PERANCANGAN SISTEM PELACAKAN (*TRACKING*) DAN PERHITUNGAN KENDARAAN PADA CITRA BERGERAK MENGGUNAKAN METODE *CONVOLUTIONAL NEURAL NETWORK*

Muhammad Arif Hudaya [*)], Imam Santoso dan Yosua Alvin Adi Soetrisno

Departemen Teknik Elektro, Fakultas Teknik, Universitas Diponegoro
Jl. Prof. Sudharto, SH, Kampus UNDIP Tembalang, Semarang 50275, Indonesia

[*)]*E-mail*: *arifhudaya@outlook.com*

Abstrak

Pengolahan citra digital kini telah banyak dimanfaatkan untuk berbagai keperluan seperti analisa citra *rontgen* pada bidang medis, penambahan efek dan desain animasi pada bidang hiburan, dan salah satunya pemanfaatan lainnya yaitu untuk pengawasan visual lalu lintas. Namun, penggunaan kamera CCTV di beberapa ruas jalan saat ini hanya berfungsi sebagai perangkat pemantauan kondisi lalu lintas saja, sedangkan tindakan aktif sebagai respon atas kondisi yang terjadi masih didominasi oleh operator. Pada penelitian ini, dirancang suatu sistem *tracking* dan perhitungan kendaraan pada citra bergerak menggunakan metode *Convolutional Neural Network* (CNN). Masukkan sistem berupa citra bergerak dari video lalu lintas, dimana sistem akan mendeteksi, *tracking* dan menghitung jumlah kendaraan. Algoritma untuk mendeteksi kendaraan menggunakan YOLO (*You Only Look Once*). Algoritma YOLO menggunakan pendekatan dengan membagi citra masukan menjadi beberapa area yang disebut *grid*, kemudian memprediksi kotak pembatas dan probabilitas kelas objek. Selanjutnya kotak pembatas dengan nilai *confidence* paling tinggi dijadikan pemisah antara objek satu dengan lainnya. Data latih untuk proses pelatihan yang digunakan sebanyak 3282 gambar dengan 10000 iterasi menghasilkan nilai mAP sebesar 79,11%. Hasil pengujian menggunakan perpotongan antara objek dengan garis diperoleh tingkat akurasi sebesar 83,33%, sedangkan pada pengujian perhitungan menggunakan *bounding box* yang terdeteksi diperoleh tingkat akurasi sebesar 83,5%.

Kata kunci: Deteksi, Pelacakan, Perhitungan, Kendaraan, Citra Bergerak, YOLO, CNN.

Abstract

Digital image processing has now been widely used for various purposes such as analysis of x-ray images in the medical field, the addition of effects and animation designs in the entertainment field, and one of them is another use for visual control of traffic. However, the use of CCTV cameras in some roads now only functions as a monitoring device for traffic conditions, while active actions in response to conditions that occur are still dominated by operators. In this research, a tracking system and vehicle calculation on a moving image are designed using the Convolutional Neural Network (CNN) method. The input of the system in the form of a moving image from a traffic video, where the system will detect, track and count the number of vehicles. Algorithm for detecting vehicles using YOLO (You Only Look Once). The YOLO algorithm uses an approach by dividing the input image into several areas called grids, then predicting bounding boxes and object class probabilities. Furthermore, the bounding box with the highest confidence value used as a separator between one object with another. The training data for the training process used were 3282 images with 10000 iterations resulting in a mAP value of 79.11%. The test results using the intersection of objects with lines obtained an accuracy rate of 83.33%, while the test calculations using bounding boxes detected an accuracy rate of 83.5% was obtained.

Keywords: Detection, Tracking, Counting, Vehicle, Moving Image, YOLO, CNN.

1. Pendahuluan

Pengawasan secara visual (*visual surveillance*) merupakan salah satu riset yang banyak diminati dibidang *computer vision*. Sistem pengawasan otomatis umumnya memanfaatkan kemampuan *computer vision* dalam mengolah informasi visual seperti segmentasi citra, klasifikasi objek, pengenalan pola, deteksi gerak, dan salah satunya adalah *object tracking* yang cukup banyak diminati. *Object tracking* merupakan salah satu bidang penting pada *computer vision*. *Object tracking* bertujuan untuk mendeteksi benda pada citra bergerak baik melalui kamera maupun dalam bentuk *file* video.

Beberapa bidang tertentu seperti olahraga, industri, dan medis telah banyak memanfaatkan teknologi *object tracking*, begitu juga dengan sistem pengawasan lalu lintas. Pemasangan kamera CCTV (*Closed-Circuit Television*) di

beberapa ruas jalan perkotaan memungkinkan petugas untuk mencegah tindak pelanggaran, memantau arus lalu lintas dan mengamati tingkat kepadatan pengguna kendaraan. Namun, penggunaan sistem kamera CCTV hanya berfungsi sebagai sistem pemantauan pasif saja [1]. Penelitian terkini telah menghasilkan sistem yang mampu mengolah informasi dari suatu citra kondisi lalu lintas secara spesifik seperti klasifikasi kendaraan, pelacakan arah gerak, perhitungan dan perkiraan kecepatan.

Tugas akhir ini bertujuan untuk merancang sistem yang dapat melakukan pengenalan dan *tracking* kendaraan serta melakukan perhitungan pada citra bergerak menggunakan arsitektur *convolutional neural network* yang diklasifikasikan dalam empat kelas, yaitu kelas 1 untuk jenis transportasi bis, kelas 2 untuk jenis transportasi mobil, kelas 3 untuk jenis transportasi sepeda motor dan kelas 4 untuk jenis transportasi truk. Algoritma yang digunakan untuk pengenalan kendaraan adalah YOLO (*You Only Look Once*).

Algoritma YOLO menggunakan pendekatan dengan membagi citra masukan menjadi beberapa area yang disebut *grid*, kemudian memprediksi kotak pembatas dan probabilitas kelas objek. Selanjutnya kotak pembatas dengan nilai *confidence* paling tinggi dijadikan pemisah antara objek satu dengan lainnya[2][3].

Agar dapat diolah menggunakan perangkat elektronik, suatu citra harus direpresentasikan menjadi nilai diskrit. Proses merubah citra kontinyu menjadi diskrit disebut digitalisasi[4][5]. Citra dapat dikelompokkan menjadi dua, yaitu citra diam (*image*) dan citra bergerak (*moving image*) atau video. Citra bergerak ini terdiri dari gabungan citra diam yang ditampilkan secara beruntun sehingga memberikan kesan pada mata sebagai gambar yang bergerak. Setiap gambar yang tampil pada citra bergerak disebut dengan *frame*. *Tracking* atau pelacakan objek merupakan suatu cara untuk mengolah citra bergerak dengan mengikuti elemen yang telah ditargetkan sebagai objek yang bergerak[6]. Beberapa faktor yang mempengaruhi kinerja algoritma pada *object tracking* diantaranya seperti hilangnya informasi karena proyeksi 3 matra dalam citra 2 matra, faktor pencahayaan, variasi objek, *noise* pada citra, serta latar belakang video itu sendiri [7][8].

2. Metode

2.1. YOLO

YOLO (*You Only Look Once*) adalah sebuah algoritma yang dikembangkan untuk mendeteksi suatu objek secara *real-time*. YOLO menggunakan pendekatan jaringan syaraf tiruan (JST) untuk mendeteksi objek pada sebuah citra. Dengan membagi citra menjadi beberapa wilayah, kemudian memprediksi setiap *bounding box* dan probabilitas untuk setiap wilayah. YOLO membagi citra masukan menjadi suatu *grid* berukuran S x S dimana tiap sel dari *grid* tersebut akan memprediksi *bounding box* dan menghasilkan nilai untuk tiap kelas. Masing-masing *bounding box* terdiri dari lima nilai prediksi yaitu pusat koordinat *x*, pusat koordinat *y*, lebar sel (*w*), tinggi sel (*h*) dan nilai *confidence* [2][3].

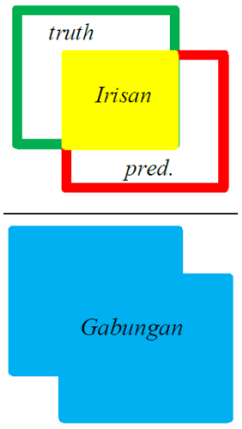
Pembagian citra masukan menjadi ukuran *grid* S x S, bagian pusat yang menjadi titik tengah dari *grid cell* bertanggungjawab untuk proses pendeteksian objek. Masing-masing *grid cell* memprediksi *bounding box* dan nilai *confidence* untuk setiap *box*[9]. Nilai *confidence* ini menyatakan tingkat akurasi sistem terhadap *bounding box* yang berisi inti informasi objek. Secara matematis nilai *confidence* dapat dinyatakan sebagai berikut:

$$Confidence\ Score = P_r(Object) * IOU_{pred}^{truth} \quad (1)$$

$P_r(Object)$ merupakan probabilitas objek yang terdapat di dalam *grid*, sedangkan IOU_{pred}^{truth} (*Intersection Over Union*) merupakan metode evaluasi untuk mengukur akurasi deteksi objek terhadap suatu data set. IOU_{pred}^{truth} juga dapat diartikan sebagai selisih nilai perpotongan (*intersection*) dan gabungan (*union*) antara *ground truth box* hasil anotasi dengan *prediction box*. [9][10]. Persamaan IOU dapat dinyatakan sebagai berikut:

$$IOU = \frac{Irisan}{Gabungan} \quad (2)$$

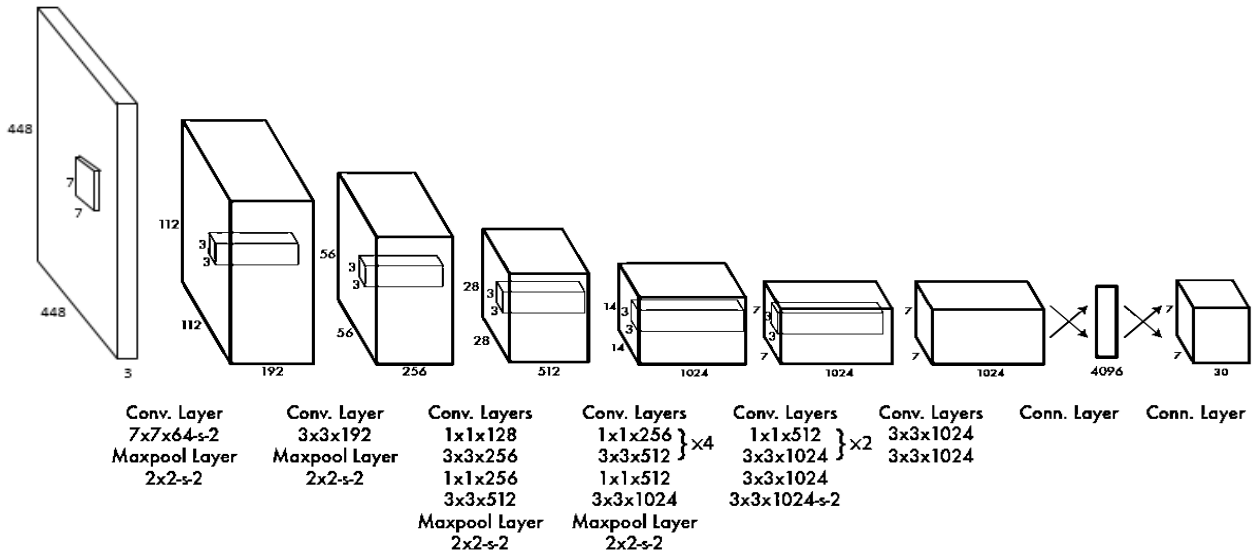
Ilustrasi persamaan IoU digambarkan seperti pada gambar 2 berikut.



Gambar 1. *Intersection over Union*[9]

Arsitektur YOLO terdiri dari 27 *layer* CNN, yaitu 24 *layer* yang terdiri dari *convolutional layer* dengan ukuran kernel 3x3 dan *maxpool layer* dengan ukuran kernel 2x2, kemudian diikuti 2 *fully connected layer* dan sebuah *final decision layer* dengan ukuran kernel 1x1 seperti pada gambar 2. *Input* pada *convolutional layer* berupa citra yang telah dibuah ukurannya menjadi 448x448. *Convolutional layer* menggunakan *filter* sebagai parameter yang akan di*update* dalam proses pembelajaran. *Filter* diinisialisasi dengan nilai tertentu, kemudian dari kalkulasi *grid* pada

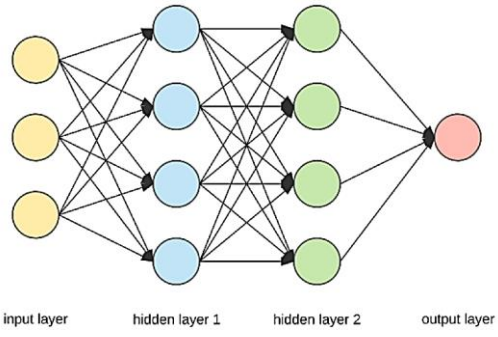
citra dengan nilai *filter* dihasilkan nilai keluaran dalam bentuk *array*. Kemudian dengan menggeser (konvolusi) *filter* di setiap kemungkinan posisi pada citra untuk dihasilkan sebuah *activation map. Fully connected layer* menggunakan ciri yang telah diambil dari *layer* konvolusi berupa nilai *array* untuk memprediksi probabilitas objek dan membuat *bounding box* diwaktu yang bersamaan. Sedangkan *final detection layer* pada YOLO memetakan *output* dari *fully connected layer* dan menetukan kelas serta *bounding box*[10].



Gambar 2. Arsitektur YOLO [2]

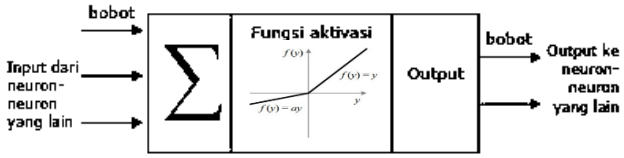
2.2. Jaringan Syaraf Tiruan

Artificial Neural Network (ANN) atau yang dikenal dengan nama Jaringan Syaraf Tiruan (JST) merupakan sistem proses komputasi yang terinspirasi dari sistem syaraf biologis yang merupakan representasi tiruan dari otak manusia [11]. JST terdiri dari sejumlah interkoneksi node yang disebut dengan *neuron*. JST umumnya terdiri dari tiga *layer* utama, yaitu *Input layer, hidden layer* dan *Output layer*. Gambar 2.3 menunjukkan ilustrasi model dasar jaringan syaraf tiruan.



Gambar 3 Ilustrasi model dasar JST [11]

Seperti sistem kerja otak manusia, fungsi dari tiap jaringan ditentukan oleh hubungan antar *neuron* yang disebut bobot (*weight*). Pada jaringan syaraf tiruan terdapat komponen-komponen utama fungsi yaitu, fungsi penjumlah, fungsi aktivasi dan *output* [12][13].



Gambar 4 Struktur Jaringan Syaraf Tiruan

Informasi (*Input*) dikirim menuju *neuron* dengan bobot tertentu. Kemudian *input* akan diproses oleh fungsi penambahan yang menjumlahkan nilai-nilai semua bobot. Hasil penjumlahan ini kemudian dibandingkan dengan nilai ambang tertentu melalui fungsi aktivasi setiap *neuron*. Apabila *input* melewati nilai ambang, maka *neuron* akan diaktifkan dengan fungsi aktivasi, pada penelitian kali ini digunakan jenis fungsi aktivasi *leaky* ReLu. Setelah *neuron* tersebut diaktifkan, selanjutnya *neuron* yang aktif ini akan mengirimkan *output* bobot ke semua *neuron* yang berhubungan dengannya.

2.3. *Convolutional Neural Network* (CNN)

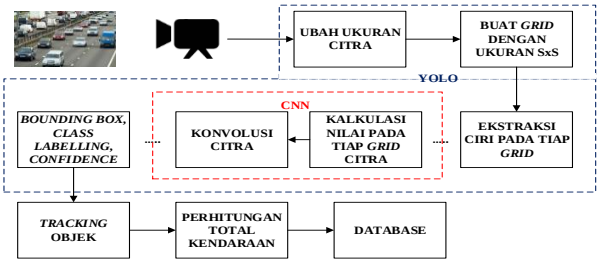
Convolutional Neural Network (CNN) adalah sebuah jaringan syaraf yang merupakan pengembangan dari *Multilayer Perceptron* (MLP) yang didesain untuk mengolah data dua dimensi. CNN dikembangkan pertama kali oleh seorang peneliti dari Jepang yaitu Kunihiko Fukushima dengan nama *NeoCognitron*[14]. CNN bekerja dengan menerapkan pergeseran kernel pada *layer* konvolusi. Tipe *layer* pada CNN dibagi menjadi dua bagian[15], yaitu :

a. *Layer* Ekstraksi Fitur
Lapisan pada jenis pertama yaitu adalah *Convolutional Layer* dan lapisan kedua adalah *pooling layer*. Pada lapisan ini terjadi proses konvolusi antara *grid cell* pada citra dengan *filter* yang telah diinisialisasi sebelumnya. Proses konvolusi ini bertujuan untuk mengambil ciri dari suatu citra. Ciri yang diambil berupa nilai hasil konvolusi yang merepresentasikan citra itu sendiri.

b. *Layer* Klasifikasi
Layer klasifikasi ini tersusun dari *fully connected layer* dan *decision layer*. *Input layer* pada lapisan ini berupa hasil *output* pada *layer* ekstraksi yang kemudian ditransformasikan dengan tambahan beberapa *hidden layer*. Hasil *output* berupa nilai akurasi kelas untuk klasifikasi.

2.4. Spesifikasi Sistem *Tracking* dan Perhitungan Kendaraan

Perancangan sistem ini terdiri dari tiga tahapan utama, yaitu penentuan spesifikasi komputer, perancangan perangkat lunak, dan perancangan program yang ditulis menggunakan bahasa pemrograman *python* versi 3.6.8. Sistem dirancang untuk dapat mendeteksi, melacak pergerakkan dan melakukan perhitungan objek berupa kendaraan. Informasi visual yang menjadi masukkan sistem berupa *file* video yang direkam secara manual di beberapa ruas jalan menggunakan kamera ponsel pintar Vivo Y35 dan video yang didapatkan dari dataset video online. Diagram blok sistem *tracking* dan perhitungan kendaraan dapat dilihat pada Gambar 5 berikut.



Gambar 5. Diagram alir sistem *tracking* dan perhitungan kendaraan

2.5. Pengumpulan Data Latih

Data yang digunakan sebagai data latih berupa kumpulan gambar kendaraan dari empat kelas yakni bis, mobil, sepeda motor dan truk. Total data yang didapatkan untuk empat kelas sebanyak 3282 data gambar yang terdiri dari 675 gambar bis, 1070 gambar mobil, 700 gambar sepeda motor, 700 gambar truk, dan 137 gambar kondisi lalu lintas yang berisi gabungan dari keempat kelas tersebut.

2.6. Pelatihan data

Pelatihan data latih dilakukan untuk proses ekstraksi ciri dari masing-masing kelas yang akan diklasifikasikan. Proses pelatihan dibagi menjadi dua bagian yaitu proses anotasi dan *training*. Gambar 6 merupakan contoh gambar dalam proses anotasi.

Gambar 6. Proses anotasi data latih

Proses anotasi gambar dilakukan berdasarkan jenis kelas kendaraan. Indeks 0 untuk mewakili adalah bis, indeks 1 mewakili mobil, indeks 2 mewakili sepeda motor dan indeks 3 adalah truk. Proses *training* dilakukan pada sistem operasi Ubuntu 18.04. Jumlah iterasi yang dilakukan pada proses *training* adalah sebanyak 10.000 kali iterasi. Setelah dilakukan proses *training* maka akan dihasilkan *file* bobot dengan format *weights* sesuai dengan jumlah iterasi terakhir yang ditentukan.

Setelah proses *training* selesai, proses selanjutnya adalah menguji tingkat akurasi bobot menggunakan data uji. Pengujian dilakukan dengan membandingkan hasil perhitungan secara manual untuk menentukan nilai akurasi (*Acc Score*) dengan hasil perhitungan program dengan parameter berupa *true positive (TP), false positive (FP),* dan *false negative (FN)*. Nilai *FP* dan *FN* merupakan nilai *error* sistem. *TP (True Positive)* merupakan target piksel yang dikenali sebagai objek, *FP (False Positive)* adalah *noise* yang terdeteksi sebagai objek sedangkan *FN (False Negative)* adalah target piksel yang tidak terdeteksi sebagai objek atau target deteksi dengan nilai *confidence* fluktuatif diantara batas nilai ambang (*threshold*) (Lipton dkk, 2014). Untuk menentukan nilai *Acc Score* dapat diihitung menggunakan persamaan 1 berikut,

$$F1\ Score = \left(\frac{TP}{TP+FP+FN}\right) 100\% \qquad (3)$$

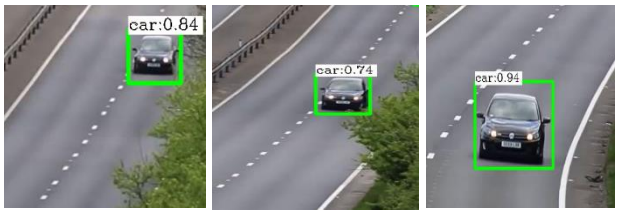
Sedangkan proses *tracking* citra dilakukan dengan mendeteksi target objek berupa kendaraan pada tiap *frame* yang dijalankan. Sehingga *bounding box* muncul pada tiap *frame* selama target objek muncul dan terdeteksi.

3. Pengujian dan Analisis

Pengujian pada program sistem *tracking* dan perhitungan kendaraan dibagi menjadi tiga tahap, yaitu pengujian performa progam *tracking*, pengujian akurasi perhitungan menggunakan *intersection*, pengujian akurasi perhitungan menggunakan *bounding box*.

3.1. Pengujian performa program *tracking*

Berbeda dengan citra diam atau gambar, nilai *confidence* dari suatu target objek pada citra bergerak selalu berubah tiap waktu. Perbedaan kondisi pada *frame* saat tampil, *frame* sebelumnya dan *frame* berikutnya sangat mempengaruhi performa program dalam mendeteksi objek. Pada pengujian performa *tracking* ini program akan mendeteksi objek kendaraan pada citra bergerak dan melakukan *tracking* terhadap objek yang terdeteksi.



(a) (b) (c)

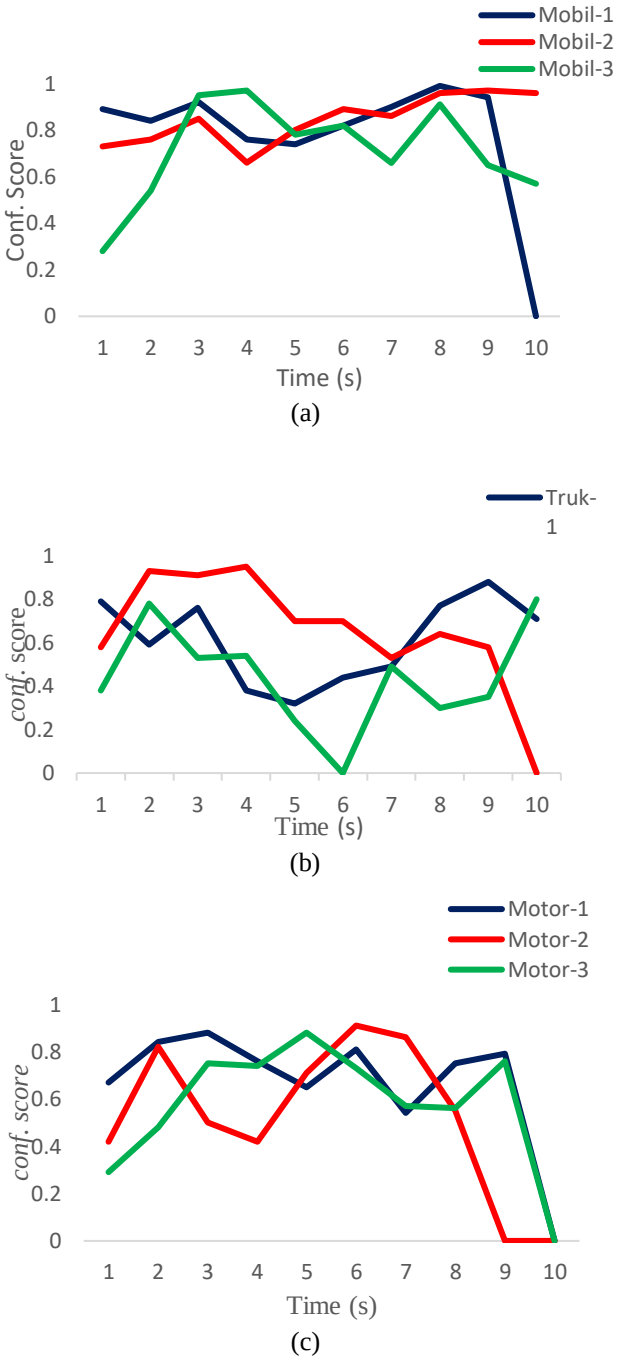
Gambar 7. Proses *tracking* mobil pada detik ke (a) 2, (b) 5, dan (c) 9

Gambar 7 menunjukkan proses *tracking* kendaraan pada kelas mobil. Nilai *confidence* pada tampilan *frame* mobil berubah setap waktu. Perubahan nilai *confidence* terhadap waktu selama proses *tracking* kendaraan disajikan pada Tabel 1 berikut.

Tabel 1. Data Perubahan Nilai *Confidence* terhadap Waktu

Time	Nilai *confidence*								
(s)	C1	C2	C3	T1	T2	T3	M1	M2	M3
1	0.89	0.73	0.28	0.79	0.58	0.38	0.67	0.42	0.29
2	0.84	0.76	0.54	0.59	0.93	0.78	0.84	0.82	0.48
3	0.92	0.85	0.95	0.76	0.91	0.53	0.88	0.50	0.75
4	0.76	0.66	0.97	0.38	0.95	0.54	0.76	0.42	0.74
5	0.74	0.80	0.78	0.32	0.70	0.24	0.65	0.71	0.88
6	0.82	0.89	0.82	0.44	0.70	0	0.81	0.91	0.73
7	0.90	0.86	0.66	0.49	0.53	0.49	0.54	0.86	0.57
8	0.99	0.96	0.91	0.77	0.64	0.30	0.75	0.55	0.56
9	0.94	0.97	0.65	0.88	0.58	0.35	0.79	0	0.76
10	0	0.96	0.57	0.71	0	0.80	0	0	0

Dari Tabel 1 diketahui bahwa nilai *confidence* program dalam mendeteksi objek pada kelas mobil (C), truk (T), dan sepeda motor (M) selalu berubah tiap waktu. Garfik perubahan nilai *confidence* terhadap waktu selama pengamatan disajikan pada gambar 8.



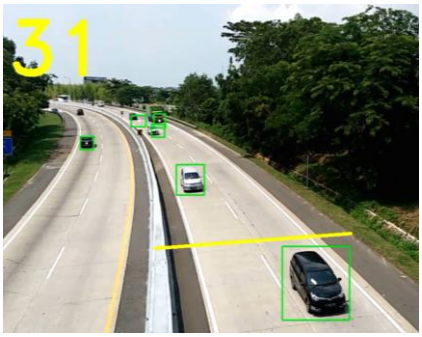
Gambar 8. Grafik perubahan nilai *confidence* terhadap waktu, (a) Mobil, (b) Truk, (c) Motor

.

3.2. Pengujian Perhitungan Menggunakan *Intersection* antar Objek dan Garis

Pada subbab ini akan dibahas pengujian perhitungan kendaraan menggunakan kondisi *intersection* antara garis batas dan titik pusat *bounding box* Pengujian dilakukan dengan membandingkan jumlah aktual objek pada *frame* yang aktif (*Vehicle/V*) dengan hasil perhitungan program (*Counting System/CS*) untuk menentukan nilai akurasi

(*Acc*) dengan parameter berupa *true positive (TP), false positive (FP),* dan *false negative (FN)*. Nilai *FP* dan *FN* merupakan nilai *error*.



Gambar 9. Program perhitungan menggunakan *intersection* objek dengan garis pada video di ruas Jalan Tol Semarang – Solo

Hasil program perhitungan menggunakan *intersection* objek dengan garis disajikan pada Tabel 2 berikut,

Tabel 2. Hasil perhitungan kendaraan menggunakan *intersection* antara objek dengan garis.

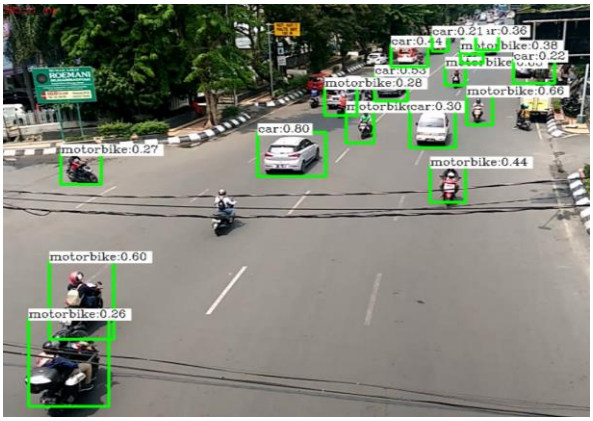
Data	*Vehicle*	CS	TP	FN	FP
Video 1	51	43	43	8	0
Video 2	93	62	62	31	0
Video 3	31	31	31	0	0
Video 4	20	19	19	1	0
Video 5	51	50	50	1	0

Dari tabel 2 diketahui bahwa selama proses perhitungan, terdapat beberapa kendaraan yang tidak terhitung. Kendaraan tidak terhitung ini dikategorikan sebagai parameter *false negative*. Kendaraan tidak terhitung ini disebabkan karena perhitugan nilai *confidence* terhadap objek kurang dari batas ambang karena pengaruh nilai *intersection over union* yang kecil yang menyebabkan hilangnya *bounding box* ketika objek bersentuhan dengan garis, sehingga kendaraan yang bersentuhan dengan garis tidak dianggap sebagai objek dan program tidak menambah jumlah perhitungan.

Akurasi program perhitungan kendaraan menggunakan perpotongan (*intersection*) antara objek dengan garis ini berdasarkan tabel 2 didapatkan nilai sebesar 83.33%

3.3. Pengujian Perhitungan Menggunakan Bounding Box

Pada subbab ini akan dibahas pengujian program perhitungan kendaraan menggunakan *bounding box* yang terdeteksi. Setelah program berjalan, kemudian dilakukan pengamatan jumlah kendaraan yang terhitung tiap 10 detik. Hasil perhitungan program selanjutnya dianalisa menggunakan parameter seperti jumlah aktual kendaraan (*Vehicle*), jumlah objek yang terdeteksi (*True Positive*), jumlah objek yang tidak dideteksi (*False Negative*) dan jumlah *noise* yang terdeteksi sebagai objek (*False Positive*). Nilai *FP* dan *FN* merupakan nilai *error*.

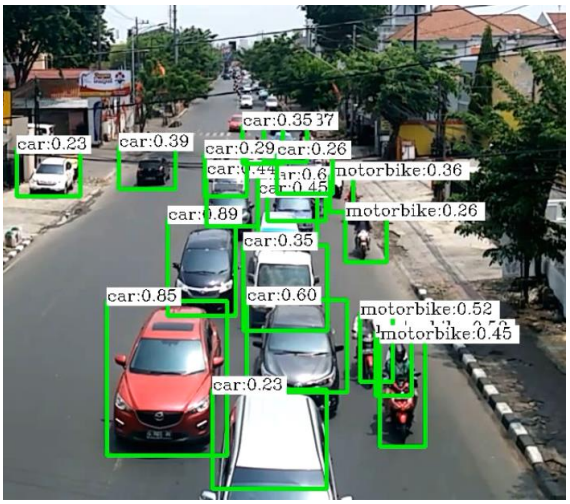


Gambar 10. Perhitungan *bounding box* pada video di ruas Jalan Ahmad Yani

Hasil program perhitungan menggunakan *bounding box* pada video di ruas Jalan Ahmad Yani disajikan pada Tabel 3 berikut:

Tabel 3. Hasil perhitungan kendaraan menggunakan *bounding box* pada video di ruas Jalan Ahmad Yani.

Time (s)	*Vehicle*	*CS*	*TP*	*FP*	*FN*
10	21	15	15	0	6
20	18	15	15	0	3
30	24	16	16	0	8
40	13	8	8	0	5
50	19	16	16	0	3
60	16	13	13	0	3



Gambar 11. Perhitungan *bounding box* pada video di ruas Jalan Brigjen Katamso

Kendaraan tidak terhitung ini disebabkan karena perhitugan nilai *confidence* terhadap objek kurang dari batas ambang karena pengaruh nilai *intersection over*

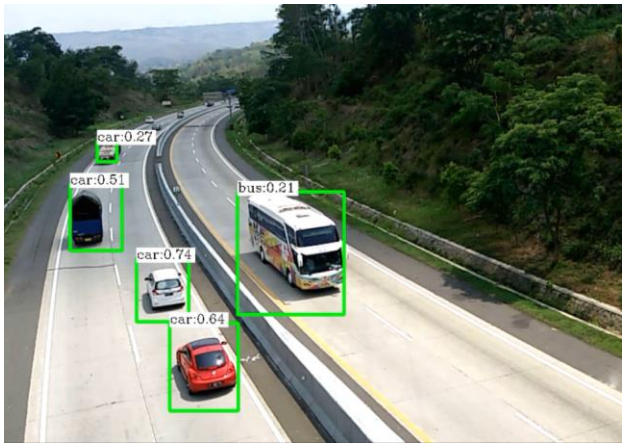
union yang kecil yang menyebabkan hilangnya *bounding box* dan kendaraan tidak terdeteksi. Disamping itu, faktor lain seperti pengaruh pada saat proses anotasi data latih yang tidak merata menyebabkan beberapa citra kendaraan pada jarak yang relatif jauh dari sudut pandang kamera tidak dapat terdeteksi.

Tingkat akurasi program perhitungan pada video di ruas Jalan Ahmad Yani berdasarkan nilai parameter yang dihasilkan pada tabel 3 didapatkan nilai sebesar 74,77%
Hasil program perhitungan menggunakan *bounding box* pada video di ruas Jalan Brigjend Katamso disajikan pada Tabel 4 berikut:

Tabel 4. Hasil perhitungan kendaraan menggunakan *bounding box* pada video di ruas Jalan Brigjen Katamso.

Time (s)	*Vehicle*	*TP*	*FP*	*FN*
10	15	12	1	2
20	16	13	2	1
30	13	11	2	0
40	16	12	2	2
50	14	12	2	0

Kendaraan tidak terhitung ini disebabkan karena perhitugan nilai *confidence* terhadap objek kurang dari batas ambang karena pengaruh nilai *intersection over union* yang kecil yang menyebabkan hilangnya *bounding box*. Disamping itu, faktor lain seperti pengaruh pada saat proses anotasi data latih yang tidak merata menyebabkan beberapa citra kendaraan pada jarak yang relatif jauh dari sudut pandang kamera tidak dapat terdeteksi.



Gambar 12. Perhitungan *bounding box* pada video di Jalan Tol Semarang-Solo

Hasil pengujian pada video di ruas Jalan Brigjend Katamso terdapat nilai pada parameter *false positive*. Kondisi *false positive* terjadi karena pada *frame* muncul *noise* yang terbentuk dari objek terhadap *background* yang menghasilkan bentuk yang menyerupai target objek sehingga program mendeteksi adanya kendaraan pada posisi tersebut.

Tingkat akurasi program perhitungan pada video di ruas Jalan Brigjen Katamso berdasarkan nilai parameter pada tabel 4 didapatkan nilai sebesar 81%

Hasil program perhitungan menggunakan *bounding box* pada video di Jalan Tol Semarang-Solo disajikan pada Tabel 5 berikut:

Tabel 5. Hasil perhitungan kendaraan menggunakan *bounding box* pada video di Jalan Tol Semarang-Solo.

Time (s)	*Vehicle*	*TP*	*FP*	*FN*
10	5	5	0	0
20	5	5	0	0
30	10	9	0	1
40	5	4	0	1
50	10	10	0	0
60	3	3	0	0

Kendaraan tidak terhitung ini disebabkan karena perhitugan nilai *confidence* terhadap objek kurang dari batas ambang karena pengaruh nilai *intersection over union* yang kecil yang menyebabkan hilangnya *bounding box*. Disamping itu, faktor lain seperti pengaruh pada saat proses anotasi data latih yang tidak merata menyebabkan beberapa citra kendaraan pada jarak yang relatif jauh dari sudut pandang kamera tidak dapat terdeteksi.

Tingkat akurasi program perhitungan pada video di ruas Jalan Tol Semarang-Solo berdasarkan nilai parameter pada tabel 5 didapatkan nilai sebesar 94,73%

Berdasarkan tingkat akurasi dari pengujian perhitungan *bounding box* dengan tiga video dari tiga tempat berbeda, dapat dihitung rata-rata tingkat akurasi program sebesar 83.5 %.

4. Kesimpulan

Pada pengujian performa *tracking*, nilai *confidence* yang didapatkan selama proses *tracking* selalu berubah tiap detiknya. Hal ini disebakan karena posisi objek yang berubah terhadap kamera yang menyebabkan nilai *confidence* fluktuatif. Sementara pada pengujian proses perhitungan, pengujian menggunakan *intersection* antara objek yang dikenali dengan garis diperoleh tingkat akurasi sebesar 83.33%. Sedangkan pada pengujian perhitungan kendaraan menggunakan *bounding box* didapatkan tingkat akurasi sebesar 83.5%.

Dari hasil penelitian ini, disarankan untuk menggunakan data latih dan iterasi pelatihan yang lebih banyak untuk mendapatkan tingkat akurasi.yang lebih tinggi. Disamping itu penggunaan kamera serta jarak dan sudut perekaman citra serta proses anotasi citra yang optimal dapat meningkatkan akurasi.

Referensi

[1] M. H. B. Pratama, A. Hidayatno dan A. A. Zahra, "Aplikasi Deteksi Gerak Pada Kamera Keamanan Menggunakan Metode Background Subtraction dengan Algoritma Gaussian Mixture Model," *Transient* vol. 6 no.2. ISSN: 2302-9927.

[2] J. Redmon, S. Divvala, R. Girshick dan A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," dipresentasikan di CVPR, Las Vegas, Amerika Serikat, Jun. 26 – Jul. 1, 2016.

[3] K. A. Shianto, K. Gunadi dan E. Setyati, "Deteksi Jenis Mobil Menggunakan Metode YOLO dan Faster R-CNN," *Jurnal Infra*, vol. 7 no. 1 pp. 157-163, 2019.

[4] Arfian dan F. Hakim, "Implementasi Convolutional Neural Network Terhadap Transportasi Tradisional Menggunakan Keras," Tugas Akhir, Program Studi Statistika, Universitas Islam Indonesia, 2018.

[5] R. Munir, *Pengolahan Citra Digital Dengan Pendekatan Algoritmik*, Bandung: Informatika, 2004

[6] H. V. Al-Kautsar dan K. Adi, "Implementesi Object Tracking Untuk Mendeteksi dan Menghitung Kendaraan Secara Otomatis Menggunakan Metode Kalman Filter dan Gaussian Mixture Model," *Youngster Physics Journal*, vol. 5 no. 1, pp 13-20. ISSN: 2302 -7371

[7] Y. Xu, X. Zho, S. Chen dan F. Li, "Deep Learning for Multiple Object Tracking," *IET Journal* vol. 13 pp 355-368, ISSN: 1751-9632.

[8] D. A. Prabowo, D. Abdullah, A. Manik, "Deteksi dan Perhitungan Objek Berdasarkan Warna Menggunakan Color Object Tracking," Jurnal Pseudocode vol. 5, no. 2, September 2018, ISSN: 2355-5920.

[9] S. Jupiyandi, F. Rizqullah, Y. Pratama, M. R. Dharmawan, dan I. Cholissodin,"Pengembangan Deteksi Citra Mobil Untuk Mengetahui Jumlah Tempat Parkir Menggunakan Cuda dan Modified Yolo." Jurnal Teknologi Informasi dan Ilmu Komputer, vol.6, no.4, pp. 413-419, 2018, ISSN: 2528-6579.

[10] M. H. Putra, Z. M. Yussof, S. I. Salim dan K. C. Lim, "Convolutional Neural Network For Person Detection Using YOLO Framework," *Journal of Telecommunication, Elecronic and Computer Engineering*, vol. 9, 2-13, 2017, ISSN: 2289-8131.

[11] R. D. Nurfita, "Implementasi Deep Learning erbasis Tensorflow untuk Pengenalan Sidik Jari," Tugas Akhir, Jurusan Informatika, Universitas Muhammadiyah Surakarta, Surakarta, Indonesia, 2018.

[12] S. R. Dewi dan F. Hakim, "Deep Learning Object Detection Pada Video Menggunakan Tensorflow dan Convolutional Neural Network," Tugas Akhir Program Studi Statistika, Universitas Islam Indonesia, Yogyakarta, 2018.

[13] Z. A. Fikriya, M. I. Irawan dan Soetrisno, "Implementasi Extreme Learning Machine untuk Pengenalan Objek Citra Digital," Jurnal Sains dan Seni ITS, vol. 6, no. 1, 2017, ISSN: 2337-3520.

[14] I. S. E. Putra, A. Y. Wijaya dan R. Soelaiman, "Klasifikasi Citra Menggunakan Convolutional Neural Network (CNN) Pada Caltech 101," *Jurnal Teknik ITS*, vol. 5 no.1, pp. A65-A69 ISSN: 2337-3539.

[15] K. O'Shea dan N. Ryan, An Introduction to Convolutional Neural Network, ArXiv: 1511.08458, 2015.