

SURVEI APLIKASI SEGMENTASI CITRA UNTUK *AUTONOMOUS VEHICLE*

Stevanus Darwin^{1*}, Nico Saputro²

¹Jurusan Teknik Elektro Konsentrasi Mekatronika, Universitas Katolik Parahyangan, Bandung, Indonesia

²Jurusan Teknik Elektro Konsentrasi Mekatronika, Universitas Katolik Parahyangan, Bandung, Indonesia

*E-mail: 6151801024@student.unpar.ac.id

ABSTRAK

Autonomous Vehicle (AV) merupakan sebuah teknologi yang dapat mengotomasi kendaraan berdasarkan level *driving otomation*. Dalam otomasi kendaraan akan dibutuhkan perangkat keras sebagai alat untuk berinteraksi dengan lingkungan seperti sensor dan aktuator. Sementara perangkat lunak dibutuhkan untuk memproses data yang sudah didapat dari sensor. Terdapat tiga komponen perangkat lunak dalam otomasi kendaraan yaitu persepsi, perencanaan dan kontrol. Persepsi merupakan kemampuan AV untuk berinteraksi dengan lingkungan sekitarnya. Tanpa adanya persepsi, tahap perencanaan dan kontrol tidak dapat dilakukan. Agar AV dapat membedakan satu objek dengan yang lain, maka diperlukan segmentasi citra yang dapat mengklasifikasi setiap objek pada sebuah frame. Dalam makalah ini akan dibahas aplikasi dari segmentasi citra untuk persepsi AV dan juga teknik segmentasi yang dapat digunakan dalam kondisi keterbatasan alat komputasi untuk aplikasi tersebut. Teknik segmentasi yang akan dibahas adalah segmentasi semantik untuk deteksi objek penghalang, dan segmentasi garis dengan menggunakan algoritma *Canny Edge Detector (CED)* untuk mendeteksi jalur yang akan dilewati.

Kata kunci: Autonomous Vehicle, Segmentasi citra, Persepsi

1. PENDAHULUAN

Seiring dengan rencana kepindahan ibukota negara Indonesia ke Kalimantan, pemerintah Indonesia juga telah merencanakan untuk mengadopsi penggunaan *Autonomous Vehicle (AV)* atau kendaraan otonom untuk ibukota baru Indonesia nantinya (BPPT,2020). Kendaraan otonom pada dasarnya merupakan kendaraan yang beroperasi melintasi lingkungan berkendaranya dengan melibatkan dua aktor utama yaitu manusia (sopir) dan suatu sistem otomasi (*automated system*). Bergantung dari seberapa besar peran dari kedua aktor utama ini saat mengemudi, *Society of Automotive Engineers (SAE)* mendefinisikan enam level *driving automation* yaitu level 0 (*no driving automation*), level 1 (*driver assistance*), level 2 (*partial driving automation*), level 3 (*conditional driving automation*), level 4 (*high driving automation*), dan level 5 (*full driving automation*) (ORAD Committee, 2021). Level 0 adalah level dimana manusia adalah satu-satunya aktor yang berperan saat mengemudi. Sebagai contoh adalah kendaraan yang umumnya kita pakai saat ini dimana manusia (sopir) mengontrol sepenuhnya kendaraan yang dikendarainya. Namun seiring dengan meningkatnya level *driving automation*, semakin berkurangnya peran manusia dan semakin meningkatnya peran sistem otomasi saat mengemudi. Di level tertinggi dari level *driving automation* yaitu level 5, sistem otomasi adalah satu-satunya aktor yang berperan saat mengemudi (*self-driving vehicle*).

Sistem otomasi (*automated system*) pada AV pada prinsipnya terdiri dari 2 komponen utama yaitu komponen perangkat keras dan perangkat lunak. Komponen perangkat keras yang terdapat dalam AV antara lain berupa sensor dan aktuator untuk berinteraksi dengan lingkungannya, baik lingkungan eksternal kendaraan maupun lingkungan internal. Beragam sensor dipergunakan untuk mendapatkan informasi dari lingkungan seperti misalnya kondisi jalan, tanda-tanda lalu lintas, kondisi rem, dan kondisi ban. Beragam actuator dipergunakan untuk melakukan aksi terhadap lingkungan berkendara. Komponen perangkat lunak dari sistem otomasi dapat dikelompokkan menjadi tiga yaitu untuk perangkat lunak sistem persepsi, perangkat lunak sistem perencanaan, dan perangkat lunak sistem kontrol (Pendleton, dkk., 2017). Persepsi adalah cara AV berinteraksi dengan lingkungan sekitarnya berdasarkan data yang diperoleh dari berbagai sensor. Data tersebut diolah untuk menghasilkan pemahaman kontekstual terhadap lingkungan sekitarnya (*environmental perception*) dan lokasi kendaraan terhadap lingkungannya (*localization*). Selanjutnya informasi kontekstual yang diperoleh dari perangkat lunak persepsi ini dipergunakan oleh perangkat lunak perencanaan untuk membawa kendaraan dari posisi awal ke posisi tujuan. Perangkat lunak kontrol selanjutnya dipergunakan untuk mengeksekusi hasil dari perangkat lunak perencanaan dalam bentuk *path tracking* dan *trajectory tracking*.

Sistem persepsi visual merupakan salah satu sistem otomasi terpenting pada AV karena berbagai keputusan penting diambil bergantung kepada persepsi visual dari lingkungan sekitarnya. Setiap keputusan yang diambil oleh sistem otomasi dapat menghasilkan keputusan yang tidak aman bila tanpa adanya persepsi yang tepat. AV biasanya akan dilengkapi dengan berbagai peralatan penginderaan seperti misalnya radar, LIDAR (*Light Detection and Ranging*), *ultra-sonic*, GPS (*Global Positioning Satellite*), kamera, dsb untuk mendapatkan data persepsi visual lingkungan sekitarnya dan lokalisasi. Data dari berbagai peralatan tersebut selanjutnya akan diproses oleh perangkat lunak berbasis *machine-learning* dan *computer vision* untuk memperoleh pemahaman yang akurat tentang lingkungan sekitar AV seperti misalnya mendeteksi kendaraan, pejalan kaki, permukaan jalan, jalur yang akan dilalui, dsb.

Variasi penggunaan dari berbagai kombinasi perangkat keras dan berbagai algoritma berbasis *machine-learning* dan *computer vision* untuk AV menentukan di *level driving automation* mana suatu AV berada. Mengingat adanya beragam algoritma yang dapat digunakan untuk AV, pada makalah ini penulis menyajikan hasil studi literatur mengenai penggunaan *computer vision* algoritma untuk AV, khususnya penggunaan teknik segmentasi citra untuk melakukan klasifikasi obyek berdasarkan fitur yang dimiliki oleh obyek tersebut. Studi literatur dilakukan dengan mencari melalui *google scholar* makalah konferensi dan/atau jurnal berdasarkan kriteria seperti tahun penerbitan 2012 hingga 2021. Selanjutnya berdasarkan makalah yang diperoleh, dilakukan proses seleksi berdasarkan relevansi terhadap topik yang akan dibahas pada makalah ini.

2. APLIKASI SEGMENTASI CITRA

Aplikasi segmentasi citra untuk AV pada umumnya digunakan untuk persepsi lingkungan dan lokalisasi dan pemetaan seperti diuraikan pada subbab berikut ini.

2.1 Persepsi Lingkungan

Persepsi Lingkungan adalah kemampuan AV untuk mendapatkan informasi yang kontekstual tentang lingkungan di sekelilingnya (Pendleton, dkk., 2017). Seperti halnya bila kendaraan dikemudikan oleh manusia, sistem otomasi pada AV juga memerlukan informasi kontekstual tentang jalan dan marka jalan (Hillel dkk, 2014), dan obyek-obyek yang ada di jalan dan sekitarnya seperti kendaraan lain (Sivaraman dan Trivedi, 2013) dan pejalan kaki (Dollar dkk, 2012). Informasi kontekstual ini selalu diperlukan oleh AV baik saat AV berada di daerah perkotaan, pedesaan, maupun di jalan bebas hambatan (*highway*).

Sehubungan dengan jalan dan marka jalan, informasi penting yang diperlukan oleh AV antara lain tentang jumlah dan posisi lajur jalan yang tersedia, penggabungan lajur jalan misalnya dari empat lajur menjadi dua lajur, pemisahan lajur misalnya dari satu lajur menjadi dua lajur, akhir dari suatu lajur atau jalan, dan bentuk dari marka lajur jalan misalnya marka garis putus-putus, marka garis sambung, marka garis ganda (garis marka sambung dan putus), dan marka garis melintang (*zebra cross*). Sedangkan informasi kontekstual yang berhubungan dengan kendaraan yang ada di jalan, bukan hanya sebatas fitur-fitur kendaraan saja (*vehicle detection*) seperti bentuk, warna, ukuran, dan kecepatan, tetapi AV juga memerlukan informasi kontekstual yang berhubungan dengan *vehicle tracking* (estimasi pergerakan kendaraan yang telah terdeteksi sebelumnya) dan *vehicle behavior* (*maneuver* kendaraan, deskripsi aktivitas di jalan, dan perkiraan gerak kendaraan) (Sivaraman dan Trivedi, 2013).

Beragam sensor, baik berupa sensor tunggal maupun gabungan dari beberapa sensor (*sensor fusion*) dapat dipergunakan untuk mendapatkan data tentang lingkungan sekitar kendaraan. Ada tiga tipe sensor yaitu sensor yang bekerja menggunakan gelombang radio (radar), laser (LIDAR), dan cahaya tampak (*vision* dengan menggunakan kamera) (Sivaraman dan Trivedi, 2013). Setiap sensor tersebut memiliki keunggulan dan kelemahannya masing-masing. Menurut Sivaraman dan Trivedi (2013), radar dapat bekerja dengan baik untuk aplikasi dengan *narrow field-of-view* dan tidak terpengaruh oleh cuaca dan pencahayaan. Namun radar mempunyai beberapa keterbatasan seperti banyaknya noise dari hasil pengukurannya sehingga memerlukan proses *filtering* dan *cleaning*, dan radar tidak melakukan deteksi kendaraan secara langsung, melainkan melakukan interpretasi bahwa suatu obyek yang tertangkap oleh radar itu adalah kendaraan berdasarkan pergerakan relatif dari obyek tersebut. LIDAR bekerja dengan baik untuk aplikasi yang memerlukan *wide field-of-view*. LIDAR, melakukan interpretasi dan melakukan klasifikasi obyek adalah suatu kendaraan berdasarkan ukuran dan gerak, dan menghasilkan hasil pengukuran yang lebih jernih (lebih sedikit *noise*-nya) dibandingkan radar. Namun, LIDAR lebih sensitif terhadap cuaca dan harga lebih mahal. Kamera merupakan sensor yang dapat menghasilkan gambar atau video secara langsung dengan biaya terjangkau meskipun memerlukan biaya komputasi (*computation cost*) yang lebih tinggi karena menggunakan berbagai teknik *computer vision* untuk mendeteksi adanya kendaraan pada gambar atau video beserta informasi-informasi tambahan lainnya yang berguna bagi AV. Kamera dapat menyediakan *wide view-of-angle* meskipun hasilnya terpengaruh oleh oleh

pencahayaan dan cuaca. Gambar yang dihasilkan oleh kamera dapat diolah dengan suatu algoritma segmentasi citra untuk menghasilkan klasifikasi dari berbagai obyek yang ada di gambar tersebut.

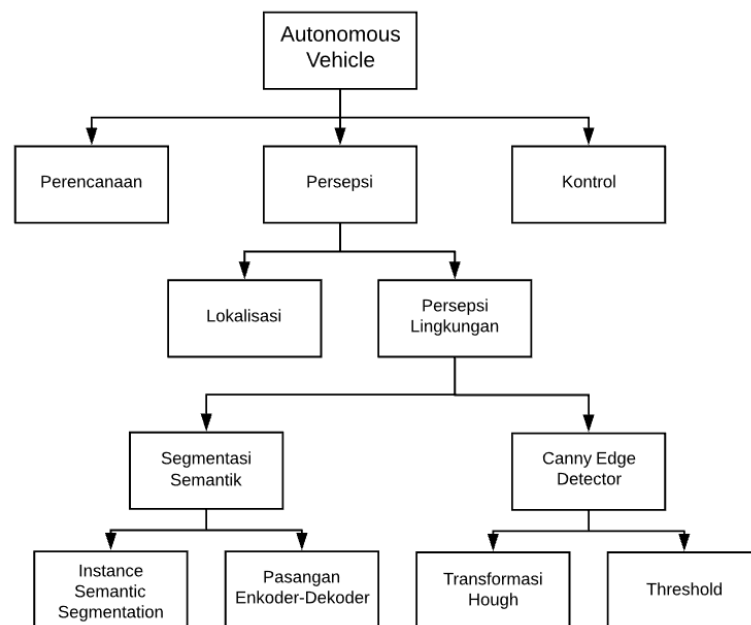
2.2 Lokalisasi & Pemetaan

Lokalisasi adalah kemampuan untuk menentukan posisi AV didalam sebuah lingkungan, sementara pemetaan adalah menggabungkan berbagai observasi menjadi sebuah model dan navigasi melalui jalur yang paling baik (Fuentes-Pacheco, dkk., 2015). Setelah segmentasi citra dilakukan terhadap foto dari kamera, maka software dapat menghitung posisi AV berdasarkan jarak kamera dengan *landmark* yang terdeteksi, kemudian sistem akan membandingkan informasi tersebut dengan peta yang sudah dimasukan kedalam sistem.

Pemetaan juga dapat dilakukan dengan menggunakan informasi hasil dari persepsi dengan menggabungkan deteksi jalan dan persepsi lingkungan untuk mengidentifikasi lingkungan sekitar pada jalan tersebut. Setelah diproses, maka perangkat lunak mendapatkan informasi mengenai jalan yang dapat dilewati oleh AV dengan menggunakan algoritma *Simultaneous Localization and Mapping (SLAM)* yang dimana melalui LIDAR, perangkat lunak mampu membuat sebuah peta dalam skala global dari tempat yang sudah pernah dikunjungi sekaligus menentukan posisi AV dari peta tersebut (Fuentes-Pacheco, dkk., 2015). Namun lokalisasi dan pemetaan dengan menggunakan LIDAR juga memiliki banyak masalah, seperti resolusi kamera yang kurang, sensitif terhadap cahaya, permukaan dengan kekurangan tekstur, foto yang kabur akibat pergerakan yang cepat, dan faktor lainnya yang menyebabkan kurangnya informasi yang dibutuhkan.

Lokalisasi dan pemetaan pada AV biasanya menggunakan sensor gabungan yang lebih akurat dan dapat diandalkan seperti GPS, atau *Inertial Measurement Unit (IMU)* (Fayyad, dkk., 2020). Walaupun GPS dapat melakukan pengukuran dengan akurat dan dapat diandalkan, ada kondisi dimana GPS tidak dapat mengambil data. Untuk itu biasanya GPS digabungkan dengan sensor IMU yang biayanya cukup murah. Dengan menggunakan gabungan GPS dan IMU, kondisi AV dapat selalu ditentukan dengan akurat dan beban komputasi pada perangkat keras yang kemampuan komputasinya terbatas menjadi lebih ringan.

3. TEKNIK SEGMENTASI CITRA

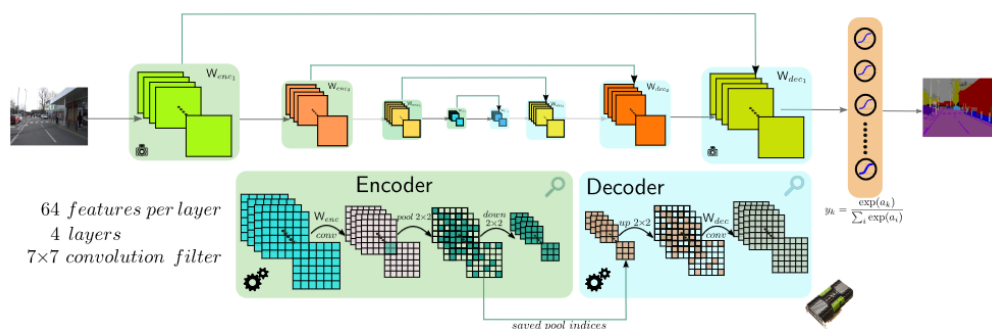


Gambar 1. Skema algoritma yang digunakan untuk persepsi lingkungan

Teknik yang akan dibahas meliputi persepsi lingkungan mengenai deteksi objek penghalang dan juga deteksi jalur yang akan dilewati oleh AV. Gambar 1 merupakan skema algoritma yang akan digunakan untuk melakukan persepsi lingkungan, yang dimana untuk deteksi objek penghalang, segmentasi semantik akan digunakan. Segmentasi ini memiliki berbagai macam metode dengan memecah gambar kedalam skala pixel dan mengklasifikasi *pixel* tersebut. Sementara untuk deteksi garis jalur, algoritma *Canny Edge Detector* akan

digunakan. Algoritma ini hanya mendeteksi garis sehingga perlu dibantu oleh algoritma lain untuk mendeteksi dan juga memprediksi jalur yang akan dilewati AV. Namun, sebelum masuk kedalam teknik segmentasi citra yang menggunakan *machine learning*, perlu diketahui bahwa segmentasi dilakukan secara bertahap dari klasifikasi kasar menjadi klasifikasi yang lebih halus (Garcia-Garcia, dkk., 2017). Klasifikasi kasar dapat berupa klasifikasi yang memisahkan objek dari latar belakang. Setelah itu dapat dilakukan langkah selanjutnya yaitu klasifikasi yang memberikan informasi spasial seperti pusat massa, koordinat, maupun bentuk dari objek tersebut. Untuk mencapai hasil klasifikasi yang baik, terdapat banyak arsitektur jaringan dengan berbagai macam strategi yang dijadikan standar dalam klasifikasi. Maka dari itu perlu diketahui struktur strategi dari arsitektur jaringan yang digunakan untuk membantu pembaca mengerti lebih dalam prinsip kerja dari teknik-teknik segmentasi berbasis *machine learning*.

3.1 Arsitektur Jaringan Klasifikasi



Gambar 2. Contoh Arsitektur Jaringan SegNet (Badrinarayanan dkk., 2015)

Sebelum masuk kedalam teknik segmentasi citra yang menggunakan *machine learning*, perlu diketahui bahwa segmentasi dilakukan secara bertahap dari klasifikasi kasar menjadi klasifikasi yang lebih halus (Garcia-Garcia, dkk., 2017). Gambar 2 merupakan sebuah contoh arsitektur jaringan dimana decoder melakukan klasifikasi kasar dengan membagi gambar menjadi bagian-bagian kecil, lalu encoder melakukan klasifikasi yang lebih halus dengan memberikan label untuk setiap bagian kecil. Setelah diberi label, bagian digabungkan kembali untuk mendapatkan informasi mengenai masing-masing objek yang harus di klasifikasi.

Klasifikasi kasar dapat berupa klasifikasi yang memisahkan objek dari latar belakang. Setelah itu dapat dilakukan langkah selanjutnya yaitu klasifikasi yang memberikan informasi spasial seperti pusat massa, koordinat, maupun bentuk dari objek tersebut. Untuk mencapai hasil klasifikasi yang baik, terdapat banyak arsitektur jaringan dengan berbagai macam strategi yang dijadikan standar dalam klasifikasi. Maka dari itu perlu diketahui struktur strategi dari arsitektur jaringan yang digunakan untuk membantu pembaca mengerti lebih dalam prinsip kerja dari teknik-teknik segmentasi berbasis *machine learning*.

3.1.1 SqueezeNet

SqueezeNet merupakan sebuah arsitektur jaringan yang dilatih khusus untuk klasifikasi dengan tujuan untuk membuat jaringan *Convolutional Neural Network* (CNN) dengan parameter input yang lebih kecil namun tetap mempertahankan akurasi dari hasil klasifikasi (Iandola dkk., 2017). Untuk mencapai tujuan tersebut, SqueezeNet menggunakan 3 strategi utama yaitu Mengganti filter input dari 3x3 menjadi 1x1, mengurangi *channel* input menjadi filter 3x3, dan melakukan *downsampling* dilangkah akhir dari arsitektur untuk mendapatkan peta aktivasi yang besar. Berdasarkan Iandola, SqueezeNet mampu menyaingi akurasi AlexNet dengan parameter input 50x lebih sedikit dan ukuran model yang dibutuhkan kurang dari 0.5 MB.

3.1.2 AlexNet

AlexNet merupakan arsitektur jaringan yang memiliki akurasi *state-of-the-art* pada tahun 2012. AlexNet memiliki strategi bahwa untuk meningkatkan akurasi, mesin harus mempelajari berjuta-juta gambar sehingga jaringan memiliki pengetahuan yang sangat banyak (Krizhevsky, dkk., 2012). AlexNet memiliki 5 *convolutional layers* dan 3 *fully-connected layers*. Walaupun memiliki akurasi yang sangat tinggi, AlexNet memiliki struktur yang sangat besar dengan parameter input mencapai 224x224x3.

3.1.3 VGG-16

VGG-16 merupakan arsitektur jaringan yang terinspirasi dari AlexNet namun dengan akurasi yang jauh lebih tinggi. VGG memiliki setumpuk *convolutional layers* dan 3 *fully-connected layers*. Perbedaan VGG-16 dari pendahulunya adalah tumpukan *convolutional layers* menggunakan ukuran yang lebih kecil sehingga input yang dibutuhkan lebih sedikit dan mempermudah fungsi arsitektur dalam membedakan fitur dari gambar (Simonyan, Zisserman, 2015).

3.1.4 Enet

Enet merupakan arsitektur jaringan khusus untuk segmentasi semantik. Enet memiliki tujuan untuk mempertahankan akurasi yang tinggi dan mempercepat waktu eksekusi dengan melewati pasca proses segmentasi sehingga mengorbankan resolusi hasil gambar (Paszke, dkk., 2016). Enet memiliki cabang utama yang berisi *Max Pooling*, dan cabang samping yang berisi filter konvolusi. Setelah itu hasil dari kedua cabang digabung dengan menggunakan operasi *Bitwise* yang melakukan operasi AND pada kedua gambar.

3.1.5 Segnet

Segnet merupakan arsitektur jaringan khusus untuk segmentasi semantik yang menggunakan algoritma pasangan dekoder-encoder. Karena Segnet menggunakan encoder dengan resolusi yang kecil, hasil yang didapat lebih mulus dibanding arsitektur lainnya (Badrinarayanan dkk., 2015). Segnet juga menggunakan beban komputasi yang kecil dengan akurasi yang kompetitif dengan arsitektur lainnya.

3.1.6 Resnet38

ResNet38 atau *Residual Network* merupakan arsitektur jaringan yang dibuat untuk menyelesaikan masalah degradasi akurasi yang disebabkan oleh gradien yang menghilang (He dkk., 2016). Untuk mengatasi masalah tersebut, ResNet memberikan referensi untuk operasi yang akan dilakukan berdasarkan hasil operasi sebelumnya. Dengan menggunakan referensi, maka jaringan lebih mudah untuk dioptimisasi dan dapat mengatasi masalah gradien yang menghilang.

3.2 Segmentasi Semantik

Segmentasi semantik telah berkembang pesat seiring berkembangnya teknologi *deep learning*. Selama ini, klasifikasi gambar dilakukan dengan mengekstrak fitur-fitur unik seperti intensitas cahaya, kontinuitas garis, ataupun daerah yang memiliki fitur yang mirip pada gambar. Berbeda dari segmentasi konvensional, segmentasi semantik melakukan pemetaan dan klasifikasi fenomena lingkungan sekitar dalam skala pixel (Kaymak dan Ucar, 2019). Semantik itu sendiri adalah bidang yang menghubungkan arti dari sebuah kata dengan bagaimana kita menggambarkan kata tersebut. Segmentasi semantik ini dapat digunakan untuk menghindari halangan.

Segmentasi semantik memiliki keuntungan ketika digunakan untuk AV dibanding segmentasi konvensional. *Output* dari hasil segmentasi semantik adalah sebuah distribusi diskrit yang berdasarkan kelas klasifikasi untuk setiap pixel. Dengan output diskrit ini, beberapa objek dapat diklasifikasi secara langsung dan juga dapat ditentukan posisinya didalam sebuah gambar (Kaymak dan Ucar, 2019). Selanjutnya akan dibahas beberapa algoritma yang dapat digunakan untuk melakukan segmentasi semantik.

3.2.1 Pasangan Encoder-Decoder

Trembl dkk. (2016) menggunakan arsitektur pasangan encoder-decoder, dimana pertama gambar dikecilkan resolusinya dengan menggunakan encoder arsitektur SqueezeNet 1.1 yang telah dimodifikasi. Gambar yang sudah diturunkan resolusinya dibagi menjadi beberapa bagian. Kemudian Gambar tersebut di konvolusi secara parallel untuk mendapatkan kecepatan eksekusi yang cepat. Setelah di konvolusi, masing-masing gambar yang sudah dilabel digabungkan kembali untuk mendapatkan resolusi gambar seperti semula dengan menggunakan dekoder. Untuk meningkatkan akurasi dari encoder, pengembalian resolusi tidak hanya menggunakan informasi dari layer sebelumnya, namun juga dibandingkan dengan informasi dari gambar sebelum diturunkan resolusinya oleh encoder.

Setelah dilakukan simulasi dengan menggunakan dataset *Cityscapes*, akurasi dari segmentasi Trembl dkk. Memiliki akurasi yang lebih tinggi dibanding hasil Enet kecuali pada identifikasi label tembok, truk, bus, kereta, dan sepeda motor. Segmentasi ini memiliki pengembangan pesat dalam identifikasi tiang, lampu dan rambu lalu lintas, dan juga orang dengan masing-masing peningkatan akurasi sebanyak 7,4%, 17,9%, 17,7%, dan 8,4%.

Kaymak dan Ucar (2019) menggunakan arsitektur jaringan yang sudah dilatih dengan tujuan klasifikasi seperti AlexNet dan VGG-16. Berbeda dari arsitektur jaringan SqueezeNet, Arsitektur AlexNet dan VGG-16 memiliki tujuan untuk mengidentifikasi gambar dengan tingkat keakuratan yang sangat tinggi, walaupun beban komputasi yang dibutuhkan juga menjadi lebih besar. Kemudian arsitektur tersebut diubah menjadi *Fully*

Convolutional Network (FCN). Kaymak dan Ucar menggunakan arsitektur AlexNet dan VGG-16 yang telah diubah dan diberi nama FCN-AlexNet untuk arsitektur AlexNet, dan FCN-8s, FCN-16s, dan FCN32s untuk VGG-16. 8s, 16s, dan 32s merepresentasikan jumlah pixel yang dinaikan resolusinya dalam satu step.

Setelah dilakukan Simulasi pada dataset SYNTHIA-Rand-CVPR16, FCN-AlexNet memiliki akurasi 92%, sementara FCN-8s, FCN-16s, dan FCN-32s memiliki akurasi 96%, 95%, dan 94% secara berurutan. Jika dilihat dari waktu pelatihan, AlexNet memiliki 34 μ s, sementara FCN-8s, FCN-16s dan FCN-32s memiliki waktu pelatihan \pm 129 μ s. Walaupun, AlexNet memiliki waktu pelatihan yang jauh lebih pendek, namun untuk AV, yang paling penting adalah akurasi dan waktu eksekusi. Sehingga FCN-8s merupakan algoritma yang dapat diimplementasikan untuk AV karena memiliki akurasi yang paling tinggi.

3.2.2 Instance Semantic Segmentation

Segmentasi semantik juga memiliki kelemahan, yaitu segmentasi semantik tidak dapat membedakan objek dengan klasifikasi sama yang bertabrakan. Untuk mengatasi masalah ini, Brabandere dkk. (2017) menggunakan sebuah *loss function* dengan dua istilah yang berlawanan yaitu *variance term* yang mendekati rata-rata dari label yang sama, dan *distance term* yang mendorong pixel yang memiliki label berbeda menjauhi satu dengan yang lainnya. *Loss function* juga dilengkapi dengan fungsi regulasi untuk memastikan tidak ada pixel dari sebuah grup yang saling berjauhan. Algoritma ini tahan terhadap oklusi yang kompleks, sehingga sangat berguna untuk adaptasi terhadap lingkungan AV. Algoritma ini telah digabungkan oleh arsitektur jaringan klasifikasi seperti ENet, Segnet, dan Resnet38 yang dibuat khusus untuk identifikasi gambar. Hasilnya adalah arsitektur Resnet38 memiliki akurasi rata-rata yang paling tinggi, namun membutuhkan memori yang lebih besar.

3.3 Canny Edge Detector

Canny Edge Detector (CED) adalah salah satu algoritma terbaik untuk mendeteksi garis dari suatu gambar karena tidak membutuhkan filter lebih lanjut sebelum memproses hasil pendeteksian garis. CED mendeteksi semua garis hingga garis yang dihasilkan oleh gangguan seperti bayangan, ataupun kerikil yang berada di jalan. Oleh karena itu dibutuhkan algoritma lain untuk membantu CED agar dapat mengklasifikasi garis-garis yang sudah dideteksi oleh CED.

3.3.1 Transformasi Hough

Bounini dkk. (2015) mengatasi masalah ini dengan menggunakan Transformasi Hough (HT) untuk membantu CED mendeteksi jalur. Algoritma HT tidak terpengaruh posisi pixel pada sebuah frame, juga tahan terhadap noise dan oklusi, membuat HT merupakan sebuah algoritma yang cocok untuk AV. Namun HT memiliki beban komputasi yang besar sehingga sulit untuk digunakan pada AV yang memiliki kemampuan komputasi yang terbatas. Untuk mengurangi beban komputasi dari HT, Bounini dkk. Mendefinisikan daerah yang harus di proses secara eksplisit (*Region of Interest*). Dengan mengurangi daerah yang harus diproses dengan HT, maka beban komputasi yang dibutuhkan akan berkurang. Setelah diklasifikasi garis lajur, maka bentuk dari jalur dapat diprediksi dengan menggunakan *Least-Squared Method* dan *Kalman Filter*.

3.3.2 Threshold

Li dkk. (2012) pertama melakukan pemetaan dengan teknik *Inverse Perspective Mapping* untuk mengubah perpektive kamera seolah-olah gambar diambil dari atas, Lalu menggunakan *adaptive thresholding* dan juga CED untuk mendeteksi garis pada jalur. Setelah dideteksi, kedua hasil output diintegrasikan dengan menggunakan menggunakan segmentasi kontur lalu hasil kontur dibandingkan dengan kedua hasil deteksi. Untuk setiap bagian dari hasil threshold dan CED berada didalam hasil segmentasi kontur, maka bagian tersebut diubah menjadi *binary*. Jika tidak cocok, maka pixel dianggap bagian dari latar belakang gambar. Setelah didapat sebuah gambar *binary*, maka gambar dianalisis lebih lanjut untuk menghilangkan elemen yang tidak dibutuhkan.

4. KESIMPULAN

Segmentasi citra merupakan sebuah cara untuk *Autonomous Vehicle* mengumpulkan informasi mengenai lingkungan sekitarnya secara *real-time*. Terdapat beberapa segmentasi citra yang dapat digunakan untuk melakukan persepsi pada AV. Segmentasi semantik mampu mengklasifikasi setiap objek dalam skala pixel dengan menggunakan teknologi *machine learning*. Segmentasi ini digunakan untuk mendeteksi objek penghalang. Terdapat berbagai macam algoritma seperti pasangan dekoder-encoder, dan *instance semantic segmentation*. Selain segmentasi semantik, segmentasi juga dapat menggunakan *Canny Edge Detector* yang menggunakan fitur garis untuk melakukan klasifikasi. Segmentasi ini digunakan untuk deteksi garis jalur yang

akan dilewati oleh AV. Hasil dari CED ditingkatkan lagi lebih jauh dengan menggunakan metode Transformasi Hough, ataupun *Thresholding* untuk meningkatkan keakuratan hasil segmentasi.

PUSTAKA

- Badrinarayanan, V., Handa, A., & Cipolla, R. (2015). *Segnet: A deep convolutional encoder-decoder architecture for robust semantic pixel-wise labelling*. *arXiv preprint arXiv:1505.07293*.
- Bounini, F., Gingras, D., Lapointe, V., & Pollart, H. (2015, October). *Autonomous vehicle and real time road lanes detection and tracking*. In *2015 IEEE Vehicle Power and Propulsion Conference (VPPC)* (pp. 1-6). IEEE.
- BPPT, Autonomous Vehicle (AV) akan diadopsi untuk ibu kota baru Indonesia. (Online) (<https://www.bppt.go.id/teknologi-hankam-transportasi-manufaktur/4017-autonomous-vehicle-av-akan-diadopsi-untuk-ibu-kota-baru-indonesia>, diakses 20 Juni 2021).
- De Brabandere, B., Neven, D., & Van Gool, L. (2017). *Semantic instance segmentation for autonomous driving*. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops* (pp. 7-9).
- Dollar, P.; Wojek, C.; Schiele, B.; Perona, P. (2012). *Pedestrian detection: An evaluation of the state of the art*. *IEEE Trans. Pattern Anal. Mach. Intell.* 34, 743–761.
- Fayyad, J., Jaradat, M. A., Gruyer, D., & Najjaran, H. (2020). *Deep learning sensor fusion for autonomous vehicle perception and localization: A review*. *Sensors*, 20(15), 4220.
- Fuentes-Pacheco, J., Ruiz-Ascencio, J., & Rendón-Mancha, J. M. (2015). *Visual simultaneous localization and mapping: a survey*. *Artificial intelligence review*, 43(1), 55-81.
- He, K., Zhang, X., Ren, S., & Sun, J. (2016). *Deep residual learning for image recognition*. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770-778).
- Hillel, A.B.; Lerner, R.; Levi, D.; Raz, G. (2014). *Recent progress in road and lane detection: A survey*. *Mach. Vis. Appl.* 2014, 25, 727–745.
- Iandola, F. N., Han, S., Moskewicz, M. W., Ashraf, K., Dally, W. J., & Keutzer, K. (2016). SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and < 0.5 MB model size. *arXiv preprint arXiv:1602.07360*.
- Kaymak, Ç., & Uçar, A. (2019). *A brief survey and an application of semantic image segmentation for autonomous driving*. In *Handbook of Deep Learning Applications* (pp. 161-200). Springer, Cham.
- Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). *Imagenet classification with deep convolutional neural networks*. *Advances in neural information processing systems*, 25, 1097-1105.
- Li, Z., Cai, Z. X., Xie, J., & Ren, X. P. (2012, May). *Road markings extraction based on threshold segmentation*. In *2012 9th International Conference on Fuzzy Systems and Knowledge Discovery* (pp. 1924-1928). IEEE.
- On-Road Automated Driving (ORAD) Committee. *Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles*. 2021. (Online) (https://saemobilus.sae.org/content/j3016_202104, tanggal akses 20 Juni 2021).
- Paszke, A., Chaurasia, A., Kim, S., & Culurciello, E. (2016). *Enet: A deep neural network architecture for real-time semantic segmentation*. *arXiv preprint arXiv:1606.02147*.
- Pendleton, S. D., Andersen, H., Du, X., Shen, X., Meghjani, M., Eng, Y. H., ... & Ang, M. H. (2017). *Perception, planning, control, and coordination for autonomous vehicles*. *Machines*, 5(1), 6
- Simonyan, K., & Zisserman, A. (2014). *Very deep convolutional networks for large-scale image recognition*. *arXiv preprint arXiv:1409.1556*.
- Sivaraman, S.; Trivedi, M.M. (2013). *Looking at Vehicles on the Road: A Survey of Vision-Based Vehicle Detection, Tracking, and Behavior Analysis*. *IEEE Trans. Intell. Transp. Syst.* 14, 1773–1795.
- Treml, M., Arjona-Medina, J., Unterthiner, T., Durgesh, R., Friedmann, F., Schuberth, P., & Hochreiter, S. (2016, December). *Speeding up semantic segmentation for autonomous driving*. In *MLITS, NIPS Workshop* (Vol. 2, No. 7).
- Van Brummelen, J., O'Brien, M., Gruyer, D., & Najjaran, H. (2018). *Autonomous vehicle perception: The technology of today and tomorrow*. *Transportation research part C: emerging technologies*, 89, 384-406.