

Implementasi *Framework Tensorflow Object Detection* Dalam Mengklasifikasi Jenis Kendaraan Bermotor

Dufan J. P. Manajang ¹⁾, Sherwin R.U.A. Sompie ²⁾, Agustinus Jacobus ³⁾
Jurusan Teknik Elektro, Universitas Sam Ratulangi Manado, Jl. Kampus Bahu, 95115, Indonesia
joiepatsy@gmail.com ¹⁾, aldo@unsrat.ac.id ²⁾, a.jacobus@unsrat.ac.id ³⁾
Diterima: 3 Agustus 2020; direvisi: 1 September 2020; disetujui: 15 September 2020

Abstract — *The type of vehicle that passes a road traffic is very influential on the density of road traffic. If the number of each type of vehicle that passes a road traffic can be known, then a good traffic strategy can be arranged by the authorities. By utilizing technological developments in the field of computer vision and deep learning, this study aims to detect and count vehicle objects that pass through road traffic based on the classification of vehicle types. This study uses the Tensorflow Object Detection API Framework and the algorithm of the YOLO v3 pretrained object detection model to make prediction and classification. Based on the test results on the five test videos, the system that was built successfully produced a data that contained the number of each type of vehicle passing on a crossroads with an average object detection accuracy level of 90.8%.*

Keywords — *Deep Learning, Object Detection, Tensorflow; YOLO.*

Abstrak — Jenis kendaraan yang melewati suatu lalu lintas jalan sangat berpengaruh terhadap kepadatan lalu lintas jalan tersebut. Jika jumlah setiap jenis kendaraan yang melewati suatu lalu lintas jalan dapat diketahui, maka strategi lalu lintas yang baik dapat diatur oleh pihak yang berwenang. Dengan memanfaatkan perkembangan teknologi di dalam bidang *computer vision* dan *deep learning*, penelitian ini bertujuan untuk mendeteksi dan menghitung objek kendaraan yang melewati suatu lalu lintas jalan berdasarkan klasifikasi jenis kendaraan. Penelitian ini menggunakan *Framework Tensorflow Object Detection API* dan algoritma dari *pretrained model object detection YOLO v3* untuk melakukan deteksi dan klasifikasi. Berdasarkan hasil pengujian pada kelima video uji coba, sistem yang dibangun berhasil menghasilkan suatu data yang berisi jumlah setiap jenis kendaraan yang lewat di suatu lintas jalan dengan tingkat akurasi deteksi objek rata-rata adalah 90.8%.

Kata kunci — *Deep Learning; Deteksi Objek; Tensorflow; YOLO.*

I. PENDAHULUAN

Lancarnya suatu lalu lintas dipengaruhi oleh beberapa faktor, antara lain luas jalan, jumlah kendaraan dan jenis kendaraan yang lewat [1]. Seiring berjalannya waktu, lalu lintas semakin dipenuhi dengan berbagai jenis kendaraan. Ditambah dengan luas jalan yang sudah tidak dapat menampung jumlah

kendaraan yang berlalu lintas dan strategi lalu lintas yang kurang baik membuat arus lalu lintas suatu jalan semakin padat dan kemacetan semakin berbondong-bondong [2]. Hal ini sangat berpengaruh besar di dalam kehidupan sehari-hari manusia, misalkan waktu yang terbuang cukup banyak.

Suatu hal yang dapat dilakukan untuk mengurangi kemacetan adalah mengatur strategi lalu lintas yang baik [3]. Jika terdapat data jenis kendaraan yang lewat di suatu jalan selama waktu tertentu, maka pihak yang berwenang dapat menganalisa dan mengambil keputusan untuk mengatur strategi lalu lintas yang baik. Misalkan jika di suatu jalan yang macet ternyata jumlah kendaraan mobil berjenis truk yang lewat memiliki kuantitas yang lebih banyak dibandingkan jenis kendaraan yang lain, maka pihak yang berwenang dapat memberikan jalur sendiri terhadap kendaraan mobil berjenis truk. Oleh karena itu, untuk mengatur strategi lalu lintas yang baik, maka pihak yang berwenang membutuhkan data yang berisi jumlah kendaraan yang lewat disuatu jalan berdasarkan jenis kendaraan. Pekerjaan menghitung jumlah kendaraan berdasarkan jenis kendaraan yang lewat di suatu jalan setiap hari merupakan pekerjaan yang tidak mudah bagi manusia, karena harus membutuhkan kesabaran dan ketelitian.

Dalam perkembangan zaman dan teknologi, *deep learning* semakin berkembang dengan sangat pesat di dalam menciptakan program-program *artificial intelligence*. *Deep learning* menggunakan *artificial neural network* yang membuat *deep learning* mampu untuk terus melatih diri sendiri dalam mengenali pola yang benar berdasarkan data atau input yang diberikan. Mulai dari *face detection*, *face recognition*, *object detection* dan *object recognition* mulai diciptakan menggunakan *deep learning*. Seiring dengan berjalannya waktu, *framework-framework deep learning* pun mulai dibuat. *Tensorflow* merupakan salah satu *framework deep learning* yang dapat digunakan untuk membuat berbagai program *artificial intelligence*. Oleh karena itu, dengan memanfaatkan *framework deep learning*, yaitu *tensorflow*, maka dapat memungkinkan untuk membuat sebuah sistem deteksi objek dalam menghitung jumlah kendaraan yang lewat di suatu jalan berdasarkan klasifikasi jenis kendaraan

Dari latar belakang penelitian ini, maka dibangunlah sebuah sistem atau aplikasi untuk mendeteksi dan menghitung jumlah kendaraan yang lewat di suatu jalan berdasarkan klasifikasi jenis kendaraan menggunakan *framework tensorflow object detection API*. Klasifikasi jenis kendaraan yang akan dideteksi antara lain, jenis kendaraan mobil, truk, bus dan sepeda motor. Data yang dihasilkan berupa jumlah kendaraan yang lewat di suatu jalan berdasarkan klasifikasi jenis kendaraan, di mana data tersebut dapat digunakan oleh pihak yang berwenang untuk mengatur strategi lalu lintas yang baik. Adapun topik yang di angkat adalah “Implementasi *Framework Tensorflow Object Detection API* Dalam Mengklasifikasi Jenis Kendaraan Bermotor”.

A. Penelitian Terkait

Terdapat beberapa penelitian yang terkait dengan penelitian saat ini mengenai implementasi deteksi objek untuk mendeteksi dan mengklasifikasi objek kendaraan, serta menghitung jumlah objek kendaraan yang melewati suatu jalan.

- 1) Penelitian oleh Lazaro, Buliali dan Amaliah pada tahun 2017 mengenai Deteksi Jenis Kendaraan di Jalan Menggunakan OpenCV. Penelitian ini bertujuan untuk menghitung jumlah kendaraan berdasarkan jenis kendaraan yang melewati suatu lalu lintas menggunakan library OpenCV dengan tingkat akurasi rata-rata 77.8% untuk kondisi jalan sepi, 47.5% untuk kondisi jalan normal dan 28.2% untuk kondisi jalan padat [4].
- 2) Penelitian oleh Janrao, Gupta dan Chandwani pada tahun 2017 mengenai *Real Time Traffic Density Count Using Image Processing*. Penelitian ini bertujuan untuk mengestimasi kepadatan lalu lintas di suatu jalan menggunakan *image processing* [5].
- 3) Penelitian oleh Kautsar dan Adi pada tahun 2016 mengenai Implementasi *Object Tracking* Untuk Mendeteksi dan Menghitung Jumlah Kendaraan Secara Otomatis Menggunakan Metode *Kalman Filter* dan *Gaussian Mixture Model*. Penelitian ini bertujuan untuk menghitung jumlah kendaraan di suatu lalu lintas. Penelitian ini memiliki hasil yang akurat pada pagi hari, namun sangat tidak akurat pada malam hari [6].
- 4) Penelitian oleh Tobi pada tahun 2015 mengenai Rancang Bangun Purwarupa Sistem Pendeteksi Kendaraan Menggunakan Pustaka *OpenCV*. Penelitian ini menghasilkan jumlah dari kendaraan yang terdeteksi dengan tingkat *error* 11%, kecepatan kendaraan yang lewat dengan tingkat *error* 2.44% sampai 4%, serta indeks kendaraan dengan tingkat *error* 26.7% [7].
- 5) Penelitian oleh Purnomo, Cholissodin dan Utamingrum pada tahun 2018 mengenai *Intellegence Vehicle Counting* Menggunakan Metode *Combination Value Saturation* pada Video Lalu Lintas. Penelitian ini bertujuan untuk menghitung jumlah kendaraan yang ada di suatu jalan. Telah disimpulkan bahwa penelitian ini mampu mendeteksi, serta menghitung jumlah kendaraan yang lewat di suatu lalu lintas jalan. Penelitian ini menggunakan metode *combination value saturation*

dengan tingkat akurasi 68.38% dari 2 video yang sama, namun memiliki ukuran aspek rasio yang berbeda berdurasi 33 detik [8].

B. Convolutional Neural Networks

Artificial neural networks (ANNs) atau biasa disebut dengan jaringan saraf tiruan adalah sebuah sistem yang digunakan untuk proses komputasi yang terinspirasi dari sistem saraf biologis, seperti otak manusia. *Artificial neural networks* (ANNs) terdiri dari sejumlah besar node komputasi yang saling berhubungan (*neuron*) yang secara kolektif belajar dari *input* untuk mengoptimalkan hasil akhir. Cara kerja *convolutional neural networks* (CNNs) sama dengan tradisional *artificial neural networks* di mana keduanya terdiri dari sekumpulan *neuron* yang mengoptimalkan diri dari pembelajaran. Mulai dari *input* awal sampai pada hasil akhir, setiap jaringan akan tetap mengekspresikan sebuah skor atau yang disebut dengan *weight* dan pada *layer* terakhir terdapat fungsi kehilangan (*loss functions*) yang terkait dengan kelas yang ada [9].

C. Deep Learning

Teknologi *Machine Learning* telah sangat banyak membantu dalam perkembangan dan kemajuan berbagai aspek komunitas *modern* saat ini, seperti pencarian *web* atau pun penyaringan konten dalam sosial media. *Machine learning* juga membantu dalam pengembangan produk-produk saat ini, seperti kamera, smartphone dan lain sebagainya. Sistem *machine learning* digunakan untuk mengidentifikasi objek dalam gambar, mengubah suara ke dalam bentuk teks, menampilkan hasil yang relevan terhadap pencarian yang dilakukan dan masih banyak lagi. Seiring dengan berjalannya waktu, aplikasi-aplikasi *machine learning* tersebut memanfaatkan teknik yang disebut *deep learning*. *Conventional machine learning* memiliki keterbatasan dalam kemampuannya untuk memproses data-data dalam bentuk mentah. Membangun sebuah sistem *machine learning* membutuhkan keahlian dan rekayasa yang cermat dalam membuat fitur ekstraksi untuk mengubah data yang mentah menjadi suatu representasi yang diperlukan. *Representation learning* memungkinkan mesin dapat secara otomatis mengolah data mentah untuk menemukan representasi yang diperlukan dalam melakukan deteksi atau klasifikasi. Metode *deep learning* juga dapat disebut sebagai metode *representation learning* yang memiliki beberapa tingkat representasi. *Deep learning* memungkinkan model komputasi yang terdiri dari beberapa lapisan proses pengolahan untuk dapat mempelajari representasi data melalui beberapa tingkat abstraksi. Metode ini secara drastis meningkatkan pengembangan dalam bidang pengenalan suara (*voice recognition*), pengenalan objek visual (*image recognition*), deteksi objek (*object detection*) dan begitu banyak hal lainnya seperti dalam penemuan obat dan dalam bidang genomika. Saat ini *Deep learning* telah membawa terobosan dalam berbagai bidang *artificial intelligence*. *Deep learning* dapat menemukan struktur yang begitu rumit di dalam sebuah dataset yang sangat besar dengan menggunakan algoritma *backpropagation* [10].

TABEL I VIDEO UJI COBA

Nama File	Durasi	Total Frame	Jumlah Kendaraan				Jumlah
			Mobil	Truk	Bus	S.Motor	
Video1	00:50	1500	15	10	2	10	37
Video2	01:00	1807	14	10	2	21	47
Video3	00:10	250	4	1	1	14	20
Video4	00:20	481	10	1	0	15	26
Video5	00:15	360	8	2	0	5	15

D. Computer Vision

Computer Vision adalah suatu ilmu yang mempelajari tentang cara pandang komputer terhadap objek disekelilingnya dengan mampu menganalisanya. *Computer Vision* merupakan gabungan dari *digital image processing*, *pattern recognition*, *computer graphics* dan *machine learning*. *Computer Vision* berada dalam naungan ilmu computer yang mencakup algoritma, pemrosesan data dan grafik, ilmu fisika yang mencakup optik dan sensor, ilmu matematika yang mencakup kalkulus dan teori informasi, serta ilmu biologi yang mencakup visual stimuli dan *neural processing*. *Computer vision* juga merupakan kombinasi antara sistem pencahayaan (*lightning system*) dan analisa citra/gambar (*image analysis*) [11]. Otak manusia dapat mengenali wajah sesama manusia meskipun baru pertama kali bertemu dan dapat mengenali objek dengan cepat. Bagi komputer, sebuah gambar hanya merupakan sekumpulan pixel. *Computer Vision* bertujuan untuk mengenali sebuah gambar dengan baik, sehingga dapat menghasilkan informasi-informasi yang penting di dalam sebuah gambar. Dengan kata lain tujuan *Computer Vision* adalah untuk mengajarkan komputer bisa membuat *pixel-pixel* menjadi hidup dan terasa seperti di dunia nyata [12].

E. Tensorflow

Tensorflow merupakan salah satu *framework deep learning* dan juga salah satu *library* untuk *data science* yang bersifat *free open source* yang dikembangkan oleh para peneliti dari tim Google. *Tensorflow* dapat digunakan dalam berbagai bidang. Dalam bidang *object detection* terdapat *framework tensorflow object detection API* yang merupakan suatu alat yang dapat digunakan untuk mempermudah proses *constructing*, *training* dan *deployment* pada suatu model *object detection*. *Framework tensorflow object detection API* menyediakan *pretrained object detection* model bagi *user*, namun memungkinkan jika *user* ingin menggunakan *pretrained object detection* model yang lain, seperti *Faster R-CNN*, *SSD*, *Retinanet*, *Resnet50* dan masih banyak lagi.

F. You Only Look Once v3

Terinovasi dari ide-ide para komunitas penelitian *computer vision*, *YOLO v3* merupakan versi ketiga dari algoritma *object detection YOLO* yang mulai diperkenalkan pada tahun 2015 oleh Joseph Redmon.

TABEL II BENTUK OBJEK KENDARAAN

Jenis Kendaraan	Contoh Gambar
Mobil	
Truk	
Bus	
Sepeda Motor	

Dengan menerapkan jaringan saraf tunggal, algoritma dari *YOLO v3* akan membagi suatu gambar ke dalam bentuk *grid* dengan ukuran yang telah ditentukan dan untuk setiap *cell* akan dilakukan prediksi tiga *bounding boxes* dengan *class probabilities* Namun, *YOLO v3* akan melakukan prediksi *bounding box* pada tiga ukuran/skala *grid* yang berbeda, untuk itu setiap *cell* pada akhirnya akan memprediksi sembilan *bounding boxes* dengan *class probabilities*. *Bounding boxes* yang memiliki *score* di atas batas nilai *confidence threshold* dan di atas batas nilai *IoU threshold* dikategorikan sebagai sebuah objek dengan sebuah *probability class* [13].

II. METODE

A. Data Yang Digunakan

Terdapat 5 *file* video yang berisi rekaman lalu lintas yang digunakan sebagai bahan untuk menguji sistem yang dibangun dan sebuah *pretrained weights* dari *YOLO v3* yang berisi hasil *training* terhadap objek-objek yang hendak dideteksi.

1) Video Uji Coba

Video uji coba yang pertama diambil dari perekaman langsung di jalan Raya Manado Tomohon dan video uji coba yang kedua diambil dari perekaman langsung di jalan Sam Ratulangi. Video uji coba yang ketiga didapatkan dari sebuah CCTV yang ada di jalan A. Yani Dolog dengan kualitas yang rendah untuk menguji kualitas dari video. Untuk video uji coba yang keempat dan video uji coba yang kelima didapatkan dari sebuah CCTV yang sama yang ada di jalan Piere Tendean pada waktu yang berbeda dalam rangka menguji tingkat kecahayaan dari video. Informasi dari masing-masing video uji coba dapat dilihat pada tabel I dan sudut pengambilan masing-masing video uji coba dapat dilihat pada gambar 1.



Gambar 1 Sudut Pengambilan Video Uji Coba

2) *Pretrained Weights*

Pretrained weights dari *YOLO v3* diunduh dari *website official YOLO*. *Pretrained weights* dari *YOLO v3* dilatih menggunakan *dataset* yang diambil dari *Google Open Images Dataset*, dalam arti *dataset* yang digunakan oleh *YOLO* bukan bersifat lokal, sehingga dapat dikatakan bahwa ada beberapa kendaraan khususnya pada jenis kendaraan mobil dan truk yang ada pada *dataset* yang digunakan oleh *YOLO* sedikit berbeda dengan biasa yang kita kenal dalam versi lokal. Bentuk-bentuk objek kendaraan mobil, truk, bus dan motor dapat dilihat pada tabel II.

B. *Rancangan Aplikasi*

Proses kerja dari sistem yang dibangun mulai dari pemrosesan *input* sampai pada *output* yang dihasilkan dapat dilihat pada *block diagram* yang ada pada gambar 2. Pada awalnya terdapat sebuah *input* yaitu berkas video yang berisi rekaman lalu lintas suatu jalan. Selanjutnya terdapat sebuah proses untuk membaca seluruh *frame* yang membangun *input* video dan untuk menandai *frame* yang hendak dilakukan proses deteksi. Untuk melakukan kedua hal tersebut, maka diperlukan sebuah *library OpenCV*. *Frame* yang telah ditandai untuk dilakukan proses deteksi pertama-tama dilakukan perubahan ukuran gambar sesuai dengan ukuran yang telah ditentukan dan kemudian dilakukan proses deteksi. *Library OpenCV* kembali digunakan untuk mengubah ukuran dari suatu *frame* dan untuk melakukan proses deteksi dibutuhkan *framework tensorflow object detection API* dan *pretrained model* dari *YOLO v3*. Proses deteksi menghasilkan prediksi *bounding boxes* pada setiap *frame* yang telah ditandai. Seluruh *bounding box* yang terprediksi kemudian dilakukan *confidence thresholding* dan *intersection over union thresholding* untuk menghasilkan *bounding box* yang bertanggung jawab terhadap objek sebenarnya yang ada di dalam *frame* atau dengan kata lain proses *thresholding* menghasilkan objek yang dinyatakan terdeteksi. Selanjutnya, objek yang dinyatakan terdeteksi dilakukan proses pemberian label. Dalam proses pemberian label digunakan beberapa tambahan *library*, seperti *library Pillow* dan *library Seaborn* untuk menggambar persegi pada objek yang dinyatakan terdeteksi dan untuk menambahkan teks

label pada setiap persegi. Pada akhirnya terdapat 3 proses, yaitu proses menghitung jumlah objek yang dinyatakan terdeteksi pada setiap *frame* yang telah ditandai, proses untuk menyimpan video hasil deteksi ke dalam bentuk berkas video yang baru dan proses untuk menyimpan *frame* yang telah ditandai yang telah dilakukan proses deteksi ke dalam bentuk berkas gambar. Proses menghitung jumlah objek yang telah dinyatakan terdeteksi menghasilkan jumlah kendaraan yang terdeteksi pada masing-masing kategori, di mana terdapat empat kategori yaitu, kategori kendaraan mobil, truk, bus dan sepeda motor. Hasil tersebut kemudian dibawa ke proses selanjutnya untuk ditambahkan dengan informasi terkait proses deteksi, seperti id deteksi, tanggal dan waktu deteksi, nama berkas video yang dideteksi, serta durasi dari berkas video yang dideteksi.

C. *Teknik Evaluasi*

Evaluasi di dalam penelitian ini merupakan suatu kegiatan untuk memeriksa kinerja dari sistem yang telah dibangun, di mana sistem yang dibangun di dalam penelitian ini mulai dari input sampai pada *output* yang dihasilkan hanya menggunakan pemrosesan CPU dengan prosesor AMD A9-9425. Sistem yang dibangun dipasang dalam sistem operasi windows 10 64-bit dengan 4GB kapasitas RAM.

Pada proses evaluasi dilakukan pengukuran secara manual terhadap kecepatan proses deteksi dan perhitungan jumlah kendaraan yang terdeteksi berdasarkan klasifikasi jenis kendaraan mobil, truk, bus dan sepeda motor. Di dalam proses evaluasi juga akan diukur tingkat akurasi deteksi objek dalam mendeteksi dan mengklasifikasi objek kendaraan. Untuk mengukur tingkat akurasi deteksi objek, maka akan dihitung berapa banyak jumlah objek kendaraan yang terdeteksi benar dari keseluruhan objek yang terdeteksi oleh sistem. Pengukuran tingkat akurasi deteksi objek digambarkan dalam bentuk tabel *confusion matrix 5x5*. Berikut merupakan parameter-parameter yang digunakan dalam pengukuran tingkat akurasi deteksi objek:

1) *True Positive (TP)*

True Positive atau biasa disingkat dengan TP merupakan objek sebenarnya yang terdeteksi benar oleh sistem

- 2) *False Positive (FP)*
False Positive atau biasa disingkat dengan FP merupakan noise yang terdeteksi oleh sistem sebagai objek
- 3) *False Negative (FN)*
False Negative atau biasa disingkat dengan FN merupakan objek sebenarnya yang tidak terdeteksi oleh sistem
- 4) *Accuracy*
Accuracy adalah hasil perhitungan tingkat akurasi *object detection* terhadap objek kendaraan pada keseluruhan *frame* yang dilakukan proses deteksi. Rumus dari perhitungan *accuracy* dapat dilihat pada (1).

$$Accuracy = \frac{\text{Jumlah Objek Yang Terdeteksi Benar}}{\text{Jumlah Keseluruhan Objek Yang Terdeteksi}} \quad (1)$$

- 5) *Precision*
Precision adalah jumlah prediksi yang benar dibandingkan keseluruhan hasil yang terprediksi oleh sistem. Dalam hal ini *precision* akan menjawab berapa jumlah objek yang terdeteksi benar dari keseluruhan jumlah objek yang terdeteksi oleh sistem. Rumus dari perhitungan *precision* dapat dilihat pada (2).

$$Precision = \frac{TP}{(TP+FP)} \quad (2)$$

- 6) *Recall*
Recall adalah jumlah prediksi yang benar dibandingkan dengan keseluruhan hasil yang sebenarnya. Dalam hal ini

recall akan menjawab berapa jumlah objek yang terdeteksi benar dari keseluruhan jumlah objek sebenarnya. Rumus dari perhitungan *recall* dapat dilihat pada (3).

$$Recall = \frac{TP}{(TP+FN)} \quad (3)$$

III. HASIL DAN PEMBAHASAN

A. Tampilan Tatap Muka Sistem

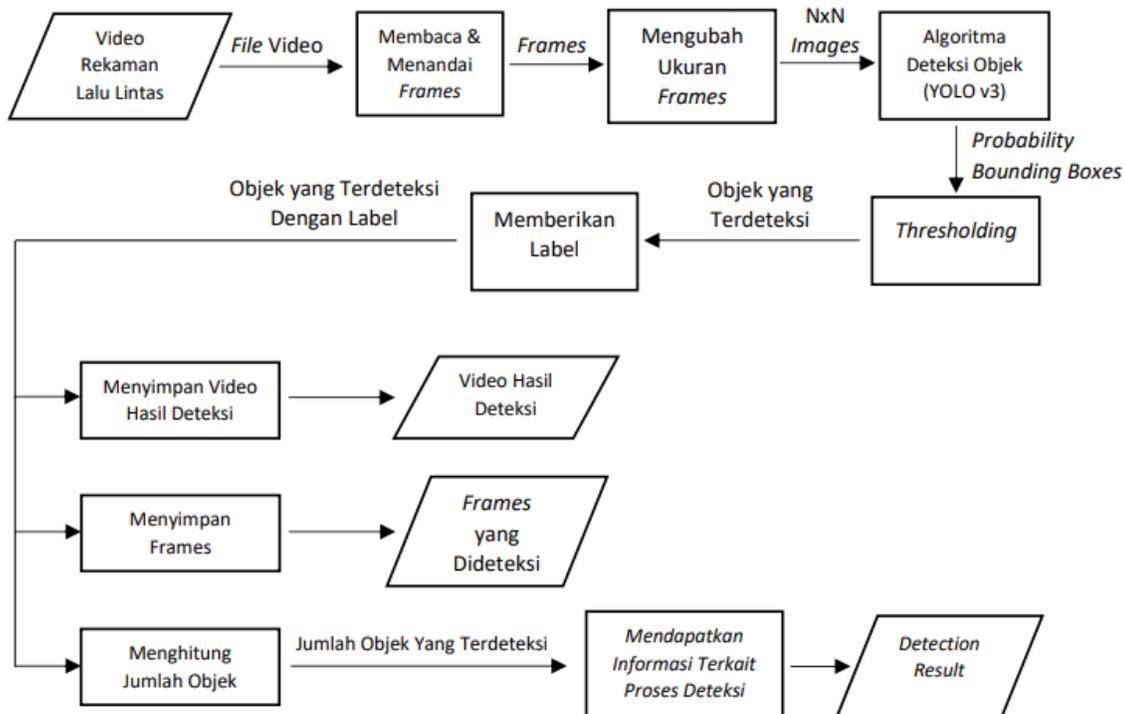
Tampilan tatap muka adalah tampilan visual dari suatu sistem untuk membuat *user* dapat berinteraksi dengan sistem dengan mudah. Tampilan tatap muka yang dibangun dalam penelitian ini berbasis web dan dibangun menggunakan *framework* Django. Tampilan halaman utama sistem dapat dilihat pada gambar 3, sedangkan tampilan halaman untuk memutar video hasil deteksi dan untuk melihat *frame* yang telah ditandai dapat dilihat pada gambar 4 dan 5.

B. Evaluasi Sistem

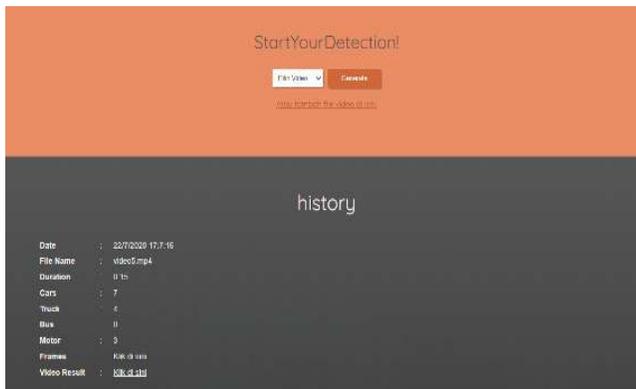
Berikut merupakan hasil evaluasi kecepatan proses deteksi, jumlah kendaraan yang terdeteksi, serta tingkat akurasi deteksi objek dalam mendeteksi dan mengklasifikasi objek kendaraan:

1) Kecepatan Proses Deteksi

Telah dilakukan pengujian sebanyak dua kali secara manual untuk menghitung kecepatan proses deteksi pada kelima *file* video uji coba dengan durasi dan total *frame* yang berbeda yang dapat dilihat pada tabel III. Hasil pengujian



Gambar 2 Diagram Blok Prinsip Kerja Sistem



Gambar 3 Antar Muka Halaman Utama



Gambar 4 Antar Muka Untuk Memutar Video Hasil Deteksi

kecepatan proses deteksi menyatakan bahwa semakin banyak *frame* yang akan dideteksi menunjukkan kecepatan proses deteksi lebih lama, namun kecepatan proses deteksi tidak menentu pada saat dilakukan dua kali pengujian pada *file* video yang sama.

2) Jumlah Kendaraan Yang Terdeteksi

Hasil perhitungan jumlah kendaraan yang terdeteksi pada kelima *file* video yang digunakan sebagai video uji coba berdasarkan klasifikasi mobil, truk, bus dan sepeda motor dapat dilihat dalam tabel IV. Setiap video uji coba memiliki tiga parameter pada masing-masing *class*, yaitu parameter yang menyimpan jumlah kendaraan sebenarnya yang ada di dalam video (*Act*), parameter yang menyimpan jumlah kendaraan yang ada di dalam seluruh *frame* yang dilakukan proses deteksi (*In*) dan parameter yang menyimpan jumlah kendaraan yang terdeteksi oleh sistem (*Pr*).

Konsistensi jumlah kendaraan sebenarnya yang ada di dalam video dan jumlah kendaraan yang ada di dalam *frame* pada tabel IV kurang efektif dikarenakan teknik dalam mendeteksi dan menghitung objek yang ada pada kelima *file* video tidak dilakukan pada seluruh *frame* di dalam video melainkan hanya dilakukan setiap interval 5 detik pada durasi video. Dengan demikian ada beberapa objek yang terlewatkan atau yang tidak berada di dalam *frame* dan bahkan ada beberapa objek yang sama yang sudah berada di dalam dua *frame*. Untuk itu, selanjutnya dilakukan perhitungan secara manual terhadap objek-objek yang tidak berada di dalam *frame* dan objek yang sama yang sudah berada di dalam dua *frame*, serta objek-objek yang tertutupi oleh objek yang lain yang membuat sistem deteksi tidak dapat mendeteksinya, hasil perhitungan tersebut dapat dilihat pada tabel V.

Berdasarkan hasil perhitungan jumlah kendaraan yang lewat dapat dikatakan bahwa pengambilan sudut dan kualitas rekaman video lalu lintas berpengaruh dalam perhitungan objek kendaraan. Hal ini dikarenakan jika pengambilan sudut terlalu rendah, maka akan terdapat objek kendaraan yang tertutupi oleh objek kendaraan yang lain dan jika kualitas rekaman video lalu lintas terlalu rendah maka akan ada objek yang tidak dapat terlihat jelas oleh sistem deteksi. Tingkat kecerahan cahaya juga berpengaruh dalam sistem deteksi di mana telah dilakukan pengujian terhadap *file* video4.mp4 pada waktu malam hari dan *file* video5.mp4 pada waktu siang hari, di mana *file* video4.mp4 dan *file* video5.mp4 memiliki lokasi

Gambar 5 Antar Muka Untuk Menampilkan *Frame* Yang Telah Ditandai

dan pengambilan sudut yang sama, namun pengujian tersebut memiliki hasil perhitungan objek yang rendah pada waktu malam hari, dalam hal ini khususnya objek kendaraan sepeda motor sulit untuk dideteksi oleh sistem.

3) Tingkat Akurasi Deteksi Objek

Hasil pengujian tingkat akurasi deteksi objek yang dilakukan pada kelima *file* video yang digunakan sebagai video uji coba dapat dilihat pada tabel VI. Rata-rata tingkat akurasi deteksi objek pada pengujian kelima *file* video uji coba adalah 90.8%. Nilai *overall accuracy* yang tinggi belum menjamin ketepatan dari algoritma deteksi objek yang digunakan dalam melakukan deteksi dan klasifikasi, hal ini dapat dilihat dari nilai *precision* dan nilai *recall* setiap *class*. Misalkan, meskipun nilai *overall accuracy* yang ada pada *file* video4.mp4 adalah 100%, namun *file* video4.mp4 memiliki nilai *recall* yang rendah pada masing-masing *class* dikarenakan ada beberapa objek yang tidak dapat terdeteksi oleh algoritma deteksi objek. Berdasarkan perhitungan sebelumnya mengenai jumlah kendaraan pada masing-masing *class* menyatakan bahwa algoritma deteksi objek sulit untuk mendeteksi objek yang ada pada video yang memiliki kualitas rendah atau yang memiliki tingkat kecahayaan yang kurang.

IV. KESIMPULAN DAN SARAN

A. Kesimpulan

Berdasarkan hasil penelitian dan pengujian terhadap sistem yang telah dibangun, maka diambil kesimpulan bahwa *Framework Tensorflow Object Detection API* dapat diimplementasikan untuk membangun sebuah sistem yang

dapat mendeteksi dan menghitung jumlah kendaraan berdasarkan jenis kendaraan yang melewati suatu jalan. Pengujian tingkat akurasi dalam mendeteksi objek kendaraan berdasarkan klasifikasi jenis kendaraan yang dilakukan pada kelima *file* video uji coba memiliki nilai rata-rata, yaitu 90.8% akurat dan perhitungan jumlah kendaraan berdasarkan klasifikasi jenis kendaraan yang melewati suatu jalan yang dilakukan setiap interval 5 detik di dalam video kurang efektif untuk diterapkan.

B. Saran

Beberapa saran penelitian untuk mengembangkan penelitian yang telah dilakukan, antara lain sistem yang telah

dibangun baru bisa mengklasifikasikan kendaraan bermotor dalam 4 kelas, sehingga masih dimungkinkan dilakukan penelitian untuk mengklasifikasikan jenis kendaraan dalam kelas yang lebih spesifik. *Pretrained model* yang digunakan dalam penelitian ini adalah pretrained model dari YOLO v3 dan pengklasifikasian kendaraan yang digunakan masih belum sesuai dengan klasifikasi kendaraan yang digunakan di Indonesia, sehingga diperlukan sebuah penelitian untuk membangun model objek kendaraan yang sesuai dengan klasifikasi kendaraan di Indonesia. Sistem masih dapat dikembangkan untuk mendeteksi kendaraan secara *real-time* dengan memanfaatkan kamera CCTV yang dipasang di jalan.

TABEL III KECEPATAN PROSES DETEKSI

Nama <i>File</i>	Durasi (m:d)	Total <i>Frame</i>	Kecepatan (m:d) (1)	Kecepatan (m:d) (2)
video1	00:50	1500	04:25	04:05
video2	01:00	1807	04:38	04:29
video3	00:10	250	01:36	01:19
video4	00:20	481	01:59	01:44
video5	00:15	360	01:43	01:31

TABEL IV HASIL PERHITUNGAN JUMLAH KENDARAAN

Nama <i>File</i>	Mobil			Truk			Bus			Sepeda Motor			<i>Overall</i>		
	Act	In	Pr	Act	In	Pr	Act	In	Pr	Act	In	Pr	Act	In	Pr
video1	15	5	5	10	8	7	2	2	2	10	1	1	37	16	15
video2	14	10	11	10	8	7	2	2	2	21	11	9	47	31	29
video3	4	3	2	1	1	2	1	1	1	14	9	0	20	14	5
video4	10	8	4	1	1	0	0	0	0	15	11	0	26	20	4
video5	8	7	7	2	2	4	0	0	0	5	3	3	15	12	14

TABEL V JUMLAH KENDARAAN YANG TIDAK ADA DALAM *FRAME*

Nama <i>File</i>	Keseluruhan Jumlah Objek	Objek Yang Ada Dalam <i>Frame</i>	Objek Yang Terulang	Objek Yang Tidak Ada Dalam <i>Frame</i>	Objek Yang Tersembunyi
video1	37	16	0	21	0
video2	47	31	1	15	2
video3	20	14	0	4	2
video4	26	20	0	6	0
video5	15	12	0	3	0

TABEL VI HASIL PENGUJIAN TINGKAT AKURASI DETEKSI OBJEK

Nama File	Mobil		Truk		Bus		Sepeda Motor		Overall Accuracy
	P	R	P	R	P	R	P	R	
video1	100%	100%	100%	87%	100%	100%	100%	100%	100%
video2	91%	100%	100%	87%	100%	100%	100%	81%	96%
video3	100%	66%	50%	100%	100%	100%	N/A	0%	80%
video4	100%	50%	N/A	0%	N/A	N/A	N/A	0%	100%
video5	100%	87%	25%	100%	N/A	N/A	100%	100%	78%

V.KUTIPAN

- [1] I. Wijanarko and M. A. Ridlo, "Faktor-Faktor Pendorong Penyebab Terjadinya Kemacetan Studi Kasus : Kawasan Sukun Banyumanik Kota Semarang," *J. Planol.*, vol. 14, no. 1, p. 63, 2019, doi: 10.30659/jpsa.v14i1.3859.
- [2] F. Marwan, "Analisis Dampak Kemacetan Lalu Lintas Dengan Pendekatan Willingness To Accept (Studi Kasus : Kecamatan Bogor Barat)," *Skripsi Fak. Ekon. dan Manaj. Inst. Pertan. Bogor*, 2011.
- [3] A. Alhadar, "Analisis Kinerja Jalan dalam Upaya Mengatasi Kemacetan Lalu Lintas pada Ruas Simpang Bersinyal di Kota Palu," *J. SMARTek, Nop. 2011*, vol. 9, no. 4, pp. 327–336, 2011.
- [4] A. Lazaro, J. L. Buliali, and B. Amaliah, "Deteksi Jenis Kendaraan di Jalan Menggunakan OpenCV," *J. Tek. ITS*, vol. 6, no. 2, 2017, doi: 10.12962/j23373539.v6i2.23175.
- [5] A. Janrao, "Real Time Traffic Density Count using Image Processing," vol. 162, no. 10, pp. 8–12, 2017.
- [6] H. Vazirani, A. Kautsar, J. Fisika, F. Sains, and U. Diponegoro, "IMPLEMENTASI OBJECT TRACKING UNTUK MENDETEKSI DAN MENGGUNAKAN METODE KALMAN FILTER DAN GAUSSIAN MIXTURE MODEL," vol. 5, no. 1, 2016.
- [7] M. D. Tobi, P. Katolik, and S. Paul, "RANCANG BANGUN PURWARUPA SISTEM PENDETEKSI KENDARAAN," no. July 2015, 2018, doi: 10.5281/zenodo.1249925.
- [8] G. A. Purnomo, I. Cholissodin, and F. Utaminingrum, "Intellegence Vehicle Counting Menggunakan Metode Combination Value Saturation Pada Video Lalu Lintas," vol. 2, no. 6, pp. 2192–2199, 2018.
- [9] K. O. Shea and R. Nash, "An Introduction to Convolutional Neural Networks," pp. 1–11.
- [10] Y. Lecun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015, doi: 10.1038/nature14539.
- [11] V. Wiley and T. Lucas, "Computer Vision and Image Processing : A Paper Review," vol. 2, no. 1, pp. 28–36, 2018, doi: 10.29099/ijair.v2i1.42.
- [12] B. Planche and E. Andres, *Hands-On Computer Vision with TensorFlow 2.*
- [13] J. Redmon, "YOLOv3: An Incremental Improvement."

melanjutkan pendidikan ke perguruan tinggi Universitas Sam Ratulangi Manado, dengan mengambil Program Studi S1 Teknik Informatika, Jurusan Elektro, Fakultas Teknik dan pada tahun 2020 berhasil meraih gelar Sarjana Komputer (S.Kom).

TENTANG PENULIS



Penulis bernama Dufan Joie Patsy Manajang yang merupakan anak pertama dari pasangan Stenly Manajang dan Imelda Lomboan, lahir di Manado pada tanggal 10 Mei 1999. Penulis mulai menempuh pendidikan di sekolah dasar SD Negeri 08 Manado (2005-2010) dan kemudian melanjutkan studi tingkat

pertama di SMP Katolik St. Rafael Manado (2010-2013). Selanjutnya penulis menempuh pendidikan tingkat atas di SMA Negeri 1 Manado (2013-2016). Pada tahun 2016 penulis