

Peningkatan Akurasi Penghitungan Jumlah Kendaraan dengan Membangkitkan Urutan Identitas Deteksi Berbasis *Yolov4 Deep Neural Networks*

Faqih Rofii^{1*}, Gigih Priyandoko¹, M. Ifan Fanani¹, Aji Suraji²

¹Teknik Elektro, Fakultas Teknik, Universitas Widyagama Malang

²Teknik Sipil, Fakultas Teknik, Universitas Widyagama Malang
Jl. Taman Borobudur Indah, Malang, Jawa Timur, Indonesia 65142

Abstrak

Model untuk deteksi, klasifikasi dan penghitungan jumlah kendaraan berbasis pada visi komputer dan kecerdasan buatan terus berkembang. Penelitian ini mempresentasikan pendekatan model deteksi, klasifikasi dan penghitungan jumlah kendaraan berbasis Yolov4. Penghitungan jumlah kendaraan dilakukan dengan membangkitkan nomor urut identitas tiap kendaraan setelah objek dideteksi dan diklasifikasi yang ditandai dengan ditampilkannya bounding box, kelas dan skor kepercayaan. Input sistem berupa dataset video yang diambil dengan mempertimbangkan posisi kamera, intensitas cahaya dan kepadatan trafik kendaraan. Metode yang dilakukan telah mampu menghitung jumlah kendaraan berupa mobil, sepeda motor, bus dan truk. Evaluasi kinerja model didasarkan pada parameter akurasi, presisi dan recall total dari confusion-matrix. Hasil pengujian dataset dan perhitungan parameter performa model telah didapat nilai akurasi, presisi dan recall total terbaik ketika pengujian model dilakukan pada siang hari dimana posisi kamera pada ketinggian 6 m dan susut 50⁰ sebesar 83%, 93% dan 94%. Sedangkan akurasi, presisi dan recall total terendah didapat ketika pengujian model dilakukan pada malam hari dimana posisi kamera pada ketinggian 1,5 m dan susut 90⁰ sebesar 68%, 77% dan 78%.

Kata kunci: deteksi dan klasifikasi; penghitungan kendaraan; Yolov4; deep neural networks; akurasi

Abstract

[**Title: Vehicle Counting Accuracy Improvement By Identity Sequences Detection Based on Yolov4 Deep Neural Networks**] Models for vehicle detection, classification, and counting based on computer vision and artificial intelligence are constantly evolving. In this study, we present the Yolov4-based vehicle detection, classification, and counting model approach. The number of vehicles was calculated by generating the serial number of the identity of each vehicle. The object is detected and classified, marked by the display of bounding boxes, classes, and confidence scores. The system input is a video dataset that considers the camera position, light intensity, and vehicle traffic density. The method has counted the number of vehicles: cars, motorcycles, buses, and trucks. Evaluation of model performance is based on accuracy, precision, and total recall of the confusion matrix. The results of the dataset test and the calculation of the model performance parameters had obtained the best accuracy, precision. Total recall values when the model testing was carried out during the day where the camera position was at the height of 6 m and the loss of 500 was 83%, 93%, and 94%. Meanwhile, the lowest total accuracy, precision, and recall were obtained when the model was tested at night. The camera position was at the height of 1.5 m, and 900 losses were 68%, 77%, and 78%.

Keywords: detection and classification; vehicles counting; Yolov4; deep neural networks; accuracy

^{*)} Penulis Korespondensi.
E-mail: faqih@widyagama.ac.id

1. Pendahuluan

Perkembangan jumlah kendaraan berdasarkan jenis kendaraan berupa: mobil penumpang, bis, mobil

barang dan sepeda motor sampai dengan akhir tahun 2018 adalah sejumlah 146.858.759 unit, dimana data tersebut bersumber dari Korps Lalu Lintas Kepolisian Republik Indonesia (Badan Pusat Statistik, n.d.). Pesatnya peningkatan jumlah kendaraan berakibat pada meningkatnya trafik kendaraan di jalan raya. Di sisi lain lebar jalan tidak mengalami penambahan secara signifikan sehingga mengakibatkan kemacetan di jalan terutama di dalam kota. Untuk mengatasi penumpukan trafik kendaraan perlu dirumuskan strategi rekayasa trafik agar kemacetan tidak semakin parah. Rekayasa trafik membutuhkan sejumlah data terkait dengan kondisi jalan dan kendaraan, salah satu faktor penting adalah mengetahui jumlah kendaraan pada suatu ruas jalan dari waktu ke waktu.

Saat ini penghitungan jumlah kendaraan masih dilakukan secara manual dengan mencatat setiap jenis kendaraan yang lewat. Cara lain adalah dengan memasang suatu sensor pada ruas jalan tertentu dimana sensor akan mendeteksi setiap kendaraan yang lewat berdasarkan kuat tekan, infra merah, ultras sonic (Yuniarto, 2008) atau jenis sensor yang lain (Balid dkk., 2017). Sejalan dengan perkembangan metode kecerdasan buatan, visi komputer dan murahnya perangkat kamera, metode deteksi dan penghitungan jumlah kendaraan juga semakin berkembang. Sensor berbasis penglihatan (*vision*) dapat memberikan informasi yang lebih banyak daripada sensor lain, sehingga deteksi trafik berbasis kamera memiliki peran penting pada deteksi kendaraan dan pemantauan lalu lintas (Mashudi dkk., 2020). Namun, *video* yang ditangkap oleh kamera membutuhkan pemrosesan yang canggih untuk mendeteksi kendaraan dikarenakan berbagai macam bentuk, warna dan perspektif.

Deteksi objek berupa kendaraan berbasis visi komputer terbagi menjadi dua, yaitu metode mesin tradisional dan *deep learning*. Metode tradisional meliputi subtraksi latar belakang, perbedaan *frame* dan aliran optik (Antony & Suchetha, 2016). Metode *deep learning* banyak dikembangkan berbasis *Convolutional Neural Networks* (CNN) yang memiliki kemampuan mempelajari fitur untuk deteksi dan klasifikasi suatu objek dalam gambar (Gomaa dkk., 2019). Salah satu metode yang penting pada deteksi objek adalah *framework* dari *You Only Look Once* (YOLO). Sistem deteksi pada YOLO melalui tiga tahap utama yaitu : *resize* gambar, menjalankan jaringan konvolusi dan *Non-max suppression* (Redmon dkk., 2016).

YOLO adalah algoritma deteksi objek berbasis CNN yang diperkenalkan pertama kali oleh Joseph Redmon. YOLO menyatukan komponen terpisah dari sistem deteksi menjadi terpadu dalam satu jaringan *neural* dengan menggunakan fitur dari gambar input untuk memprediksi *bounding box*. Jaringan tersebut memprediksi semua *bounding box* dari semua kelas

yang terdapat dalam gambar secara simultan. Ide dasar YOLO adalah membagi input gambar atau *video* menjadi $S \times S$ grid. Suatu sel grid yang kecil bertanggungjawab melakukan prediksi objek berupa *bounding box* B dan skor kepercayaan C, jika titik tengah suatu objek jatuh di sel grid tersebut. Skor kepercayaan menggambarkan seberapa besar model percaya bahwa di dalam *box* terdapat suatu objek dan seberapa akurat dia memprediksi objek.

Masing-masing *bounding box* mengandung 5 komponen prediksi: x, y, w, h dan skor kepercayaan. Koordinat (x, y) mewakili titik tengah dari *box* relatif terhadap batas-batas dari sel grid. Lebar dan tinggi (w, h) diperkirakan relatif terhadap keseluruhan gambar. Skor prediksi menyatakan *Intersection Over Union* (IOU) antara kotak yang diprediksi dan *ground truth*. Setiap sel grid juga memprediksi probabilitas kelas bersyarat $P_r(Class_i|Object)$ pada sel grid yang berisi objek dikalikan dengan prediksi keyakinan kotak individu untuk mendapatkan skor keyakinan khusus kelas untuk setiap kotak. Skor ini mengkodekan probabilitas kelas tersebut muncul di *box* dan seberapa cocok *box* prediksi dengan objek.

$$\frac{P_r(Class_i|Object) * P_r(Object) * IOU_{pred}^{truth}}{P_r(Class_i) * IOU_{pred}^{truth}} = \quad (1)$$

Input gambar pada YOLO dibagi menjadi *grid* 13×13 sel, sehingga terdapat 169 sel kotak dan setiap sel memprediksi 5 *bounding box*. Total terdapat $169 \times 5 = 845$ *bounding box* pada satu gambar. Setiap kotak memiliki skor kepercayaan masing-masing, ada yang rendah hingga yang tinggi. YOLO hanya menyimpan skor kepercayaan di atas *threshold* yang nilainya dapat kita tentukan sendiri, misal 30%. *Bounding box* yang memiliki skor terendah kemudian dihilangkan, proses ini disebut *non-max Suppression*. Sebagaimana pada Gambar 1, hanya terdapat 3 objek dengan skor kepercayaan tinggi yang kemudian ditampilkan hasil prediksinya. Walaupun terdapat 845 prediksi yang terpisah, namun semuanya dibuat dalam waktu bersamaan, CNN hanya berjalan dalam sekali proses, sehingga menjadikan YOLO sebagai algoritma deteksi yang kuat dan cepat.

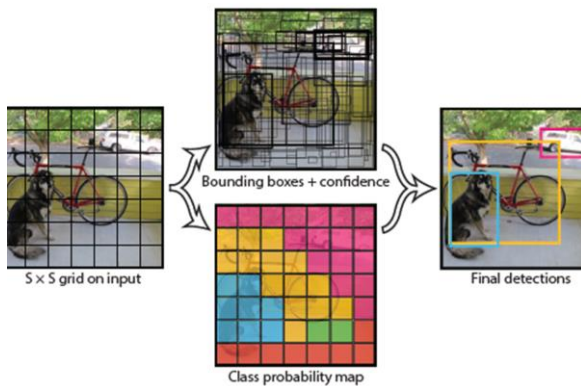
Penelitian terkait deteksi dan penghitungan kendaraan telah banyak dikembangkan, antara lain dengan menggunakan metode: *Fuzzy C-means* dan segmentasi deteksi tepi *Canny* (Wibowo dkk., 2014), algoritma *Backpropagation* dan *Sobel* (Adistya & Muslim, 2016), analisa *Blob* (Setyawan dkk., 2019), *Framework Tensorflow Object Detection API* (Manajang dkk., 2020), CNN dan *optical flow* (Gomaa dkk., 2019) dan algoritma *Deep Learning YOLO* (Rachmawati & Widhyaestoeti, 2020). Penelitian

terdahulu yang dilakukan telah mampu mendeteksi dan menghitung kendaraan, namun tidak menjelaskan dan menampilkan urutan identitas kelas kendaraan yang dideteksi dan dihitung. Tujuan penelitian ini adalah menghasilkan sistem deteksi dan penghitung masing-masing kelas kendaraan dan jumlah totalnya melalui analisis *video* berbasis algoritma YOLO versi 4. Untuk menandai kendaraan yang telah terdeteksi pada suatu *frame* dalam *video*, maka kendaraan yang telah terdeteksi ditandai dengan urutan nomor identitas berdasarkan kelasnya. Adapun kelas kendaraan yang dideteksi dalam penelitian ini adalah: mobil, bus, truk, sepeda, sepeda motor dan kereta api.

2. Metode Penelitian

Penelitian yang dilakukan berbasis algoritma YOLO versi 4. Kebanyakan orang saat ini masih menggunakan Yolov3 sebagai algoritma deteksi objek karena menghasilkan output yang bagus. Yolov3 mempunyai 33 layer konvolusi sebagai ekstraktor fitur dan Darknet 53 sebagai backbone jaringan (Bochkovskiy *dkk.*, 2020). Yolov4 telah memperbaiki dua parameter penting dalam menilai kualitas dari algoritma deteksi objek yaitu: akurasi (*average precision /AP*) dan kecepatan (*frames per second/FPS*). Selain itu *training model* dapat dilakukan pada GPU tunggal. Untuk memperbaiki jaringan terdapat spesifikasi teknis yaitu *Bag-of-Freebies* dan *Bag-Of-Specials*. *Backbone* sebagai arsitektur ekstraksi fitur pada Yolov4 menggunakan CSPDarknet53. Jika dibandingkan dengan detektor obek lain seperti EfficientDet, Yolov4 dapat bekerja dua kali lebih cepat, dan juga memperbaiki AP dan FPS dari Yolov3 sebesar masing-masing 10% dan 12%.

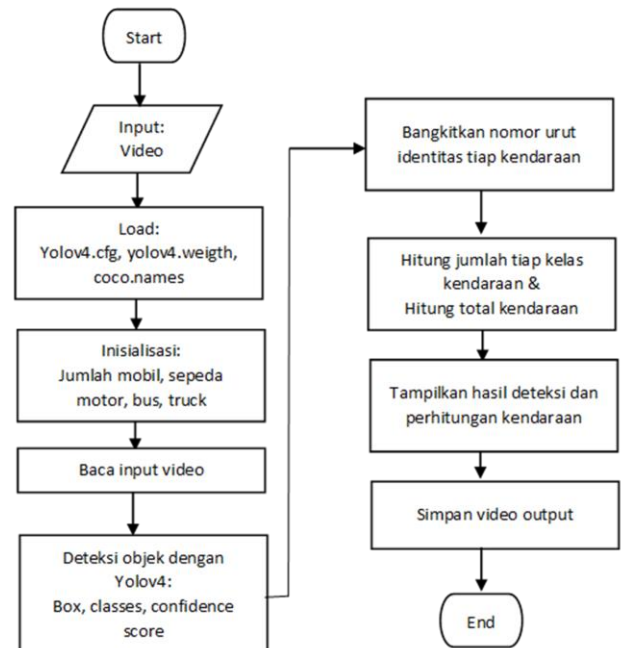
Kim dalam *paper*nya telah memaparkan metode pengenalan jenis kendaraan dengan model *deep learning* YOLO dibandingkan dengan *Recurrent-Convolutional Neural Networks* (R-CNN) dan *Single Shot Detector*



Gambar 1. Model deteksi objek YOLO (Redmon *dkk.*, 2016)

(SSD). R-CNN menggunakan basis CNN untuk deteksi dan klasifikasi jenis kendaraan melalui dua langkah, hasilnya adalah akurasi relatif tinggi, namun kecepatan deteksinya sangat lambat. SSD karena menggunakan model yang ringan menghasilkan deteksi kendaraan yang lebih cepat namun akurasi rendah, bahkan terkadang gagal mendeteksi kendaraan. YOLO memproses deteksi kendaraan lebih cepat dari R-CNN dengan akurasi yang lebih baik dari SSD (Kim *dkk.*, 2020). Kinerja AP dari model Yolov4, R-CNN dan SSD dalam mendeteksi mobil adalah 98,8%, 93,2% dan 92.7%. Sedangkan kinerja FPS dari masing-masing model secara berturut-turut adalah 82,1, 105,14 dan 36,32 FPS. Paper lain yang dipresentasikan oleh Wang membandingkan Yolov3, R-CNN, dan SSD. Hasilnya dapat dinyatakan bahwa kinerja AP dan FPS dari YOLO adalah *moderate* (Wang *dkk.*, . 2019). Berdasarkan beberapa paper sebelumnya, maka pada penelitian ini dipilih Yolov4 untuk digunakan deteksi dan klasifikasi jenis kendaraan.

Diagram alir penelitian yang dilakukan ditunjukkan oleh Gambar 2. Input detektor adalah *video* hasil rekaman dalam format .MP4, sebanyak 10 *video* yang diambil pada waktu berbeda: pagi, siang, petang dan malam, serta kondisi trafik kendaraan pada waktu padat, normal dan lengang. Pengambilan data *video* pada waktu yang berbeda adalah mempertimbangkan kepadatan trafik kendaraan dan juga intensitas cahaya pada waktu yang berbeda untuk mengetahui kinerja detektor.



Gambar 2. Diagram alir proses deteksi dan penghitungan jumlah kendaraan

Sistem penghitung jumlah kendaraan terdiri dari tiga proses utama, yaitu : deteksi, pelacakan (*tracking*) dan penghitungan. Proses deteksi dengan detektor Yolov4 membutuhkan beberapa file utama yaitu *yolov4.cfg*, *yolov4.weights* dan *coco.names*. Detektor ini dijalankan pada python 3.8 dengan dukungan *frameworks* dan library pendukung lainnya, yaitu: *tensorflow* 1.15.4, *keras* 2.2.4, *numpy* 1.18.5, *imutils* 0.5.3, *opencv* 4.4, *scipy* 1.5.4 dan lainnya. Untuk mulai menghitung jumlah kendaraan yang akan dideteksi maka dilakukan inisialisasi nilai awal 0 pada fungsi penjumlahan mobil, sepeda motor, bus dan truk.

Pada penelitian ini digunakan *yolov4.cfg* dan *yolov4.weights*. *Yolov4.cfg* adalah file yang berisi parameter jaringan CNN yang dapat disesuaikan dengan kebutuhan input, training dan output. *Yolov4.weights* berisi nilai-nilai bobot jaringan yang telah diupdate setelah proses training. Penelitian ini menggunakan bobot *pre-trained networks* sesuai paper yang disajikan oleh Bochkovskiy. *Training* menggunakan *dataset imagenet*, iterasi *training* sebanyak 1.000 kali dan terus diupdate tiap kelipatan seribu, ukuran *batch* dan *mini batch* adalah 128 dan 32, laju *training* 0,1 dan momentum 0,9 (Bochkovskiy dkk., 2020). *Pre-trained networks* yang telah didapat, selanjutnya kita modifikasi agar dapat mendeteksi dan mengklasifikasi objek gambar dalam *frame video* serta menampilkan nama objek kendaraan berupa mobil, truk, bus dan sepeda motor.

Untuk menguji kinerja *modified pre-trained networks* Yolov4 yang telah didapat, diambil *video* trafik kendaraan sebagai bahan uji. Bahan uji di sini adalah *video* dataset yang diambil dengan merekam trafik kendaraan secara langsung di Wilayah perbatasan Kota Malang menuju ke arah Surabaya. Titik pengambilan *video* ini dipilih karena merupakan jalan yang sibuk dan jenis kendaraan yang lewat dapat mewakili kelas kendaraan yang akan dideteksi dan diklasifikasi. Peralatan yang digunakan untuk mengambil *video* adalah kamera *handphone* dengan resolusi gambar 1080 pixels/30 fps. Lama pengambilan tiap *video* adalah 300 detik, dengan jumlah pengambilan sebanyak 30 kali pada waktu pagi, siang dan malam. Teknik pengambilan *video* dilakukan pada dua posisi kamera, yaitu posisi lurus di depan objek dan posisi 1,5 m di atas objek. Proses pengujian dataset *video* dilakukan melalui komputer dengan spesifikasi prosesor core i5 generasi 6, 16 GB RAM dan GPU 2 GB, dengan *python* dan *darknet* (Redmon, 2016).

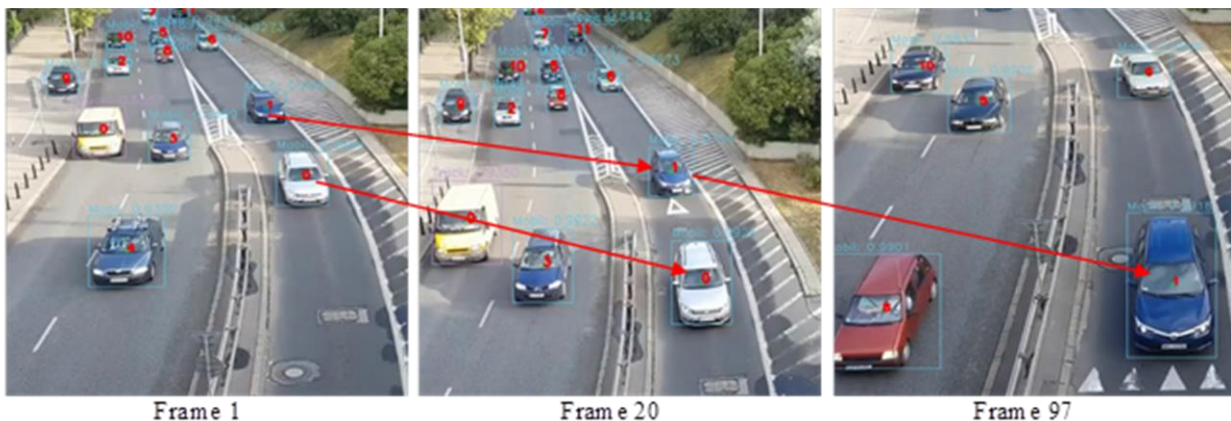
Pembacaan input *video* menggunakan perintah *VideoCapture* dari *opencv* dimana *video* input dalam format *.mp4*. Detektor akan mendeteksi objek pada

frame yang telah dibaca dari *video* input untuk mendapatkan *bounding box*, kelas dan skor kepercayaan. Pelacakan adalah proses mengikuti jalur atau pergerakan suatu objek dengan tujuan untuk menemukan atau mengamati jalurnya. Proses tracking direpresentasikan dengan menampilkan nomor identitas kendaraan dan menjaganya hingga objek keluar dari area deteksi dalam *frame video*. Untuk membangkitkan nomor urut identitas tiap kendaraan yang terdeteksi digunakan alur program berikut:

```

Input: frame video input
Output: Identitas kendaraan terdeteksi
Initialization idx, current_detcetion
Get ID_kendaraan
if len(idxs) > 0:
if (LABELS[classIDs[i]]== "Mobil" in vehicles ):
    current_detections[(centerX, centerY)] =
    jumlah_mobil
    jumlah_mobil += 1
if (LABELS[classIDs[i]]== "Motor" in vehicles ):
    current_detections[(centerX, centerY)] =
    jumlah_SpdMotor
    jumlah_SpdMotor += 1
if (LABELS[classIDs[i]]== "Bus" in vehicles ):
    current_detections[(centerX, centerY)] =
    jumlah_bus
    jumlah_bus += 1
if (LABELS[classIDs[i]]== "Truck" in vehicles ):
    current_detections[(centerX, centerY)] =
    jumlah_truck
    jumlah_truck += 1
ID = current_detections.get((centerX, centerY))
End
    
```

Sub-program di atas bertujuan untuk membangkitkan urutan identitas kelas kendaraan yang terdeteksi dalam bentuk angka 0,1,2,3 dan seterusnya sampai dengan objek kendaraan yang terdeteksi di frame terakhir. Objek kendaraan yang terdeteksi selanjutnya diklasifikasi berdasarkan jenis kendaraan dan dilakukan penghitungan dengan menambahkan +1 untuk tiap-tiap jenis kendaraan yang terdeteksi dalam frame dan mendapatkan nomor urut identitas kendaraan. Ketika masing-masing jenis kendaraan dijumlahkan, pada saat yang sama total kendaraan yang telah dideteksi juga dijumlahkan. Hasil deteksi, tracking dan penghitungan selanjutnya ditampilkan sepanjang *frame video* selama ketiga proses tersebut berlangsung, dan di saat yang sama tiap-tiap *frame video* yang telah diproses akan disimpan menjadi output dalam format *.avi*.



Gambar 3. Deteksi dan pelacakan dengan membangkitkan urutan identitas objek pada tiap *frame*

3. Hasil dan Pembahasan

Penelitian ini menggunakan *pre-trained network* dari Yolov4 untuk proses deteksi objek kendaraan (Mahto, 2020). Secara khusus penelitian ini bertujuan untuk mendeteksi dan melacak kendaraan dengan membangkitkan nomor urut identitas kendaraan yang terdeteksi dalam suatu *frame video* dan tetap menjaganya hingga kendaraan yang terdeteksi tidak muncul dalam video. Setiap objek kendaraan yang telah terdeteksi kemudian ditampilkan label kelas, nomor identitas dan skor kepercayaan tertinggi. Setelah proses deteksi selanjutnya dilakukan penghitungan sesuai kelas kendaraan, yaitu: mobil, sepeda motor, bus dan truk.

Pada penelitian ini dilakukan analisis video trafik lalu lintas yang berdurasi 5 menit untuk menguji program deteksi dan pelacakan gerak kendaraan dengan membangkitkan nomor identitas kendaraan. Video tersebut menjadi input bagi program, kemudian proses deteksi objek mulai dilakukan dari *frame* pertama hingga *frame* terakhir.

Gambar 3 menunjukkan contoh hasil dari proses yang dilakukan. Pada *frame* pertama objek telah terdeteksi dengan menampilkan label kelas dan skor kepercayaan. Pada saat yang sama nomor urut identitas dimunculkan di titik tengah *bounding box* berupa angka 0, 1, 2, 3, dan seterusnya. Nomor identitas tersebut muncul setelah detektor mendapatkan skor kepercayaan tertinggi hasil deteksi.

Nomor identitas ini akan tetap dijaga (dilacak) dari *frame* saat awal deteksi hingga *frame* di mana objek tersebut tidak muncul. Sebagaimana ditunjukkan oleh Gambar 3, objek mobil yang terdeteksi dengan nomor urut 0 muncul di *frame* pertama dan tetap terjaga hingga *frame* 20 hingga 97 dan tidak muncul lagi setelah *frame* berikutnya. Sedangkan objek mobil dengan nomor identitas 1 karena berbeda kecepatan dengan mobil 0 nomor identitas tetap dijaga hingga *frame* lebih dari 100

dimana objek tersebut sudah tidak muncul lagi. Nomor identitas akan tetap dibangkitkan untuk setiap objek baru yang terdeteksi di setiap kelas kendaraan dan bersifat unik (tidak berulang) hingga *frame* akhir dari *video*.

Untuk menguji kemampuan sistem melakukan deteksi dan penghitungan kendaraan, dilakukan pengujian dengan memberi input *video* yang diambil berdasarkan kriteria: tinggi dan sudut kamera terhadap area objek, intensitas cahaya dan kepadatan trafik. *Video* input diambil di jalan raya jalur masuk dan keluar Kota Malang. Lokasi pengambilan *video* mempertimbangkan kepadatan trafik dan mewakili seluruh jenis kendaraan yang akan dideteksi dan dihitung. Waktu pengambilan *video* dilakukan pada jam 6-8 pagi, jam 12 – 13 siang, 16 – 17 sore dan 19 – 20 malam. Hasil deteksi dan penghitungan kendaraan dapat dilihat pada *capture* Gambar 4 serta Tabel 1 dan 2.

Gambar 4 menunjukkan hasil deteksi dan penghitungan jumlah kendaraan berupa mobil, sepeda motor, bus, truk dan totak kendaraan keseluruhan yang merupakan sampel dari pengambilan *video* pada waktu siang, sore dan malam hari. Kecepatan proses deteksi dan penghitungan kendaraan sangat ditentukan oleh spesifikasi hardware komputer yang digunakan terutama RAM dan GPU. Masing-masing kendaraan yang dideteksi dan diklasifikasi pada Gambar 4 yang ditandai dengan adanya *bounding box*, label kelas kendaraan dan skor kepercayaan.

Untuk menghindari *overfitting pre-trained networks* telah dilatih menggunakan *dropout* dan augmentasi data ekstensif. *Layer dropout* dengan laju 0,5 berada setelah *layer* koneksi pertama untuk mencegah koadaptasi di antara *layer-layer*nya. Untuk augmentasi data diperkenalkan penskalaan acak dan translasi hingga 20% dari gambar asli dan juga penyesuaian eksposur dan saturasi gambar hingga faktor 1:5 dalam warna HSV (Redmon *dkk.*, 2016).



Gambar 4. Contoh hasil deteksi dan penghitungan jumlah kendaraan

Confusion matrix digunakan sebagai basis untuk mengukur performa pengujian model terhadap dataset. Confusion matrix memberikan informasi perbandingan hasil klasifikasi yang dilakukan oleh sistem (model) dengan hasil klasifikasi sebenarnya. Parameter performa model adalah akurasi, presisi dan recall. Akurasi merupakan rasio prediksi benar (positif dan negatif) dengan keseluruhan data. Akurasi menggambarkan berapa persen kelas objek yang benar diprediksi dan tidak benar dari keseluruhan objek. Presisi merupakan rasio prediksi benar positif dibandingkan dengan keseluruhan hasil yang diprediksi positif. Presisi menggambarkan berapa persen kelas objek yang benar dari keseluruhan objek yang diprediksi. Recall merupakan rasio prediksi benar positif dibandingkan dengan keseluruhan data yang benar positif. Recall menggambarkan berapa persen kelas objek yang benar dibandingkan keseluruhan objek yang sebenarnya.

Untuk mendapatkan parameter tersebut, maka perlu diketahui nilai prediksi dan aktual data. Nilai tersebut adalah True Positive (TP) merupakan data positif yang diprediksi benar. True Negative (TN) merupakan data negatif yang diprediksi benar. False Positive (FP) merupakan data negatif namun diprediksi sebagai data positif. False Negative (FN) merupakan

data positif namun diprediksi sebagai data negatif. Terdapat dua dataset yang telah dibuat untuk validasi dataset, yaitu: 1) posisi kamera 1, dataset video yang diambil oleh kamera dengan ketinggian 6 meter di atas permukaan jalan dengan sudut kamera terhadap tanah sebesar 50°, dan 2) posisi kamera 2, dataset video yang diambil oleh kamera dengan ketinggian 1.5 meter di atas permukaan jalan dan sudut kamera terhadap tanah sebesar 90°. Dataset pertama memiliki total data 520 buah untuk pengambilan data siang hari, sedangkan pada malam hari sejumlah 420 buah. Dataset kedua memiliki total data sejumlah 539 buah untuk pengambilan data siang hari, sedangkan malam hari sejumlah 431 buah

Untuk mendapatkan nilai akurasi, presisi dan recall dari confusion matrix pada Tabel 2, maka dapat dihitung menggunakan persamaan 2 – 6.

$$\text{Akurasi total} = \frac{TP}{\text{Total dataset}} \tag{2}$$

$$\text{Presisi} = \frac{TP}{TP+FP} \tag{3}$$

$$\text{Presisi Total} = \frac{\sum_{i=1}^N \text{Presisi}(i)}{\sum \text{kelas}} \tag{4}$$

$$\text{Recall} = \frac{TP}{TP+FN} \tag{5}$$

Tabel 1. Dataset pengujian

		Objek Kendaraan			
		Mobil	Sepeda Motor	Bus	Truk
Posisi kamera 1	Siang	135	285	40	60
	Malam	124	237	34	25
Posisi Kamera 2	Siang	135	316	46	42
	Malam	118	235	40	38

Tabel 2. Confusion matrix hasil deteksi dan penghitungan pada tinggi kamera 6 m dan sudut 50°

		Prediksi			
		Mobil	Sepeda Motor	Bus	Truk
Aktual	Mobil	130	0	5	0
	Sepeda Motor	0	229	0	0
	Bus	0	0	34	6
	Truk	0	0	4	56

Tabel 3. *Confusion matrix* hasil deteksi dan penghitungan pada tinggi kamera 6 m dan sudut 50⁰ malam hari

		Prediksi			
		Mobil	Sepeda Motor	Bus	Truk
Aktual	Mobil	112	0	12	0
	Sepeda Motor	0	138	0	0
	Bus	3	0	21	10
	Truk	5	0	5	15

$$\text{Recall Total} = \frac{\sum_{i=1}^N \text{Recali}(i)}{\sum \text{kelas}} \quad (6)$$

Berdasarkan *confusion matrix* pada Tabel 1, maka nilai akurasi, presisi dan *recall* adalah:

$$\begin{aligned} \text{Akurasi} &= \frac{130+229+34+56}{520} = 0,86 = 86\% \\ \text{Presisi(Mobil)} &= \frac{130}{130+5} = 0,96 \\ \text{Presisi(Sepeda Motor)} &= \frac{229}{229+0} = 1 \\ \text{Presisi(Bus)} &= \frac{34}{34+6} = 0,85 \\ \text{Presisi(Truk)} &= \frac{56}{56+4} = 0,93 \\ \text{Presisi Total} &= \frac{0,96+1+0,85+0,93}{4} = 0,93 = 93\% \\ \text{Recall(Mobil)} &= \frac{130}{130+0} = 1 \\ \text{Recall(Sepeda Motor)} &= \frac{229}{229+0} = 1 \\ \text{Recall(Bus)} &= \frac{34}{34+4} = 0,89 \\ \text{Recall(Truk)} &= \frac{56}{56+6} = 0,90 \\ \text{Recall Total} &= \frac{1+1+0,89+0,90}{4} = 0,94 = 94\% \end{aligned}$$

Berdasarkan *confusion matrix* pada Tabel 3, maka nilai akurasi total, presisi total dan *recall* total masing-masing adalah sebesar 68%, 83% dan 85%.

Tabel 5. *Confusion matrix* hasil deteksi dan penghitungan pada tinggi kamera 1.5 m dan sudut 90⁰ malam hari

		Prediksi			
		Mobil	Sepeda Motor	Bus	Truk
Aktual	Mobil	112	0	6	0
	Sepeda Motor	0	129	0	0
	Bus	12	0	16	12
	Truk	5	0	14	19

Tabel 4. *Confusion matrix* hasil deteksi dan penghitungan pada tinggi kamera 1.5 m dan sudut 90⁰ siang hari

		Prediksi			
		Mobil	Sepeda Motor	Bus	Truk
Aktual	Mobil	129	0	6	0
	Sepeda Motor	0	233	0	0
	Bus	3	0	31	12
	Truk	5	0	11	26

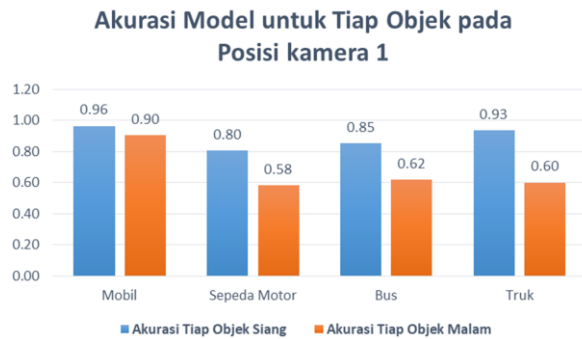
Berdasarkan *confusion matrix* pada Tabel 4, maka nilai akurasi total, presisi total dan *recall* total masing-masing adalah sebesar 77%, 84% dan 85%. Hasil pengujian sistem dirangkum dalam Tabel 6. Hasil pengujian pada Tabel 2 dan 3 adalah deteksi dan penghitungan jumlah kendaraan dari input *video* yang diambil oleh kamera dengan ketinggian 6 meter di atas permukaan jalan dan sudut kamera terhadap tanah sebesar 50⁰ pada waktu siang dan malam hari. Pada ketinggian dan sudut tersebut objek berada di bawah kamera sehingga terlihat bagian depan/belakang dan atas. Sedangkan hasil pengujian pada Tabel 4 dan 5 adalah deteksi dan penghitungan jumlah kendaraan dari input *video* yang diambil oleh kamera dengan ketinggian 1.5 meter di atas permukaan jalan dan sudut kamera terhadap tanah sebesar 90⁰ pada waktu siang dan malam. Pada ketinggian dan sudut tersebut objek berada di depan kamera sehingga terlihat bagian depan/belakang dan samping. Resolusi ketika melakukan pengambilan *video* adalah sebesar 1080 pixel/30fps.

Berdasarkan *confusion matrix* pada Tabel 5, maka nilai akurasi total, presisi total dan *recall* total masing-masing adalah sebesar 66%, 77% dan 78%. Performa pengujian model terangkum pada tabel 6. Evaluasi performa model meliputi akurasi total, presisi total dan *recall* total.

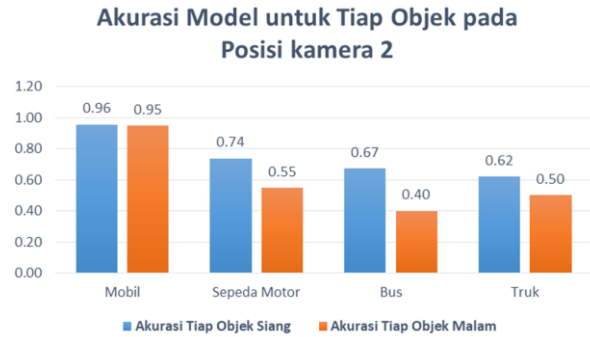
Sebagaimana diperlihatkan oleh Tabel 6, akurasi, presisi dan *recall* total terbaik didapat ketika

Tabel 6. Performa pengujian model

		Akurasi total (%)	Presisi total (%)	Recall total (%)
Posisi kamera 1	Siang	83	93	94
	Malam	68	83	85
Posisi Kamera 2	Siang	77	84	85
	Malam	68	77	78



Gambar 5. Akurasi deteksi tiap objek pada tinggi kamera 6 m dan susut 50°



Gambar 6. Akurasi deteksi tiap objek pada tinggi kamera 1.5 m dan susut 90°

pengujian model dilakukan pada siang hari dimana posisi kamera pada ketinggian 6 m dan susut 50°. Sedangkan akurasi, presisi dan *recall* total terendah didapat ketika pengujian model dilakukan pada malam hari dimana posisi kamera pada ketinggian 1,5 m dan susut 90°.

Dari Tabel 6 dan Gambar 5 dapat dinyatakan bahwa pada kondisi cerah akurasi tertinggi didapat pada saat model mendeteksi objek mobil sebesar 96% dan akurasi terendah untuk objek bus sebesar 85%. Sedangkan pada kondisi malam akurasi terendah didapat ketika model mendeteksi objek sepeda motor sebesar 58%. Pada Gambar 6 ketika kondisi cerah akurasi tertinggi didapat pada saat model mendeteksi objek mobil sebesar 96% dan akurasi terendah untuk objek truk sebesar 62%. Sedangkan pada kondisi malam akurasi terbaik didapat ketika model mendeteksi objek mobil sebesar 95% dan terendah ketika mendeteksi objek bus sebesar 40%. Pada saat pengambilan *video* dengan ketinggian 6 meter dan sudut 50° terhadap permukaan tanah dan kondisi cerah, deteksi terhadap objek mobil, truk dan sepeda motor tidak bermasalah karena sesuai dengan *bounding box*. Namun ketika mendeteksi bus, objek terlihat lebih panjang dan *bounding box* tidak sesuai dengan ukuran objek, sehingga deteksi menjadi bias antara bus atau truk.

Pengambilan *video* dengan ketinggian 1.5 meter dan sudut 90° terhadap permukaan tanah dan kondisi cerah, deteksi terhadap objek mobil, truk, bus dan sepeda motor tidak bermasalah karena permukaan objek sesuai dengan *bounding box*. Akurasi deteksi bermasalah ketika ukuran bus atau truk lebih besar dari rata-rata ukuran normal. Pada saat malam hari deteksi objek lebih banyak bermasalah ketika mendeteksi objek sepeda motor, hal ini disebabkan oleh bentuk objek yang terlihat samar dan gelap sehingga seringkali objek tidak terdeteksi.

4. Kesimpulan

Penelitian ini telah mengembangkan penggunaan detektor objek berbasis Yolov4 untuk mendeteksi objek kendaraan. Penelitian mengusulkan penghitungan jumlah kendaraan dengan membangkitkan nomor urut identitas unik untuk tiap kendaraan. Penghitungan jumlah masing-masing kendaraan telah dapat dilakukan berupa: mobil, sepeda motor, bus dan truk. Akurasi deteksi dihasilkan berdasarkan hasil pengukuran dan analisis *video* input yang didapat dari hasil perekaman kamera yang ditempatkan pada ketinggian dan sudut yang berbeda. Pengukuran juga dilakukan pada kondisi siang/ cerah dan malam hari. Kekurangan dari penelitian adalah tidak adanya *bounding area* untuk membatasi area perhitungan dan menghindari objek kendaraan di luar area agar tidak terdeteksi dan terhitung. Pengujian model belum dilakukan secara *real-time*. Klasifikasi jenis kendaraan belum didasarkan pada sumbu kendaraan sebagaimana klasifikasi jenis kendaraan yang ditetapkan oleh dinas perhubungan.

Ucapan Terimakasih

Penulis menyampaikan ucapan terima kasih kepada Kementerian Riset dan Teknologi/Badan Riset dan Inovasi Nasional yang telah membiayai penelitian ini dengan Nomor Kontrak 8/E1/KPT/2020. Ucapan terima kasih juga disampaikan kepada LPPM Universitas Widyagama Malang yang telah memfasilitasi penelitian ini.

Daftar Pustaka

Adistya, R., & Muslim, M. A. (2016). Deteksi Dan Klasifikasi Kendaraan Menggunakan Algoritma Backpropagation Dan Sobel. *Journal of Mechanical Engineering and Mechatronics* 1(02), 282312.

- Antony, J. J., & Suchetha, M (2016). Vision Based Vehicle Detection: A Literature Review. *International Journal of Applied Engineering Research* 11(5):3128–33.
- Badan Pusat Statistik (n.d.). Perkembangan Jumlah Kendaraan Bermotor Menurut Jenis, 1949-2018. Retrieved November 23, 2020 (<https://www.bps.go.id/linkTableDinamis/view/id/1133>).
- Balid, Walid, Tafish, H., & Refai, H. H. (2017). Intelligent Vehicle Counting and Classification Sensor for Real-Time Traffic Surveillance. *IEEE Transactions on Intelligent Transportation Systems* 19(6):1784–94.
- Bochkovskiy, A., Wang, C., & Liao, H. M. (2020). YOLOv4: Optimal Speed and Accuracy of Object Detection. *ArXiv:2004.10934 [Cs, Eess]*.
- Gomaa, A., Abdelwahab, M. M., Abo-Zahhad, M., Minematsu, T., & Taniguchi, R. (2019). Robust Vehicle Detection and Counting Algorithm Employing a Convolution Neural Network and Optical Flow. *Sensors* 19(20):4588.
- Kim, J., Sung, J., & Park, S. (2020). Comparison of Faster-RCNN, YOLO, and SSD for Real-Time Vehicle Type Recognition. *2020 IEEE International Conference on Consumer Electronics-Asia (ICCE-Asia)*. 1–4
- Mahto, Pooja and Garg, Priyamm and Seth, Pranav and Panda, J, Refining Yolov4 for Vehicle Detection (June 16, 2020). *International Journal of Advanced Research in Engineering and Technology (IJARET)*, 11(5), 2020, pp. 409-419., Available at SSRN: <https://ssrn.com/abstract=3628439>
- Manajang, D., Sompie, S. R. U. A., & Jacobus, A.(2020). Implementasi Framework Tensorflow Object Detection API Dalam Mengklasifikasi Jenis Kendaraan Bermotor. *Jurnal Teknik Informatika* 15(3):171–78.
- Mashudi, A., Rofii, F., & Mukhsim, M. (2020). Sistem Kamera Cerdas Untuk Deteksi Pelanggaran Marka Jalan. *JASEE Journal of Application and Science on Electrical Engineering* 1(01):15–25.
- Rachmawati, F., & Widhyaestoeti, D. (2020) Deteksi Jumlah Kendaraan Di Jalur SSA Kota Bogor Menggunakan Algoritma Deep Learning YOLO. *Prosiding LPPM UIKA BOGOR*.
- Redmon, J., Divvala, S., Girshick, R., & Farhadi, A.(2016). You Only Look Once: Unified, Real-Time Object Detection. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 779–88.
- Redmon, J. (2016). Darknet: open source neural networks in C. 2013–2016. URL <http://pjreddie.com/darknet>.
- Setyawan, G. E., Adiwijaya, B., & Fitriyah, H. (2019). Sistem Deteksi Jumlah, Jenis Dan Kecepatan Kendaraan Menggunakan Analisa Blob Berbasis Raspberry Pi. *Jurnal Teknologi Informasi Dan Ilmu Komputer (JTIIK)* 6(2).
- Wang, H., Yu, Y., Cai, Y., Chen, Xi., Chen, L., & Liu, Q.(2019). A Comparative Study of State-of-the-Art Deep Learning Algorithms for Vehicle Detection. *IEEE Intelligent Transportation Systems Magazine* 11(2):82–95.
- Wibowo, D. W., Muslim, M. A., & Sarosa, M. (2014). Perhitungan Jumlah Dan Jenis Kendaraan Menggunakan Metode Fuzzy C-Means Dan Segmentasi Deteksi Tepi Canny. *Jurnal EECCIS* 7(2):103–10.
- Yuniarto, A. (2008). Deteksi Kepadatan Lalu Lintas Menggunakan Sensor Ultrasonik Pada Persimpangan Jalan Berbasis Mikrokontroler. *PhD Thesis*, Universitas Muhammadiyah Surakarta.