Analisis Metoda Estimasi Peta Kedalaman Dari Citra RGB 2D Untuk Penerapan Teknik Persepsi Pada Kendaraan Otonom

[1]Yan Setiaji*), [1]Muhammad Raihan Miransyahputra**) & [1,2]Yul Yunazwin Nazaruddin***)

[1]Program Studi Teknik Fisika, Fakultas Teknologi Industri, Institut Teknologi Bandung, Indonesia

[2]Kelompok Keahlian Instrumentasi dan Kontrol, Fakultas Teknologi Industri, Institut Teknologi Bandung, Indonesia

*) *yan.setiaji@gmail.com*, **) *raihanmiransyah@gmail.com*, ***) *yul@tf.itb.ac.id*

Abstrak

Seiring berjalannya perkembangan teknologi kendaraan otonom, dilakukan berbagai usaha untuk dapat menciptakan metode pengolahan data sebagai masukkan pada sistem kontrolnya. Salah satunya adalah pengembangan metode pada sistem persepsi visual, yaitu estimasi peta kedalaman dari pengolahan citra RGB 2D. Metode ini dikembangkan untuk dapat memperkuat penggunaan kamera 2D yang tidak mampu memberikan informasi ruang 3D dari lingkungan layaknya LiDAR, Radar, atau teknik ultrasonik. Metode yang akan dipresentasikan pada makalah ini menerapkan teknik Jaringan Syaraf Tiruan untuk dapat mengestimasi jarak objek dari titik perekaman. Analisis yang tajam mengenai persepsi yang dihasilkan telah dilakukan dengan menggunakan tiga arsitektur, penggunaan jaringan penskalaan global dan lokal pada arsitektur pertama, penambahan jaringan prediksi menengah pada arsitektur kedua, serta penggunaan empat modul jaringan yang terdiri dari pengkode, dekoder, gabungan fitur multi skala, dan jaringan perbaikan pada arsitektur ketiga. Pelatihan dan pengujian dicoba dengan menggunakan beberapa data set yang merepresentasikan kondisi di dalam dan di luar ruangan. Hasil simulasi dan analisis yang diperoleh memperlihatkan bahwa arsitektur ketiga memberikan hasil yang paling baik dalam memberikan informasi ruang 3D dari lingkungannya.

Kata Kunci: kendaraan otonom, peta kedalaman, visi komputer, jaringan syaraf, tiruan

1 Pendahuluan

Salah satu aspek penting pada penggunaan kendaraan otonom adalah persepsi. Aspek persepsi merupakan kemampuan sistem otonom untuk mengekstrak informasi penting dari lingkungan dan menentukan lokasi [1]. Pada saat kendaraan otonom berjalan dibutuhkan perangkat atau instrumen yang mampu memetakan atau merekam kondisi lingkungan sehingga dapat dilakukan ekstraksi informasi tersebut. Pada penelitian terdahulu dilakukan studi terkait penggunaan instrumen *Light Detection and Ranging* (LiDAR) oleh Royo dan Ballesta-Garcia [2], pengajuan penggunaan *Radio Detection and Ranging* (RADAR) oleh Bilik et al. [3] dan Roos et al. [4], penggunaan sensor ultrasonik oleh Xu dan Yan [5], dan kamera dilengkapi visi komputer oleh Pidurkar et al. [6]. Namun saat ini penggunaan LiDAR mendominasi sebagai perangkat yang dipilih, karena kelebihannya dari segi presisi, akurasi, dan kemampuan merekam ruang 3D dari lingkungan. Namun LiDAR memiliki kekurangan pada segi ketersediaan dan biaya perangkat yang relatif lebih mahal, sehingga dibutuhkan alternatif lain untuk dapat berfungsi layaknya LiDAR. Solusi alternatif yang ditawarkan adalah penggunaan kamera RGB 2D dilengkapi algoritma komputer visi untuk melakukan estimasi peta kedalaman. Estimasi ini digunakan untuk memberikan informasi ruang 3D dari lingkungan.

Pada makalah ini dilakukan komparasi performa dari 3 (tiga) arsitektur yang berbeda untuk metode tersebut. Selanjutnya juga dilakukan analisis kemungkinan penerapan metode estimasi peta kedalaman pada kendaraan otonom.

2 Dasar Teori

2.1 Peta Kedalaman

Peta kedalaman merupakan istilah yang diberikan pada hasil pencitraan 3D baik dengan perekaman kamera stereo ataupun pemindaian 3D dengan laser. Penggunaan kata kedalaman merupakan penjelasan dari jarak objek pada citra dengan titik perekaman sensor. Besar jarak diwakili dengan nilai piksel untuk menentukan jauh dekatnya, sehingga gabungan piksel yang menyusun citra hasil rekam menjadikan citra memberikan informasi posisi objek layaknya peta [7]. Pada metode estimasi peta kedalaman dari citra RGB 2D, dilakukan sebuah pendekatan untuk dapat mengekstrak aspek kedalaman dari sebuah citra 2D, seperti yang diperlihatkan pada Gambar 1. Untuk melakukan pendekatan tersebut dibutuhkan suatu algoritma yang mampu mempelajari fitur citra, dan untuk itu akan digunakan algoritma Jaringan Syaraf Tiruan (JST).

Gambar 1. (kiri) Citra RGB 2D, (kanan) Depth map

2.2 Jaringan Syaraf Tiruan (JST)

Jaringan Syaraf Tiruan adalah suatu pengolah informasi atau prosesor yang mencoba untuk memodelkan otak manusia, yang terdistribusi secara paralel dan memiliki kemampuan natural untuk menyimpan pengetahuan berdasarkan pengalaman, sehingga dapat digunakan untuk penyelesaian berbagai permasalahan yang memerlukan informasi-informasi masa lampau [8,9]. Algoritma JST dirancang untuk memiliki kemampuan belajar, mensarikan, dan memberikan kesimpulan. JST sendiri terbagi menjadi 3 model berdasarkan data belajarnya, yaitu perseptron multi-lapis (*Multi Layer Perceptron*/MLP) untuk data tabular, jaringan syaraf konvolusi (*Convolutional Neural Network*/CNN) untuk data citra, dan jaringan syaraf berulang (*Recurrent Neural Network*/RNN) untuk data sekuensial. Pada metode estimasi peta kedalaman digunakan jaringan syaraf konvolusi (CNN) untuk mengekstrak fitur citra.

3 Perancangan Model

3.1 Arsitektur

Untuk proses estimasi peta kedalaman dari citra RGB 2D dibutuhkan model Jaringan Syaraf Konvolusi yang perlu dilatih terlebih dahulu. Pada penelitian analisis metode ini digunakan 3 varian arsitektur Jaringan Syaraf Konvolusi dari Eigen et al. (2014) [10], yang selanjutnya disebut arsitektur 1, Eigen dan Fergus (2015) [11], disebut arsitektur 2, dan Hu et al. (2019) [12] disebut arsitektur 3. Juga pada pengembangan perangkat lunak yang diperlukan untuk analisis digunakan PyTorch sebagai pustaka pembelajaran mesin, karena lebih mudah dalam proses implementasi model.

Pada arsitektur 1, struktur lapisan JST yang digunakan mempunyai 2 kelompok lapisan, yaitu kelompok lapisan kasar (coarse) dan halus (fine). Kasar dan halus ini dibedakan berdasarkan proses ekstraksi fitur citra masukan. Citra masukan terlebih dahulu melewati kelompok x` kasar, lalu terjadi proses konvolusi, pooling, dan ekstraksi fitur melewati lapisan linier. Selanjutnya keluaran fitur dari kelompok lapisan kasar digabung (concatenate) dengan citra masukan awal untuk menjadi masukan pada kelompok lapisan halus. Penggabungan kedua citra tersebut digunakan untuk proses penghalusan citra, sehingga fitur yang didapat dari proses kelompok lapisan kasar dapat dikoreksi kembali terkait informasi spasialnya dengan citra masukan awal. Proses detil ukuran citra setiap lapisan telah diinformasikan pada [10].

Selanjutnya, arsitektur 2 merupakan pengembangan lanjutan dari arsitektur yang pertama, masih dengan prinsip lapisan konvolusi dan pooling dengan mengganti kelompok lapisan kasar (Scale 1) dengan model VGG [13] yang sudah dilatih dengan dataset Imagenet [14] untuk mengekstraksi fitur-fitur yang terdapat pada gambar. Sedangkan pada kelompok lapisan tengah (Scale 2) digunakan untuk memprediksi pada resolusi sedang dengan gambar yang lebih jelas digabungkan (concatenate) dengan informasi yang dihasilkan oleh Scale 1. Hasil dari Scale 2 digabungkan dengan citra masukan untuk diolah pada kelompok lapisan halus. [11].

Perbedaan yang cukup signifikan dibandingkan dengan kedua arsitektur yang sebelumnya dimiliki oleh arsitektur 3, yang terdiri dari empat bagian utama, yaitu *encoder*, *decoder*, *multiscale feature fusion*, dan *refinement module*. Pada arsitektur 3 selain menggunakan *pre-train* dari ResNet [15] yang berada pada blok *encoder* untuk mengekstraksi fitur-fitur yang terdapat pada citra 2D, model ini juga menerapkan skema *up-projection* [16] pada blok *decoder*. *Up-projection* merupakan pengembangan dari metode *unpooling* yang berguna untuk meningkatkan skala (*upscale*) dari blok terakhir *encoder*. Keluaran-keluaran dari masing-masing lapisan *encoder* digabung pada bagian *multiscale feature fusion* untuk dilakukan penghalusan pada *refinement module* seperti dua arsitektur yang sebelumnya. [12]

3.2 Dataset

Untuk penelitian ini telah digunakan 2 jenis dataset, yaitu *NYU Depth Dataset* dan *Kitti Dataset*. *NYU Depth Dataset* berisi kumpulan perekaman citra pada lingkungan dalam ruang, terdapat berbagai furnitur yang berperan sebagai objek (Gambar 1) [17]. Kemudian *Kitti Dataset* berisi kumpulan perekaman citra lingkungan luar ruang dari sudut pandang kendaraan, terdapat mobil, pembatas jalan, dan lain-lain sebagai objek (seperti diperlihatkan pada Gambar 2) [18].

Gambar 2. Contoh-contoh Citra pada *Kitti Dataset*

3.3 Fungsi rugi-rugi (*loss function*)

Untuk ketiga arsitektur yang disebutkan sebelumnya, maka fungsi rugi-rugi (*loss function*) yang digunakan adalah persamaan yang memberikan nilai selisih dari citra keluaran model (hasil estimasi) D dengan citra *ground-truth* D^*. Nilai selisih ini dilakukan untuk setiap nilai piksel pada citra, yang dapat dinyatakan dalam bentuk persamaan berikut

$$L_{depth}(D, D^*) = \frac{1}{n}\sum_i d_i^2 - \frac{1}{2n^2}\left(\sum_i d_i\right)^2 + \frac{1}{n}\sum_i \left[(\nabla_x d_i)^2 + (\nabla_y d_i)^2\right] \quad (1)$$

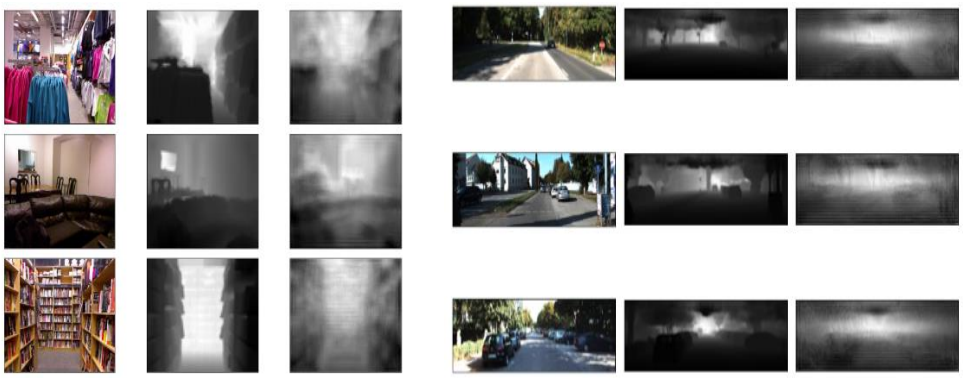
dimana $d = D - D^*$, i adalah indeks piksel, dan n adalah jumlah piksel. Kemudian pada ruas $\nabla_x d_i$ dan $\nabla_y d_i$ dilakukan perhitungan selisih gradien kedua sumbu (x dan y) citra prediksi dan citra ground-truth agar perhitungan rugi-rugi juga mempertimbangkan kesamaan struktur citra [11].

3.4 Pelatihan Model

Pada arsitektur 1 dilakukan modifikasi parameter pelatihan yang diajukan pada [10]. Modifikasi dilakukan karena pada penggunaan parameter awal didapatkan nilai rugi-rugi yang melonjak tinggi dan tidak stabil. Untuk itu digunakan *optimizer* Adam dan *learning rate* 0.0001 untuk kedua kelompok lapisan, dan kemudian ditambahkan *learning rate decay* secara eksponensial dengan nilai *gamma* 0.9. *Optimizer* Adam merupakan metode optimasi stokastik yang mudah diterapkan dengan penggunaan memori yang ringan [19]. Selanjutnya untuk jumlah *epoch* diberikan 25, karena tidak ada perubahan *error* yang signifikan setelah itu. Kemudian untuk arsitektur 2 dan 3 digunakan *optimizer* Adam dan *learning rate* 0.00001. Nilai *learning rate* ditentukan berdasarkan uji coba variasi, karena apabila terlalu kecil komputasi menuju rugi-rugi yang optimal cukup lama dicapainya dan jika terlalu besar maka nilai rugi-rugi menjadi tidak stabil.

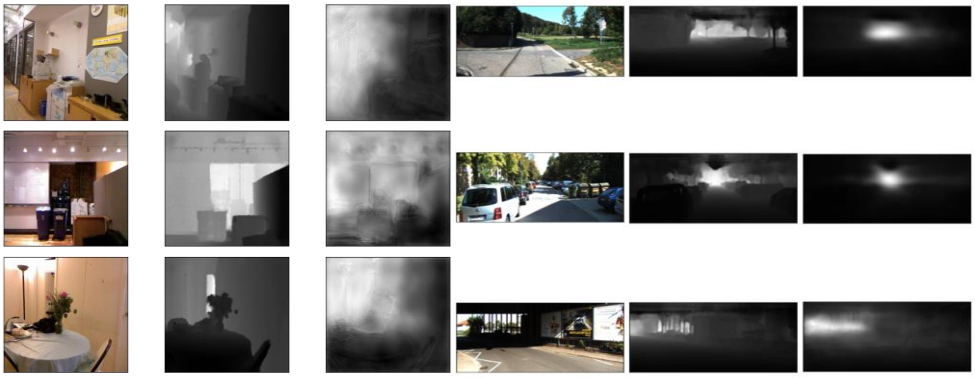
4 Hasil Simulasi dan Analisis

Gambar 3 ,4, dan 5 berturut-turut memperlihatkan hasil pengujian dengan menggunakan arsitektur 1, 2, dan 3 yang telah dilatih. Pada gambar-gambar tersebut maka kolom 1 adalah citra masukan, kolom 2 adalah citra *ground-truth*, dan kolom 3 adalah citra hasil estimasi model.



Gambar 3. (kiri) Keluaran dengan *NYU Depth Dataset*, (kanan) Keluaran dengan *Kitti Dataset*

Simulasi dan pelatihan dilakukan dengan menggunakan GPU *accelerator* NVIDIA Tesla T4 16 GB.



Gambar 4. (kiri) Keluaran dengan *NYU Depth Dataset*, (kanan) Keluaran dengan *Kitti Dataset*

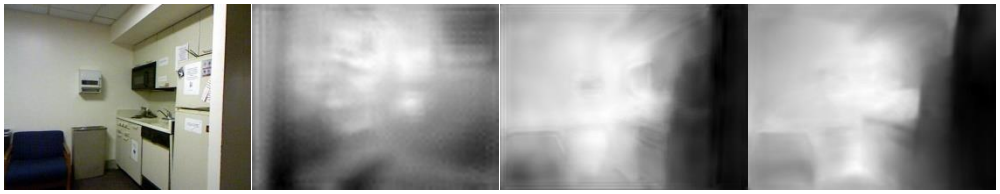
Gambar 5. (kiri) Keluaran dengan *NYU Depth Dataset*, (kanan) Keluaran dengan *Kitti Dataset*

Selanjutnya, setelah dilakukan pelatihan model diuji untuk menerima citra tunggal untuk pengujian durasi komputasi dan nilai rugi-rugi dari *ground-truth* nya. Citra diambil disesuaikan dengan kondisi dalam ruang dan kondisi luar ruang, seperti diperlihatkan pada Gambar 6 dan 7.

Tabel 1. Hasil pelatihan model dengan *NYU Depth Dataset* dan *Kitti Dataset*

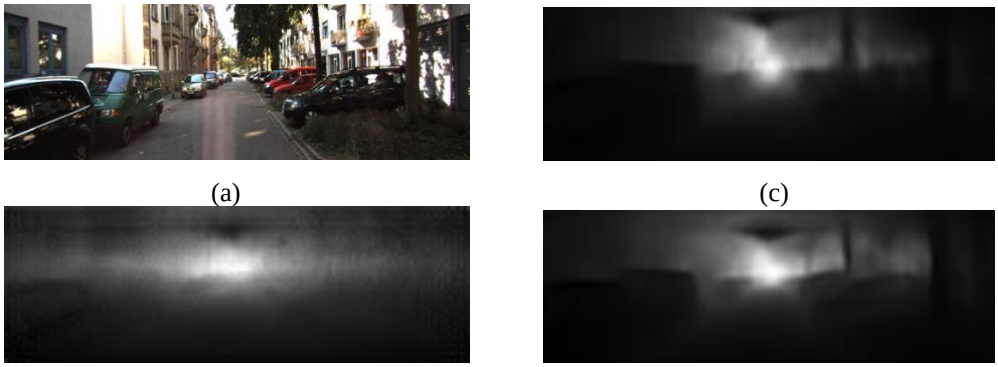
Arsitektur	GPU	*NYU Depth Dataset*		*Kitti Dataset*	
		Durasi Pelatihan	Rugi-rugi per batch	Durasi Pelatihan	Rugi-rugi per batch
1	Tesla T4	1 j 20 menit	2,4707	54 m 10 detik	4,1981
2	Tesla T4	2 j 44 menit	0,8056	1 j 32 menit	3,2540
3	Tesla T4	11 j 1 menit	0,6914	3 j 45 menit	2,3708

Dari hasil simulasi ketiga arsitektur diperoleh informasi terkait perkembangan yang terjadi pada sistem visi komputer. Perkembangan yang dilakukan diantaranya adalah pengajuan arsitektur baru, fungsi rugi-rugi baru, dan diimplementasikannya teknik *pre training* dengan menggunakan *Imagenet* [14].



(a) (b) (c) (d)

Gambar 6. Hasil pengujian penerimaan citra RGB tunggal (dalam ruang) (a) citra masukan, (b) arsitektur 1, (c) arsitektur 2, (d) arsitektur 3



(a) (c)

(b) (d)

Gambar 7. Hasil pengujian penerimaan citra RGB tunggal (luar ruang) (a) citra masukan, (b) arsitektur 1., (c) arsitektur 2., (d) arsitektur 3.

Tabel 2. Hasil Pengujian dengan data citra tunggal

Arsitektur	GPU	Durasi pengolahan citra tunggal		Rugi-rugi	
		NYU Depth Dataset	*Kitti Dataset*	*NYU Depth Dataset*	*Kitti Dataset*
1	Tesla T4	0,06 detik	0,05 detik	0,3117	0,6317
2	Tesla T4	0,1 detik	0,06 detik	0,3420	0,6815
3	Tesla T4	0,28 detik	0,14 detik	0,0714	0,4207

Berdasarkan hasil yang ditunjukkan pada tabel 1, arsitektur 3 menghasilkan citra kedalaman dengan rugi-rugi per *batch* paling kecil. Kemudian pada gambar 6 dan 7, terlihat sangat jelas bahwa citra keluaran model yang paling baik diberikan juga oleh arsitektur 3, kemudian arsitektur 2, dan terakhir arsitektur 1. Hasil pada gambar 6 dan 7 diperkuat dengan hasil kuantitatif perhitungan nilai rugi-rugi dari keluaran model pada tabel 2. Hasil yang baik didapatkan dari kemampuan arsitektur JST untuk mengekstrak fitur citra dan mengolah informasi. Namun untuk meningkatkan kemampuan tersebut stuktur lapisan menjadi semakin kompleks dan hal ini mengakibatkan peningkatan durasi pelatihan dan pengolahan masukan pada pengujian model. Hal tersebut terbukti oleh arsitektur 3, dengan pengajuan lapis up-projection untuk menjaga informasi spasial pada penskalaan citra, sedangkan pada arsitektur 1 dan 2 masih digunakan struktur jaringan syaraf konvolusi sederhana. Lalu apabila ditinjau pada tabel 2, nilai rugi-rugi yang dihasilkan dari pengolahan citra tunggal tidak berselisih jauh untuk ketiga arsitektur, namun apabila ditinjau secara visual pada gambar 6 dan 7 perbedaan hasil antara ketiga arsitektur tersebut dapat terlihat sangat jelas. Disini dapat dipahami bahwa perubahan nilai rugi-rugi sangat mempengaruhi hasil estimasi peta kedalaman. Dapat disimpulkan metode yang digunakan pada arsitektur 3 memberikan hasil estimasi peta kedalaman paling baik, namun dengan konsekuensi durasi komputasi lebih lama.

Dengan permasalahan tersebut, maka dibutuhkan sumber daya komputasi yang cukup besar apabila sistem ini hendak diimplementasikan pada kendaraan otonom. Jika ditinjau pada tabel 2, terlihat waktu yang

dibutuhkan untuk model mengolah citra masukan, namun waktu tersebut didapatkan dengan citra beresolusi kecil. Kemudian perlu dipertimbangkan juga waktu untuk sistem kendaraan otonom mengolah informasi persepsi visual dan melakukan pengambilan keputusan, karena kendaraan otonom harus bertindak responsif untuk keselamatan pengendara dan lingkungan.

Dari segi fungsional didapatkan bahwa metode estimasi peta kedalaman melalui citra RGB 2D mampu memberikan informasi kondisi lingkungan khususnya terkait jarak objek dengan titik perekaman. Metode ini mampu digunakan untuk memperkuat penggunaan kamera RGB sebagai instrumen persepsi di segala bidang baik penggunaan dalam ruang atau luar ruang. Hal tersebut menjadikan metode ini sangat berpotensi untuk diterapkan pada kendaraan otonom, dengan hanya menggunakan kamera 2D tetap didapatkan informasi ruang 3D.

5 Kesimpulan

Analisis estimasi peta kedalaman yang ditunjukkan pada makalah ini berhasil menyimpulkan bagaimana pengujian yang dilakukan pada tiga arsitektur, berdasarkan parameter rugi-rugi per batch dan rugi-rugi sampel citra tunggal diperoleh bahwa arsitektur ketiga memberikan hasil yang paling baik dalam memberikan informasi mengenai persepsi terhadap lingkungan, Hal ini juga menunjukkan bahwa perkembangan yang cukup pesat dari tahun ke tahun mengenai peningkatan penggunaan metoda persepsi yang dikembangkan. Dengan sumber daya komputasi yang memadai metode ini dapat menjadi pilihan untuk memberikan informasi persepsi pada ranah robotika khususnya pada penerapan kendaraan otonom, dimana dengan melalui informasi 2 dimensi dapat diperoleh informasi tambahan aspek 3 dimensi pada citra.

6 Referensi

[1] S. D. Pendleton *et al.*, "Perception, planning, control, and coordination for autonomous vehicles", *Machines*, vol. 5, no. 1, pp. 1–54, 2017, doi: 10.3390/machines5010006.

[2] S. Royo and M. Ballesta-Garcia, "An overview of lidar imaging systems for autonomous vehicles", *Appl. Sci.*, vol. 9, no. 19, 2019, doi: 10.3390/app9194093.

[3] I. Bilik, O. Longman, S. Villeval, and J. Tabrikian, "The Rise of Radar for Autonomous Vehicles: Signal processing solutions and future research direction", *IEEE Signal Process. Mag.*, vol. 36, no. 5, pp. 20–31, 2019, doi: 10.1109/MSP.2019.2926573.

[4] F. Roos, J. Bechter, C. Knill, B. Schweizer, and C. Waldschmidt, "Radar sensors for autonomous driving", *IEEE Microw. Mag.*, vol. 20, no. 9, pp. 58–72, 2019, doi: 10.1109/MMM.2019.2922120.

[5] W. Xu, C. Yan, W. Jia, X. Ji, and J. Liu, "Analyzing and Enhancing the Security of Ultrasonic Sensors for Autonomous Vehicle", *IEEE Internet Things J.*, vol. 5, no. 6, pp. 5015–5029, 2018, doi: 10.1109/JIOT.2018.2867917.

[6] A. Pidurkar, R. Sadakale, and A. K. Prakash, "Monocular Camera based Computer Vision System for Cost Effective Autonomous Vehicle", *2019 10th Int. Conf. Comput. Commun. Netw. Technol. ICCCNT 2019*, pp. 1–5, 2019, doi: 10.1109/ICCCNT45670.2019.8944496.

[7] C. Solomon, and T. Breckon, *Fundamentals of Digital Image Processing: A Practical Approach with Examples in Matlab (1st. ed.)*, Wiley Publishing, 2011.

[8] S. Haykin, *Neural Networks and Learning Machines (3rd. ed.*), Prentice Hall, 2009

[9] Y.Y. Nazaruddin, *Sistem Kontrol Cerdas*, Program Studi Teknik Fisika, ITB, 2020

[10] D. Eigen, C. Puhrsch, and R. Fergus, "Depth map prediction from a single image using a multi-scale deep network", *Adv. Neural Inf. Process. Syst.*, vol. 3, no. January, pp. 2366–2374, 2014.

[11] D. Eigen and R. Fergus, "Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture", *Proc. IEEE Int. Conf. Comput. Vis.*, vol. 2015 Inter, pp. 2650–2658, 2015, doi: 10.1109/ICCV.2015.304.

[12] J. Hu, M. Ozay, Y. Zhang, and T. Okatani, "Revisiting single image depth estimation: Toward higher resolution maps with accurate object boundaries", *Proc. - 2019 IEEE Winter Conf. Appl. Comput. Vision, WACV 2019*, pp. 1043–1051, 2019, doi: 10.1109/WACV.2019.00116.

[13] K. Simonyan and A. Zisserman, "VERY DEEP CONVOLUTIONAL NETWORKS FOR LARGE-SCALE IMAGE RECOGNITION", 2015. Accessed: Dec. 05, 2020. [Online]. Available: http://www.robots.ox.ac.uk/.

[14] Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., & Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In 2009 IEEE conference on computer vision and pattern recognition (pp. 248–255).

[15] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition." Accessed: Dec. 05, 2020. [Online]. Available: http://image-net.org/challenges/LSVRC/2015/.

[16] I. Laina, C. Rupprecht, V. Belagiannis, F. Tombari, and N. Navab, "Deeper Depth Prediction with Fully Convolutional Residual Networks." 3D Vision (3DV), 2016 Fourth International Conference on, 239-248

[17]N. Silberman, D. Hoiem, P. Kohli, and R. Fergus. Indoor Segmentation and Support Inference From Rgbd Images. In ECCV, 2012.
[18]A. Geiger, P. Lenz, C. Stiller, and R. Urtasun. Vision meets robotics: The Kitti Dataset. International Journal of Robotics Research (IJRR), 2013.
[19]Kingma, Diederik P., and Jimmy Ba. “Adam: A Method for Stochastic Optimization”. ArXiv:1412.6980 [Cs], Jan. 2017. arXiv.org, http://arxiv.org/abs/1412.6980.