu pengereman yang cukup.

Kata kunci: Rambu Lalu Lintas, Lampu Lalu Lintas, *Autonomous Car*, *Faster R-CNN*.

KATA PENGANTAR

Segala puji dan syukur penulis panjatkan kepada Tuhan Yang Maha Esa, yang telah memberikan rahmat dan karunia-Nya, sehingga dapat menyelesaikan tugas akhir dengan judul “SISTEM DETEKSI RAMBU DAN LAMPU LALU LINTAS UNTUK AUTONOMOUS CAR MENGGUNAKAN FASTER R-CNN” sebagai salah satu syarat dalam menyelesaikan jenjang Strata-1 di Departemen Teknik Informatika, Fakultas Teknik, Universitas Hasanuddin.

Penulis menyadari bahwa penyusunan dan penulisan tugas akhir ini tidak dapat terselesaikan dengan baik tanpa adanya bantuan, bimbingan, serta dukungan dari berbagai pihak. Oleh karena itu, penulis ingin menyampaikan ucapan terima kasih banyak kepada:

1. Kedua orang tua penulis, Bapak Taufik Ali dan Ibu Nurwahida yang selalu memberikan dukungan, doa, dan semangat yang tiada hentinya, serta selalu sabar dalam mendidik penulis sejak kecil.
2. Bapak Dr. Indrabayu, S.T., M.T., M.Bus.sys., selaku pembimbing I dan Ibu Anugrayani Bustamin, S.T., M.T. selaku pembimbing II, yang senantiasa menyediakan waktu, tenaga, pikiran, dan perhatian yang luar biasa dalam mengarahkan penulis untuk menyelesaikan tugas akhir.
3. Bapak Dr. Indrabayu, S.T., M.T., M.Bus.sys., selaku ketua Departemen Teknik Informatika Fakultas Teknik Universitas Hasanuddin.
4. Muh Fadhil, Nublan Azqalani, dan Taslinda sebagai tim *Autonomous Car* yang membantu penulis dalam pengumpulan data dan pengujian sistem.

5. Muh Alfarabi Alif Putra yang senantiasa memberikan nasihat dan masukan sebagai mentor.
6. Segenap keluarga *AIMP Research Group* Universitas Hasanuddin yang telah memberikan begitu banyak bantuan selama penelitian, pengambilan data dan diskusi *progress* penyusunan tugas akhir serta memberikan semangat di masa-masa sulit.
7. Ilmi Nurrahma Ismail, Muhammad Zulfahmi Sadrah, Muh. Ikbal, Muh. Ghalib, Ikhsan Jihadi, Ilham, Irzam Kasyfillah, Irfan Ripat, dan teman-teman Recognizer 2017 yang telah membantu penulis sejak awal perkuliahan dan selalu membantu dalam penyelesaian tugas akhir.

Akhir kata, penulis menyadari bahwa tugas akhir ini masih jauh dari kata sempurna, oleh karenanya diharapkan segala bentuk saran serta masukan yang membangun dari berbagai pihak. Semoga tugas akhir ini dapat memberikan sumbangsih dan manfaat besar bagi kepentingan bersama.

Makassar, September 2022

Penulis

DAFTAR ISI

ABSTRAK	1
KATA PENGANTAR	2
DAFTAR ISI	4
DAFTAR GAMBAR	6
DAFTAR TABEL	8
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Tujuan Penelitian	3
1.4 Manfaat Penelitian	3
1.5 Batasan Masalah	3
1.6 Sistematika Penulisan	4
BAB II TINJAUAN PUSTAKA	6
2.1 Rambu Lalu Lintas dan Lampu Lalu Lintas	6
2.2 <i>Autonomous Car</i>	9
2.3 Visi Komputer	11
2.3.1 <i>Image Augmentation</i>	12
2.3.2 <i>Bounding Box</i>	12
2.3.3 <i>Anchor Boxes</i>	13
2.4 <i>Region Based CNN</i>	13
2.4.1 R-CNN	13
2.4.2 <i>Fast R-CNN</i>	15
2.4.3 <i>Faster R-CNN</i>	18

2.5 <i>Residual Network</i>	21
2.6 <i>Confusion Matrix</i>	24
2.7 Jarak Aman Pengereman	29
BAB III METODOLOGI PENELITIAN	31
3.1 Tahap Penelitian	31
3.2 Waktu dan Lokasi Penelitian	32
3.3 Instrumen Penelitian	32
3.4 Teknik Pengambilan Data	33
3.5 Perancangan Sistem	34
3.5.1 Data Input	36
3.5.2 <i>Data Preparation</i>	36
3.5.3 Implementasi <i>Faster R-CNN</i>	41
3.6 Evaluasi Sistem	44
BAB IV HASIL DAN PEMBAHASAN	46
4.1 Hasil Penelitian	46
4.1.1 Pengujian 1	46
4.1.2 Pengujian 2	47
4.1.3 Pengujian Jarak pada Kecepatan 40km/jam	54
4.2 Pembahasan	55
BAB V KESIMPULAN DAN SARAN	61
5.1 Kesimpulan	61
5.2 Saran	62
DAFTAR PUSTAKA	63
LAMPIRAN	65

DAFTAR GAMBAR

Gambar 2.1. Ukuran Rambu Peringatan	6
Gambar 2.2. Ukuran Rambu Larangan	7
Gambar 2.3. Ukuran Rambu Perintah	7
Gambar 2.4. Ukuran Rambu Petunjuk	8
Gambar 2.5. Alat pemberi isyarat lalu lintas (lampu lalu lintas)	9
Gambar 2.6. Model R-CNN (Sumber: Zhang et al., 2022).	14
Gambar 2.7. Model <i>Fast</i> R-CNN (Sumber: Zhang et al., 2022).	15
Gambar 2.8. 2×2 region of interest pooling layer (Sumber: Zhang et al., 2022).	17
Gambar 2.9. Diagram Prinsip RPN (Sumber: Kong et al., 2020).	20
Gambar 2.10. <i>Faster</i> R-CNN <i>basic operation process</i> (Sumber: Kong et al., 2020).	21
Gambar 2.11. <i>ResNet50 Architecture</i>	22
Gambar 2.12. Blok <i>Regular</i> (Kiri) dan Blok <i>Residual</i> (Kanan) (Sumber: Zhang et al., 2022).	23
Gambar 2.13. <i>Multi-Class Confusion Matrix</i> (Sumber: Grandini et al., 2020).	25
Gambar 3.1 Tahapan Penelitian	31
Gambar 3.2. Penempatan <i>dashcam</i> (kiri), Sudut pandang <i>dashcam</i> (kanan)	34
Gambar 3.3. Diagram Blok Sistem	35
Gambar 3.4. LabelImg	37
Gambar 3.5. Augmentasi Grayscale.	38
Gambar 3.6. Augmentasi ColorJitter.	38
Gambar 3.7. Augmentasi Invert.	39

Gambar 3.8. Augmentasi Posterize	39
Gambar 3.9. Augmentasi Solarize	40
Gambar 3.10. Augmentasi AdjustSharpness	40
Gambar 3.11. Augmentasi Equalize	41
Gambar 3.12. Arsitektur <i>Faster R-CNN</i>	42
Gambar 4.1. Perbandingan <i>precision</i> , <i>recall</i> , dan <i>F1-score</i> pada pengujian 1	56
Gambar 4.2. Grafik perbandingan nilai <i>F1-score</i> pada pengujian 2	58
Gambar 4.3. Contoh rambu larangan	59

DAFTAR TABEL

Tabel 2.1 Nilai Confusion Matrix pada Kelas A	27
Tabel 2.2 Nilai Confusion Matrix pada Kelas B	27
Tabel 2.3 Nilai Confusion Matrix pada Kelas C	28
Tabel 2.4 Nilai Confusion Matrix pada Kelas D	29
Tabel 2.5 Jarak pengereman	30
Tabel 4.1 Hasil <i>Confusion Matrix</i>	46
Tabel 4.2. Hasil pengujian 1	47
Tabel 4.3. Hasil <i>confusion matrix</i> pengujian 2 dengan kecepatan 10km/jam	48
Tabel 4.4. Hasil pengujian 2 dengan kecepatan 10km/jam	49
Tabel 4.5. Hasil <i>confusion matrix</i> pengujian 2 dengan kecepatan 20km/jam	50
Tabel 4.6. Hasil pengujian 2 dengan kecepatan 20km/jam	50
Tabel 4.7. Hasil <i>confusion matrix</i> pengujian 2 dengan kecepatan 30km/jam	51
Tabel 4.8. Hasil pengujian 2 dengan kecepatan 30km/jam	52
Tabel 4.9. Hasil <i>confusion matrix</i> pengujian 2 dengan kecepatan 40km/jam	53
Tabel 4.10. Hasil pengujian 2 dengan kecepatan 40km/jam	53
Tabel 4.11. Hasil pengujian jarak dengan kecepatan 40km/jam	54
Tabel 4.12. Perbandingan Nilai Akurasi Tiap Skneario	59

BAB I

PENDAHULUAN

1.1 Latar Belakang

Mobilitas merupakan awal dari perkembangan sosial, budaya dan ekonomi. Sejak dulu manusia terus bergerak berpindah atas tuntutan kehidupan. Berbagai macam teknologi untuk memudahkan manusia dapat berpindah dari satu tempat ke tempat lain terus dikembangkan. Teknologi tersebut biasa kita sebut dengan alat transportasi. Arahnya, transportasi di masa depan tidak lagi dikendalikan oleh manusia melainkan mesin yang akan mengambil alih peran tersebut.

Pada pertengahan abad ini, kita akan menyaksikan bagaimana perubahan revolusioner transportasi akan berlangsung. Kendaraan akan dapat mengemudi sendiri dari tempat A ke tempat B, dan peran pengemudi hanya memberi perintah untuk menuju ke tujuan (Sirbu et al., 2018). Kendaraan tersebut biasa kita sebut dengan *self-driving car* atau *autonomous car*. *Autonomous car* biasanya dilengkapi dengan berbagai macam sensor dan aktuator yang dapat menghasilkan banyak data secara *real-time*. Kemudian data tersebut diproses dan dianalisis agar dapat mengambil keputusan secara tepat. Namun, syarat utama dari desain *autonomous car* adalah kemampuan untuk dapat berfungsi secara mandiri (Hussain & Zeadally, 2019).

Dalam sistem *autonomous car*, fitur dasar yang wajib dimiliki yaitu kemampuan untuk mendeteksi dan mengidentifikasi jalur lintasan, rintangan serta rambu lalu lintas. Dengan mendeteksi rambu lalu lintas akan menyediakan berbagai informasi, dari batas kecepatan hingga informasi tentang kemana dan kapan harus

berbelok. Mengikuti rambu lalu lintas dapat membantu memastikan keselamatan semua orang di jalan dengan meminimalkan risiko pengemudi menabrak kendaraan lain, pejalan kaki, atau pengendara sepeda.

Meskipun rambu lalu lintas merupakan komponen keselamatan di jalan raya, namun rambu lalu lintas masih tetap sering dilanggar oleh pengguna jalan itu sendiri. Menurut Handayani (Handayani et al., 2017), faktor yang berpengaruh terhadap potensi kecelakaan lalu lintas yaitu faktor pelanggaran lampu dan rambu sebesar 39.51%. Artinya, pola perilaku manusia yang abai terhadap rambu dan lampu lalu lintas masih cukup tinggi. Selain itu, data dari *U.S National Highway Traffic Safety Administration* juga menyatakan bahwa 94% kecelakaan lalu lintas serius disebabkan karena kesalahan pengambilan keputusan atau kelalaian oleh pengemudi. Dengan adanya sistem deteksi rambu lalu lintas dan juga sistem *autonomous car* akan mengurangi faktor kesalahan manusia serta memberi keamanan yang lebih untuk pengendara.

Maka dari itu, penulis mengusulkan judul “Sistem Deteksi Rambu dan Lampu Lalu Lintas untuk *Autonomous Car* Menggunakan *Faster R-CNN*”. Penelitian ini akan menggunakan *Faster R-CNN* untuk dapat mendeteksi dan mengklasifikasikan objek rambu dan lampu lalu lintas. Kemudian untuk performa sistem, penelitian ini menggunakan *Confusion Matrix* untuk mengetahui tingkat akurasinya.

1.2 Rumusan Masalah

1. Bagaimana penerapan arsitektur *Faster R-CNN* pada sistem deteksi rambu dan lampu lalu lintas?

2. Bagaimana unjuk kerja sistem deteksi rambu dan lampu lalu lintas menggunakan *Faster R-CNN*?

1.3 Tujuan Penelitian

Tujuan dari pengerjaan tugas akhir ini adalah:

1. Untuk menerapkan arsitektur *Faster R-CNN* pada sistem deteksi rambu dan lampu lalu lintas.
2. Untuk mengetahui unjuk kerja sistem deteksi rambu dan lampu lalu lintas menggunakan *Faster R-CNN*.

1.4 Manfaat Penelitian

Manfaat dari pengerjaan tugas akhir ini adalah:

1. Memberikan informasi terhadap bidang ilmu pengetahuan yang relevan serta bahan referensi ilmiah untuk penelitian selanjutnya.
2. Adanya sistem pendeteksi rambu dan lampu lalu lintas pada sistem *autonomous car*.

1.5 Batasan Masalah

Yang menjadi Batasan masalah dalam penelitian ini adalah:

1. Objek yang dideteksi adalah rambu stop, rambu batas kecepatan 40km/jam, rambu dilarang melintas, rambu dilarang stop, rambu dilarang parkir, rambu dilarang putar balik, rambu putar balik, rambu penyeberangan orang, rambu

perhatian, dan alat pemberi isyarat lalu lintas tiga warna (lampu merah, lampu kuning, dan lampu hijau).

2. Waktu pengambilan data dilakukan pada siang hari pukul 12.00 hingga pukul 15.00.
3. Kamera yang digunakan satu buah, diletakkan pada kaca mobil bagian depan, di balik spion tengah, dan menghadap ke depan.
4. Jenis mobil yang digunakan yaitu mobil penumpang dengan model MPV (*Multi Purpose Vehicle*).

1.6 Sistematika Penulisan

Berikut adalah gambaran singkat mengenai isi tulisan secara keseluruhan:

BAB I PENDAHULUAN

Bab ini berisi latar belakang masalah, rumusan masalah, tujuan penelitian, manfaat penelitian, batasan masalah, dan sistematika penulisan.

BAB II TINJAUAN PUSTAKA

Pada bab ini akan dijelaskan landasan teori yang menunjang setiap elemen percobaan yang dilakukan.

BAB III METODOLOGI PENELITIAN

Bab ini berisi tahapan penelitian, instrumen penelitian, proses pengumpulan data, perancangan sistem, implementasi algoritma, dan evaluasi kinerja sistem deteksi rambu dan lampu lalu lintas.

BAB IV HASIL DAN PEMBAHASAN

Bab ini menjelaskan secara keseluruhan hasil pengujian sistem deteksi rambu dan lampu lalu lintas.

BAB V PENUTUP

Bab ini berisi kesimpulan dari hasil penelitian dan uraian saran untuk pengembangan lebih lanjut

BAB II

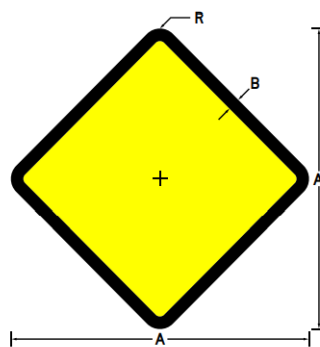
TINJAUAN PUSTAKA

2.1 Rambu Lalu Lintas dan Lampu Lalu Lintas

Berdasarkan *Peraturan Menteri Perhubungan Nomor PM 13 Tahun 2014* (Indonesia), Rambu Lalu Lintas adalah bagian perlengkapan Jalan yang berupa lambang, huruf, angka, kalimat, dan/atau perpaduan yang berfungsi sebagai peringatan, larangan, perintah, atau petunjuk bagi Pengguna Jalan. Rambu lalu lintas dapat dibedakan berdasarkan jenisnya yaitu:

1. Rambu Peringatan

Rambu peringatan digunakan untuk memberi peringatan kemungkinan ada bahaya di jalan atau tempat berbahaya pada jalan dan menginformasikan tentang sifat bahaya.



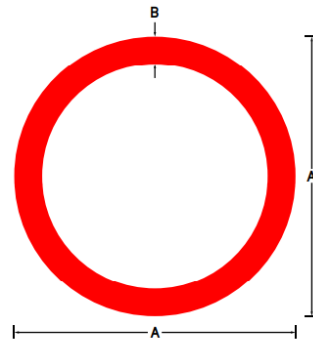
Jenis Ukuran	A	B	r
Kecil	450	25	37
Sedang	600	25	37
Besar	750	31	47
Sangat Besar	900	38	56

(dalam mm)

Gambar 2.1. Ukuran Rambu Peringatan

2. Rambu Larangan

Rambu larangan digunakan untuk menyatakan perbuatan yang dilarang dilakukan oleh Pengguna Jalan.



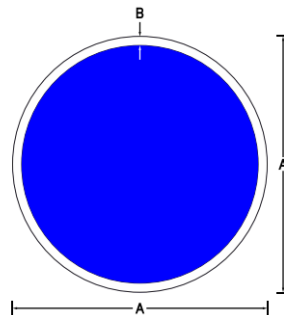
Jenis Ukuran	A	B
Kecil	450	45
Sedang	600	60
Besar	750	75
Sangat Besar	900	90

(dalam mm)

Gambar 2.2. Ukuran Rambu Larangan

3. Rambu Perintah

Rambu perintah digunakan untuk menyatakan perintah yang wajib dilakukan oleh Pengguna Jalan.



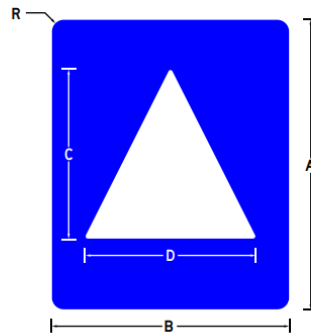
Jenis Ukuran	A	B
Kecil	450	20
Sedang	600	20
Besar	750	25
Sangat Besar	900	30

(dalam mm)

Gambar 2.3. Ukuran Rambu Perintah

4. Rambu Petunjuk

Rambu petunjuk digunakan untuk memandu Pengguna Jalan saat melakukan perjalanan atau untuk memberikan informasi lain kepada Pengguna Jalan.



Jenis Ukuran	A	B	C	D	r
Kecil	500	400	260	230	37
Sedang	600	500	350	350	37
Besar	750	600	430	460	47
Sangat Besar	900	750	520	580	56

(dalam mm)

Gambar 2.4. Ukuran Rambu Petunjuk

Penempatan dan pemasangan Rambu Lalu Lintas harus sesuai dengan jarak penempatan, ketinggian penempatan, jenis rambu, ukuran daun rambu, serta ukuran huruf, angka, dan simbol. Rambu Lalu Lintas dapat ditempatkan pada sisi jalan, dengan tinggi 265 (dua ratus enam puluh lima) sentimeter dan paling rendah 175 (seratus tujuh puluh lima) sentimeter diukur dari permukaan jalan tertinggi sampai dengan sisi daun rambu bagian bawah.

Selain rambu lalu lintas, lampu lalu lintas atau bisa juga disebut alat pemberi isyarat lalu lintas juga merupakan salah satu bagian perlengkapan jalan. Sesuai dengan Pasal 1, ayat (1) *Peraturan Menteri Perhubungan No. PM 49 Tahun 2014* (Indonesia), bahwa Alat Pemberi Isyarat Lalu Lintas adalah perangkat elektronik yang menggunakan isyarat lampu yang dapat dilengkapi dengan isyarat bunyi untuk mengatur lalu lintas orang dan/ atau Kendaraan di persimpangan atau pada ruas jalan. Ada tiga jenis alat pemberi isyarat lalu lintas, dibagi berdasarkan banyak warnanya yaitu lampu tiga warna, lampu dua warna, dan lampu satu warna.

Alat Pemberi Isyarat Lalu Lintas dengan lampu tiga warna dipergunakan untuk mengatur Kendaraan yang terdiri dari lampu berwarna merah, kuning, dan hijau. Lampu berwarna merah untuk menyatakan Kendaraan harus berhenti dan tidak boleh melewati marka melintang yang berfungsi sebagai garis henti. Lampu berwarna kuning untuk memberikan peringatan bagi pengemudi: (a) lampu berwarna kuning yang menyala sesudah lampu berwarna hijau padam, menyatakan lampu berwarna merah akan segera menyala, Kendaraan bersiap untuk berhenti; dan (b) lampu berwarna kuning yang menyala bersama dengan lampu berwarna merah, menyatakan lampu berwarna hijau akan segera menyala, Kendaraan bersiap untuk bergerak. Lampu berwarna hijau menyatakan Kendaraan berjalan.



Gambar 2.5. Alat pemberi isyarat lalu lintas (lampu lalu lintas)

2.2 *Autonomous Car*

Penjelasan atau gambaran tentang *Self-driving car* atau *Autonomous car* akan berbeda tiap orang. Beberapa orang mungkin mengartikannya sebagai kendaraan

yang berjalan sepenuhnya secara mandiri, tanpa memerlukan pengemudi sama sekali. Orang lain mungkin membayangkan kendaraan yang berjalan sendiri membuat keputusan otonom, tetapi itu masih membutuhkan pengemudi manusia yang duduk di belakang kemudi yang siap bertindak jika terjadi keadaan darurat (Sjafrie, 2020).

Menurut Thrun (Thrun, 2010), dalam perangkat lunak *Autonomous Car* umumnya memasukkan tiga area fungsional utama: persepsi, perencanaan, dan kontrol. Persepsi membahas masalah pemetaan data sensor ke dalam keyakinan internal dan prediksi tentang lingkungan. Perencanaan membahas masalah membuat keputusan mengemudi. Kontrol kemudian menggerakkan roda kemudi, *throttle*, rem, dan kontrol kendaraan lainnya. Antarmuka modul perangkat lunak tambahan ke kendaraan dan sensornya, dan menyediakan layanan menyeluruh seperti pencatatan data dan fungsionalitas pengawas.

Ada beberapa definisi untuk berbagai tingkat otomatisasi dan untuk beberapa waktu ada kebutuhan akan standarisasi untuk membantu kejelasan dan konsistensi. Tingkatan ini mengadopsi aturan *SAE International (Society of Automotive Engineers)* dan membagi kendaraan menjadi beberapa tingkatan berdasarkan “siapa melakukan apa, kapan”. Umumnya:

- Pada SAE Level 0, pengemudi manusia melakukan segalanya;
- Pada SAE Level 1, sistem otomatis pada kendaraan terkadang dapat membantu pengemudi manusia melakukan beberapa bagian dari tugas mengemudi;

- Pada SAE Level 2, sistem otomatis pada kendaraan benar-benar dapat melakukan beberapa bagian dari tugas mengemudi, sementara manusia terus memantau lingkungan mengemudi dan melakukan sisa tugas mengemudi;
- Pada SAE Level 3, sistem otomatis dapat benar-benar melakukan beberapa bagian dari tugas mengemudi dan memantau lingkungan mengemudi dalam beberapa kasus, tetapi pengemudi manusia harus siap untuk mengambil kembali kendali saat sistem otomatis meminta;
- Pada SAE Level 4, sistem otomatis dapat melakukan tugas mengemudi dan memantau lingkungan mengemudi, dan manusia tidak perlu mengambil kembali kendali, tetapi sistem otomatis hanya dapat beroperasi di lingkungan tertentu dan dalam kondisi tertentu; dan
- Pada SAE Level 5, sistem otomatis dapat melakukan semua tugas mengemudi, dalam semua kondisi yang dapat dilakukan oleh pengemudi manusia. (*Federal Automated Vehicles Policy Accelerating the Next Revolution In Roadway Safety*, 2016).

2.3 Visi Komputer

Baik itu *medical diagnosis*, *self-driving vehicles*, *camera monitoring*, atau *smart filters*, banyak aplikasi di bidang visi komputer terkait erat dengan kehidupan kita saat ini dan masa depan. Dalam beberapa tahun terakhir, pembelajaran mendalam telah menjadi kekuatan transformatif untuk memajukan kinerja sistem visi komputer. Dapat dikatakan bahwa aplikasi *computer vision* tercanggih hampir tidak dapat dipisahkan dari *deep learning* (Zhang et al., 2022).

2.3.1 Image Augmentation

Augmentasi gambar menghasilkan contoh pelatihan yang serupa tetapi berbeda setelah serangkaian perubahan acak pada gambar latih, sehingga memperluas ukuran *training set*. Sebagai alternatif, *augmentasi* gambar dapat dimotivasi oleh fakta bahwa perubahan acak dari contoh pelatihan memungkinkan model untuk tidak terlalu bergantung pada atribut tertentu, sehingga meningkatkan kemampuan generalisasinya. Misalnya, gambar dapat dipotong dengan cara yang berbeda untuk membuat objek yang diinginkan muncul di posisi yang berbeda, sehingga mengurangi ketergantungan model pada posisi objek. *Augmentasi* juga dapat menyesuaikan faktor seperti kecerahan dan warna untuk mengurangi sensitivitas model terhadap warna (Zhang et al., 2022).

2.3.2 Bounding Box

Dalam pendeteksian objek, untuk menggambarkan lokasi spasial suatu objek biasanya menggunakan *bounding box*. *Bounding Box* berbentuk persegi panjang, yang ditentukan oleh koordinat x dan y dari sudut kiri atas persegi panjang dan koordinat sudut kanan bawah tersebut. Representasi *bounding box* yang umum digunakan lainnya adalah koordinat sumbu (x, y) dari pusat *bounding box*, dan lebar dan tinggi kotak (Zhang et al., 2022).

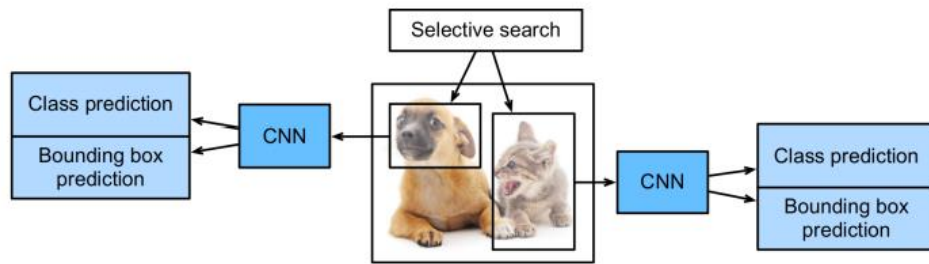
2.3.3 Anchor Boxes

Algoritma pendeteksian objek biasanya mengambil sampel sejumlah besar *region* dalam gambar masukan, menentukan apakah wilayah ini berisi objek yang menarik, dan menyesuaikan batas wilayah untuk memprediksi kebenaran dari *bounding box* objek dengan lebih akurat. Model yang berbeda dapat mengadopsi skema pengambilan sampel *region* yang berbeda. Salah satu metode tersebut menghasilkan beberapa *bounding box* dengan berbagai skala dan rasio aspek yang berpusat pada setiap piksel. *Bounding Box* ini disebut *Anchor Box* (Zhang et al., 2022).

2.4 Region Based CNN

2.4.1 R-CNN

R-CNN pertama-tama mengekstrak banyak (misalnya, 2000) *region proposals* dari gambar input (misalnya, *anchor boxes* juga dapat dianggap sebagai *region proposals*), memberi label pada kelas dan *bounding box* (misalnya, *offset*) (Girshick et al., 2013). Kemudian CNN digunakan untuk melakukan *forward propagation* pada setiap *region proposals* untuk mengekstrak fitur-fiturnya. Selanjutnya, fitur masing-masing *region proposals* digunakan untuk memprediksi kelas dan *bounding box* pada *region proposals* ini.



Gambar 2.6. Model R-CNN (Sumber: Zhang et al., 2022).

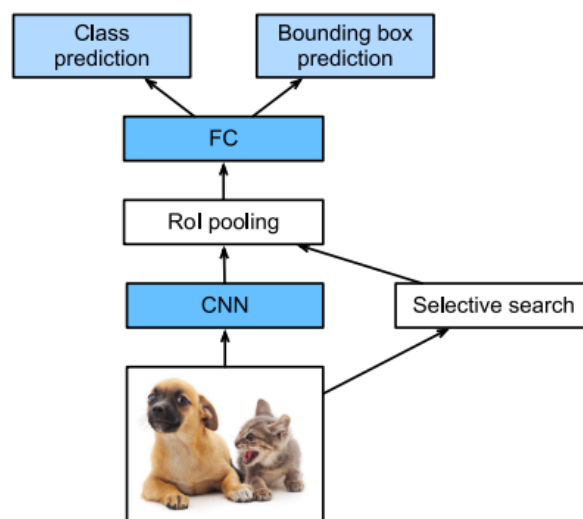
Lebih konkretnya, R-CNN terdiri dari empat langkah berikut:

1. Melakukan pencarian selektif untuk mengekstrak beberapa proposal wilayah berkualitas tinggi pada gambar input. Wilayah yang diusulkan ini biasanya dipilih pada beberapa skala dengan bentuk dan ukuran yang berbeda. Setiap proposal wilayah akan diberi label dengan kelas dan *bounding box*.
2. Memilih CNN yang telah dilatih sebelumnya dan potong sebelum lapisan keluaran. Ubah ukuran setiap proposal wilayah ke ukuran input yang diperlukan oleh jaringan, dan keluarkan fitur yang diekstraksi untuk proposal wilayah melalui *forward propagation*.
3. Mengambil fitur yang diekstraksi dan kelas berlabel dari setiap proposal wilayah sebagai contoh. Latih beberapa mesin vektor pendukung untuk mengklasifikasikan objek, di mana setiap mesin vektor pendukung secara individual menentukan apakah contoh berisi kelas tertentu.
4. Mengambil fitur yang diekstraksi dan *bounding box* berlabel setiap proposal wilayah sebagai contoh. Latih model regresi linier untuk memprediksi *bounding box*.

Meskipun model R-CNN menggunakan CNN yang telah dilatih sebelumnya untuk mengekstrak fitur gambar secara efektif, namun lambat. Bayangkan kita memilih ribuan proposal wilayah dari satu gambar input: ini membutuhkan ribuan CNN *forward propagation* untuk melakukan deteksi objek. Beban komputasi yang besar ini membuat R-CNN tidak dapat digunakan secara luas dalam aplikasi dunia nyata. (Zhang et al., 2022).

2.4.2 Fast R-CNN

Hambatan performa R-CNN terletak pada *forward propagation* CNN untuk setiap *regional proposals*, tanpa berbagi perhitungan. Karena region ini biasanya tumpang tindih, ekstraksi fitur independen menyebabkan banyak perhitungan berulang. Salah satu peningkatan besar *Fast R-CNN* dari R-CNN adalah bahwa CNN *forward propagation* dilakukan untuk setiap proposal objek dengan saling berbagi komputasi (Girshick, 2015).



Gambar 2.7. Model *Fast R-CNN* (Sumber: Zhang et al., 2022).

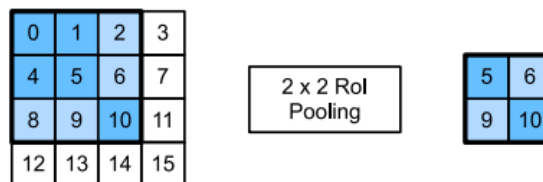
Berdasarkan Gambar 2.7. Perhitungan utamanya adalah sebagai berikut:

1. Dibandingkan dengan R-CNN, pada *fast* R-CNN, input dari CNN untuk ekstraksi fitur adalah seluruh citra, bukan proposal region individual. Apalagi CNN ini bisa dilatih. Diberikan gambar input, biarkan bentuk *output* CNN menjadi $1 \times c \times h1 \times w1$ dimana c , h , dan w adalah *channel*, *height* dan *width* dari *feature map*.
2. Misalkan pencarian selektif menghasilkan n proposal wilayah. Proposal wilayah ini (dengan bentuk yang berbeda) menandai wilayah yang diminati (dengan bentuk yang berbeda) pada keluaran CNN. Kemudian daerah yang menarik ini mengekstrak fitur lebih lanjut dari bentuk yang sama (misalnya tinggi $h2$ dan lebar $w2$ ditentukan) agar mudah digabungkan. Untuk mencapai hal ini, *Fast* R-CNN memperkenalkan lapisan *region of interest pooling* (RoI): *output* CNN dan proposal region dimasukkan ke dalam lapisan ini, menghasilkan fitur gabungan dari bentuk $n \times c \times h2 \times w2$ yang selanjutnya diekstraksi untuk semua usulan daerah.
3. Menggunakan lapisan yang terhubung penuh, ubah fitur gabungan menjadi *output* bentuk $n \times d$, di mana d bergantung pada desain model.
4. Prediksi kelas dan *bounding box* untuk masing-masing dari n proposal wilayah. Lebih konkretnya, dalam prediksi kelas dan *bounding box*, ubah output lapisan yang terhubung penuh menjadi

output bentuk $n \times q$ (q adalah jumlah kelas) dan output bentuk $n \times 4$, masing-masing. Prediksi kelas menggunakan regresi *softmax*.

Pada *pooling* layer, kita secara tidak langsung mengontrol bentuk output dengan menentukan ukuran *pooling window*, *padding*, dan *stride*. Sebaliknya, kita dapat secara langsung menentukan bentuk keluaran di *ROI pooling layer*.

Sebagai contoh, mari kita tentukan tinggi dan lebar keluaran untuk setiap wilayah sebagai h_2 dan w_2 , masing-masing. Untuk setiap daerah yang diinginkan jendela berbentuk $h \times w$, jendela ini dibagi menjadi $h_2 \times w_2$ grid subwindows, di mana bentuk setiap *subwindow* kira-kira $(h/h_2) \times (w/w_2)$. Dalam prakteknya, tinggi dan lebar setiap subwindow harus dibulatkan ke atas, dan elemen terbesar harus digunakan sebagai output dari subwindow. Oleh karena itu, region of interest pooling layer dapat mengekstrak fitur dari bentuk yang sama bahkan ketika region yang diinginkan memiliki bentuk yang berbeda.



Gambar 2.8. 2×2 region of interest pooling layer (Sumber: Zhang et al., 2022).

Sebagai contoh ilustratif, pada Gambar 2.8, wilayah 3×3 kiri atas yang diinginkan dipilih pada input 4×4 . Untuk wilayah yang diminati ini,

menggunakan *pooling layer* wilayah 2×2 untuk mendapatkan *output* 2×2 . Perhatikan bahwa masing-masing dari empat *subwindow* yang dibagi berisi elemen 0, 1, 4, dan 5 (5 adalah maksimum); 2 dan 6 (6 adalah maksimum); 8 dan 9 (9 adalah maksimum); dan 10 (Zhang et al., 2022).

2.4.3 *Faster R-CNN*

Agar lebih akurat dalam pendeteksian objek, model *Fast R-CNN* biasanya harus menghasilkan banyak proposal wilayah dalam pencarian selektif. Untuk mengurangi proposal wilayah tanpa kehilangan akurasi, *Faster R-CNN* mengusulkan untuk menggantikan pencarian selektif dengan *Region Proposal Network* (Ren et al., 2017).

Dibandingkan dengan *Fast R-CNN*, *Faster R-CNN* hanya mengubah metode proposal wilayah dari pencarian selektif ke *Region Proposal Network*. Sisa model tetap tidak berubah. *Region Proposal Network* bekerja dalam langkah-langkah berikut:

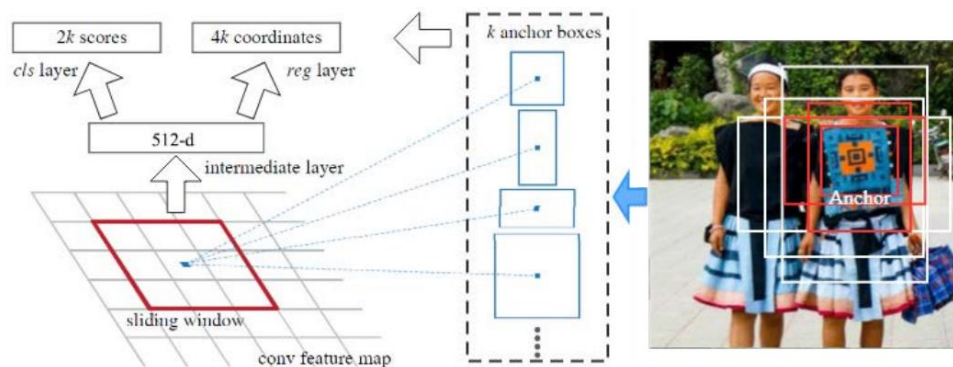
1. Menggunakan *convolutional layer* 3×3 dengan *padding* 1 untuk mengubah *output CNN* menjadi *output* baru dengan *c channel*. Dengan cara ini, setiap unit di sepanjang dimensi spasial peta fitur yang diekstraksi CNN mendapatkan vektor fitur baru dengan panjang *c*.
2. Berpusat pada setiap piksel peta fitur, buat beberapa *anchor boxes* dengan skala dan rasio aspek yang berbeda dan beri label.

3. Menggunakan vektor fitur panjang-c di tengah setiap *anchor boxes*, prediksi kelas biner (latar belakang atau objek) dan *bounding box* untuk *anchor boxes* ini.
4. Pertimbangkan *bounding box* yang diprediksi yang kelas prediksinya adalah objek. Hapus hasil yang tumpang tindih menggunakan penekanan non-maksimum. *Bounding box* prediksi yang tersisa untuk objek adalah proposal wilayah yang diperlukan oleh *pooling layer* wilayah yang diminati.

Perlu dicatat bahwa, sebagai bagian dari model *Faster R-CNN*, *Region Proposal Network* dilatih bersama dengan model lainnya. Dengan kata lain, fungsi tujuan dari *Faster R-CNN* tidak hanya mencakup prediksi kelas dan *bounding box* dalam deteksi objek, tetapi juga prediksi kelas biner dan *bounding box* dari *anchor boxes* di *Region Proposal Network*. Sebagai hasil dari pelatihan *end-to-end*, jaringan proposal region mempelajari cara menghasilkan proposal region berkualitas tinggi, agar tetap akurat dalam pendeteksian objek dengan pengurangan jumlah proposal region yang dipelajari dari data (Zhang et al., 2022).

Faster R-CNN menyediakan metode konfigurasi VGG16, ResNet_v1 dan model pra-pelatihan lainnya. Pertama, peta fitur dihasilkan oleh ekstraksi fitur dari gambar dalam kumpulan data. Kemudian ditransfer ke Region Proposal Network (RPN) untuk diproses. Proses utama adalah: menghasilkan *anchor*, klasifikasi *softmax*, *bounding box regression*, dan menghasilkan proposal. Pada peta fitur, setiap piksel dapat dianggap sebagai

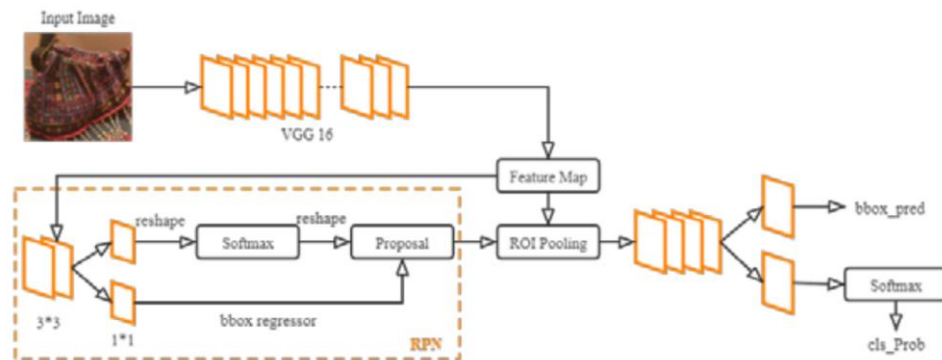
anchor. RPN menghasilkan beberapa *anchor box* untuk setiap titik *anchor* dengan menyesuaikan koefisien. Area kandidat ini diubah menjadi fitur yang terhubung penuh dengan konvolusi, dan pemetaan dari fitur ke lapisan klasifikasi dan lapisan regresi diwujudkan dengan menggunakan *kernel* konvolusi $2*k$ dan $4*k$. Pada lapisan klasifikasi, klasifikasi *softmax* digunakan untuk menentukan apakah area kandidat termasuk dalam latar depan (yaitu target); di lapisan regresi, regresi *bbox* digunakan untuk memodifikasi area kandidat, sehingga lebih dekat ke area nyata yang telah ditetapkan (yaitu kotak yang ditandai selama *preprocessing*). Kemudian, lapisan proposal akan menyesuaikan titik *anchor* sesuai dengan hasil regresi, dan memilih beberapa kandidat area dengan skor prospek tinggi sebagai proposal. Prinsip RPN ditunjukkan pada Gambar 2.9.



Gambar 2.9. Diagram Prinsip RPN (Sumber: Kong et al., 2020).

Setelah itu, *ROI pooling layer* akan mengumpulkan proposal dan peta fitur untuk menghasilkan peta fitur proposal. Peta fitur proposal keluaran dihitung melalui lapisan koneksi penuh dan klasifikasi *softmax* untuk menentukan kategori mana yang dimiliki setiap proposal, keluaran vektor probabilitas *cls_prob*, dan mendapatkan posisi *offset bbox_pred* dari setiap

proposal dengan regresi *bbox* untuk mengekstrak bingkai deteksi target yang lebih akurat. Parameter keluaran seperti *cls_prob* dan *bbox_pred* akan digunakan untuk mengoptimalkan model data keluaran pelatihan CNN dan meningkatkan kinerja deteksi *Faster R-CNN*. Proses operasi dasar *Faster R-CNN* ditunjukkan pada Gambar 2.10 (Kong et al., 2020).

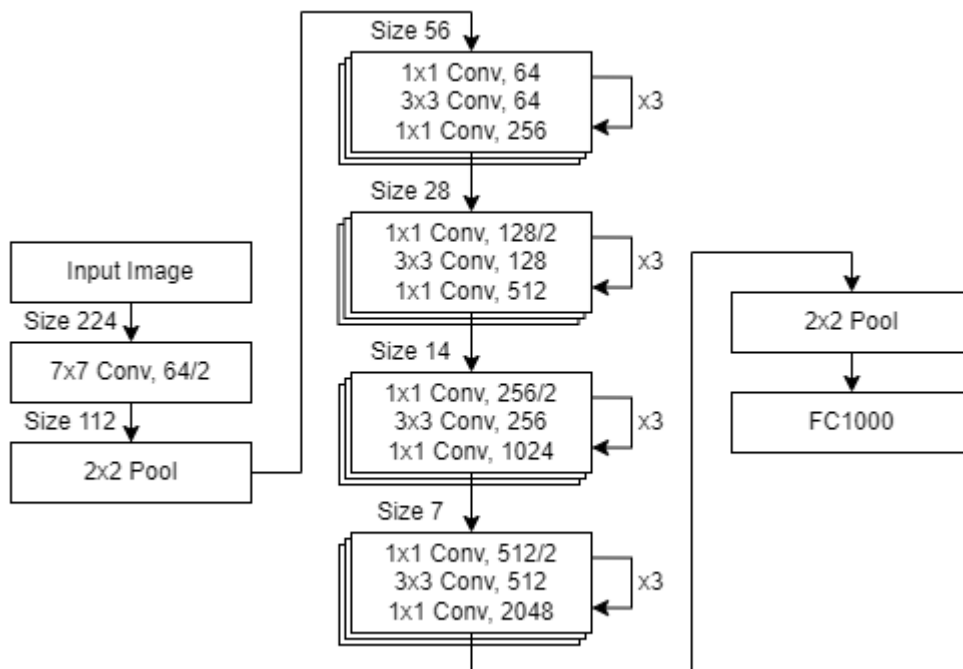


Gambar 2.10. *Faster R-CNN basic operation process* (Sumber: Kong et al., 2020).

2.5 Residual Network

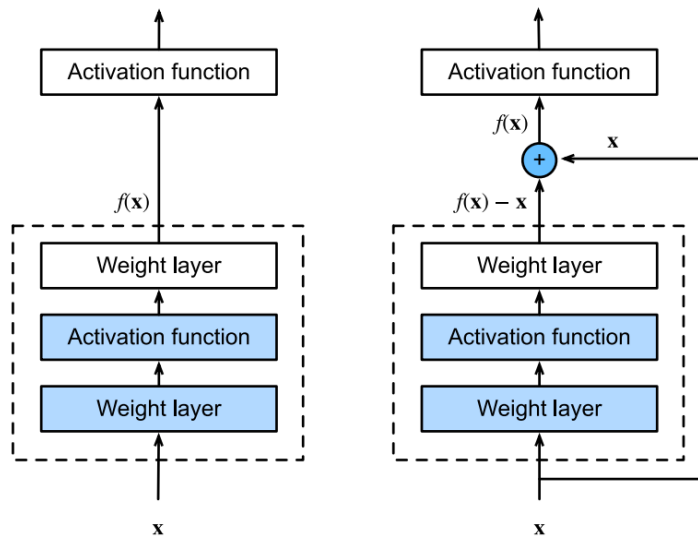
Operasi yang paling penting di CNN adalah *Convolutional Layer*. *Convolutional Layer* merupakan blok inti dari *Convolutional Neural Network* yang melakukan sebagian besar komputasi berat. Salah satu contoh model *Convolutional Layer* yang cukup populer adalah *ResNet*. *ResNet50* adalah Jaringan *Residual* yang berisi 50 layer. Seperti VGG, ukuran lapisan konvolusi adalah filter 3×3 , ukuran input model ini ditetapkan sebagai 224×224 . dan mereka mengikuti beberapa desain sederhana seperti: Untuk lapisan yang memiliki jumlah filter yang sama memiliki keluaran yang sama. Jumlah filter menjadi dua kali lipat jika ukuran keluaran yang terbelit dibelah dua sehingga kompleksitas waktu per lapisan

dipertahankan. Model diakhiri dengan *pooling layer* rata-rata dan 1000-way *fully-connected layer* dengan *softmax*. Model ini memiliki lebih sedikit filter dan kompleksitas yang lebih rendah daripada jaring VGG dan ada varian lain seperti *ResNet101* dan *ResNet152* (Bendjillali et al., 2020). Gambar 2.11 menunjukkan lapisan konfigurasi jaringan *ResNet50*.



Gambar 2.11. *ResNet50 Architecture*

Inti dari *Residual Network* (ResNet) adalah gagasan bahwa setiap lapisan tambahan harus lebih mudah memuat fungsi identitas sebagai salah satu elemennya. Pertimbangan ini agak mendalam tetapi mengarah pada solusi yang sangat sederhana, *Residual Block*.



Gambar 2.12. Blok *Regular* (Kiri) dan Blok *Residual* (Kanan)

(Sumber: Zhang et al., 2022).

Seperti yang digambarkan pada Gambar 2.12. Nilai input dinyatakan dengan x . Di sebelah kiri Gambar 2.12, bagian dalam kotak garis putus-putus harus langsung mempelajari pemetaan $f(x)$. Di sebelah kanan, bagian dalam kotak garis putus-putus perlu mempelajari pemetaan *residual* $f(x) - x$, yang merupakan cara blok residual mendapatkan namanya. Jika pemetaan identitas $f(x) = x$ adalah pemetaan dasar yang diinginkan, pemetaan residual lebih mudah dipelajari: kita hanya perlu mendorong bobot dan bias lapisan bobot atas (misalnya, *fully connected layer* dan *convolutional layer*) di dalam kotak garis putus-putus menjadi nol. Gambar kanan pada Gambar 2.12 mengilustrasikan blok *residual* ResNet, di mana garis padat yang membawa input lapisan x ke operator tambahan disebut sambungan residual (atau sambungan pintasan). Dengan blok residual, input dapat menyebar lebih cepat melalui koneksi residual di seluruh lapisan.

ResNet mengikuti desain lapisan konvolusi 3×3 penuh VGG. Blok residual memiliki dua lapisan konvolusi 3×3 dengan jumlah saluran keluaran yang sama. Setiap lapisan konvolusi diikuti oleh lapisan normalisasi batch dan fungsi aktivasi ReLU. Kemudian, kita lewati dua operasi konvolusi ini dan menambahkan input langsung sebelum fungsi aktivasi ReLU terakhir. Jenis desain ini mensyaratkan bahwa output dari dua lapisan konvolusi harus memiliki bentuk yang sama dengan input, sehingga dapat dijumlahkan. Jika kita ingin mengubah jumlah saluran, kita perlu memperkenalkan lapisan konvolusi 1×1 tambahan untuk mengubah input menjadi bentuk yang diinginkan untuk operasi penambahan (Zhang et al., 2022).

2.6 Confusion Matrix

Confusion Matrix adalah matriks yang menunjukkan klasifikasi prediksi dan aktual. Sebuah matriks konfusi berukuran $l \times l$, di mana l adalah jumlah nilai label atau kelas yang berbeda (Kohavi & Provost, 1998). Dalam *multi-class confusion matrix*, kelas-kelas dicantumkan dalam urutan yang sama dalam baris seperti pada kolom Gambar 2.13, oleh karena itu elemen yang diklasifikasikan dengan benar (True Positif) terletak pada diagonal utama dari kiri atas ke kanan bawah dan merupakan titik pertemuan antara prediksi dan aktual (Grandini et al., 2020).

		PREDICTED classification				Total
		Classes	a	b	c	
ACTUAL classification	a	6	0	1	2	9
	b	3	9	1	1	14
	c	1	0	10	2	13
	d	1	2	1	12	16
Total		11	11	13	17	52

Gambar 2.13. *Multi-Class Confusion Matrix* (Sumber: Grandini et al., 2020).

Precision adalah pecahan elemen *True Positive* dibagi dengan jumlah total unit yang diprediksi secara positif (jumlah kolom dari prediksi positif). Secara khusus, *True Positive* adalah elemen yang telah diberi label positif oleh model dan sebenarnya positif, sedangkan *False Positive* adalah elemen yang telah diberi label positif oleh model, tetapi sebenarnya negatif.

$$Precision = \frac{TP}{TP+FP} \quad (2.1)$$

Recall adalah pecahan elemen *True Positive* dibagi dengan jumlah total unit yang diklasifikasikan secara positif (jumlah baris yang sebenarnya bernilai positif). Khususnya *False Negative* adalah elemen yang telah diberi label negatif oleh model, tetapi sebenarnya positif.

$$Recall = \frac{TP}{TP+FN} \quad (2.2)$$

F1-Score menilai kinerja model klasifikasi dari *confusion matrix*. Rumus *F1-score* dapat diartikan sebagai rata-rata antara *Precision* dan *Recall*, di mana *F1-score* mencapai nilai terbaiknya pada 1 dan skor terburuk pada 0. Kontribusi relatif

dari *precision* dan *recall* adalah sama pada *F1-score* dan rata-rata harmonik berguna untuk menemukan *trade-off* terbaik antara dua kuantitas (Grandini et al., 2020).

$$F_1 \text{ Score} = 2 \times \left(\frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \right) \quad (2.3)$$

Akurasi adalah salah satu metrik yang paling populer dalam klasifikasi *multi-class*. Rumus Akurasi mempertimbangkan jumlah elemen *True Positive* dan *True Negative* pada pembilang dan jumlah semua entri *confusion matrix* pada penyebut.

$$\text{Akurasi} = \frac{TP+TN}{TP+TN+FP+FN} \quad (2.4)$$

Akurasi mengembalikan ukuran keseluruhan tentang seberapa banyak model memprediksi dengan benar pada seluruh kumpulan data. Elemen dasar metrik adalah individu tunggal dalam kumpulan data: setiap unit memiliki bobot yang sama dan mereka berkontribusi sama pada nilai Akurasi. Ketika kita berpikir tentang kelas daripada individu tunggal (keseluruhan isi tabel), akan ada kelas dengan jumlah unit yang tinggi dan yang lain hanya dengan sedikit. Dalam situasi ini, kelas yang padat akan memiliki bobot yang lebih tinggi dibandingkan dengan kelas yang terkecil. Oleh karena itu, Akurasi paling cocok ketika kita hanya peduli pada individu tunggal daripada beberapa kelas. (Grandini et al., 2020).

Dalam menentukan nilai sel pada *confusion matrix*, nilai TP, FP, FN, dan TN berbeda untuk setiap kelas. Berikut merupakan cara dalam menentukan nilai pada *confusion matrix*:

a. Menentukan Nilai pada Kelas A

Tabel 2.1. Nilai *Confusion Matrix* pada Kelas A

		Predicted Classification			
		Classes	a	b	c
Actual Classification	a	TP	FN	FN	FN
	b	FP	TN	TN	TN
	c	FP	TN	TN	TN
	d	FP	TN	TN	TN

Dalam menentukan nilai pada setiap sel pada *confusion matrix*, dapat dilihat pada tabel 2.1. Kelas sel didefinisikan dengan (kolom prediksi, baris aktual). Nilai TP untuk kelas A terdapat pada sel (a, a). Nilai FP untuk kelas A yaitu sel pada kolom prediksi (a) yang bukan merupakan nilai TP, yaitu sel (a, b), (a, c) dan (a, d). Nilai FN untuk kelas A yaitu sel pada baris aktual (a) yang bukan merupakan nilai TP, yaitu sel (b, a), (c, a) dan (d, a). Nilai TN untuk kelas A yaitu sel yang bukan merupakan TP, FP, dan FN, yaitu sel (b, b), (b, c), (b, d), (c, b), (c, c), (c, d), (d, b), (d, c) dan (d, d).

b. Menentukan Nilai pada Kelas B

Tabel 2.2. Nilai *Confusion Matrix* pada Kelas B

		Predicted Classification			
		Classes	a	b	c
Actual Classification	a	TN	FP	TN	TN
	b	FN	TP	FN	FN
	c	TN	FP	TN	TN
	d	TN	FP	TN	TN

Dalam menentukan nilai pada setiap sel pada *confusion matrix*, dapat dilihat pada tabel 2.2. Kelas sel didefinisikan dengan (kolom prediksi, baris aktual). Nilai TP untuk kelas B terdapat pada sel (b, b). Nilai FP untuk kelas B yaitu sel pada kolom prediksi (b) yang bukan merupakan nilai TP, yaitu sel (b, a), (b, c) dan (b, d). Nilai FN untuk kelas B yaitu sel pada baris aktual (b) yang bukan merupakan nilai TP, yaitu sel (a, b), (c, b) dan (d, b). Nilai TN untuk kelas B yaitu sel yang bukan merupakan TP, FP, dan FN, yaitu sel (a, a), (c, a), (d, a), (a, c), (c, c), (d, c), (a, d), (c, d) dan (d, d).

c. Menentukan Nilai pada Kelas C

Tabel 2.3. Nilai *Confusion Matrix* pada Kelas C

		Predicted Classification			
		Classes	a	b	c
Actual Classification	a	TN	TN	FP	TN
	b	TN	TN	FP	TN
	c	FN	FN	TP	FN
	d	TN	TN	FP	TN

Dalam menentukan nilai pada setiap sel pada *confusion matrix*, dapat dilihat pada tabel 2.3. Kelas sel didefinisikan dengan (kolom prediksi, baris aktual). Nilai TP untuk kelas C terdapat pada sel (c, c). Nilai FP untuk kelas C yaitu sel pada kolom prediksi (c) yang bukan merupakan nilai TP, yaitu sel (c, a), (c, b) dan (c, d). Nilai FN untuk kelas C yaitu sel pada baris aktual (c) yang bukan merupakan nilai TP, yaitu sel (a, c), (b, c) dan (d, c). Nilai TN untuk kelas C yaitu sel yang bukan merupakan TP, FP, dan FN, yaitu sel (a, a), (b, a), (d, a), (a, b), (b, b), (d, b), (a, d), (b, d) dan (d, d).

d. Menentukan Nilai pada Kelas D

Tabel 2.4. Nilai *Confusion Matrix* pada Kelas D

		Predicted Classification			
		Classes	a	b	c
Actual Classification	a	TN	TN	TN	FP
	b	TN	TN	TN	FP
	c	TN	TN	TN	FP
	d	FN	FN	FN	TP

Dalam menentukan nilai pada setiap sel pada *confusion matrix*, dapat dilihat pada tabel 2.4. Kelas sel didefinisikan dengan (kolom prediksi, baris aktual). Nilai TP untuk kelas D terdapat pada sel (d, d). Nilai FP untuk kelas D yaitu sel pada kolom prediksi (d) yang bukan merupakan nilai TP, yaitu sel (d, a), (d, b) dan (d, c). Nilai FN untuk kelas D yaitu sel pada baris aktual (d) yang bukan merupakan nilai TP, yaitu sel (a, d), (b, d) dan (c, d). Nilai TN untuk kelas C yaitu sel yang bukan merupakan TP, FP, dan FN, yaitu sel (a, a), (b, a), (c, a), (a, b), (b, b), (c, b), (a, c), (b, c) dan (c, c).

2.7 Jarak Aman Pengereman

Pada sistem *autonomous car*, jarak aman pengereman berpengaruh pada proses pengambilan keputusan oleh sistem agar tidak menabrak. Jarak pengereman harus dipahami oleh pembuat sistem sebagai parameter dasar yang penting untuk memperkirakan visibilitas jarak yang terjangkau. Jarak pengereman kendaraan dipengaruhi oleh sejumlah faktor yang berkaitan dengan kendaraan, jalan dan perilaku pengemudi. Faktor yang paling penting adalah: Kecepatan kendaraan, gesekan ban-jalan, teknik pengereman, sistem dan kondisi pengereman, tekanan

ban, kedalaman ulir ban dan kemampuan permukaan jalan, serta kemiringan vertikal jalan (Sabri et al., 2018). Berikut merupakan Jarak aman pengereman dapat dilihat pada Tabel 2.5.

Tabel 2.5. Jarak pengereman (Sumber: Sabri et al., 2018)

No.	Kecepatan (km/jam)	Jarak Pengereman (m)
1	10	0.54
2	20	2.24
3	30	5.55
4	40	10.91
5	50	14.51