



Seperation Deteksi Kendaraan Pada Citra Digital Dengan Menggunakan AlgoritmaYOLO (You Only Look Once)

Abdurrahman Faqih ^{1*}, Khonsa Mutmainnah^{2†}, Muthiah Afifah R^{3‡}

^{1,2,3}Program Studi Ilmu Komputer STIMIK ESQ Jakarta, Kampus Menara 165, Lantai 18 & 19,
Jl. Tb. Simatupang Kav. 1, Jakarta, Indonesia

Abstract. Pertumbuhan sektor transportasi di Indonesia sangat tinggi sehingga padatnya kendaraan yang berada di jalan raya. Dengan jumlah kendaraan yang semakin bertambah, fasilitas jalan saat ini tidak dapat mengimbangi sehingga terjadinya permasalahan-permasalahan lalu lintas. Salah satu cara untuk mengatasi permasalahan tersebut dengan cara pengawasan lalu lintas. Maka dari itu dibutuhkannya informasi kendaraan yang melintasi jalan raya dan program yang mampu mendeteksi dan menghitung jumlah kendaraan yang lewat di jalan raya. Pada penelitian ini, kami memakai metode YOLO (You Only Look Once) untuk dapat menghitung kendaraan yang ada pada citra input.

Keywords: YOLO; citra; kendaraan; lalu lintas

1. Introduction

Peningkatan pemakaian alat transportasi yang semakin hari semakin bertambah mengakibatkan penumpukan jumlah kendaraan pada ruas jalan. Hal itu juga didukung dengan tidak bertambahnya luas jalan yang signifikan. Penumpukan jumlah kendaraan, atau kemacetan tersebut, seringkali terjadi di beberapa daerah di Indonesia. Hal ini terbukti dengan Jakarta sempat masuk 10 besar kota termacet di dunia pada tahun 2017-2019.

Maka dari itu, perlu adanya penindakan dari berbagai pihak untuk menjadi pencegah hal tersebut dapat diminimalisir. Pencegahan tersebut membutuhkan suatu data statistik tentang pertumbuhan kendaraan di wilayah tertentu. Pertumbuhan kendaraan ini dapat dilihat dari seberapa banyak kendaraan yang melewati jalan tertentu. Dengan melihat pertumbuhan tersebut, barulah data dapat digunakan sebagai tolak ukur perlu tidaknya penambahan fasilitas jalan raya atau fasilitas lainnya.

Sebelumnya, untuk pengambilan data tersebut dilakukan secara manual. Dengan perkembangan teknologi yang sudah maju saat ini, hal tersebut dapat dilakukan dengan mudah menggunakan teknologi. Dalam hal ini, penulis membuat sebuah program deteksi citra digital untuk menghitung alat transportasi yang ada pada sebuah input citra. Program yang dibuat ini diharapkan dapat dikembangkan pada kemudian hari untuk penelitian terkait.

*Corresponding author's email: a.faqih@students.esqbs.ac.id

†Corresponding author's email: k.mutmainnah@students.esqbs.ac.id

‡Corresponding author's email: m.afifah@students.esqbs.ac.id

2. Methods

2.1 Pengambilan Citra

Sistem yang dibuat pada penelitian ini terdiri dari pengambilan citra dengan menggunakan gambar untuk menentukan citra kendaraan yang melintasi jalan raya.

2.2 Anotasi Data

Selanjutnya dilakukan anotasi data. Anotasi data bertujuan untuk memberikan label pada gambar dengan memberikan kotak pembatas (bounding box) dan nama kelas pada setiap objek. Dalam hal ini kami sudah memiliki pelabelan objek yang kami unduh dari sumber yang telah meneliti mengenai objek yang kami deteksi. Tahap pelabelan objek dilakukan ketika data yang berupa gambar-gambar kendaraan dikumpulkan dengan cukup banyak agar data yang diolah dapat keakuratan yang efisien. Proses pelabelan mobil dimulai dari atap mobil sampai roda mobil, pelabelan pada sepeda motor dimulai dari kepala sampai roda sepeda motor, kemudian pelabelan pada truk dimulai dari bagian atas bak truk sampai roda truk, lalu pelabelan pada bus dimulai dari atap bus sampai roda bus, dan pelabelan pada becak dimulai dari bagian atas becak sampai roda becak.

Proses anotasi citra menggunakan software LabelImg. Pelabelan ini menggunakan anotasi dalam format anotasi YOLO. Hasil dari anotasi tersebut adalah data yang terdapat informasi letak kotak pembatas dan labelnya dalam bentuk .txt. Pada .txt file terdapat baris file yang memiliki format `<object-class> <x_center><y_center> <width> <height>`, dimana pada `<object-class>` merupakan bilangan bulat yang menyatakan kelas objek, `<x_center>` dan `<y_center>` adalah koordinat pusat persergi kotak pembatas, `<width>` dan `<height>` adalah nilai float relatif terhadap dimensi gambar.

2.3 Proses Training

Training data bertujuan untuk melatih komputer dengan mengolah data yang telah di anotasi agar terbentuk suatu karakteristik sebagai pertimbangan untuk mencapai sebuah prediksi.

Langkah selanjutnya adalah melakukan download file weight dan file configuration YOLOv4 yang sudah selesai di-training. File ini adalah inti dari algoritma YOLO untuk mendeteksi objek dan file configuration adalah pengaturan dari algoritma YOLO.

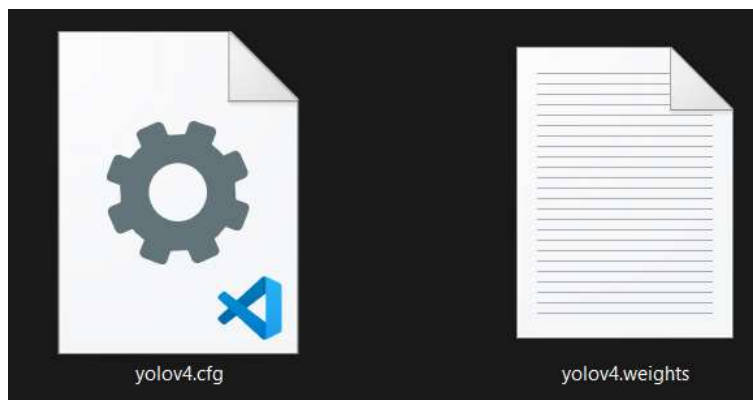


Figure 1 Experiment set-up and apparatus

2.4 Implementasi algoritma You Only Look Once (YOLO)

Selanjutnya adalah mengimplementasikan algoritma You Only Look Once (YOLO) untuk melakukan deteksi dan klasifikasi kendaraan. Implementasi dilakukan dengan mengimplementasikan hasil desain ke dalam perangkat lunak menggunakan bahasa yang dapat dibaca oleh komputer.

Pada bagian ini menggunakan pre-trained weights YOLOv4. Dengan menggunakan teknik transfer learning. Transfer learning adalah suatu teknik yang menggunakan model yang telah di-training sebelumnya (pre-trained model) yang dapat digunakan untuk klasifikasi dataset baru tanpa harus melakukan training data dari awal. Proses transfer learning pada Darknet menggunakan data file, cfg file, dan pre-trained weights. Data file berisi lokasi gambar yang digunakan untuk train dan test. Cfg file berisi bentuk jaringan yang digunakan untuk training, dan pre-trained weights berisi model weight yang telah dilatih sebelumnya pada jaringan YOLO.

Pada proses deteksi, setelah semua software yang dibutuhkan telah ter-instalasi dengan baik. Langkah selanjutnya adalah menggabungkan file weight, configuration, dan dataset pada satu folder.

2.1.1 Import Library

Menggunakan library cv2, numpy, dan glob untuk membaca folder gambar. Library cv2 berfungsi untuk melakukan computer vision task. Library numpy berfungsi untuk pengolahan data numerical.

```
import cv2
import glob
from deteksi_kendaraan import KendaraanDetector

import numpy as np
```

2.1.2 Membaca Masukan Gambar

Pada bagian ini dilakukan membaca gambar dengan mendefinisikan cv2.imread (img.path) untuk mendapatkan objek dari image pada yang telah dikumpulkan/di unduh sesuai dengan lokasi penyimpanan. Kami menggunakan library glob agar dapat membaca gambar-gambar dari folder

```
images_folder = glob.glob("gambar/*.jpg")
for img_path in images_folder:
    print("Img path", img_path)
    img = cv2.imread(img_path)
```

2.1.3 Membuat Network YOLOv4

Menggunakan file weight, cfg, dan name files. Weight file adalah model yang sudah di training, inti dari algoritma YOLO untuk mendeteksi objek. Cfg file adalah file konfigurasi, dimana semua pengaturan algoritma YOLO terdapat pada file tersebut. Dan Name files adalah file yang berisi nama-nama objek yang dapat dideteksi menggunakan algoritma YOLO. Disini penulis menggunakan name files yang berisikan lima kelas yang akan dideteksi. Setelah itu mengatur minimum probabilitas untuk mengeliminasi prediksi

rendah dengan nilai 0,5. Serta melakukan pengaturan untuk menyaring kotak pembatas yang rendah dengan nilai threshold 0,5 jika lebih besar maka objek akan terdeteksi dengan benar, jika tidak maka akan dilewatkan. Nilai threshold berubah dari 0 ke 1. Semakin dekat ke 1 maka semakin besar akurasi deteksi, jika semakin dekat ke 0 maka semakin sedikit akurasi tetapi juga semakinbesar jumlah objek yang terdeteksi.

```
def __init__(self):
    # Load Network
    net=cv2.dnn.readNet("dnn_model/yolov4.weights",
"dnn_model/yolov4.cfg")
    self.model = cv2.dnn_DetectionModel(net)
    self.model.setInputParams(size=(832, 832), scale=1 / 255)
```

2.1.4 Menentukan koordinat dan menghitung kendaraan

Koordinat yang ditentukan adalah dari sebuah box yang telah diinisialisasikan untuk mendeteksi kendaraan, koordinat akan direpresentasikan dengan x dan y sebagai titik dan h sebagai tinggi(height) wsebagai lebar(width). setelah mendapatkan koordinatnya maka box yang didapat akan di hitung.

```
#Menentukan koordinat box
kendaraan_boxes = kd.detect_kendaraan(img)
#print(kendaraan_boxes)

#menghitung kendaraan dari box yang ada
kendaraan_count = len(kendaraan_boxes)
```

3. Results and Discussion

Berikut ini akan ditampilkan bebrapa gambar hasil dari percobaan yang peneliti lakukan, diantaranya yaitu:



Figure 2 Contoh hasil program ketika dijalankan

Pada gambar 2 diatas, nampak terdeteksi 4 kendaraan yang terdiri dari 1 motor dan 3 mobil. Pada gambar 3 seperti ditunjukkan dibawah ini, terdapat 6 kendaraan yang terdeteksi yang terdiri dari 6 mobil



Figure 3 Contoh hasil program ketika dijalankan

Pada gambar 4, terlihat gambar yang memperlihatkan ada kendaraan dan ada seseorang yang sedang berjalan, karena algoritma yang dibangun hanya mendeteksi kendaraan maka ada 3 kendaraan yang terdeteksi yaitu kendaraan mobil

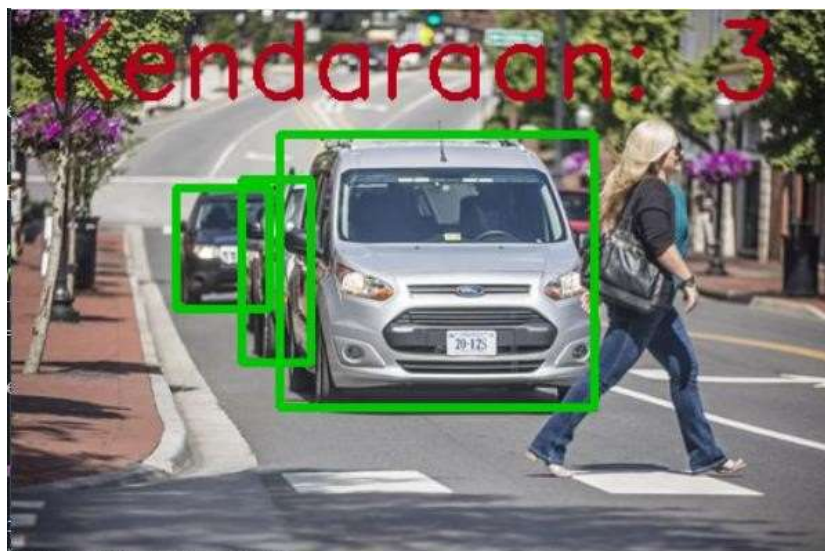


Figure 4 Contoh hasil program ketika dijalankan

Pada gambar 5 terdeteksi sebuah gambar dan terhitung ada 3 kendaraan yaitu 2 truk dan 1 bus



Figure 5 Contoh hasil program ketika dijalankan

Pada gambar 6 terdapat kendaraan yang terparkir di depan halaman rumah dan terdeteksi 11 kendaraan yang terdiri dari 11 mobil



Figure 6 Contoh hasil program ketika dijalankan

Pada gambar 7 terdapat banyak kendaraan yang berada di jalan tol karena banyaknya kendaraan maka pendeteksian kurang akurat, maka dari gambar yang ada hanya terdeteksi 12 kendaraan yang terdiri dari truk dan mobil.



Figure 7 Contoh hasil program ketika dijalankan

Pada gambar 8 terdapat gambar kendaraan pada malam hari, dalam kasus ini kendaraan dapat terdeteksi akan tetapi karena terbatasnya jarak maka kendaraan yang cukup jauh tidak dapat terdeteksi, makahasilnya hanya ada 10 kendaraan yang terdeteksi.

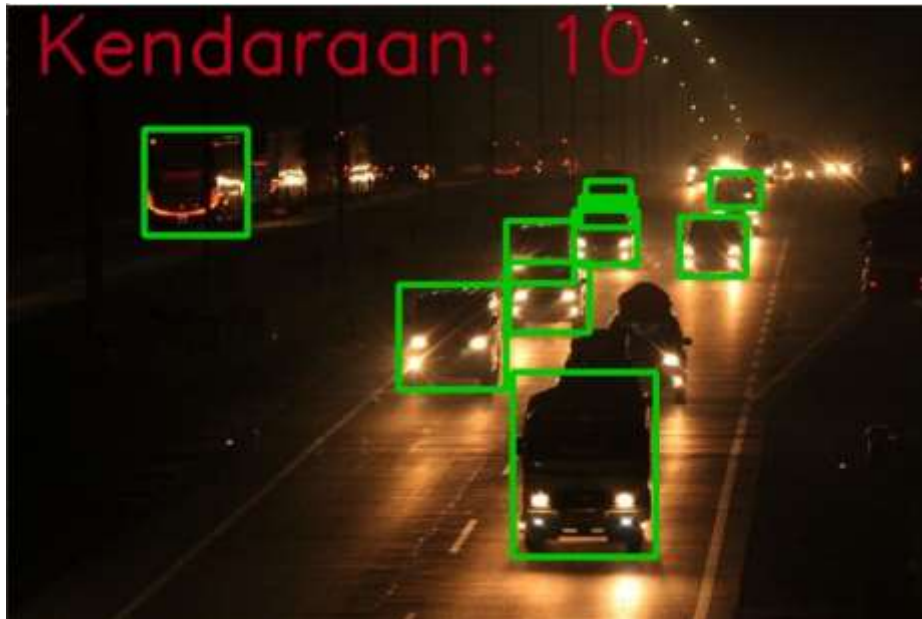


Figure 8 Contoh hasil program ketika dijalankan

4. Conclusions

Kesimpulan yang dapat kami ambil dari penelitian ini adalah bahwa program yang kami bangun untuk dapat mendeteksi kendaraan darat seperti mobil, motor, bus, dan truk dengan menggunakan algoritma YOLO (You Only Look Once) tersebut kurang akurat apabila keadaan kendaraan penuh atau blur.

References

- Jumadi, Juju, Yupianti & Sartika, Devi. 2021. Pengolahan Citra Digital Untuk Identifikasi Objek Menggunakan Metode Hierarchical Agglomerative Clustering. *Jurnal Sains dan Teknologi*, Vol.10 No 2 Tahun 2021
- Lazaro, Alvin., Buliali, Joko Lianto., Amaliah, Bilqis. 2017. Deteksi Jenis Kendaraan di Jalan Menggunakan OpenCV. *JURNAL TEKNIK ITS* Vol. 6, No. 2 (2017)
- Liunanda, Christopher Nathanel., Rostianingsih, Silvia., Purbowo, Anita Nathania. 2020. Implementasi Algoritma YOLO pada Aplikasi Pendeteksi Senjata Tajam di Android. *JURNAL INFRA*, Vol.8.No.2
- Wahyudinur, Alfarisi., 2016. Pencarian Jalur Terpendek Pengiriman Barang Menggunakan Algoritma A* Studi Kasus Kantor Pos Besar Medan, *Jurnal RisetKomputer (JURIKOM)*, Vol. 3.