

SKRIPSI

**DETEKSI LAJUR JALAN SECARA *REAL-TIME* PADA *AUTONOMOUS*
CAR MENGGUNAKAN METODE *SEMANTIC SEGMENTATION***

Disusun dan diajukan oleh
NUBLAN AZQALANI MUIS
D121171521



DEPARTEMEN TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS HASANUDDIN
MAKASSAR

2022

LEMBAR PENGESAHAN SKRIPSI

**DETEKSI LAJUR JALAN SECARA REAL-TIME PADA AUTONOMOUS
CAR MENGGUNAKAN METODE SEMANTIC SEGMENTATION**

Disusun dan diajukan oleh

NUBLAN AZQALANI MUIS

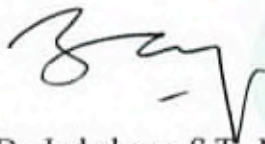
D121171521

Telah dipertahankan di hadapan Panitia Ujian yang dibentuk dalam rangka Penyelesaian Studi Program Sarjana Program Studi Teknik Informatika Fakultas Teknik Universitas Hasanuddin pada tanggal 16 November 2022 dan dinyatakan telah memenuhi syarat kelulusan.

Menyetujui,

Pembimbing Utama,

Pembimbing Pendamping,



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



A. Ais Prayogi, ST., M.Eng
Nip. 198305102014041001

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 : Nublan Azqalani Muis

NIM : D121171521

Departemen : Teknik Informatika

Jenjang : S1

Menyatakan dengan ini karya tulisan saya berjudul:

DETEKSI LAJUR JALAN SECARA *REAL-TIME* PADA *AUTONOMOUS CAR* MENGGUNAKAN METODE *SEMANTIC SEGMENTATION*

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, 17 November 2022

Yang menyatakan,



Nublan Azqalani Muis

KATA PENGANTAR

Segala puji syukur penulis panjatkan atas ke hadirat Allah SWT yang telah memberikan berkat, rahmat, karunia dan pertolongan-Nya dalam menyelesaikan skripsi ini dengan judul “**DETEKSI LAJUR JALAN SECARA *REAL-TIME* PADA *AUTONOMOUS CAR* MENGGUNAKAN METODE *SEMANTIC SEGMENTATION*”** sehingga dapat terselesaikan dengan baik. Skripsi ini merupakan syarat menyelesaikan jenjang pendidikan Strata-I pada program studi Teknik Informatika Universitas Hasanuddin.

Sholawat serta salam kepada Nabi Muhammad SAW beserta junjungannya yang membawa kita dari zaman kebodohan menuju zaman yang penuh dengan ilmu pengetahuan.

Penulis menyadari bahwa selama pengerjaan skripsi ini banyak kendala yang dihadapi, namun berkat bantuan dan dukungan dari berbagai pihak, skripsi ini dapat selesai di waktu yang tepat. Oleh karena itu pada kesempatan ini penulis ingin berterima kasih dan memberikan apresiasi sebesar-besarnya kepada:

1. Kedua Orang tua penulis, Bapak Abdul Muis Said dan Ibu Sri Wahyuni yang selalu memberikan dukungan moril maupun materil selama penulis menempuh pendidikan dan sampai menyelesaikan studi. Tak lupa pula saudara kandung penulis.
2. Kepada Bapak Dr. Indrabayu, S.T., M.T., M.Bus.sys., selaku pembimbing satu dan sebagai penasihat akademik penulis yang tidak hanya memberikan ilmu

tetapi senantiasa memberikan nasihat dan motivasi kepada penulis selama proses pengerjaan skripsi.

3. Kepada Bapak A. Ais Prayogi Alimuddin, S.T., M.Eng. selaku pembimbing dua yang menyediakan waktu, tenaga dan pikiran yang luar biasa dalam menyelesaikan skripsi.
4. Kepada ibu Anugrayani Bustamin, S.T., M.T. dan ibu Dr. Eng. Intan Sari Areni, S.T., M.T yang selalu memberikan dukungan kepada penulis dan kepada seluruh Tenaga Pendidik dan Staff Teknik Informatika Universitas Hasanuddin.
5. Kepada keluarga Bantaeng yang telah sangat banyak membantu penulis dalam segala masalah selama proses penyelesaian skripsi
6. Kepada keluarga Sidrap yang begitu banyak pula membantu penulis sehingga kelancaran pembuatan skripsi dapat tercapai.
7. Teman-teman Autonomous Car penulis: Muh. Fadhil, Kahirul Hidayat dan Taslinda yang membantu penulis selama pengambilan data hingga bersama-sama menyelesaikan skripsi ini hingga selesai.
8. Ilmi, Rieka, Fahmi, Alfa, Irzam, Fauzan, Herlina, Ikbal, Ikhsan, Wahyudi, Ilpes, dan saudara RECOGN17ER atas segala cerita dan dukungan tanpa ada kamu, aku dan kalian penulis dapat menyelesaikan skripsi ini.
9. Kepada kakanda, saudara dan adinda AIMP Research Group yang telah menyediakan ruang dan tempat diskusi selama proses penyelesaian tugas akhir

10. Kepada teman-teman dan adinda CINEMA09: Nanda, Fitri, Fikram, Mage, Faidah, Mardiani, Ojan, Ali dan Heri yang telah banyak membantu selama masa kepengurusan penulis dan meneruskan kepengurusan UKMT.
11. Dimas, Muflih, Amar, Faiz, Adel, Rahmah, Ica dan Cania sebagai teman penulis dan senantiasa berbagi cerita.
12. Dan seluruh pihak yang telah banyak membantu dalam penyelesaian skripsi ini yang namanya tidak dapat disebutkan satu per satu.

Akhir kata dengan segala hormat dan kerendahan hati, penulis menyadari tugas akhir ini masih memiliki banyak kekurangan, oleh karena itu segala saran dan kritik yang membangun diharapkan demi kesempurnaan sistem dan tugas akhir. Penulis berharap semoga skripsi ini dapat bermanfaat bagi semua pihak yang membaca.

Makassar, Juli 2022

Penulis

ABSTRAK

Kiblat inovasi kendaraan sekarang telah berganti, yang awalnya industri berlomba-lomba untuk menciptakan mesin yang bertenaga besar dengan berkecepatan tinggi, sekarang menjadi masalah keselamatan. Deteksi lajur jalan merupakan salah satu unsur keselamatan pada *Autonomous Car* dimana kendaraan dapat mengenali lajur jalan dan mengambil keputusan sendiri dengan aman. Penulis melakukan penelitian deteksi lajur jalan secara *real-time* dengan menggunakan metode *Semantic Segmentation*. Penulis menggunakan arsitektur U-Net dengan parameter epoch sebanyak 10.000 epoch serta ukuran gambar 128x128 pixel. Data sebanyak 374 gambar berupa lingkungan jalan Kota Makassar terdiri dari empat kelas yakni jalan, marka garis membujur penuh, marka garis membujur putus-putus dan *background*. Hasil pengujian model menggunakan *Intersection over Union* (IoU) dengan akurasi 79.8%. Sistem yang dikembangkan juga melakukan estimasi jarak antara ban dengan marka. Fitur ini bertujuan untuk memperingatkan pengemudi bila kendaraanya mulai bergerak keluar lajur. Hasil pengujian system estimasi jarak terhadap jarak ban dan marka garis dengan metode RMSE sebesar 0,069962636.

Kata kunci: *Autonomous car, real-time, lajur jalan, Semantic Segmentation, U-Net*

DAFTAR ISI

| | |
|---|-----|
| KATA PENGANTAR..... | i |
| ABSTRAK | iv |
| DAFTAR ISI..... | v |
| DAFTAR GAMBAR | vii |
| DAFTAR TABEL | ix |
| BAB I PENDAHULUAN | 1 |
| 1.1. Latar Belakang..... | 1 |
| 1.2. Rumusan Masalah..... | 3 |
| 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. Marka Jalan..... | 6 |
| 2.2. <i>Autonomous Car</i> | 8 |
| 2.3. Visi Komputer | 10 |
| 2.4. <i>Semantic Segmentation</i> | 12 |
| 2.5. Convolutional Neural Network (CNN)..... | 14 |
| 2.6. <i>Fully Convolutional Network For Semantic Segmentation</i> | 19 |
| 2.7. U-Net | 21 |
| 2.8. Inverse Perspective Matpping (IPM) | 23 |
| 2.9. Intersection over Union (IoU)..... | 27 |
| 2.10. Root-Mean-Square Error (RMSE)..... | 29 |
| 2.11. Penelitian Terkait..... | 30 |
| BAB III METODE PENELITIAN..... | 34 |
| 3.1. Tahapan Penelitian..... | 34 |
| 3.2. Waktu dan Lokasi Penelitian | 35 |
| 3.3. Instrumen Penelitian | 36 |
| 3.4. Teknik Pengambilan Data..... | 36 |

| | | |
|-----------------------------------|-----------------------------|----|
| 3.5. | Perancangan Sistem | 37 |
| 3.6. | Analisis Kerja Sistem | 48 |
| BAB IV HASIL DAN PEMBAHASAN | | 52 |
| 4.1. | Hasil Penelitian | 52 |
| 4.2. | Pembahasan | 61 |
| BAB V PENUTUP | | 64 |
| 5.1. | Kesimpulan | 64 |
| 5.2. | Saran | 65 |
| DAFTAR PUSTAKA | | 66 |
| LAMPIRAN | | 68 |

DAFTAR GAMBAR

| | |
|--|----|
| Gambar 1. 1 Peraturan marka garis bujur penuh (Permenhub, 2018) | 6 |
| Gambar 1. 2 Marka garis membujur putus-putus sesuai dengan peraturan Menteri Perhubungan jalan 2 jalur (Permenhub, 2018) | 7 |
| Gambar 1. 3 Marka garis membujur putus-putus sesuai dengan peraturan Menteri Perhubungan jalan lebih dari 2 jalur (Permenhub, 2018) | 7 |
| Gambar 1. 4 Marka garis membujur kombinasi penuh dan putus-putus (Permenhub, 2018) | 7 |
| Gambar 1. 5 (a) Warna marka membujur jalan Nasional (b) warna marka garis membujur jalan selain jalan Nasional (Permenhub, 2018) | 8 |
| Gambar 2. 1 Level Autonomous (SAE J3016)..... | 10 |
| Gambar 2. 2 Perbedaan tugas visi komputer | 12 |
| Gambar 2. 3 Semantic Segmentation secara Holistic | 13 |
| Gambar 2. 4 Arsitektur CNN Dasar LeNet (Ajit dkk., 2020)..... | 14 |
| Gambar 2. 5 Nilai gambar dalam bentuk matrix (Ajit dkk., 2020)..... | 15 |
| Gambar 2. 6 Cara kerja Konvolusi (Ajit dkk., 2020) | 16 |
| Gambar 2. 7 Layer Konvolusi (Ajit dkk., 2020)..... | 17 |
| Gambar 2. 8 Cara kerja pooling (Ajit dkk., 2020)..... | 18 |
| Gambar 2. 9 Fully Connected Layer..... | 19 |
| Gambar 2. 10 Arsitektur Fully Convolutional Network untuk Semantic Segmentation (Shelhamer dkk., 2016)..... | 20 |
| Gambar 2. 11 Arsitektur U-Net (Ronneberger dkk., 2015)..... | 21 |
| Gambar 2. 12 (a) Jalan bebas hambatan (b) Jalan dengan hambatan (c) IPM jalan bebas hambatan (d) IPM jalan dengan hambatan (Oliveira dkk., 2015) | 25 |
| Gambar 2. 13 IPM Model (Muad dkk., 2004)..... | 26 |
| Gambar 2. 14 Implementasi IPM (Muad dkk., 2004)..... | 27 |
| Gambar 2. 15 Algoritma IoU..... | 28 |
| Gambar 3. 1 Diagram tahapan penelitian..... | 34 |
| Gambar 3. 2 Lokasi Dashcam..... | 37 |
| Gambar 3. 3 Flowchart perancangan sistem..... | 38 |

| | |
|---|----|
| Gambar 3. 4 Contoh Anotasi gambar menggunakan Labelme | 39 |
| Gambar 3. 5 Contoh Dataset dengan standar Pascal VOC | 40 |
| Gambar 3. 6 Arsitektur U-Net | 42 |
| Gambar 3. 7 Contoh IPM..... | 44 |
| Gambar 3. 8 Flowchart Lane Departure Warning System..... | 45 |
| Gambar 3. 9 Hasil gambar IPM yang tersegmentasi | 45 |
| Gambar 3. 10 Nilai maksimal marka | 46 |
| Gambar 3. 11 Ilustrasi penentuan koordinat ban | 47 |
| Gambar 3. 12 (a) Skenario 1 (b) Skenario 2 (c) Skenario 3 (d) Skenario 4 (e) Skenario 5 | 50 |
| Gambar 4. 1 Visualisasi Gambar (a) Citra Input (b) Citra Segmentasi Aktual (c) Prediksi..... | 53 |
| Gambar 4. 2 Hasil Segmentasi pada saat pengujian Real-time | 54 |
| Gambar 4. 3 Contoh hasil salah segmentasi | 62 |
| Gambar 4. 4 Grafik Jarak dan Estimasi jarak pada sistem | 63 |

DAFTAR TABEL

| | |
|---|----|
| Tabel 2. 1 Hasil Segmentasi (IoU) pada ISBI cell tracking challenge 2015 (Ronneberger dkk., 2015)..... | 23 |
| Tabel 2. 2 Perbandingan performa pada dataset Massachusetts road (Norelyaqine Abderrahim dkk., 2020)..... | 23 |
| Tabel 3. 1 Label encode tiap kelas | 41 |
| Tabel 3. 2 Tabel setiap pengujian | 50 |
| Tabel 4. 1 Akurasi tiap kelas..... | 53 |
| Tabel 4. 2 Hasil Skenario 1..... | 55 |
| Tabel 4. 3 Hasil skenario 2 | 56 |
| Tabel 4. 4 Hasil skenario 3 | 57 |
| Tabel 4. 4 Hasil skenario 3 | 58 |
| Tabel 4. 5 Hasil Perhitungan Root Mean Square Error | 60 |

BAB I

PENDAHULUAN

1.1. Latar Belakang

Perkembangan industri mobil dalam beberapa tahun terakhir ini kian pesat dan menjadi perhatian bagi pihak akademik dan industri. Pada hari ini, banyak bermunculan inovasi bagaimana kendaraan dapat jalan dengan sendirinya tanpa bantuan manusia. Kiblat inovasi mobil telah berganti, bukan lagi tentang masalah kecepatan melainkan ke masalah keselamatan. Saat ini Tesla adalah perusahaan yang berinovasi dalam kendaraan otonom (*Autonomous Car*), meskipun tidak sepenuhnya otonom (*Semi-Autonomous*), Tesla telah menjadi perusahaan dengan valuasi setara dengan valuasi gabungan antara Ford dan GM yang menjadi perusahaan mobil lama. Tidak hanya Tesla beberapa perusahaan juga berinovasi dalam kendaraan otonom seperti Google dan Mercedes-Benz yang sementara membangun kendaraan yang sepenuhnya otonom (*Fully Autonomous*). Oleh karena itu, penelitian *Autonomous Car* masih menjadi inovasi yang besar kedepannya.

Salah satu fitur *Autonomous Car* yang paling menantang adalah memahami kondisi lajur jalan. Deteksi lajur membantu memandu kendaraan dan dapat digunakan dalam sistem kendaraan tanpa awak (Urmson dkk., 2009). Sistem lajur jalan biasanya mendeteksi marka sebagai objek untuk memandu kendaraan agar tetap pada lajurnya. Sedangkan, di Indonesia, terdapat jalan yang memiliki marka garis dan jalan yang tidak memiliki marka garis akibat belum adanya pemerataan pembangunan. Sehingga penelitian ini dilakukan agar dapat mengenali jalan yang

memiliki marka garis dan jalan yang tidak memiliki marka garis tersebut dengan menggunakan kamera secara *real-time*. Tujuannya yaitu memahami lingkungan di sekitar mobil agar dapat menempatkan mobil agar berada tepat di lajurnya. Ini juga penting dalam pengambilan keputusan saat ingin berpindah lajur sendiri. Dengan demikian pentingnya mendeteksi lajur jalan berbasis kamera secara *real-time* dan akurat menjadi faktor utama dalam pembuatan mobil otonom sendiri.

Sangat banyak kecelakaan terjadi akibat dari kelalaian manusia sendiri. Tercatat 61% penyebab kecelakaan terbesar karena kemampuan serta karakter pengemudi, 9 % disebabkan karena faktor kendaraan (terkait dengan pemenuhan persyaratan teknik laik jalan) dan 30 % disebabkan oleh faktor prasarana dan lingkungan (Marroli, 2017). Pengemudi harus menyesuaikan kecepatan kendaraan dengan lajur yang dipilih, dan menggunakan lajur sesuai peruntukannya. Pengendara juga harus bisa memperkirakan dan menjaga jarak aman dengan kendaraan lain agar bisa menghindar jika sewaktu-waktu kendaraan tiba-tiba berpindah lajur dan menghindari segala hambatan yang berkaitan dengan lajur kendaraan.

Melihat permasalahan tersebut penulis mengusul judul “Deteksi Lajur Jalan Secara *Real-Time* pada *Autonomous Car* menggunakan metode *Semantic Segmentation*”. *Semantic Segmentation* adalah proses klasifikasi setiap pixel dari sebuah citra sebagai sebuah label kelas untuk memahami citra dalam tingkat per pixel. Label kelas yang dimaksud adalah kelas objek, seperti rumah, buku, manusia, dan lainnya agar kendaraan mampu mengenali lingkungan sekitarnya.

1.2. Rumusan Masalah

1. Bagaimana mendeteksi lajur jalan secara *real-time* pada *Autonomous Car*?
2. Bagaimana performa *Semantic Segmentation* dalam mendeteksi lajur jalan?

1.3. Tujuan Penelitian

Tujuan yang ingin dicapai dalam penelitian ini adalah sebagai berikut:

1. Merancang dan membangun sistem pendeteksi lajur jalan secara *Real-Time* pada *Autonomous Car*.
2. Menunjukkan performa penggunaan *Semantic Segmentation* dalam mendeteksi lajur jalan.

1.4. Manfaat Penelitian

Adapun manfaat dari penelitian ini adalah sebagai berikut :

1. Adanya sistem pendeteksi lajur jalan dalam menurunkan risiko terjadinya kecelakaan yang disebabkan oleh manusia.
2. Memberikan informasi penelitian dalam pengembangan *autonomous car* khususnya deteksi lajur jalan selanjutnya

1.5. Batasan Masalah

Yang menjadi batasan masalah dalam penelitian ini adalah:

1. Kelas obyek yang dideteksi adalah jalan, marka garis sambung dan marka garis putus.
2. Waktu penelitian dilakukan di siang hari dengan kondisi cuaca cerah
3. Kamera yang digunakan berupa kamera *dashcam* yang berjumlah satu buah dan diposisikan menghadap ke depan.
4. Jenis mobil yang dipakai yaitu mobil *low-MPV (Multi Purpose Vehicle)*
5. Kecepatan kendaraan pada saat pengukuran estimasi jarak antara marka dengan ban yaitu 10 km/jam

1.6. Sistematika Penulisan

Berikut adalah gambaran singkat mengenai isi tulisan secara keseluruhan:

BAB 1 PENDAHULUAN

Pada bab ini diuraikan mengenai latar belakang masalah, rumusan masalah, tujuan penelitian, manfaat penelitian, batasan masalah penelitian, metode penelitian, serta sistematika penulisan.

BAB 2 TINJAUAN PUSTAKA

Pada bab ini membahas landasan teori yang digunakan untuk menganalisis masalah yang akan diteliti serta hal-hal lain yang berhubungan dengan variabel-variabel data yang akan digunakan, *Semantic Segmentation*, *Inverse Perspective Mapping*, dan arsitektur U-Net.

BAB 3 METODOLOGI PENELITIAN

Pada bab ini berisi mengenai tahap penelitian, instrumen penelitian, pengumpulan data, penerapan algoritma, teknik pengolahan data, serta hasil segmentasi.

BAB 4 HASIL DAN PEMBAHASAN

Bab ini berisi tentang sistem yang telah berhasil dibangun serta pembahasannya.

BAB 5 PENUTUP

Bab ini berisi tentang kesimpulan yang didapatkan berdasarkan hasil penelitian yang telah dilakukan serta saran-saran untuk pengembangan lebih lanjut.

BAB II

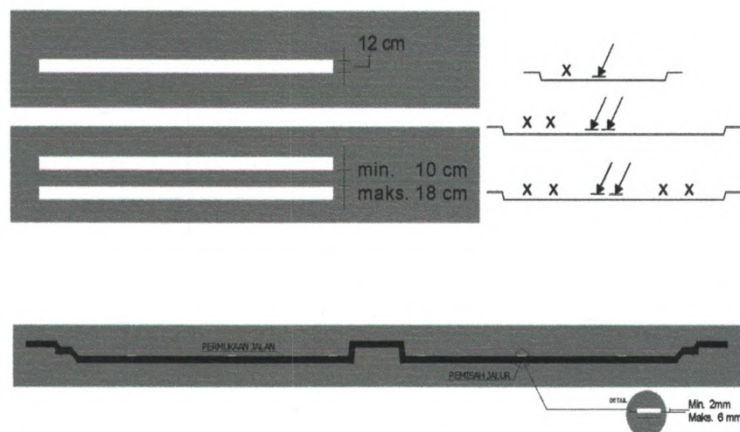
TINJAUAN PUSTAKA

2.1. Marka Jalan

Berdasarkan Peraturan Menteri Perhubungan Republik Indonesia Nomor PM 67 Tahun 2018 Tentang Marka Jalan. Marka jalan adalah suatu tanda yang berada di permukaan jalan atau di atas permukaan jalan yang meliputi peralatan atau tanda yang membentuk garis membujur, garis melintang, garis serong serta lambang lainnya yang berfungsi untuk mengarahkan arus lalu lintas dan membatasi daerah kepentingan lalu lintas. Berikut jenis-jenis marka jalan:

1. Marka Garis Membujur penuh

Pengemudi dilarang melintasi marka ini. Biasanya dipasang di tempat yang mengandung bahaya, misalnya tikungan, tanjakan, turunan, atau tempat yang ramai.

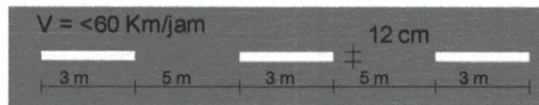


Gambar 1. 1 Peraturan marka garis bujur penuh (Permenhub, 2018)

2. Marka Garis Membujur Putus-putus

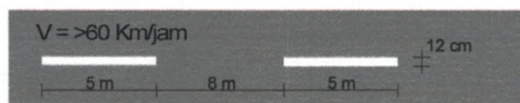
Pengemudi boleh melintasi marka ini, misalnya untuk pindah jalur atau mendahului kendaraan lain.

a. Jalan 2 jalur, 2 arah dengan lebar > 550cm



Gambar 1. 2 Marka garis membujur putus-putus sesuai dengan peraturan Menteri Perhubungan jalan 2 jalur (Permenhub, 2018)

b. Jalan lebih dari 2 jalur



Gambar 1. 3 Marka garis membujur putus-putus sesuai dengan peraturan Menteri Perhubungan jalan lebih dari 2 jalur (Permenhub, 2018)

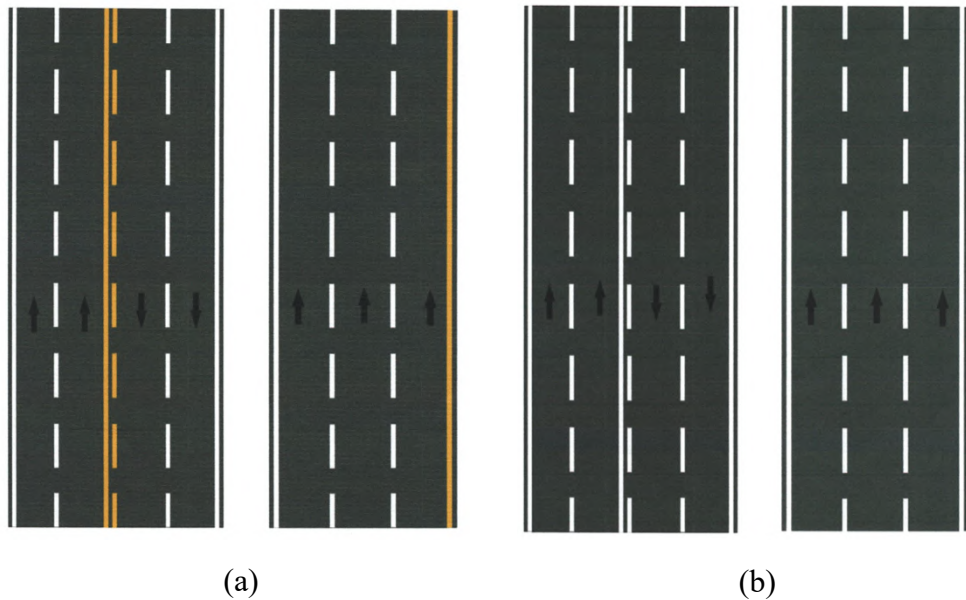
c. Marka garis membujur kombinasi penuh dan putus-putus

Pengemudi yang berada di sisi jalan yang lebih dekat dengan marka putus-putus, boleh melintasi marka kombinasi. Sebaliknya, pengemudi yang berada di sisi jalan yang lebih dekat dengan marka penuh, dilarang melintasi marka kombinasi.



Gambar 1. 4 Marka garis membujur kombinasi penuh dan putus-putus (Permenhub, 2018)

3. Warna Marka Garis



Gambar 1. 5 (a) Warna marka membujur jalan Nasional (b) warna marka garis membujur jalan selain jalan Nasional (Permenhub, 2018)

2.2. *Autonomous Car*

Autonomous Car adalah mobil yang dikendalikan oleh komputer yang dapat memandu dirinya sendiri, mengenali lingkungan sekitar, membuat keputusan dan beroperasi penuh tanpa adanya interaksi manusia. Hal yang menjadi dasar dalam munculnya ide mobil yang dikendalikan oleh komputer tersebut adalah meliputi keselamatan pengendara, pertumbuhan penduduk seiring dengan peningkatan jumlah kendaraan, pertumbuhan infrastruktur, kebutuhan akan manajemen waktu yang efisien serta pemanfaatan dan optimalisasi sumber daya (Hussain and Zeadally, 2019).

Autonomous Car dapat diklasifikasikan menurut keterlibatan manusia dengan teknologi, mulai dari manusia menjalankan sepenuhnya dengan manual hingga dikendalikan sepenuhnya oleh komputer di mana tidak ada campur tangan dari manusia. Pada tugas akhir ini, penulis mengacu pada standar yang diterapkan oleh *Society of Automotive Engineers (SAE)* yang ditunjukkan dalam gambar 2.1 di mana sangat banyak yang mengadopsi definisi mengenai *Autonomous Car* yang terdiri dari:

Level 0 : Tanpa automasi, di mana semua tugas mengemudi dilakukan oleh manusia

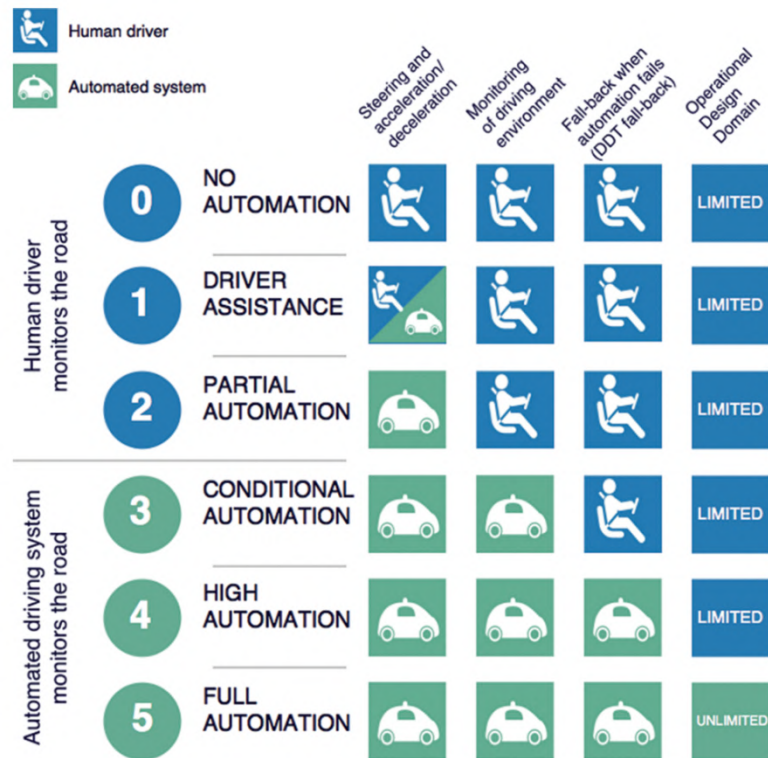
Level 1 : Asisten pengemudi, di mana tugas akselerasi kendaraan dan kemudi dilakukan oleh pengemudi yang dibantu oleh sistem automasi dan sisanya dilakukan oleh tugas manusia.

Level 2 : Automasi Parsial, di mana kemudi dan akselerasi dilakukan oleh komputer dan tugas sisanya dilakukan oleh pengemudi manusia

Level 3 : Automasi bersyarat, di mana komputer dapat memantau dan memahami kondisi lingkungan di jalan dan mengambil tindakan dalam menjalankan kemudi dan akselerasi mobil. Tetapi, pengemudi harus siap mengambil tindakan saat komputer merespon.

Level 4 : Automasi tinggi, di mana seluruh tugas seperti kemudi, akselerasi, dan memantau keadaan sekitar dilakukan sepenuhnya oleh komputer tetapi di dalam kondisi lingkungan tertentu.

Level 5 : Automasi sepenuhnya, di mana komputer melakukan tugas mengemudi sepenuhnya di kondisi lingkungan berbeda.



Gambar 2.1 Level Autonomous (SAE J3016)

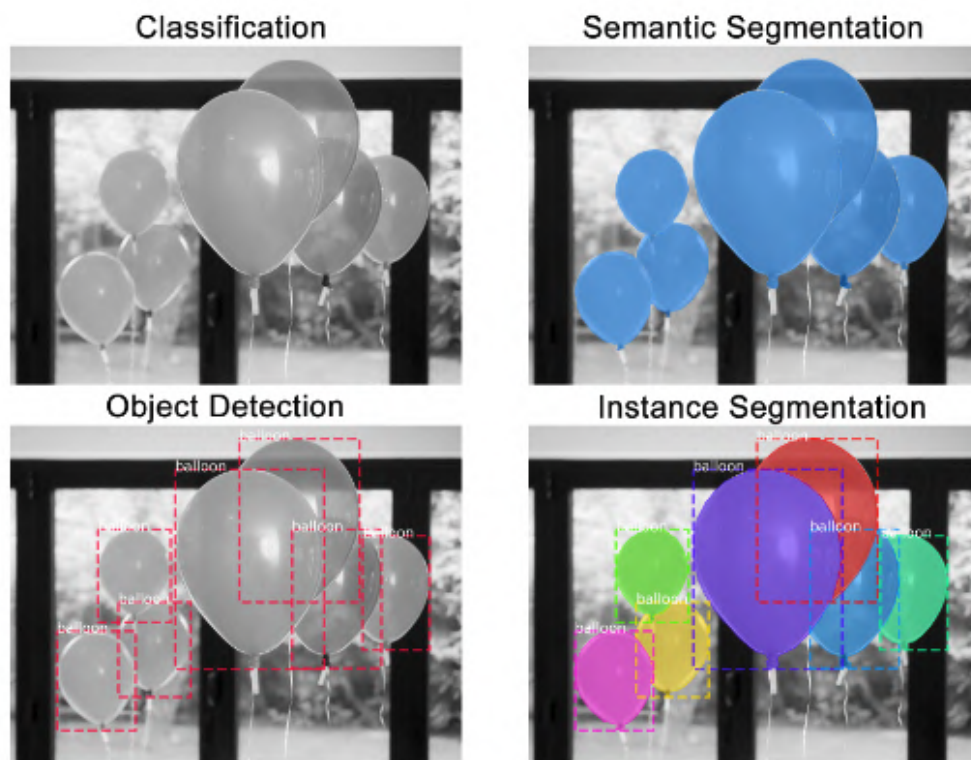
2.3. Visi Komputer

Visi komputer sebagai bidang ilmiah interdisipliner melibatkan penggunaan komputer untuk mendapatkan pemahaman rinci tentang data visual, yang merupakan pendekatan yang mirip dengan sistem visual manusia. Bidang ini termasuk dalam kategori kecerdasan buatan yang luas dan aplikasinya telah menangani berbagai masalah (Xu dkk., 2021).

Visi komputer pada mobil otonom memungkinkan kendaraan untuk melihat semua kejadian di sekitar, menggunakan kamera mono, stereo, radar, dan sensor lidar yang dipasang. (Manoharan, 2019). Visi komputer menjadi teknologi yang menjanjikan dalam beberapa tahun belakangan ini yang berfokus pada bagaimana

kemampuan sebuah mesin dapat memvisualisasikan dan menafsirkan gambar dalam format digital serta konten yang ditemukan dalam video (V., 2019).

Visi Komputer dan deteksi objek adalah dua fitur paling penting dan esensial dari mobil otonom. Untuk meniru perilaku pengemudi manusia, mobil otonom harus mengenali jalan dan mendeteksi setiap rintangan di depan dan di sekitarnya, baik itu mobil lain, pejalan kaki, lubang, atau jenis rintangan lainnya. Kedua fitur utama ini bersama dengan modul lainnya memungkinkan mobil otonom untuk mengemudi di sepanjang jalan dan merespons situasi yang tidak diinginkan dengan cara yang aman, misalnya berhenti di lampu lalu lintas, melambat jika mobil sebelumnya mengurangi kecepatan. menghindari berlari ke pejalan kaki, dan sebagainya. Sampai saat ini, banyak hasil penelitian telah dicapai baik dalam Visi Komputer dan deteksi objek untuk mobil otonom (Hussain and Zeadally, 2019).



Gambar 2. 2 Perbedaan tugas visi komputer (Abdulla, 2018)

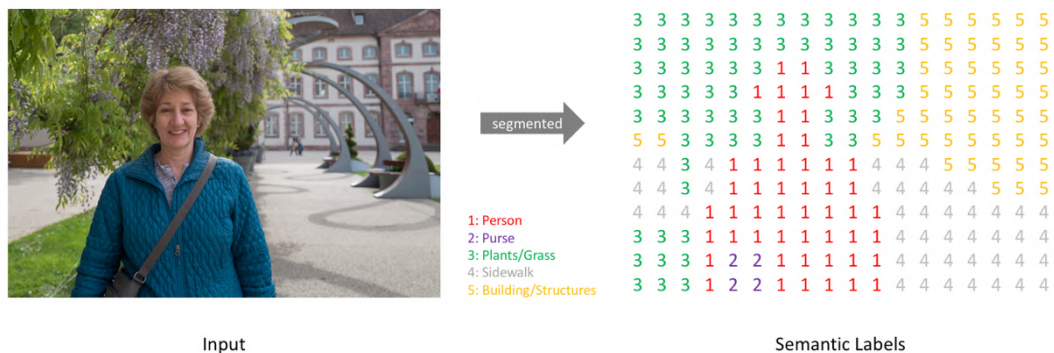
2.4. Semantic Segmentation

Semantic Segmentation, juga disebut pelabelan adegan, mengacu pada proses pemberian label semantik (misalnya mobil, orang, dan jalan) ke setiap pixel gambar. Ini adalah langkah pemrosesan data penting untuk komputer dan sistem tak berawak lainnya untuk memahami lingkungan di sekitarnya. Terlepas dari upaya selama beberapa dekade, *Semantic Segmentation* masih merupakan tugas yang sangat menantang karena terlalu banyak variasi dalam pemandangan alam (Yudha dkk., 2018).

Munculnya terminologi *Semantic Segmentation* dapat dilihat kembali pada tahun 1970-an (Ohta dkk., 1978) Pada saat itu, terminologi ini setara dengan

segmentasi citra tetapi menekankan bahwa daerah yang tersegmentasi harus “bermakna secara semantik” contohnya pada citra perkotaan, sebuah gedung memiliki jendela dan memiliki ciri-ciri berbentuk persegi panjang sedangkan langit berwarna biru. Lalu penelitian berlanjut menjadi objek segmentasi yang dapat memisahkan antara objek dan latar belakang menggunakan algoritma Constrained Parametric Min-Cuts(CPMC) (Carreira and Sminchisescu, 2012).

Sederhananya, tujuannya adalah mengambil gambar warna RGB ($height \times width \times 3$) atau gambar skala abu-abu ($height \times width \times 1$) dan menampilkan peta segmentasi dimana setiap pixel berisi label kelas yang direpresentasikan sebagai bilangan bulat ($height \times width \times 1$)



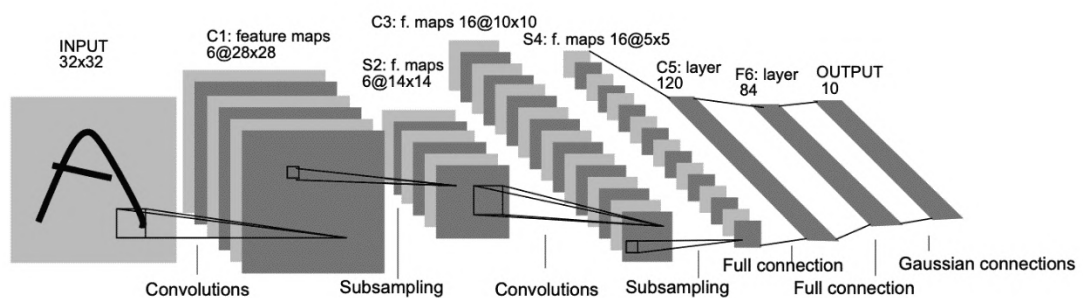
Gambar 2. 3 Semantic Segmentation secara Holistic (Jordan, 2018)

Dapat kita lihat ada dua masalah yang dapat diselesaikan oleh *Semantic Segmentation* yaitu informasi objek dan lokalisasi. Bagaimana *Semantic Segmentation* dapat mengklasifikasikan objek berupa *person*, *purse*, *plants*, *sidewalk*, *building*. Dimana objek itu berada juga dapat diketahui oleh *Semantic Segmentation*.

Dengan pengembangan *Deep Learning*, penggunaan fitur yang dipelajari dalam tugas Visi Komputer, seperti klasifikasi gambar telah mencapai sukses besar dalam beberapa tahun terakhir. Akibatnya, peneliti *Semantic Segmentation* baru-baru ini menaruh banyak perhatian pada fitur yang dipelajari yang biasanya mengacu pada Convolutional Neural Network (CNN) (Yu dkk., 2018).

2.5. Convolutional Neural Network (CNN)

Convolutional Neural Network adalah jenis jaringan saraf tiruan yang digunakan dalam pengenalan dan pemrosesan gambar. *convolutional neural network* meniru cara sel-sel saraf kita berkomunikasi dengan neuron yang saling berhubungan dan *convolutional neural network* memiliki arsitektur yang sama. Apa yang membuatnya unik dari jaringan saraf lain adalah operasi konvolusional yang menerapkan filter pada setiap bagian dari *input* sebelumnya untuk mengekstraksi pola dan *features maps*. Ahli statistik dan Peneliti telah mencari ide-ide jaringan saraf di abad ke-20 untuk mengerjakan Pengenalan Pola (Kholik, 2021).

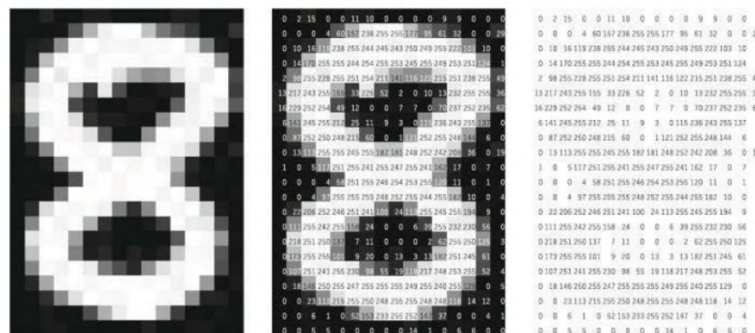


Gambar 2. 4 Arsitektur CNN Dasar LeNet (Ajit dkk., 2020)

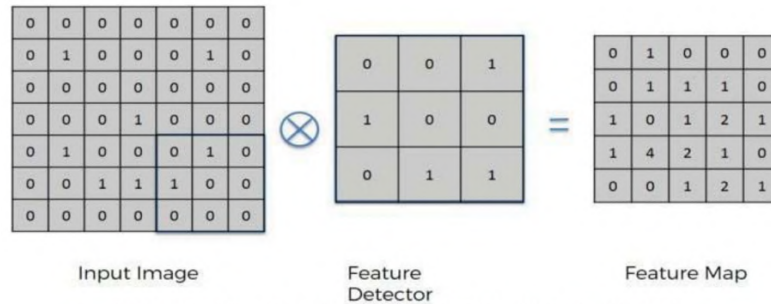
A. Convolutional Layer

Convolution Layer adalah lapisan paling dasar tetapi sekaligus paling penting di CNN. Ini pada dasarnya menggulung atau mengalikan matriks pixel yang dihasilkan untuk gambar atau objek yang diberikan untuk menghasilkan *activation map* untuk gambar yang diberikan.

Keuntungan utama dari *activation map* adalah bahwa peta ini menyimpan semua fitur pembeda dari gambar yang diberikan sementara pada saat yang sama mengurangi jumlah data yang akan diproses. Matriks dengan datanya berbelit-belit dan kompleks adalah pendeteksi fitur yang pada dasarnya adalah sekumpulan nilai yang kompatibel dengan mesin. Versi gambar yang berbeda dihasilkan menggunakan nilai pendeteksi fitur yang berbeda. Model berbelit-belit juga dilatih dengan *backpropagation* untuk memastikan kesalahan minimal di setiap lapisan. Menurut set kesalahan terendah, kedalaman dan *padding* diatur (Ajit dkk., 2020).



Gambar 2. 5 Nilai gambar dalam bentuk matrix (Ajit dkk., 2020)



Gambar 2. 6 Cara kerja Konvolusi (Ajit dkk., 2020)

Gambar 1 di atas menunjukkan cara kerja konvolusi. Langkah ini melibatkan konvolusi matriks yang berisi data gambar dan kemudian detektor fitur yang memberi kita peta aktivasi atau *feature map*. Apa yang terjadi dalam konvolusi adalah bahwa nilai pada posisi identik dalam data dan peta fitur yaitu nilai yang memiliki nilai 1 atau lebih dari 1 disimpan sementara sisanya dihapus. Matriks dari data citra dibandingkan 3x3 sekaligus. Ukuran fitur detektor bervariasi dengan jenis CNN yang digunakan. Misalnya ada versi CNN yang menggunakan filter skala 5x5 atau bahkan 7x7 untuk konvolusi. Konvolusi mengikuti

$$(f * g)(t) = \int_{-\infty}^{\infty} f(\tau)g(t - \tau)d\tau \quad (2.1)$$

Keterangan:

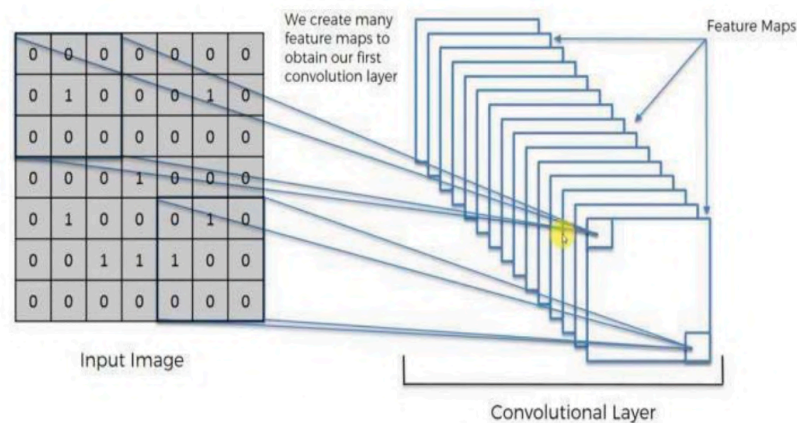
$(f * g)$ = fungsi yang sedang terkonvolusi

t = variable bilangan riil f dan g

$g(\tau)$ = konvolusi fungsi $f(t)$

$d\tau$ = turunan pertama dari fungsi $d(\tau)$

yang bertujuan untuk menunjukkan bagaimana satu fungsi memodifikasi bentuk yang lain. Menjelaskan gambar di atas, data yang dihasilkan untuk gambar ini dimodifikasi menggunakan filter untuk menghasilkan peta aktivasi.

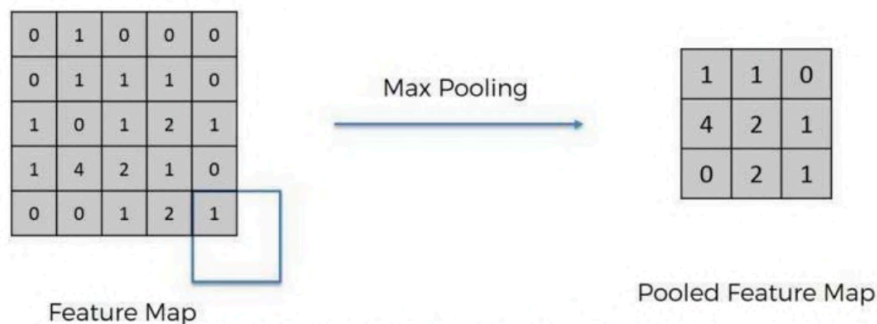


Gambar 2. 7 Layer Konvolusi (Ajit dkk., 2020)

B. Pooling

Pooling adalah langkah penting untuk lebih mengurangi dimensi *activation map*, hanya menyimpan fitur-fitur penting sambil juga mengurangi invariansi spasial. Hal ini pada gilirannya mengurangi jumlah fitur yang dapat dipelajari untuk model. Ini membantu mengatasi masalah *overfitting*. *Pooling* memungkinkan CNN untuk menggabungkan semua dimensi gambar yang berbeda sehingga berhasil mengenali objek yang diberikan meskipun bentuknya miring atau hadir pada sudut yang berbeda. Ada berbagai jenis *pooling* seperti *max pooling*, *average pooling*, *stochastic pooling*, *spatial piramida pooling*. Di antaranya yang paling populer adalah *max pooling* (Ajit dkk., 2020).

Max Pooling mengambil nilai tertinggi dari setiap sub matriks peta aktivasi dan membentuk matriks terpisah darinya. Melakukan hal ini memastikan bahwa fitur yang dapat dipelajari tetap terbatas jumlahnya sambil juga mempertahankan fitur utama dari gambar apa pun. *Max Pooling* biasanya dilakukan dengan menggunakan filter 2x2

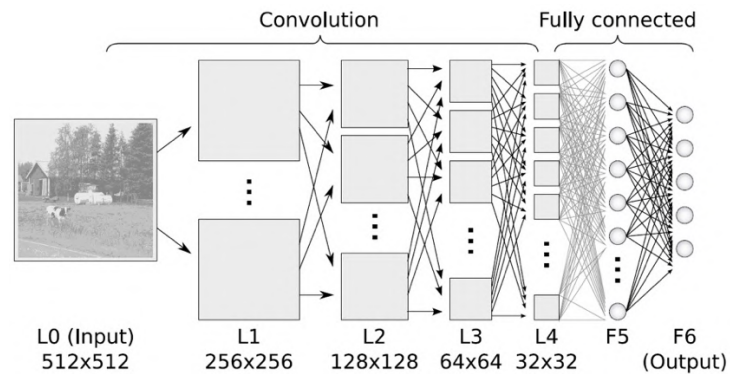


Gambar 2. 8 Cara kerja *pooling* (Ajit dkk., 2020)

C. *Fully Connected Layer*

Fully Connected Layer mirip dengan cara neuron diatur dalam jaringan saraf tradisional. Oleh karena itu, setiap *node* di lapisan yang terhubung penuh terhubung langsung ke setiap *node* di lapisan sebelumnya dan di lapisan berikutnya seperti yang ditunjukkan pada Gambar 2.9. Dari gambar ini kita dapat mencatat bahwa masing-masing *node* di *frame* terakhir di *pooling* lapisan terhubung sebagai vektor ke lapisan pertama dari lapisan yang sepenuhnya terhubung. Ini adalah parameter yang paling banyak digunakan dengan CNN dalam lapisan ini, dan membutuhkan waktu lama dalam proses pelatihan.

Kelemahan utama dari *Fully Connected Layer*, adalah bahwa ia mencakup banyak parameter yang memerlukan komputasi kompleks dalam contoh *training*. Oleh karena itu, perlu untuk menghilangkan jumlah *node* dan koneksi. *Node* dan koneksi yang dihapus dapat dipenuhi dengan menggunakan teknik *dropout*. Misalnya, LeNet dan AlexNet merancang jaringan yang dalam dan luas sambil menjaga kompleks komputasi tetap konstan (Albawi dkk., 2017).

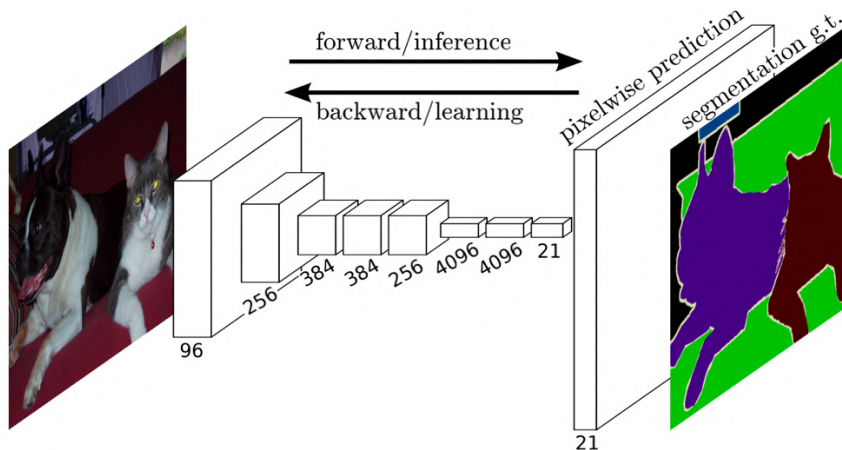


Gambar 2. 9 Fully Connected Layer (Kannoja dan Jaiswal, 2018)

2.6. Fully Convolutional Network For Semantic Segmentation

Shelhamer dan kawan-kawan menunjukkan bahwa Fully Convolutional Network (FCNs) yang dilatih ujung-ke-ujung, pixel-ke-pixel pada segmentasi semantik melebihi hasil terbaik sebelumnya tanpa mesin lebih lanjut. Menurut Shelhamer dkk, ini adalah pekerjaan pertama untuk melatih FCN *end-to-end* untuk prediksi pixel dan dari pra-pelatihan yang diawasi. Versi konvolusi penuh dari jaringan yang ada memprediksi keluaran padat dari masukan berukuran arbitrer. Baik pembelajaran dan inferensi dilakukan seluruh gambar pada satu waktu dengan

perhitungan *feedforward* yang padat dan *backpropagation*. Lapisan *upsampling* dalam jaringan memungkinkan prediksi dan pembelajaran pixel dalam jaring dengan *subsampling* (Shelhamer dkk., 2016).



Gambar 2. 10 Arsitektur Fully Convolutional Network untuk Semantic Segmentation (Shelhamer dkk., 2016)

Setiap keluaran lapisan dalam sebuah *convnet* adalah *array* tiga dimensi dengan ukuran $h \times w \times d$, di mana h dan w adalah dimensi spasial, dan d adalah dimensi fitur atau saluran. Lapisan pertama adalah citra, dengan ukuran pixel $h \times w$, dan saluran d . Lokasi di lapisan yang lebih tinggi sesuai dengan lokasi di gambar yang terhubung dengan jalurnya, yang disebut bidang reseptifnya.

Convnets secara inheren terjemahan invarian. Komponen dasarnya (fungsi konvolusi, penyatuan, dan aktivasi) beroperasi pada daerah masukan lokal, dan hanya bergantung pada koordinat spasial relatif. Menulis x_{ij} untuk vektor data pada lokasi (i, j) pada lapisan tertentu, dan y_{ij} untuk lapisan berikutnya, fungsi-fungsi ini menghitung keluaran y_{ij} dengan (Shelhamer dkk., 2016)

$$y_{ij} = f_{ks}(\{X_{si} + \delta_{i,sj} + \delta_j\} \mid 0 \leq \delta_i, \delta_j < k) \quad (2.2)$$

Keterangan

y_{ij} = Layer

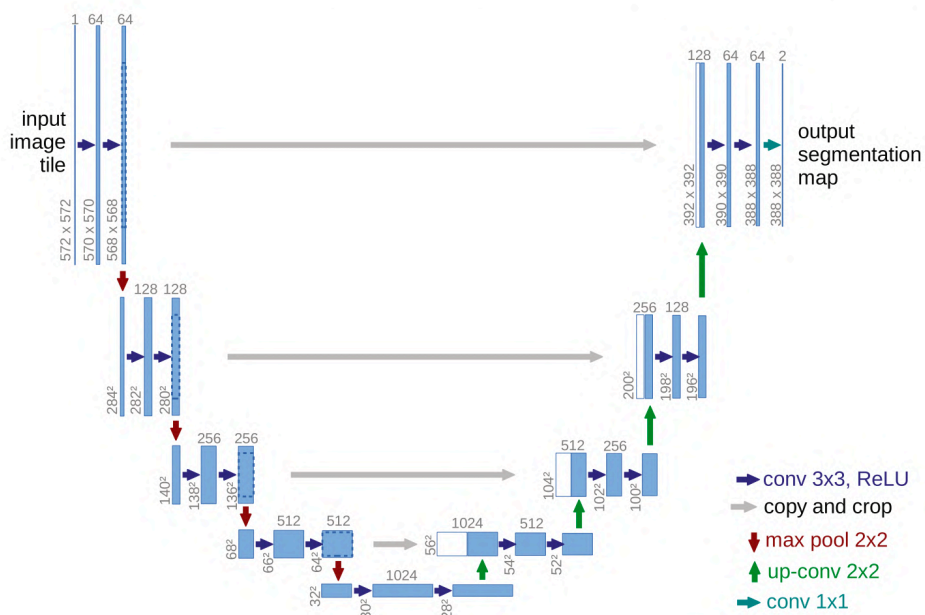
k = Ukuran kernel

s = *Stride*

f_{ks} = Jenis lapisan layer

δ = Perubahan nilai variabel

2.7. U-Net



Gambar 2. 11 Arsitektur U-Net (Ronneberger dkk., 2015)

U-Net (Ronneberger dkk., 2015) menggunakan jaringan Fully Convolutional Network untuk melakukan tugas segmentasi semantik. Arsitektur jaringan simetris, memiliki *Encoder* yang mengekstrak fitur spasial dari gambar, dan *Decoder* yang membangun peta segmentasi dari fitur yang dikodekan. *Encoder* mengikuti formasi khas jaringan konvolusi. Ini melibatkan urutan dua operasi konvolusi 3×3 , diikuti

oleh operasi *max-pooling* dengan ukuran *pooling* 2×2 dan langkah 2. Urutan ini diulang empat kali, dan setelah setiap *down-sampling*, jumlah filter di lapisan konvolusi menjadi dua kali lipat. Akhirnya, perkembangan dua operasi konvolusi 3×3 menghubungkan *Encoder* ke *Decoder*.

Di sisi lain, *Decoder* pertama-tama mengambil sampel peta fitur menggunakan operasi konvolusi 2×2 yang dialihkan (Zeiler dkk., 2010), mengurangi saluran fitur hingga setengahnya. Kemudian urutan dua operasi konvolusi 3×3 dilakukan lagi. Mirip dengan *Encoder*, sukseksi *upsampling* dan dua operasi konvolusi ini diulang empat kali, mengurangi separuh jumlah filter pada setiap tahap. Akhirnya, operasi konvolusi 1×1 dilakukan untuk menghasilkan peta segmentasi akhir. Semua lapisan konvolusi dalam arsitektur ini, kecuali yang terakhir, menggunakan fungsi aktivasi ReLU (Rectified Linear Unit) lapisan konvolusi akhir menggunakan fungsi aktivasi Sigmoid.

Aspek yang paling cerdas dari arsitektur U-Net adalah skip *connection*. Di keempat level, *output* dari *convolutional layer*, sebelum operasi penyatuan *Encoder* ditransfer ke *Decoder*. Peta fitur ini kemudian digabungkan dengan *output* dari operasi *upsampling*, dan peta fitur gabungan disebarkan ke lapisan yang berurutan. Koneksi lewati ini memungkinkan jaringan untuk mengambil informasi spasial yang hilang oleh operasi penyatuan (Carneiro dkk., 2016) Arsitektur jaringan diilustrasikan pada Gambar 2.11.

Ronneberger dkk menggunakan dataset nucleus PhC-U373 dan DIC-HeLa membandingkan kinerja U-Net untuk segmentasi gambar. Hasilnya U-Net

menempati akurasi tertinggi dengan menggunakan matriks IoU yang dapat dilihat pada tabel 2.1

Tabel 2. 1 Hasil Segmentasi (IoU) pada ISBI cell tracking challenge 2015 (Ronneberger dkk., 2015)

| Nama | PhC-U373 | DIC-HeLa |
|---------------------|---------------|---------------|
| IMCB-SG (2014) | 0.2669 | 0.2935 |
| KTH-SE (2014) | 0.7953 | 0.4607 |
| HOUS-US (2014) | 0.5323 | - |
| second-best 2015 | 0.83 | 0.46 |
| U-net (2015) | 0.9203 | 0.7756 |

Sedangkan Norelyaqine dkk menggunakan arsitektur U-Net pada dataset *Massachusetts road* yang berupa gambar yang diambil di atas langit menggunakan drone. Hasilnya U-Net menampilkan hasil yang lebih baik dari metode lainnya sebagaimana ditunjukkan pada tabel 2.2.

Tabel 2. 2 Perbandingan performa pada dataset *Massachusetts road* (Norelyaqine Abderrahim dkk., 2020)

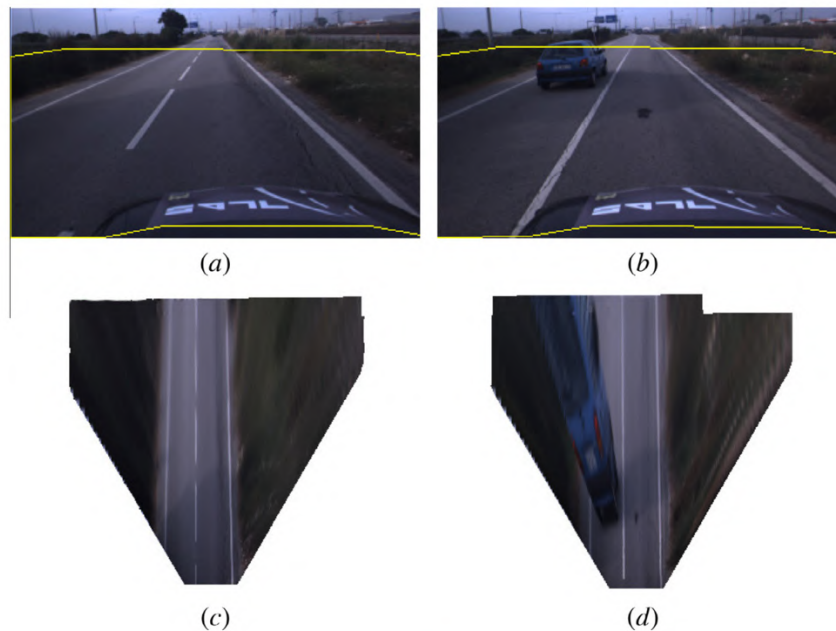
| Model | Precision | Recall | Accuracy | F-Score |
|--------------|--------------|--------------|--------------|--------------|
| FCN | 43.5% | 68.6% | 90.4% | 53.2% |
| RSRCNN | 60.6% | 72.9% | 92.4% | 66.2% |
| SegNet | 77.3% | 76.5% | 95.7% | 76.8% |
| U-Net | 86.8% | 88.3% | 97.7% | 87.5% |

2.8. *Inverse Perspective Mapping (IPM)*

Memahami pemandangan 3D dari gambar perspektif tunggal adalah salah satu masalah mendasar dalam visi komputer. Menghasilkan *Bird-eye view* (atau di atas kepala, atau ortografis) dari pemandangan berperan dalam pemahaman ini karena memungkinkan distorsi perspektif bidang tanah dihilangkan. Penyerahan bidang dasar ini memungkinkan geometri pemandangan di bidang dasar diukur langsung dari sebuah gambar (Abbas and Zisserman, 2019).

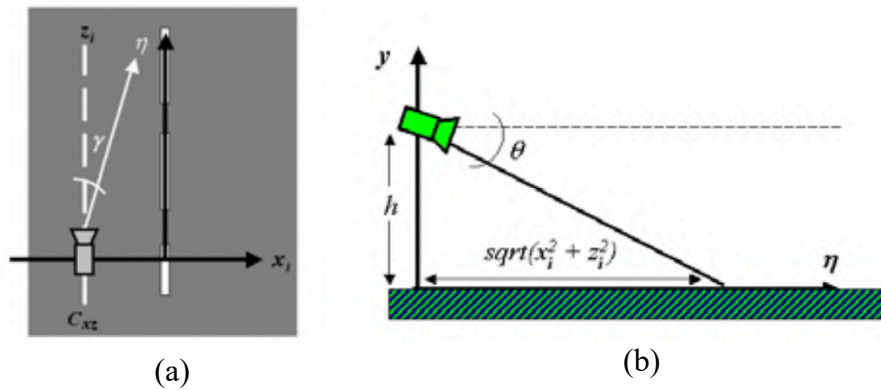
IPM (*Inverse Perspective Mapping*) merupakan metode transformasi citra untuk menghilangkan efek perspektif dari citra berdasarkan parameter intrinsik dan ekstrinsik kamera yang memiliki beberapa aplikasi, misalnya deteksi jarak, pemetaan jalan dan sistem bantuan parkir (Lin and Wang, 2012). Oleh karena itu, IPM sangat penting untuk sejumlah besar tugas otomatis yang harus ditangani oleh kendaraan cerdas. IPM bekerja di bawah tiga asumsi inti: jalan harus berupa permukaan datar, harus ada transformasi bodi yang kaku dari kamera ke jalan, dan jalan harus bebas dari rintangan. Kami fokus pada asumsi terakhir: jalan bebas hambatan. Ini adalah asumsi yang paling tidak realistis, karena sangat sering jalan dipenuhi oleh kendaraan lain, dinding pelindung, pejalan kaki, dll. Faktanya, algoritma IPM klasik gagal menghasilkan gambar proyeksi yang akurat ketika gambar input dari kamera berisi gambar lain. hal-hal lain selain jalan itu sendiri. Contohnya ditunjukkan pada Gambar. 1: di (a) jalan bebas dari rintangan, yang mengarah ke gambar IPM (c) yang sangat akurat. Pada Gambar 1(b), ada kendaraan lain di jalan. Akibatnya, gambar IPM yang dihasilkan (d) mengandung artefak yang

mungkin menyedatkan misalnya untuk algoritma pengenalan pola (Oliveira dkk., 2015).



Gambar 2. 12 (a) Jalan bebas hambatan (b) Jalan dengan hambatan (c) IPM jalan bebas hambatan (d) IPM jalan dengan hambatan (Oliveira dkk., 2015)

IPM adalah teknik transformasi geometris yang memproyeksikan setiap pixel dari tampilan perspektif 2D dari objek 3D, dan memetakannya kembali ke posisi baru dan membangun gambar baru pada planar 2D terbalik. Secara matematis, IPM dapat digambarkan sebagai proyeksi dari ruang Euclidean 3D, $W = \{(x, y, z)\} E3$ (ruang dunia) ke planar 2D, $I = \{(u, v)\} E2$. Ini akan menghasilkan pandangan mata burung dari gambar dan dengan demikian, menghilangkan efek perspektif



Gambar 2. 13 IPM Model (Muad dkk., 2004)

Model IPM seperti ditunjukkan pada Gambar 2.13(a) dan (b). Gambar 2.13(a) mewakili model geometri tampak atas sedangkan Gambar 2.13(b) mewakili model geometri tampak samping. Dua persamaan diturunkan menggunakan triangulasi dan trigonometri model IPM.

$$u(x, 0, z) = \frac{\gamma(x,0,z) - (Y - \alpha)}{\frac{2\alpha}{n-1}} \tag{2.3}$$

$$v(x, 0, z) = \frac{\theta(x,0,z) - (\Theta - \alpha)}{\frac{2\alpha}{m-1}} \tag{2.4}$$

Dimana,

$$\gamma = \tan^{-1} \left(\frac{z}{x} \right) \tag{2.5}$$

$$\theta = \tan^{-1} \left(\frac{h}{\sqrt{x^2 + y^2}} \right) \tag{2.6}$$

Keterangan

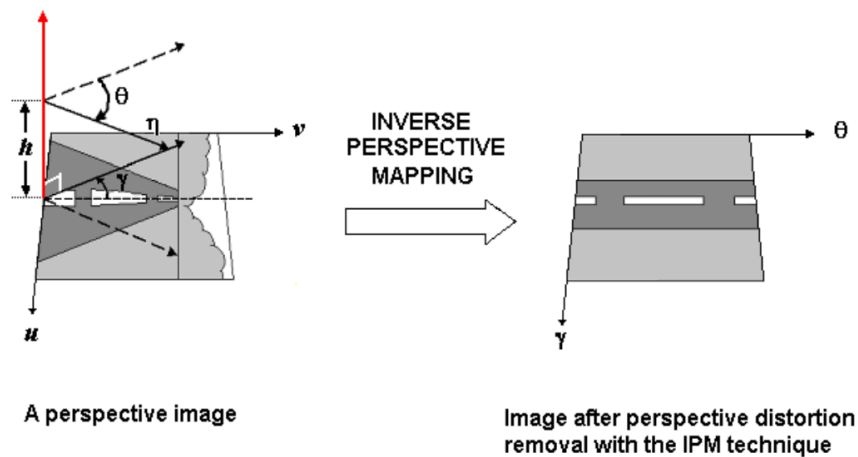
Y = Sudut antara proyeksi optic dari bidang datar

Θ = Sudut antara sumbu optik dan horizon

α = Camera aperture

n x m = resolusi gambar

Implementasi IPM mengasumsikan bahwa permukaan jalan pada citra perspektif terletak rata pada bidang x-z merujuk pada gambar 2.13. Gambar 2.14 menampilkan presentasi grafis implementasi IPM. Pada citra perspektif 2D, nilai sudut yaw (J) berubah dari negatif ke positif saat pemindaian berlangsung dari kiri ke kanan dan sebaliknya sedangkan sudut kemiringan (T) adalah nilai yang berubah secara vertikal. Baik J dan T bertindak sebagai faktor 'bobot proyeksi' untuk semua piksel saat IPM memproyeksikan dan memetakan ulang gambar perspektif.

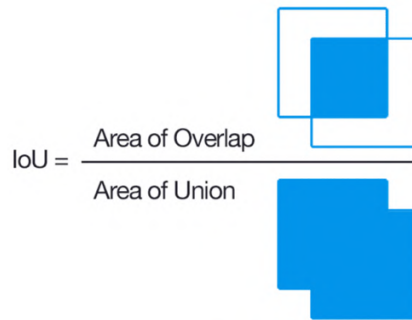


Gambar 2. 14 Implementasi IPM (Muad dkk., 2004)

2.9. Intersection over Union (IoU)

Intersection over Union (IoU), juga dikenal sebagai *Jaccard Index* dalam statistik, adalah salah satu metode paling mendasar untuk membandingkan kesamaan antara sampel data dalam Machine Learning. Dalam tugas Visi Komputer, IoU adalah salah satu metode evaluasi inti dalam berbagai tolak ukur termasuk Object Detection, Object Tracking, Semantic Segmentation, dll. Secara

umum, IoU didefinisikan untuk menghitung tumpang tindih antara dua set elemen A dan B yang diberikan: (Yu dkk., 2021)



Gambar 2. 15 Algoritma IoU (Jakhar, 2019)

Dalam matematika ditulis sebagai,

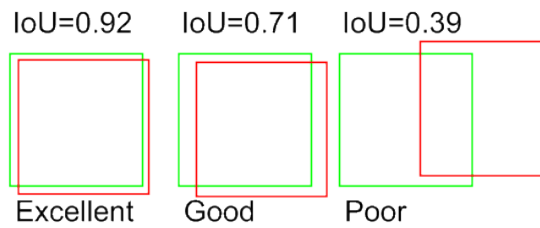
$$IoU = \frac{A \cap B}{A \cup B} \quad (2.7)$$

Keterangan

A = Nilai Prediksi

B = Nilai Aktual

Berdasarkan persamaan diatas sederhananya, matriks IoU mengukur nilai kesamaan dari objek prediksi dan objek akatual. Untuk mendapatkan nilai IoU harus memiliki kedua objek tersebut, yang kemudian area irisan kedua objek dibagi dengan area gabungan kedua objek. Setelah melalui persamaan tersebut, untuk menilai sebuah model baik atau buruk dengan melihat semakin banyak area irisan objek prediksi dan area aktual maka semakin bagus, begitupula sebaliknya. Sebagai contoh penilaian IoU yang ditunjukkan pada gambar 2.16 berikut.



Gambar 2. 16 Contoh hasil perhitungan IoU (Interstellar Engine, 2018)

Pada gambar 2.16 berikut area objek aktual ditandai dengan warna hijau dan area objek prediksi ditandai dengan warna merah. Dari ilustrasi tersebut dapat kita simpulkan bahwa semakin skor IoU semakin tinggi bila jarak antara objek aktual dan objek prediksi semakin mirip atau dekat atau dengan kata lain semakin banyak irisan antara kedua objek.

2.10. Root-Mean-Square Error (RMSE)

Root-Mean-Square-Error atau RMSE adalah salah satu ukuran paling populer untuk memperkirakan keakuratan nilai prediksi model dengan nilai aktual atau yang diamati saat melatih model. RMSE mengukur kesalahan dalam nilai prediksi ketika target atau variabel respons adalah angka kontinu. Misalnya, saat menggunakan model regresi untuk memprediksi kuantitas seperti pendapatan, nilai/volume penjualan, volume permintaan, skor, tinggi atau berat, dll.

Oleh karena itu, untuk menghitung RMSE, rumusnya adalah sebagai berikut:

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (f_i - o_i)^2} \quad (2.8)$$

Keterangan :

f_i = Nilai data Aktual

o_i = Nilai hasil prediksi

n = banyaknya data

Σ = Summation (Jumlahkan keseluruhan nilai)

Dibandingkan dengan Mean-Square-Error (MSE), MSE juga merupakan pengukuran standar evaluasi terhadap regresi, namun hasil MSE selalu positif dikarenakan data yang bersifat acak sehingga tidak dapat mengukur dengan baik. (Lehmann and Springer, 1998). Oleh karena itu penambahan *root* pada rata-rata *square error* perlu untuk mengurangi adanya bias.

2.11. Penelitian Terkait

Berikut beberapa penelitian terkait terhadap penelitian yang dilakukan:

2.11.1 Fully Convolutional Networks for Semantic Segmentation (Shelhamer dkk, 2016)

Shelhamer dkk menunjukkan bahwa Convolutional Network itu sendiri, terlatih dari ujung-ke-ujung, piksel-ke-piksel, meningkatkan hasil terbaik sebelumnya dalam Semantic Segmentation. Wawasan utama Shelhamer dkk adalah membangun jaringan "Fully Convolutional" yang mengambil input dengan ukuran arbitrer dan menghasilkan output berukuran sesuai dengan inferensi dan pembelajaran yang efisien. Shelhamer dkk mengadaptasi jaringan klasifikasi kontemporer (AlexNet, VGG net, dan GoogLeNet) ke dalam jaringan yang sepenuhnya konvolusi dan mentransfer representasi yang dipelajari dengan menyesuaikan tugas segmentasi. Shelhamer dkk kemudian mendefinisikan arsitektur lompatan yang menggabungkan informasi semantik dari lapisan yang dalam dan kasar dengan

informasi tampilan dari lapisan yang dangkal dan halus untuk menghasilkan segmentasi yang akurat dan terperinci. Fully Convolutional Network Shelhamer dkk mencapai segmentasi PASCAL VOC yang lebih baik (30% peningkatan relatif menjadi 67,2% rata-rata IU pada 2012), NYUDv2, SIFT Flow, dan PASCAL-Context, sementara inferensi membutuhkan sepersepuluh detik untuk gambar biasa.

2.11.2 U-Net: Convolutional Networks for Biomedical Image Segmentation (Ronneberger dkk, 2015)

Pada penelitiannya, Ronneberger memodifikasi arsitektur Fully Convolutional Network menjadi bentuk menyerupai huruf U. Dengan menggunakan dataset PhC-U373 yang merupakan kumpulan data gambar sel untuk kebutuhan biomedis. Ronneberger mendapatkan akurasi 92% menggunakan Intersection Over Union (IoU), yang secara signifikan lebih baik daripada algoritma terbaik kedua dengan 83%. Kumpulan data kedua “DIC-HeLa”³ adalah sel HeLa pada kaca datar direkam dengan mikroskop kontras interferensi diferensial (DIC). Ini berisi 20 gambar pelatihan beranotasi sebagian. Di sini kami mencapai rata-rata IoU 77,5% yang secara signifikan lebih baik daripada algoritma terbaik kedua dengan 46%.

2.11.3 Implementation of Inverse Perspective Mapping Algorithm for The Development of an Automatic Lane Tracking System (Muad dkk, 2004)

Muad dkk telah ditunjukkan bahwa IPM memberikan solusi unik untuk memperbaiki distorsi yang terlihat pada permukaan jalan dari tampilan

perspektif 2D. Dengan penempatan kamera yang tepat dan pengaturan sudut yaw (J) dan kemiringan (T) yang benar, IPM dapat menjadi teknik yang sangat berguna. IPM menghasilkan pandangan mata burung dari gambar dan ini bisa sangat berguna untuk tugas deteksi jalur dan pengembangan navigasi kendaraan.

2.11.4 Ultra Fast Structure-aware Deep Lane Detection (Qin, Zqundkk, 2020)

Dalam penelitian ini, Qin dkk telah mengusulkan formulasi baru dengan kehilangan struktural dan mencapai kecepatan dan akurasi yang luar biasa. Formulasi yang diusulkan menganggap deteksi jalur sebagai masalah pemilihan berbasis baris menggunakan fitur global. Dengan cara ini, masalah kecepatan dan tidak adanya petunjuk visual dapat diatasi. Selain itu, kerugian struktural yang digunakan untuk pemodelan informasi jalur sebelumnya secara eksplisit juga diusulkan. Efektivitas formulasi dan kehilangan struktural Qin dkk dibenarkan dengan baik dengan eksperimen kualitatif dan kuantitatif. Khususnya, model penulis yang menggunakan tulang punggung Resnet-34 dapat mencapai akurasi dan kecepatan tercanggih. Versi Resnet-18 yang ringan dari metode kami bahkan dapat mencapai 322,5 FPS dengan kinerja yang sebanding pada resolusi yang sama.

2.11.5 Unstructured Road Detection and Steering Assist Based on HSV Color Space Segmentation for Autonomous Car (A.A. Mahersatillah, 2020)

Penelitian ini bertujuan untuk merancang sebuah sistem deteksi jalan yang tidak terstruktur atau jalan yang tidak memiliki garis marka dan garis tepi berdasarkan hasil dari segmentasi warna objek yang berada di sisi jalan menggunakan ruang warna HSV untuk penerapan mobil otonom yang menghasilkan informasi posisi mobil dari tengah jalur dimana informasi ini kemudian yang akan menjadi parameter pada bagian kontrol. Selanjutnya deteksi tepi atau batas objek (jalan) menggunakan *edge detector* (Canny). Garis ini yang akan divisualisasikan menggunakan metode Hough Transform. Selanjutnya koordinat dari garis tepi jalan akan merepresentasikan lebar jalan. Pengujian dilakukan secara *realtime* dengan kecepatan mobil yang berbeda-beda. Hasil penelitian ini menunjukkan bahwa tingkat akurasi tertinggi dan terendah dalam penandaan batas tepi jalan yaitu 94,97% pada kecepatan 16,2 km/h dan 75,34% pada kecepatan 45,8 km/h, serta tingkat akurasi tertinggi dan terendah dalam menampilkan informasi posisi mobil dari tengah jalur yaitu 95,40% pada kecepatan 16,2 km/h dan 75,77% pada kecepatan 45,8 km/h