

Deteksi Kemacetan Menggunakan *YOLOv4* dan *Euclidean Distance Tracker* pada Jalan Raya

Traffic Detection with YOLOv4 and Euclidean Distance Tracker on Highways

Jeremia Manurung¹, Nur Azizi², Disty Anastasya³, Nicholas Valentino⁴, Aditia Sanjaya⁵, Kana Saputra⁶

^{1,2,3,4,5,6}Program Studi Ilmu Komputer, Universitas Negeri Medan, Medan, Indonesia

Email: 1jeremiamanurung@mhs.unimed.ac.id

*Penulis Koresponden

Diterima: 02 Januari 2023 | Direvisi: 03 Februari 2023 | Disetujui: 15 Februari 2023



This work is licensed under a [Creative Commons Attribution 4.0 International License](https://creativecommons.org/licenses/by/4.0/).
Copyright (c) 2023 JUSTINDO

ABSTRAK

Kemacetan lalu lintas di Kota Medan menyebabkan waktu yang hilang sebesar 35,6 menit per hari untuk sepeda motor dan 48,5 menit per hari untuk mobil. Total biaya kemacetan di Kota Medan mencapai Rp. 22.535.355.867/tahun. Dengan adanya pendeteksian kemacetan secara realtime maka diharapkan dapat mengurangi kemacetan lalu lintas apabila diintegrasikan dengan sistem pengatur lalu lintas. Penelitian ini menerapkan metode Deep Learning YOLO versi 4 Euclidean Distance Tracker. YOLOv4 digunakan untuk mendeteksi objek seperti mobil, motor, bus, dan becak. Euclidean Distance Tracker digunakan untuk melacak perpindahan objek yang telah dideteksi oleh YOLOv4. Adapun data yang digunakan adalah data lalu lintas berupa video dari CCTV yang disediakan oleh Pemerintah Kota Medan, Sumatera Utara. Dari hasil penelitian ini dapat diambil kesimpulan YOLOv4 dapat digunakan untuk mendeteksi kendaraan yang memiliki jarak kendaraan yang cukup antara kendaraan yang satu dengan kendaraan yang lainnya (Akurasi 61,3%). Dengan mengintegrasikan Euclidean Distance Tracker, pendeteksi kemacetan memiliki hasil akurasi maksimum (Akurasi 100%) pada sample frame yang diuji.

Kata kunci: *deep learning, deteksi kemacetan, euclidean distance, yolov4*

ABSTRACT

Traffic jam in Medan City causes lost time of 35.6 minutes per day for motorbikes and 48.5 minutes per day for cars. The total cost of congestion in Medan City reaches Rp. 22,535,355,867/year. With the detection of congestion in real time, it is hoped that it can reduce traffic jams when integrated with a traffic control system. This study applies the YOLO Deep Learning method version 4 to detect objects such as cars, motorcycles, buses, and pedicabs. The data used is traffic data in the form of video from CCTV provided by the City Government of Medan, North Sumatra. From the results of this study it can be concluded that YOLOv4 can be used to detect vehicles that have sufficient vehicle distance between one vehicle and another (accuracy 61.3%). By integrating the Euclidean Distance Tracker, the congestion detector has maximum accuracy results (100% accuracy) on the tested sample frame.

Keywords: *deep learning, traffic jam detection, euclidean distance, yolov4*

1. Pendahuluan

Menurut Margareth, dkk (2015) Kemacetan adalah turunnya tingkat kelancaran arus lalu lintas pada jalan yang ada, dimana terdapat banyak kendaraan tetapi kapasitas jalan tidak mencukupi sehingga menyebabkan lalu lintas menjadi sulit dan terjadi kemacetan. Terutama di kota-kota besar Indonesia, dimana pertumbuhan jumlah kendaraan terus meningkat setiap tahunnya. Kemacetan juga terjadi pada jam-jam sibuk, yaitu pada pukul 07.00 pagi dan pada pukul 16.00 sore. Pada jam ini masyarakat melakukan berbagai aktivitas, seperti berangkat bekerja, pulang bekerja, berangkat sekolah, pulang sekolah dan banyak kegiatan aktivitas lainnya (Rozari & Wibowo, 2015). Khususnya di Kota Medan, waktu yang hilang akibat kemacetan lalu lintas adalah 35,6 menit per hari untuk sepeda motor dan 48,5 menit per hari untuk mobil. (Girsang, 2020). Kondisi ini meningkatkan penggunaan bahan bakar minyak (BBM) dan pegawai yang kehilangan pekerjaan karena terlambat masuk kantor, yang juga mengurangi pendapatan (Lubis, 2017). Adapun total biaya kemacetan di Kota Medan mencapai Rp. 22.535.355.867/tahun (Susanti & Magdalena, 2015). Oleh karena itu, kemacetan lalu lintas harus dikurangi dengan deteksi kemacetan lalu lintas real-time yang dikombinasikan dengan sistem manajemen lalu lintas.

Convolution Neural Network (CNN) merupakan salah satu metode *Deep Learning* yang dapat mendeteksi dan mengenali suatu objek (Devi et al., 2022). CNN dapat melakukan perhitungan matematis pada input yang terdiri dari banyak *hidden layer* (lapisan tersembunyi). Lapisan tersembunyi memeriksa gambar input dan mengirimkan hasil olahan ke bagian output.

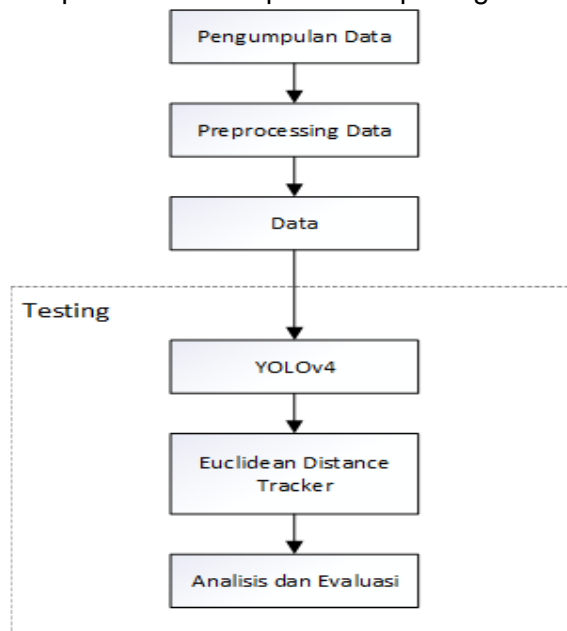
Penelitian sebelumnya menggunakan Penerapan *Haar Cascade Classifier* dan CNN untuk membuat pendeteksi kendaraan otomatis untuk mengukur kepadatan lalu lintas. Peneliti menggunakan dataset *The Comprehensive Cars Dataset*, dengan total 1000 gambar untuk dataset kendaraan. Peneliti menyesuaikan format gambar untuk pembuatan *Haar Cascade* pada tahap *preprocessing*. Didapatkan hasil dengan akurasi 82% untuk data pengujian yang sepi, akurasi 69% untuk data pengujian normal, dan akurasi 60% untuk data pengujian ramai (Hasanah et al., 2021).

Penelitian lainnya menggunakan *Deep Learning YOLOv4* secara *real-time* dalam pengendalian lalu lintas. Peneliti menggunakan 25.000 gambar untuk train, validation dan testing model yang dibagi dengan 70% *train*, 10% *validation* dan 20% *test*. Didapatkan hasil pendeteksian YOLOv4 sebesar 87,98% pada 13 kendaraan (Hasibuan et al., 2021).

Penelitian ini menggunakan deep learning YOLO versi 4 untuk mendeteksi objek seperti mobil, sepeda motor dan lainnya. Data yang digunakan adalah data lalu lintas berupa video CCTV yang disediakan oleh Pemerintah Kota Medan Sumatera Utara.

2. Metode Penelitian

Metode yang digunakan dalam penelitian ini dapat dilihat pada gambar 1.



Gambar 1. Skema Metode Penelitian

Penelitian dimulai dengan mengumpulkan data video lalu melakukan *preprocessing* pada video. Selanjutnya melakukan *object detection* dengan menggunakan YOLOv4 pada video yang dilanjutkan dengan melakukan *tracking* dengan menggunakan Euclidean Distance Tracker pada objek yang telah terdeteksi. Selanjutnya, melakukan analisis dan evaluasi dari performa YOLOv4 dan Euclidean Distance Tracker.

2.1. Convolution neural network

CNN adalah jenis jaringan saraf yang dalam karena kedalaman jaringannya yang besar dan juga banyak diterapkan pada data gambar (Hafifah et al., 2021). CNN dapat melakukan perhitungan matematis pada input yang terdiri dari banyak hidden layer (lapisan tersembunyi). Lapisan tersembunyi ini memeriksa gambar input dan mengirimkan hasil olahan ke bagian output. CNN memiliki convolution layer, layer ini berguna untuk ekstraksi fitur, dan bagian classifier untuk mengklasifikasikan fitur ke dalam kelas-kelas yang ditentukan oleh pelatihan. Fase training umumnya memakan waktu yang cukup lama dan membutuhkan performa komputer yang kuat, namun prediksi CNN relatif cepat dan cukup akurat (Hasibuan et al., 2021).

2.2. YOLO v4

YOLO v4 merupakan jaringan saraf yang paling akurat (55,8% AP) dalam dataset Microsoft COCO jika dibandingkan dengan seluruh jaringan saraf yang telah dipublikasikan hingga saat ini (Wang et al., 2020). YOLO v4 juga merupakan yang terbaik pada hal kecepatan & akurasi pada semua rentang akurasi & kecepatan hingga 1774 FPS (Wang et al., 2020).

Jaringan saraf yang diungguli oleh YOLO v4 dalam hal akurasi (Wang et al., 2020):

- 1) Google EfficientDet D7x / DetectoRS/ SpineNet-190 (dilatih sendiri dengan data tambahan)
- 2) Amazon Cascade-RCNN ResNest200
- 3) Microsoft RepPoints v2
- 4) Facebook RetinaNet SpineNet-190

Dari perbandingan dengan jaringan saraf lainnya, performa YOLO v4 terletak pada *Pareto Optimality Curve* (Wang et al., 2020).

Ketika objek terdeteksi oleh detektor YOLOv4, maka YOLOv4 akan mengembalikan *bounding box* koordinat objek dalam bentuk $(x,y,w,h,conf)$, (x,y) merupakan koordinat objek, w merupakan panjang objek, h merupakan tinggi objek dan $conf$ merupakan nilai *confidence score* (Zhao et al., 2020)

2.3. Euclidean Distance Tracker

Tracker / pelacak yang digunakan merupakan *tracker* yang menggunakan rumus *Euclidean Distance* untuk melacak perpindahan kendaraan. Rumus dari *Euclidean Distance* terdapat pada persamaan (1) (Dharma Adhinata et al., 2021).

$$d(C_1, C_2) = \sqrt{(x_{max} - x_{min})^2 + (y_{max} - y_{min})^2} \quad (1)$$

Setelah mendapatkan nilai koordinat tengah objek pada masing masing objek yang terdeteksi dengan menggunakan YOLOv4, dilanjutkan dengan pemberian ID baru apabila belum pernah dilacak dan disimpan dalam *tracked list*. Pada frame selanjutnya, YOLOv4 akan mendeteksi lagi dan mengembalikan koordinat posisi objek yang terdeteksi dan akan mengupdate *tracked list*. Dengan integrasi *Euclidean Distance*, program dapat menghitung perpindahan masing masing objek.

Di dalam penelitian ini, apabila perpindahan objek antara frame sebelum dan sesudah kurang dari 25px maka program akan menganggap objek tersebut adalah objek yang sama dengan frame sebelumnya (program memberikan ID yang sama). Apabila lebih dari 25px dianggap menjadi objek yang berbeda (program akan memberikan ID baru kepada objek tersebut). Apabila perpindahan objek lebih dari 4px maka objek tersebut dianggap bergerak oleh program. Nilai ambang batas 25px dan 4px didapat dari perpindahan objek dalam kecepatan normal (± 30 km/jam) dan dalam kecepatan rendah (± 5 km/jam) yang bergantung pada posisi CCTV, FPS video serta luas jalan.

Apabila kendaraan mengalami percepatan lebih dari ambang batas toleransi tersebut, maka program akan menganggap kendaraan tersebut sebagai kendaraan yang berbeda. Sedangkan apabila kendaraan mengalami perlambatan, program akan tetap dapat melacak kendaraan tersebut.

2.4. Uji dan Eksperimen

2.4.1 Dataset

Data yg dipakai pada penelitian ini merupakan data video dengan codec MPEG, didapatkan dari *livestream* CCTV yang disediakan oleh Pemerintah Kota Medan dengan nama "SIMP SM RAJA PELANGI".

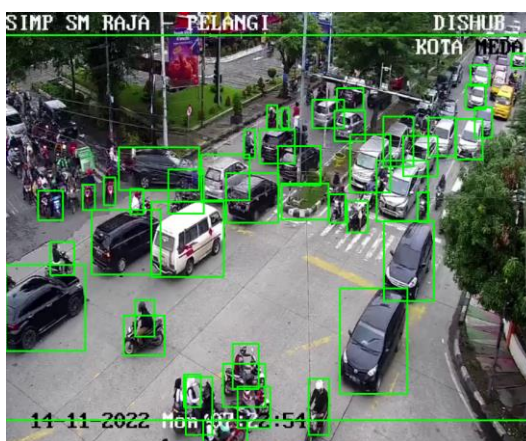


Gambar 2. Dataset CCTV SIMP. SM. RAJA PELANGI

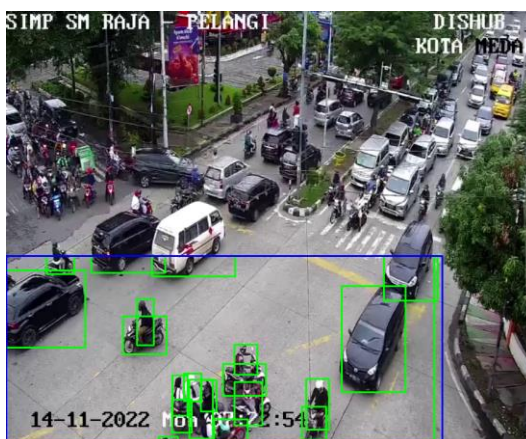
Video memiliki *bitrate* 795kbps, *width* 704px, *height* 576px dan 15 FPS dengan total 6 video masing - masing berdurasi 1 menit. Video tersebut diambil dari 2 hari yang berbeda. Gambar 2 diambil dari screenshot dataset pada saat malam, pagi, siang dan sore hari.

2.4.2 Preprocessing Data

Preprocessing yang dilakukan adalah menetapkan daerah *ROI (Region of Interest)* untuk menghemat *computing resource* dan waktu yang dibutuhkan untuk mendeteksi. ROI diletakkan sesuai dengan kotak berwarna kuning yang berada di tengah persimpangan jalan.



Gambar 3. Tanpa ROI



Gambar 4. Dengan ROI

2.4.3 Training

Dikarenakan YOLOv4 telah dilatih terlebih dahulu untuk mendeteksi 80 *classes* - termasuk mobil, motor, bis dan truk yang diperlukan dalam penelitian ini – yang telah disediakan di dalam *library cvlib*, maka peneliti tidak melakukan training lagi.

2.4.4 Testing

Pengujian dilakukan dengan menggunakan YOLOv4 dan mengimplementasikan *Euclidean Distance* untuk *tracking*. Untuk menghemat *computing resource*, peneliti juga menurunkan FPS video dari 15 FPS menjadi 1 FPS.

3. Hasil dan Pembahasan

3.1. Pengujian Deteksi Objek & Kemacetan

Pengujian ini dilakukan untuk mengetahui bagaimana performa dari YOLOv4 dalam mendeteksi objek mobil, motor dan truck dalam berbagai situasi. Pengujian dilakukan dengan 3 skenario, yaitu pada saat suasana macet, lancar dan malam hari.



Gambar 5. Jalan Macet



Gambar 6. Jalan Lancar



Gambar 7. Jalan pada malam hari

3.2. Pengujian Akurasi Deteksi Kendaraan

Akurasi dari pendeteksian kendaraan dilakukan dengan menggunakan persamaan (Indah Kusumawati et al., 2021):

$$Acc = \frac{V_{deteksi}}{V_{sebenarnya}} * 100\% \quad (2)$$

Tabel 1. Hasil deteksi dari Beberapa Sampel Frame

sampel	jumlah deteksi	jumlah sebenarnya
1	4	7
2	9	13
3	9	15
4	10	15
5	11	20
6	15	34
7	15	21
8	16	24

Tabel 2. Status dan Akurasi dari Pendeteksian

sampel	status	akurasi
1	Lancar, Malam Hari	0,57143
2	Lancar, Siang Hari	0,69231
3	Lancar, Siang Hari	0,6
4	Lancar, Siang Hari	0,66667
5	Macet, Siang Hari	0,55
6	Macet, Malam Hari	0,44118
7	Macet, Siang Hari	0,71429
8	Macet, Siang Hari	0,66667

Dari tabel 1, dapat disimpulkan bahwa pada tingkat akurasi pendeteksian objek kendaraan oleh YOLOv4 dipengaruhi oleh faktor kemacetan dan intensitas cahaya. Semakin macet dan semakin rendah intensitas cahaya, maka tingkat akurasi akan semakin rendah. Hal ini dikarenakan apabila kemacetan terjadi, kemungkinan terjadinya *overlap* antara kendaraan yang satu dan yang lainnya semakin tinggi, sehingga YOLOv4 tidak mengenali salah satu atau kedua kendaraan tersebut. Dari semua sampel, didapatkan rata rata akurasi YOLOv4 yaitu 61,3%.

3.3. Pengujian Akurasi Deteksi Kemacetan

Akurasi dari deteksi kemacetan diuji dengan persamaan (2). Akurasi diuji dengan membandingkan hasil deteksi status kemacetan (macet atau lancar) dengan keadaan sebenarnya.

Tabel 3. Hasil deteksi kemacetan dari beberapa sampel *frame*

sampel	Hasil Deteksi	Status Sebenarnya
1	Lancar	Lancar
2	Lancar	Lancar
3	Lancar	Lancar
4	Lancar	Lancar
5	Macet	Macet
6	Macet	Macet
7	Macet	Macet
8	Macet	Macet

Dengan menggunakan persamaan (2) untuk menghitung akurasi dari jumlah hasil deteksi yang sesuai dengan keadaan sebenarnya dibagi dengan jumlah sampel yang terdapat di Tabel 3, didapatkan akurasi 100%.

4. Kesimpulan

Dari hasil pengamatan dan penelitian dapat diambil kesimpulan sebagai berikut. Pertama, YOLOv4 dapat digunakan untuk mendeteksi kendaraan dengan jarak kendaraan yang cukup antara kendaraan tersebut dengan kendaraan lain dan pencahayaan yang cukup. Kedua, akurasi YOLOv4 menurun jika kendaraan memiliki jarak sangat dekat satu dengan lainnya dan dengan tingkat pencahayaan yang rendah. Ketiga, tingkat akurasi pendeteksian YOLOv4 memiliki akurasi 61,3% pada sampel frame yang diuji. Keempat, walaupun YOLOv4 memiliki performa yang kurang baik secara keseluruhan, namun dengan integrasi tracking mendeteksi adanya kemacetan memberikan hasil akurasi yang maksimum.

Daftar Pustaka

- Devi, O., Sunanto, S., & Utomo, P. H. (2022). Implementation Of Deep Learning With Convolutional Neural Network To Classification Organic And Nonorganic Waste Picture. *Seminar Nasional Matematika, Geometri, Statistika, Dan Komputasi*, 373–382. <https://magestic.unej.ac.id/>
- Dharma Adhinata, F., Putra Rakhmadani, D., Jala, A., & Segara, T. (2021). YOLO Algorithm for Detecting People in Social Distancing System. *Transformatika*, 19(1), 1–7. <https://pjreddie.com/>.
- Girsang, W. E. J. (2020). *Analisis Kerugian Pengguna Jalan Akibat Kemacetan Lalu Lintas di Kota Medan*. Universitas Sumatera Utara.
- Hafifah, F., Rahman, S., & Asih, S. (2021). Klasifikasi Jenis Kendaraan Pada Jalan Raya Menggunakan Metode Convolutional Neural Networks (CNN). *TIN: Terapan Informatika Nusantara*, 2(5), 292–301. <https://ejournal.seminar-id.com/index.php/tin>
- Hasibuan, N. N., Zarlis, M., & Efendi, S. (2021). Detection and tracking different type of cars with YOLO model combination and deep sort algorithm based on computer vision of traffic controlling. *Jurnal Dan Penelitian Teknik Informatika*, 6(1), 210–220. <https://doi.org/10.33395/sinkron.v6i1.11231>
- Indah Kusumawati, W., Pratikno, H., & Pradeska Admaja, Y. (2021). Sistem Penghitung Jumlah Pengunjung Restoran Menggunakan Kamera Berbasis Single Shot Detector (SSD). *Journal of Technology and Informatics (JoTI)*, 3(1), 19–26. <https://doi.org/10.37802/joti.v3i1.197>
- Lubis, A. M. M. (2017). *Analisis Dampak Kemacetan Lalu Lintas Terhadap Sosial Ekonomi Bagi Pengguna Jalan Raya (Studi Kasus Kota Medan)*. Universitas Muhammadiyah Sumatera Utara.
- Rozari, A. de, & Wibowo, Y. H. (2015). Faktor-faktor Yang Menyebabkan Kemacetan Lalu Lintas Di Jalan Utama Kota Surabaya. *Jurnal Penelitian Administrasi Publik*, 1(1), 42–57.
- Susanti, S., & Magdalena, M. (2015). Estimasi Biaya Kemacetan Di Kota Medan. *Jurnal Transportasi Multimoda*, 13(1), 21–30.
- Wang, C.-Y., Bochkovskiy, A., & Liao, H.-Y. M. (2020). *Scaled-YOLOv4: Scaling Cross Stage Partial Network*. <http://arxiv.org/abs/2011.08036>
- Zhao, Y., Zhou, X., Xu, X., Jiang, Z., Cheng, F., Tang, J., & Shen, Y. (2020). A novel vehicle tracking id switches algorithm for driving recording sensors. *Sensors (Switzerland)*, 20(13), 1–15. <https://doi.org/10.3390/s20133638>