JCONES Vol. 9, No. 2 (2020) 72-78

Journal of Control and Network Systems

Situs Jurnal : http://jurnal.stikom.edu/index.php/jcone

APLIKASI PENGHITUNG KENDARAAN YANG MELINTAS DI JALAN RAYA BERDASARKAN METODE YOLO OBJECT DETECTION

Rezza Ifmanur Isnaini [1] **Susijanto Tri Rasmana**[2] **Pauladie Susanto**[3]
Program Studi/Jurusan Teknik Komputer
Universitas Dinamika
Jl. Raya Kedung Baruk 98 Surabaya, 60298
Email: 1)15410200068@dinamika.ac.di, 2)susyanto@dinamika.ac.id, 3)pauladie@dinamika.ac.id

Abstrak: Berbagai macam dampak yang dihasilkan oleh kemacetan dan bersifat negatif, setelah dilihat dari berbagai aspek. Kemacetan menimbulkan banyak kerugian dari berbagai segi materi, waktu, dan tenaga. Seperti dari aspek ekonomi kemacetan menghambat proses produksi dan distribusi, sehingga menghambat laju perekonomian yang harusnya berjalan. Dari aspek kesehatan kemacetan menyebabkan dampak negatif yaitu mempengaruhi kondisi fisik dan psikis para pengguna jalan raya. Sudah banyak upaya yang dilakukan untuk mengurangi kepadatan kendaraan di jalan raya. Dengan cara membuat sebuah sistem penghitung jumlah kendaraan dengan menggunakan cctv yang ada pada salah satu lampu lalu lintas di Surabaya dilakukan secara otomatis dan diproses pada pengolahan citra digital menggunakan metode yolo *object detection*. Hasil pengujian yang didapatkan aplikasi yolo *object detection* yang sudah bisa mendeteksi dan membedakan kendaraan dengan unsur lain. Dengan pendeteksian kali ini lebih mudah untuk dapat menghitung kendaraan yang lewat pada jalan raya di video. Setelah itu yolo object detection dapat menghitung kendaraan ditandai dengan adanya tampilan di layer pada bagian pojok kanan. Kemudian dengan pengujian menggunakan 3 video yang diambil dari hasil rekam dan dari internet yang sudah bisa mendeteksi dan menghitung dengan akurat dengan akurat yang mempunyai nilai keakuratan paling tinggi 45%, 31%, dan 10%.

Kata kunci: Kemacetan, Kendaraan, Kerugian yolo object detection

PENDAHULUAN

Kendaraan merupakan alat yang digunakan hampir semua manusia untuk bermobilitas atau berpindah dari tempat yang jauh ataupun dekat. Terdapat berbagai jenis kendaraan (sepeda motor) dan beroda empat (móbil, truk, dan bus). Kemajuan teknologi transportasi berdamak pada perkembangan lalu lintas dan angkutan jalan, sehingga terjadi perubahan pada prasarana jalan, sarana angkutan, dan pangkat lalu lintas lain–lain. Faktor yang lainnya adalah pertumbuhan ekonomi yang menyebabkan pengguna jalan semakin meningkat dan semakin padat di jalan raya. Indonesia merupakan salah satu negara dengan tingkat pembelian kendaraan bermotor yang tinggi.

Berbagai macam dampak yang dihasilkan oleh kemacetan dan bersifat negatif. Setelah dilihat dari berbagai aspek, kemacetan menimbulkan banyak kerugian dari bebagai segi yaitu materi, waktu, dan tenaga. Seperti dari aspek ekonomi kemacetan menghambat proses produksi dan distribusi, sehingga menghambat laju perekonomian yang harusnya berjalan. Dari aspek kesehatan, kemacetan menyebabkan dampak negatif yaitu mempengaruhi kondisi fisik dan psikis para pengguna jalan raya, terlebih bagi mereka yang sering beraktifitas di jalan saat berangkat dan pulang dari bekerja, dan lain sebagainya. Terjadinya kemacetan adalah dampak dari ketidakseimbangan jaringan lalu lintas karena adanya penumpukan kendaraan yang mengakibatkan kepadatan lalu lintas pada suatu jalan menjadi tinggi, sehingga arus lalu lintas menjadi tersendat bahkan berhenti.

Sudah banyak upaya yang dilakukan untuk mengurangi kepadatan kendaraan di jalan raya. Salah satunya pada penelitian ini dibuat sebuah sistema penghitung jumlah kendaraan dengan menggunakan cctv yang ada pada salah satu lampu

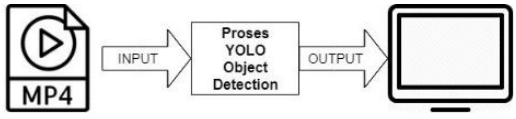
lalu lintas di Surabaya dilakukan secara otomatis dan diproses pada pengolahan citra digital menggunakan métode *yolo object* detection. Algoritma yolo ini merupakan *real object detection*, dengan kemampuan pada base model yang bisa memproses 45 *frame per second* secara *real-time* dan pada versi yang lebih kecil bisa memproses 155 *frame per second* secara *real-time.*

Sudah banyak upaya yang dilakukan pada penelitian lain yang mengarah pada permasalahan yang hampir sama. Salah satunya adalah sebagai berikut, (Karlina dan Indarti, 2019) dengan judul pengenalan objek makanan cepat saji pada video dan real time webcam menggunakan metode you only look once (YOLO). Didalam penelitian ini membahas tentang keakuratan metode yolo yang digunakan untuk mendeteksi makanan cepat saji menggunakan video maupun secara real time menggunakan webcam.

Selain itu penelitian lainnya yang menggunakan kendaraan sebagai objek pengolahan citra adalah kendali lampu lalu lintas dengan deteksi kendaraan menggunakan metode blob detection oleh (Qory Hidayati, 2017). Didalam penelitian ini membahas tentang metode selain YOLO yang bisa digunakan untuk mendeteksi kendaraan menggunakan video ataupun secara real time.

METODE PENELITIAN

Metode penelitian ini adalah implementasi algoritma yolo object detection pada kendaraan yang digunakan untuk menghitung kendaraan yang ada pada suatu ruas jalan. Pendeteksi kendaraan tersebut dilakukan oleh rekaman cctv pada lampu lalu lintas yang datanya diambil dan data berupa citra gambar di proses oleh matlab. Blok diagram aplikasi penghitung kendaraan seperti gambar 3.1



Gambar 1. Blok diagram aplikasi penghitung kendaraan

Gambar blok diagram diatas, memperoleh dari visual yang sudah diambil dari cctv diproses oleh *image processing* pada matlab menggunakan metode yolo *object detection* untuk menghitung berapa banyak kendaraan yang berada pada salah satu ruas jalan.

Proram yolo object detection hanya menghitung kendaraan jika kendaraan melewati garis berwarna kuning yang berfungsi menghitung kendaraan yang lewat seperti pada gambar 2.

Gambar 2. Garis pendeteksi kendaraan yang lewat

Perancangan Program

Penghitung kendaraan dirancang pada berbagai kondisi lingkungan dengan cahaya dan status lalu lintas berubah. Dalam sistem yang digunakan, sistem menerima lalu lintas dari video kemudian memberikan kotak perhitungan pada jalan raya untuk deteksi benda bergerak melewatinya. Sistem dijelaskan sebagai berikut.

YOLO Object Detection

Yolo (you only look once) meruapakan algoritma pendeteksian objek dilakukan dengan membingkai objek yang dideteksi sebagai masalah regresi dan memisahkan spesial pada *bounding boxes* dan kelas probabilitas. Dengan menggunakan jaringan neural tunggal untuk menprediksi bounding boxes dan kelas probabiltas dari seluruh gambar pada satu kali evaluasi. Karena metode ini menggunakan jaringan neural tunggal untuk semua pendeteksian, maka performa deteksi ini bisa dioptimasi dari *end–to–end.* Yolo memiliki kemampuan pada base model yang bisa memproses 45 frame per second secara real time pada versi yang lebih kecil bisa memproses 155 frame per second secara real time.

Sistem *Pre-processing*

Tahapan saat proses mengklasifikasi objek gambar secara manual menggunakan software. User dapat mengklasifikasi semua objek dan menyimpan hasil gambar beserta .txt ke folder data.

Learning

Tahap saat user mengandakan darknet setelah menginstall repo. Kemudian *training* kembali menggunakan model gambar yang sudah ada. Setelah proses *training* selesai, maka

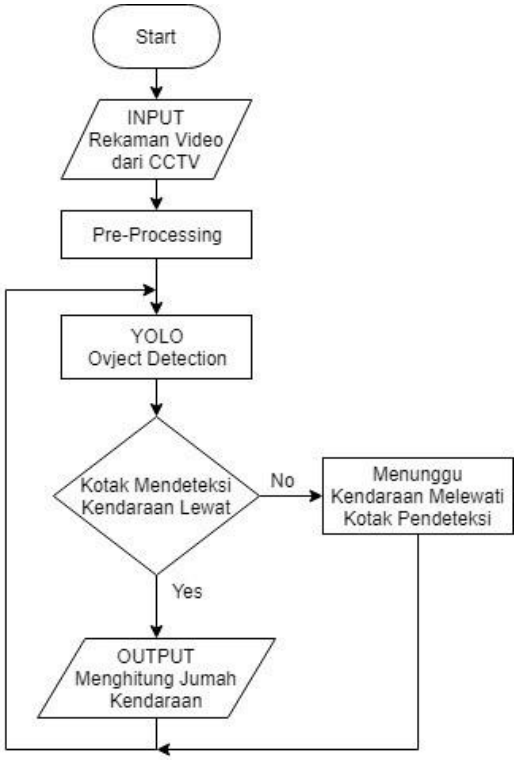
mendapatkan file model yang digunakan untuk mendeteksi objek.

Pendeteksi

Jika aplikasi mendeteksi adanya kendaraan yang sedang lewat pada kotak, maka program menambahkan *counter* +1 yang berfungsi untuk menhitung kendaraan yang lewat pada satu durasi video.

Flowchart Sistem Program Software

Untuk mempermudah pembuatan program *software* diperlukan beberapa tahapan yaitu dengan membuat *flowchart* sistem kerja program terlihat sebagai berikut.



Gambar 3. Flowchart Program Yolo Object Detection

Berikut adalah penjelasan dari gambar 3:

1. Input rekaman video adalah ketika memasukan rekaman video yang dibutuhkan program untuk mendeteksi kendaraan yang lewat.
2. Yolo object detection adalah ketika program sudah memulai bekerja mendeteksi kendaraan dan membuat garis menghitung kendaraan melewatinya.
3. Kotak mendeteksi kendaraan adalah ketika program sudah berhasil mendeteksi kendaraan yang melintas.
4. Menghitung jumlah kendaraan adalah ketika kendaraan yang sudah terdeteksi melewati garis dan counter otomatis bertambah.
5. Menunggu kendaraan melewati garis adalah ketika garis masih menunggu kendaraan yang melewatinya.

HASIL DAN PEMBAHASAN

Pada bab ini membahas tentang banyaknya hasil dari pengujian dan hasil penelitian ini. Dengan tujuan untuk megetahui tingkat keberhasilan dari perancangan sistem yang telah diajukan dan dikerjakan, pengujian dilakukan meliputi uji aplikasi menggunakan métode yolo, uji deteksi kendaraan beroda 4 dan lebih, dan penghitungan kendaraan yang terdeteksi melintas.

Uji Aplikasi Menggunakan Metode Yolo

Pengambilan data dengan cara mengamati berapa banyak kendaraan yang terdeteksi dengan gambar.

Tabel 1. Pengujian deteksi kendaraan pada gambar

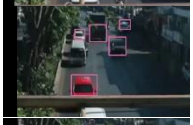
No.	Gambar	Jumlah kendaraan		Persentase
		Sebenarnya	Deteksi	
1		14	12	86%
2		6	5	83%
3		9	8	89%
4		7	7	100%
5		13	11	85%
Rata – rata				89%

Berdasarkan hasil pada tabel 1, dapat disimpulkan bahwa hasil dari deteksi kendaraan menggunakan aplikasi pada gambar mempunyai keakuratan rata–rata 89%.

Uji Deteksi Kendaraan Beroda 4 dan Lebih

Pengambilan data dengan cara mengamati berapa banyak kendaraan dengan roda 4 atau lebih yang terdeteksi dengan gambar.

Tabel 2. Pengujian deteksi kendaraan beroda 4 dan lebih pada gambar

No.	Gambar	Jumlah kendaraan		Persentase
		Sebenarnya	Deteksi	
1		7	7	100%
2		6	5	83%
3		7	6	86%
4		5	5	100%
5		8	7	88%
Rata – rata				91%

Berdasarkan hasil pada tabel 2, dapat disimpulkan bahwa hasil dari deteksi kendaraan beroda 4 dan lebih pada gambar menggunakan aplikasi mempunyai nilai keakuratan rata–rata 91%.

Pengujian Program Perhitungan Kendaraan

Pengambilan data dengan cara mengamati berapa banyak mobil yang lewat 10 detik sampai ke 2 dan melihat kecocokan antara *counter* aplikasi dengan manual. Proses pengambilan data dilakukan menggunakan 2 sampel video yang berbeda dari berbagai sumber yang ada pada internet serta rekaman penulis. Adapun data dari hasil analisis adalah sebagai berikut:

Tabel 3. Hasil pengujian dengan ketelitian aplikasi 60% pada video 1

Nomor	Detik	Jumlah Kendaraan		
		Terdeteksi	Manual	Deteksi Error
1	ke-10	23	11	12
2	ke-20	38	20	18
3	ke-30	44	27	17
4	ke-40	59	35	24
5	ke-50	71	40	31
6	ke-60	83	45	38
7	ke-70	96	54	42
8	ke-80	108	61	47
9	ke-90	117	68	49
10	ke-100	130	77	53
11	ke-110	146	83	63
12	ke-120	160	91	69

Pada tabel 3 pengambilan data dengan menguji keakuratan penghitungan deteksi aplikasi dan penghitungan manual dapat disimpulkan bahwa masih terdapat banyak deteksi error. Karena data keberhasilan sudah ditemukan, maka dapat melakukan proses perhitungan keakuratan deteksi aplikasi dengan cara.

Keakuratan Data = 100% - ($\frac{69}{91}$ x 100%)
= 100% - 76%
= 24%

Tabel 4. Hasil pengujian dengan ketelitian aplikasi 70% pada video 1

Nomor	Detik	Jumlah Kendaraan		
		Terdeteksi	Terdeteksi	Deteksi Error
1	ke-10	15	11	4
2	ke-20	28	20	8
3	ke-30	35	27	8
4	ke-40	47	35	12
5	ke-50	60	40	20
6	ke-60	68	45	23
7	ke-70	82	54	28
8	ke-80	95	61	34
9	ke-90	104	68	36
10	ke-100	116	77	39
11	ke-110	129	83	46
12	ke-120	141	91	50

Pada tabel 4 pengambilan data dengan menguji keakuratan penghitungan deteksi aplikasi dan penghitungan manual dapat disimpulkan bahwa masih terdapat banyak deteksi error. Karena data keberhasilan sudah ditemukan, maka dapat melakukan proses perhitungan keakuratan deteksi aplikasi dengan cara.

Keakuratan Data = 100% - ($\frac{50}{91}$ x 100%)
= 100% - 55%
= 45%

Tabel 5. Hasil pengujian dengan ketelitian aplikasi 80% pada video 1

Nomor	Detik	Jumlah Kendaraan		
		Terdeteksi	Manual	Deteksi Error
1	ke-10	15	11	4
2	ke-20	25	20	5
3	ke-30	34	27	7
4	ke-40	47	35	12
5	ke-50	56	40	16
6	ke-60	66	45	21
7	ke-70	77	54	23
8	ke-80	86	61	25
9	ke-90	94	68	26
10	ke-100	110	77	33
11	ke-110	121	83	38
12	ke-120	131	91	40

Pada tabel 5 pengambilan data dengan menguji keakuratan penghitungan deteksi aplikasi dan penghitungan manual dapat disimpulkan bahwa masih terdapat banyak deteksi error. Karena data keberhasilan sudah ditemukan, maka dapat melakukan proses perhitungan keakuratan deteksi aplikasi dengan cara.

Keakuratan Data = 100% - ($\frac{40}{91}$ x 100%)
= 100% - 44%
= 56%

Tabel 6. Hasil pengujian dengan ketelitian aplikasi 60% pada video 2

Nomor	Detik	Jumlah Kendaraan		
		Terdeteksi	Manual	Deteksi Error
1	ke-10	26	11	15
2	ke-20	42	23	19
3	ke-30	62	32	30
4	ke-40	82	42	40
5	ke-50	102	53	49
6	ke-60	115	60	55
7	ke-70	124	66	58
8	ke-80	143	77	66
9	ke-90	163	89	74
10	ke-100	184	100	84
11	ke-110	196	106	90
12	ke-120	211	116	95

Pada tabel 6 pengambilan data dengan menguji keakuratan penghitungan deteksi aplikasi dan penghitungan manual dapat disimpulkan bahwa masih terdapat banyak deteksi error. Karena data keberhasilan sudah ditemukan, maka dapat melakukan proses perhitungan keakuratan deteksi aplikasi dengan cara.

Keakuratan Data = 100% - ($\frac{95}{116}$ x 100%)
= 100% - 82%
= 18%

Tabel 7. Hasil pengujian dengan ketelitian aplikasi 70% pada video 2

Nomor	Detik	Jumlah Kendaraan		
		Terdeteksi	Manual	Deteksi Error
1	ke-10	21	11	10
2	ke-20	37	23	14
3	ke-30	57	32	25
4	ke-40	77	42	35
5	ke-50	96	53	43
6	ke-60	107	60	47
7	ke-70	116	66	50
8	ke-80	134	77	57
9	ke-90	151	89	62
10	ke-100	172	100	72
11	ke-110	183	106	77
12	ke-120	196	116	80

Pada tabel 7 pengambilan data dengan menguji keakuratan penghitungan deteksi aplikasi dan penghitungan manual dapat disimpulkan bahwa masih terdapat banyak deteksi error. Karena data keberhasilan sudah ditemukan, maka dapat melakukan proses perhitungan keakuratan deteksi aplikasi dengan cara.

Keakuratan Data = 100% - ($\frac{80}{116}$ x 100%)
= 100% - 69%
= 31%

Tabel 8. Hasil pengujian dengan ketelitian aplikasi 80% pada video 2

Nomor	Detik	Jumlah Kendaraan		
		Terdeteksi	Manual	Deteksi Error
1	ke-10	20	11	9
2	ke-20	35	23	12
3	ke-30	51	32	19
4	ke-40	71	42	29
5	ke-50	92	53	39
6	ke-60	102	60	42
7	ke-70	113	66	47
8	ke-80	127	77	50
9	ke-90	147	89	58
10	ke-100	169	100	69
11	ke-110	180	106	74
12	ke-120	192	116	76

Pada tabel 8 pengambilan data dengan menguji keakuratan penghitungan deteksi aplikasi dan penghitungan manual dapat disimpulkan bahwa masih terdapat banyak deteksi error. Karena data keberhasilan sudah ditemukan, maka dapat melakukan proses perhitungan keakuratan deteksi aplikasi dengan cara.

Keakuratan Data = 100% - ($\frac{76}{116}$ x 100%)
= 100% - 66%
= 34%

Tabel 9. Hasil pengujian dengan ketelitian aplikasi 60% pada video 3

Nomor	Detik	Jumlah Kendaraan		
		Terdeteksi	Manual	Deteksi Error
1	ke-10	18	9	9
2	ke-20	47	26	21
3	ke-30	80	40	40
4	ke-40	107	54	53
5	ke-50	137	71	66
6	ke-60	166	88	78
7	ke-70	196	103	93
8	ke-80	225	119	106
9	ke-90	261	135	126
10	ke-100	279	145	134
11	ke-110	302	158	144
12	ke-120	333	175	158

Pada tabel 9 pengambilan data dengan menguji keakuratan penghitungan deteksi aplikasi dan penghitungan manual dapat disimpulkan bahwa masih terdapat banyak deteksi error. Karena

data keberhasilan sudah ditemukan, maka dapat melakukan proses perhitungan keakuratan deteksi aplikasi dengan cara.

Keakuratan Data = 100% - ($\frac{158}{175}$ x 100%)
= 100% - 90%
= 10%

Tabel 10. Hasil pengujian dengan ketelitian aplikasi 70% pada video 3

Nomor	Detik	Jumlah Kendaraan		
		Terdeteksi	Manual	Deteksi Error
1	ke-10	15	9	6
2	ke-20	48	26	22
3	ke-30	82	40	42
4	ke-40	111	54	57
5	ke-50	141	71	70
6	ke-60	174	88	86
7	ke-70	206	103	103
8	ke-80	220	119	101
9	ke-90	266	135	131
10	ke-100	281	145	136
11	ke-110	304	158	146
12	ke-120	332	175	157

Pada tabel 10 pengambilan data dengan menguji keakuratan penghitungan deteksi aplikasi dan penghitungan manual dapat disimpulkan bahwa masih terdapat banyak deteksi error. Karena data keberhasilan sudah ditemukan, maka dapat melakukan proses perhitungan keakuratan deteksi aplikasi dengan cara.

Keakuratan Data = 100% - ($\frac{157}{175}$ x 100%)
= 100% - 90%
= 10%

Tabel 11. Hasil pengujian dengan ketelitian aplikasi 80% pada video 3

Nomor	Detik	Jumlah Kendaraan		
		Terdeteksi	Manual	Deteksi Error
1	ke-10	12	9	3
2	ke-20	50	26	24
3	ke-30	82	40	42
4	ke-40	111	54	57
5	ke-50	135	71	64
6	ke-60	162	88	74
7	ke-70	192	103	89
8	ke-80	215	119	96
9	ke-90	245	135	110
10	ke-100	264	145	119
11	ke-110	284	158	126
12	ke-120	310	175	135

Pada tabel 11 pengambilan data dengan menguji keakuratan penghitungan deteksi aplikasi dan penghitungan manual dapat disimpulkan bahwa masih terdapat banyak deteksi error. Karena data keberhasilan sudah ditemukan, maka dapat melakukan proses perhitungan keakuratan deteksi aplikasi dengan cara.

Keakuratan Data = 100% - ($\frac{135}{175}$ x 100%)
= 100% - 77%
= 23%

KESIMPULAN

Dari banyaknya hasil pembahasan di atas dapat disimpulkan beberapa dianataranya adalah:

1. Sistem aplikasi penghitung kendaraan yang melintas di jalan raya menggunakan bantuan dari metode YOLO object detection yang dimodifikasi, sehingga tidak hanya mendeteksi kendaraan tapi juga dapat menghitung kendaraan yang melewatinya.
2. Sistem aplikasi penghitung kendaraan yang melintas tidak hanya menggunakan metode YOLO object detection melainkan juga menggunakan beberapa libray yang ada pada aplikasi Python Anaconda.
3. Hasil dari pengujian aplikasi dengan metode YOLO object detection sudah bisa membedakan kendaraan dengan roda 4 atau lebih ditandai dengan deteksi kotak berwarna hijau pada kendaraan yang berada di frame video.
4. Hasil pengujian aplikasi penghitung kendaraan berdasarkan metode YOLO object detection telah berhasil menghitung jumlah kendaran yang melewati sensor deteksi walaupun dengan nilai keakuratan yang belum mencapai 60%.

Saran

Dalam perancangan dan pengujian yang telah dijalankan oleh penulis, terdapat beberapa hal yang dapat ditambahkan supaya hasil perancangan lebih baik dari penulis, diantaranya adalah :

1. Menambahkan semua jenis kendaraan yang terdeteksi tidak hanya mendeteksi kendaraan dengan roda 4 atau lebih.
2. Dapat menambahkan nilai akurasi dari aplikasi penghitung kendaraan berdasarkan metode YOLO object detection yang telah dibuat penulis.
3. Membandingkan nilai akurasi dari aplikasi penghitung kendaraan berdasarkan metode YOLO object detection dengan metode-metode lainnya.
4. Dapat digunakan disaat intensitas cahaya rendah dengan bantuan rekaman kamera yang mempuni.

DAFTAR PUSTAKA

Agrawal, V. (2018). Melatih Model Klasifikasi Gambar Dengan Membuat Machine Learning. Jakarta: https://code.tutsplus.com/id/articles/training-an-image-classification-model-with-create-ml--cms-31617.

Antares, A. (2019). PENGENALAN FDEEP LEARNING MENGGUNAKAN Convolutional Neural Network (CNN). Surabaya: https://medium.com/@arum.antares9/pengenalan-fdeep-learning-menggunakan-convolutional-neural-network-cnn-67d9b5f9556a.

Dadang, W. (2018). Memahami Kecerdasan Buatan berupa Deep Learning dan Machine Learning. Palembang: https://warstek.com/2018/02/06/deepmachinelearning/.

Hidayati, Q. (2017). Kendali Lampu Lalu Lintas dengan Deteksi Kendaraan. Jogjakarta: JNTETI, Vol. 6, No. 2.

Karlina, O. E., & Indarti, D. (2019). Pengenalan Objek Makanan Cepat Saji Pada Video Dan Real Time Webcam Menggunakan Metode You Look Only Once (Yolo). Jakarta: Jurnal Ilmiah Informatika Komputer, Vol 24, No 3.

MC.AI. (2018). Deep Learning : Klasifikasi Image dengan Convolutional Neural Network (CNN). Jakarta: https://mc.ai/deep-learning-klasifikasi-image-dengan-convolutional-neural-network-cnn-menggunakan-keras-di/.

R., A. Y. (2018). Fully-Connected Layer CNN dan Implementasinya. Yogyakarta: https://machinelearning.mipa.ugm.ac.id/2018/06/25/fully-connected-layer-cnn-dan-implementasinya/.

Sena, S. (2017). Pengenalan Deep Learning Part 7 : Convolutional Neural Network (CNN). Surabaya: https://medium.com/@samuelsena/pengenalan-deep-learning-part-7-convolutional-neural-network-cnn-b003b477dc94.

Syaikhoni, A., & Ariyadi, A. (2018). Deteksi Objek Dengan Tensorflow Object Detection API. Jakarta: https://mti.binus.ac.id/2018/12/26/deteksi-objek-dengan-tensorflow-object-detection-api/.