

**PENERAPAN CONVOLUTIONAL NEURAL NETWORK
UNTUK SEGMENTASI SEMANTIK CITRA PERKOTAAN**

TUGAS AKHIR

Daniel Alexander
1118023



PROGRAM STUDI INFORMATIKA
INSTITUT TEKNOLOGI HARAPAN BANGSA
BANDUNG
TAHUN 2022

PENERAPAN CONVOLUTIONAL NEURAL NETWORK UNTUK SEGMENTASI SEMANTIK CITRA PERKOTAAN

TUGAS AKHIR

**Diajukan sebagai salah satu syarat untuk memperoleh
gelar sarjana dalam bidang Informatika**

**Daniel Alexander
1118023**



**INSTITUT
TEKNOLOGI
HARAPAN
BANGSA**

Veritas vos liberabit

**PROGRAM STUDI INFORMATIKA
INSTITUT TEKNOLOGI HARAPAN BANGSA
BANDUNG
TAHUN 2022**

ABSTRAK

Nama : Daniel Alexander
Program Studi : Informatika
Judul : Penerapan Convolutional Neural Network untuk Segmentasi Semantik Citra Perkotaan

Segmentasi semantik adalah proses mengklasifikasikan setiap *pixel* pada citra ke dalam salah satu *class*. Dalam aplikasinya pada sistem swakemudi, segmentasi semantik citra perkotaan dilakukan untuk mengenali objek dan kondisi jalan guna mengambil keputusan kendali yang tepat. Pada penelitian ini, digunakan metode *convolutional neural network* dengan arsitektur DeepLabV3+ untuk melakukan segmentasi semantik. Objek penelitian adalah *dataset* citra perkotaan Bandung Cityscapes yang dikumpulkan di daerah Kota Bandung, Jawa Barat, Indonesia dan diberi anotasi secara manual. Pengujian arsitektur dilakukan dengan mengkombinasikan nilai *epoch*, *learning rate*, dan jumlah *filter* konvolusi untuk mencari nilai akurasi *mean intersection over union* yang terbaik. Model paling optimal mencapai akurasi rata-rata tertinggi sebesar 77.431% dan rata-rata waktu prediksi 74.890 ms pada dataset validasi, dicapai oleh model dengan 128 *filter* konvolusi, dilatih dengan 300 *epoch* dan *learning rate* 0.0001 (10^{-4}). Model tersebut mempertahankan rata-rata akurasi pelatihan sebesar 87.194%, rata-rata akurasi validasi sebesar 75.274%, dan rata-rata perbedaan antara keduanya sebesar 11.919% selama proses pelatihan.

Kata kunci: *computer vision*, segmentasi semantik, *convolutional neural network*.

ABSTRACT

Name : Daniel Alexander
Department : Informatics
Title : Application of Convolutional Neural Network for Semantic Segmentation of Urban Images

Semantic segmentation is the process of classifying each pixel in an image into one of the predesignated classes. In its application for self-driving systems, semantic segmentation is used to identify objects and road conditions to make the right control decisions. This work uses the convolutional neural network method to perform semantic segmentation with the DeepLabV3+ architecture. Object of study is the urban scene dataset called Bandung Cityscapes collected around the city of Bandung, West Java, Indonesia and annotated by hand. Architecture testing is performed by combining the values of epoch, learning rate, and number of convolution filters to find the best mean intersection over union accuracy. The most optimal model achieved the highest average accuracy of 77.431% and an average prediction time of 74.890 ms on the validation set, attained by the model with 128 convolutional filters, trained with 300 epochs and 0.0001 (10^{-4}) learning rate. This model maintained an average training accuracy of 87.194%, average validation accuracy of 75.274%, and average difference between the two of 11.919% throughout the training process.

Keywords: computer vision, semantic segmentation, convolutional neural network

KATA PENGANTAR

Penulis mengucapkan terima kasih kepada Tuhan Yang Maha Esa, karena berkat karunia-Nya, penulis dapat menyelesaikan Tugas Akhir yang berjudul “Penerapan Convolutional Neural Network untuk Segmentasi Semantik Citra Perkotaan” dengan baik. Penelitian ini disusun sebagai syarat kelulusan Program Studi Informatika di Institut Teknologi Harapan Bangsa. Menerapkan keilmuan Informatika dalam bidang *computer vision* merupakan tujuan utama penulis dalam penelitian ini. Kemajuan teknologi deep learning yang diperoleh dari banyaknya data yang dihasilkan melalui kamera digital, serta terus meningkatnya kemampuan pemrosesan komputer dengan hadirnya *Graphics Processing Unit* (GPU) memungkinkan pemecahan masalah yang sebelumnya tidak mungkin. Penulis sadar bahwa masih banyak kekurangan yang dibuat dalam proses penyusunan Tugas Akhir ini karena keterbatasan pengetahuan dan waktu yang dimiliki. Maka dari itu, penulis akan sangat menghargai komentar dan saran konstruktif agar penyusunan dapat dilakukan dengan lebih baik lagi di kemudian hari. Akhir kata, penulis berharap bahwa Tugas Akhir ini dapat memberikan bantuan dan inspirasi kepada pembaca yang memiliki keinginan memahami penerapan keilmuan Informatika dalam bidang *computer vision* serta mengimplementasikan hal yang serupa.

Bandung, 15 Juni 2022

Hormat penulis,

A handwritten signature in black ink, appearing to read 'Daniel Alexander', with a long horizontal stroke extending to the right.

Daniel Alexander

DAFTAR ISI

ABSTRAK	iv
ABSTRACT	v
KATA PENGANTAR	vi
DAFTAR ISI	vii
DAFTAR TABEL	x
DAFTAR GAMBAR	xii
BAB 1 PENDAHULUAN	1-1
1.1 Latar Belakang	1-1
1.2 Rumusan Masalah	1-3
1.3 Tujuan Penelitian	1-3
1.4 Batasan Masalah	1-3
1.5 Kontribusi Penelitian	1-4
1.6 Metodologi Penelitian	1-4
1.7 Sistematika Pembahasan	1-5
BAB 2 LANDASAN TEORI	2-1
2.1 Tinjauan Pustaka	2-1
2.1.1 Citra Digital	2-1
2.1.2 <i>Color Model</i>	2-1
2.1.3 <i>Computer Vision</i>	2-1
2.1.4 Fitur dan <i>Classifier</i>	2-2
2.1.5 <i>Artificial Neural Network</i>	2-2
2.1.6 <i>Multilayer Perceptron</i>	2-3
2.1.7 <i>Convolutional Neural Network</i>	2-4
2.1.8 <i>Image Data Augmentation</i>	2-19
2.1.9 <i>Intersection over Union</i>	2-21
2.2 Pustaka	2-21
2.2.1 TensorFlow	2-22
2.2.2 NumPy	2-22
2.2.3 OpenCV	2-23

2.2.4	Keras	2-23
2.2.5	Flask	2-27
2.2.6	Flutter	2-27
2.3	State-of-the-Art	2-27
2.4	Tinjauan Objek	2-29
2.4.1	Citra Perkotaan	2-30
2.4.2	Proses Pengambilan Dataset	2-30
2.4.3	Spesifikasi Data	2-31
2.4.4	Proses Anotasi dan Pembagian Dataset	2-34
BAB 3	METODOLOGI PENELITIAN	3-1
3.1	Analisis Masalah	3-1
3.2	Kerangka Pemikiran	3-1
3.3	Analisis Urutan Proses Global	3-3
3.4	Analisis Manual	3-5
3.4.1	<i>Dataset</i>	3-6
3.4.2	Konversi nilai <i>ground truth</i> RGB menjadi <i>integer</i>	3-7
3.4.3	<i>Image Data Augmentation</i>	3-8
3.4.4	<i>Convolutional Neural Network: Forward Propagation</i>	3-10
3.4.5	<i>Convolutional Neural Network: Backward Propagation</i>	3-17
3.4.6	Pengujian Hasil Prediksi	3-19
BAB 4	IMPLEMENTASI DAN PENGUJIAN	4-1
4.1	Lingkungan Implementasi	4-1
4.1.1	Spesifikasi Perangkat Keras	4-1
4.1.2	Spesifikasi Perangkat Lunak	4-1
4.2	Implementasi Perangkat Lunak	4-1
4.2.1	Implementasi Model CNN	4-1
4.2.2	Implementasi API	4-4
4.2.3	Implementasi Aplikasi	4-4
4.2.4	<i>Deployment</i>	4-8
4.3	Pengujian	4-9
4.3.1	Pengujian Nilai <i>Epoch</i>	4-10
4.3.2	Pengujian Nilai <i>Learning Rate</i>	4-10
4.3.3	Pengujian Arsitektur dan Jumlah <i>Filter</i>	4-10
4.4	Hasil Pengujian	4-13
4.4.1	Pengujian pada Arsitektur dengan 64 Filter Konvolusi	4-14
4.4.2	Pengujian pada Arsitektur dengan 128 Filter Konvolusi	4-21

4.4.3	Pengujian pada Arsitektur dengan 256 Filter Konvolusi . . .	4-28
4.4.4	Pembahasan Umum Hasil Pengujian Arsitektur CNN . . .	4-34
4.4.5	Hasil Pengujian Mean IoU	4-35
4.4.6	Hasil Pengujian Waktu Prediksi	4-36
4.4.7	Hasil Pengujian <i>Class IoU</i>	4-37
4.5	Analisis Kesalahan	4-41
BAB 5	KESIMPULAN DAN SARAN	5-1
5.1	Kesimpulan	5-1
5.2	Saran	5-2

DAFTAR TABEL

2.1	<i>Class object</i> yang digunakan dalam penelitian	2-11
2.2	Daftar <i>method</i> yang digunakan dari pustaka TensorFlow	2-22
2.3	Daftar <i>method</i> yang digunakan dari pustaka NumPy	2-22
2.4	Daftar <i>method</i> yang digunakan dari pustaka OpenCV	2-23
2.5	Daftar <i>method</i> yang digunakan dari pustaka Keras	2-24
2.6	Daftar <i>method</i> yang digunakan dari pustaka Flask	2-27
2.7	Daftar <i>method</i> yang digunakan dari pustaka Flutter	2-27
2.8	<i>State of the Art</i>	2-28
2.8	<i>State of the Art</i>	2-29
2.9	Perincian Jumlah <i>Pixel</i> untuk Masing-Masing <i>Class</i>	2-35
2.10	Perincian Proporsi <i>Pixel</i> untuk Masing-Masing <i>Class</i>	2-36
3.1	Contoh matriks citra	3-6
3.2	Contoh matriks <i>ground truth</i>	3-7
3.3	<i>Dictionary labels</i>	3-7
3.4	Contoh matriks <i>ground truth</i> setelah dikonversi ke <i>integer</i>	3-8
3.5	Contoh matriks <i>ground truth</i> setelah <i>one-hot encoding</i>	3-8
3.6	Contoh inisialisasi matriks citra	3-9
3.7	Matriks citra sebelum <i>horizontal flipping</i>	3-9
3.8	Matriks citra setelah <i>horizontal flipping</i>	3-9
3.9	Contoh <i>kernel Gaussian</i>	3-9
3.10	Contoh matriks citra RGB dengan <i>zero-padding</i>	3-10
3.11	Contoh <i>kernel</i> berukuran 3×3	3-11
3.12	Contoh <i>feature map</i> hasil konvolusi	3-11
3.13	Contoh <i>feature map</i> hasil <i>batch normalization</i>	3-12
3.14	Contoh <i>feature map</i> hasil fungsi aktivasi ReLU	3-13
3.15	Contoh <i>feature map</i> hasil operasi <i>convolution layer</i> pertama	3-14
3.16	Contoh <i>feature map</i> hasil operasi <i>convolution layer</i> ketiga	3-14
3.17	Contoh <i>feature map</i> hasil <i>skip connection</i>	3-14
3.18	Contoh <i>feature map</i> hasil <i>average pooling layer</i> level 1	3-15
3.19	Contoh <i>feature map</i> hasil <i>average pooling layer</i> level 2	3-15
3.20	Contoh <i>feature map</i> hasil <i>average pooling layer</i> level 3	3-15
3.21	Contoh <i>feature map</i> hasil <i>average pooling layer</i> level 6	3-15
3.22	Contoh <i>feature map</i> level 1 hasil <i>upsampling</i>	3-16
3.23	Contoh matriks citra RGB hasil prediksi	3-20

3.24	Contoh matriks citra <i>ground truth</i>	3-20
4.1	Daftar <i>method</i> pada <i>section DataLoader</i>	4-2
4.2	Daftar <i>method</i> pada <i>section CNN</i>	4-3
4.3	Daftar <i>endpoint</i> pada <i>API</i>	4-4
4.4	Daftar <i>method</i> pada <i>class WelcomeScreen</i>	4-7
4.5	Daftar <i>method</i> pada <i>class ArchitectureSelectionScreen</i>	4-8
4.6	Daftar <i>method</i> pada <i>class ResultScreen</i>	4-8
4.7	Daftar <i>elemen</i> pada <i>app.yaml</i>	4-9
4.8	Daftar <i>elemen</i> konfigurasi pada <i>platform Vercel</i>	4-9
4.9	Hasil Pengujian (nilai rata-rata) pada Arsitektur dengan 64 Filter Konvolusi	4-18
4.10	Hasil Pengujian (nilai maksimum) pada Arsitektur dengan 64 Filter Konvolusi	4-18
4.10	Hasil Pengujian (nilai maksimum) pada Arsitektur dengan 64 Filter Konvolusi	4-19
4.11	Hasil Pengujian (nilai rata-rata) pada Arsitektur dengan 128 Filter Konvolusi	4-25
4.12	Hasil Pengujian (nilai maksimum) pada Arsitektur dengan 128 Filter Konvolusi	4-25
4.13	Hasil Pengujian (nilai rata-rata) pada Arsitektur dengan 256 Filter Konvolusi	4-31
4.14	Hasil Pengujian (nilai maksimum) pada Arsitektur dengan 256 Filter Konvolusi	4-31
4.15	Hasil pengujian rata-rata <i>mean IoU</i> prediksi pada data validasi . . .	4-35
4.16	Hasil pengujian rata-rata waktu prediksi pada data validasi	4-36
4.17	Hasil pengujian rata-rata <i>class IoU</i> arsitektur dengan 64 <i>filter</i> konvolusi pada data validasi	4-37
4.18	Hasil pengujian rata-rata <i>class IoU</i> arsitektur dengan 128 <i>filter</i> konvolusi pada data validasi	4-37
4.19	Hasil pengujian rata-rata <i>class IoU</i> arsitektur dengan 256 <i>filter</i> konvolusi pada data validasi	4-38

DAFTAR GAMBAR

2.1	<i>Neuron</i> [6]	2-3
2.2	<i>Multilayer Perceptron</i> [6]	2-3
2.3	<i>Dua jenis signal pada neuron</i> [6]	2-4
2.4	<i>Convolutional Neural Network</i> [1]	2-5
2.5	<i>Komponen-komponen utama pada CNN</i> [19]	2-5
2.6	Operasi konvolusi. [1]	2-6
2.7	Operasi konvolusi dengan <i>stride</i> = 2 dan <i>zero-padding</i> = 1. [1] . . .	2-6
2.8	Operasi <i>max-pooling layer</i> dengan daerah <i>pooling</i> 2×2 dan <i>stride</i> 1. [1]	2-8
2.9	<i>Spatial Pyramid Pooling Layer</i> dapat menangkap informasi dalam tiga skala yang berguna dalam melakukan klasifikasi yang akurat. .	2-9
2.10	<i>Skip Connection</i> pada arsitektur ResNet [1]	2-12
2.11	<i>Loss surface</i> pada arsitektur ResNet tanpa dan dengan <i>skip</i> <i>connection</i> [13]	2-13
2.12	Ilustrasi dan cara mengatasi masalah <i>generalization</i> [22]	2-14
2.13	Grafik perbandingan antara probabilitas dan nilai <i>loss</i> dengan <i>scaling factor</i> γ	2-17
2.14	<i>Image Data Augmentation</i> [1]	2-20
2.15	Proses Akuisisi Citra [2]	2-30
2.16	Rute pengambilan dataset	2-31
2.17	Fitur <i>scene filter</i> pada kakas VLC	2-31
2.18	Skenario bebas lancar dua arah	2-32
2.19	Skenario bebas lancar satu arah	2-32
2.20	Skenario padat pejalan kaki	2-32
2.21	Skenario padat kendaraan bermotor	2-33
2.22	Skenario persimpangan	2-33
2.23	Skenario tikungan kiri	2-33
2.24	Skenario tikungan kanan	2-34
2.25	Fitur <i>pen tool</i> pada kakas <i>Adobe Photoshop</i>	2-34
2.26	Proporsi jumlah <i>pixel</i> per <i>class</i> pada data pelatihan	2-37
2.27	Proporsi jumlah <i>pixel</i> per <i>class</i> pada data validasi	2-37
2.28	Proporsi jumlah <i>pixel</i> per <i>class</i> pada gabungan data pelatihan dan validasi	2-38
2.29	Jumlah <i>pixel</i> per <i>class</i> data pelatihan	2-38

2.30	Jumlah <i>pixel</i> per <i>class</i> pada data validasi	2-39
2.31	Jumlah <i>pixel</i> per <i>class</i> pada gabungan data pelatihan dan validasi . .	2-39
3.1	Kerangka Pemikiran	3-3
3.2	<i>Flowchart</i> urutan Proses Global	3-4
3.3	<i>Flowchart</i> sistem <i>training</i>	3-4
3.4	<i>Flowchart</i> sistem <i>testing</i>	3-5
3.5	Citra RGB	3-6
3.6	Citra Ground Truth	3-6
4.1	Halaman <i>Welcome Screen</i>	4-6
4.2	Halaman <i>Architecture Selection</i>	4-6
4.3	Halaman <i>Result Screen</i>	4-6
4.4	Halaman <i>Result Screen</i> setelah melakukan prediksi	4-6
4.5	Rancangan Arsitektur dengan 64 <i>Filter</i>	4-11
4.6	Rancangan Arsitektur dengan 128 <i>Filter</i>	4-12
4.7	Rancangan Arsitektur dengan 256 <i>Filter</i>	4-13
4.8	Grafik perubahan akurasi pada pengujian arsitektur dengan 64 filter konvolusi untuk <i>epoch</i> = 100	4-14
4.9	Grafik perubahan akurasi validasi pada pengujian arsitektur dengan 64 filter konvolusi untuk <i>epoch</i> = 100	4-15
4.10	Grafik perubahan akurasi pada pengujian arsitektur dengan 64 filter konvolusi untuk <i>epoch</i> = 200	4-16
4.11	Grafik perubahan akurasi validasi pada pengujian arsitektur dengan 64 filter konvolusi untuk <i>epoch</i> = 200	4-16
4.12	Grafik perubahan akurasi pada pengujian arsitektur dengan 64 filter konvolusi untuk <i>epoch</i> = 300	4-17
4.13	Grafik perubahan akurasi validasi pada pengujian arsitektur dengan 64 filter konvolusi untuk <i>epoch</i> = 300	4-17
4.14	Grafik akurasi, akurasi validasi, dan selisihnya pada pengujian di arsitektur dengan 64 filter konvolusi dan <i>learning rate</i> = 0.001 (10^{-3})	4-19
4.15	Grafik akurasi, akurasi validasi, dan selisihnya pada pengujian di arsitektur dengan 64 filter konvolusi dan <i>learning rate</i> = 0.0001 (10^{-4})	4-20
4.16	Grafik akurasi, akurasi validasi, dan selisihnya pada pengujian di arsitektur dengan 64 filter konvolusi dan <i>learning rate</i> = 0.00001 (10^{-5})	4-20
4.17	Grafik perubahan akurasi pada pengujian arsitektur dengan 128 filter konvolusi untuk <i>epoch</i> = 100	4-22

4.18	Grafik perubahan akurasi validasi pada pengujian arsitektur dengan 128 filter konvolusi untuk $epoch = 100$	4-22
4.19	Grafik perubahan akurasi pada pengujian arsitektur dengan 128 filter konvolusi untuk $epoch = 200$	4-23
4.20	Grafik perubahan akurasi validasi pada pengujian arsitektur dengan 128 filter konvolusi untuk $epoch = 200$	4-23
4.21	Grafik perubahan akurasi pada pengujian arsitektur dengan 128 filter konvolusi untuk $epoch = 300$	4-24
4.22	Grafik perubahan akurasi validasi pada pengujian arsitektur dengan 128 filter konvolusi untuk $epoch = 300$	4-24
4.23	Grafik akurasi, akurasi validasi, dan selisihnya pada pengujian di arsitektur dengan 128 filter konvolusi dan $learning rate = 0.001$ (10^{-3})	4-26
4.24	Grafik akurasi, akurasi validasi, dan selisihnya pada pengujian di arsitektur dengan 128 filter konvolusi dan $learning rate = 0.0001$ (10^{-4})	4-26
4.25	Grafik akurasi, akurasi validasi, dan selisihnya pada pengujian di arsitektur dengan 128 filter konvolusi dan $learning rate = 0.00001$ (10^{-5})	4-27
4.26	Grafik perubahan akurasi pada pengujian arsitektur dengan 256 filter konvolusi untuk $epoch = 100$	4-28
4.27	Grafik perubahan akurasi validasi pada pengujian arsitektur dengan 256 filter konvolusi untuk $epoch = 100$	4-28
4.28	Grafik perubahan akurasi pada pengujian arsitektur dengan 256 filter konvolusi untuk $epoch = 200$	4-29
4.29	Grafik perubahan akurasi validasi pada pengujian arsitektur dengan 256 filter konvolusi untuk $epoch = 200$	4-29
4.30	Grafik perubahan akurasi pada pengujian arsitektur dengan 256 filter konvolusi untuk $epoch = 300$	4-30
4.31	Grafik perubahan akurasi validasi pada pengujian arsitektur dengan 256 filter konvolusi untuk $epoch = 300$	4-30
4.32	Grafik akurasi, akurasi validasi, dan selisihnya pada pengujian di arsitektur dengan 256 filter konvolusi dan $learning rate = 0.001$ (10^{-3})	4-32
4.33	Grafik akurasi, akurasi validasi, dan selisihnya pada pengujian di arsitektur dengan 256 filter konvolusi dan $learning rate = 0.0001$ (10^{-4})	4-32

4.34	Grafik akurasi, akurasi validasi, dan selisihnya pada pengujian di arsitektur dengan 256 filter konvolusi dan <i>learning rate</i> = 0.00001 (10^{-5})	4-33
4.35	Grafik hasil pengujian <i>mean IoU</i> pada data validasi	4-35
4.36	Grafik hasil pengujian waktu prediksi pada data validasi.	4-36
4.37	Grafik hasil pengujian <i>class IoU</i> arsitektur 64 <i>filter</i> konvolusi pada data validasi	4-39
4.38	Grafik hasil pengujian <i>class IoU</i> arsitektur 128 <i>filter</i> konvolusi pada data validasi	4-39
4.39	Grafik hasil pengujian <i>class IoU</i> arsitektur 256 <i>filter</i> konvolusi pada data validasi	4-40
4.40	Perbandingan proporsi jumlah <i>pixel</i> per <i>class</i> dengan akurasi prediksi pada data validasi	4-40
4.41	Pengujian citra dengan hasil prediksi terbaik oleh model dengan 64 <i>filter</i> konvolusi	4-42
4.42	Pengujian citra dengan hasil prediksi terbaik oleh model dengan 128 <i>filter</i> konvolusi	4-42
4.43	Pengujian citra dengan hasil prediksi terbaik oleh model dengan 256 <i>filter</i> konvolusi	4-43
4.44	Kesalahan prediksi <i>class Pedestrian</i> yang diprediksi sebagai <i>MotorcycleScooter</i>	4-44
4.45	Kesalahan prediksi pejalan kaki yang terobstruksi	4-44
4.46	Kesalahan prediksi rambu lalu lintas	4-45

DAFTAR REFERENSI

- [1] S. Khan, et al., *A Guide to Convolutional Neural Networks for Computer Vision*, 1st ed, Gérard Medioni and Sven Dickinson, Ed. California: Morgan and Claypool, 2018.
- [2] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, 2nded. Prentice Hall, 1992.
- [3] H. Zhao, et al. “Pyramid Scene Parsing Network.” 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, <https://doi.org/10.1109/cvpr.2017.660>.
- [4] R. Szeliski, *Computer Vision Algorithms and Applications*, Springer, 2011.
- [5] J. Heaton, *Artificial Intelligence for Humans: Deep Learning and Neural Networks*, Volume 3, 1st ed. CreateSpace Independent Publishing Platform, 2015., pp. 46-47, 186-191.
- [6] S. Haykin, *Neural Networks and Learning Machines*, 3rd ed. New Jersey: Pearson Education Inc., 2009, pp.10-15, 123-125, 131-141.
- [7] J. Long, et al. “Fully Convolutional Networks for Semantic Segmentation.” 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, <https://doi.org/10.1109/cvpr.2015.7298965>.
- [8] O. Ronneberger, et al. “U-Net: Convolutional Networks for Biomedical Image Segmentation.” *Lecture Notes in Computer Science*, 2015, pp. 234–241., https://doi.org/10.1007/978-3-319-24574-4_28.
- [9] V. Badrinarayanan, et al. “SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation.” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 12, 2017, pp. 2481–2495., <https://doi.org/10.1109/tpami.2016.2644615>.
- [10] L.C. Chen, et al. “Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation.” *Computer Vision ECCV 2018*, 2018, pp. 833–851., https://doi.org/10.1007/978-3-030-01234-2_49.

- [11] P. Sturges, et al. "Combining Appearance and Structure from Motion Features for Road Scene Understanding." *Proceedings of the British Machine Vision Conference 2009*, 2009, <https://doi.org/10.5244/c.23.62>.
- [12] T. Li, et al. "Semantic Segmentation of Urban Street Scene Based on Convolutional Neural Network." *Journal of Physics: Conference Series*, vol. 1682, no. 1, 2020, p. 012077., <https://doi.org/10.1088/1742-6596/1682/1/012077>.
- [13] H. Li, et al. "Visualizing the loss landscape of neural nets." *Advances in neural information processing systems* 31 (2018).
- [14] K. He, et al. "Deep Residual Learning for Image Recognition." *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, <https://doi.org/10.1109/cvpr.2016.90>.
- [15] D. P. Kingma, and J. Ba, "Adam: A Method for Stochastic Optimization.", *3rd International Conference for Learning Representations, San Diego, 2015*.
- [16] J. Jordan, "Setting the learning rate of your neural network," Jeremy Jordan, March 1st 2018. [Online]. Available: <https://www.jeremyjordan.me/nn-learning-rate/>.
- [17] X. Glorot and Y. Bengio. "Understanding the Difficulty of Training Deep Feedforward Neural Networks.", 2010, <http://proceedings.mlr.press/v9/glorot10a/glorot10a.pdf>.
- [18] K. He, et al. "Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification." In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1026–1034, 2015a.
- [19] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning, An MIT Press book*, 2016. [Online]. Available: <http://www.deeplearningbook.org>.
- [20] T. Lin , et al. "Focal Loss for Dense Object Detection." *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017, <https://doi.org/10.1109/iccv.2017.324>.
- [21] K. He, et al. "Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 9, 2015, pp. 1904–1916., <https://doi.org/10.1109/tpami.2015.2389824>.

DAFTAR REFERENSI

- [22] A. Shrivastava, “Underfitting Vs Just Right Vs Overfitting in Machine learning: Data Science and Machine Learning,” Kaggle. [Online]. Available: <https://www.kaggle.com/getting-started/166897>.
- [23] M. Nixon and A. S. Aguado. Feature Extraction and Image Processing. Academic Press, 2012.