



TEKNOTIKA Vol. XX, No. X, 20XX, XX-XX

Pendeteksian Mobil di Jalan Raya Menggunakan Algoritma YOLOv3

Ahmad Ibnu Widya Attamimi¹, Ahmad Hilman Dani² & Wahyu SJ Saputra³

Universitas Pembangun Nasional “Veteran” Jawa Timur; Jl. Rungkut Madya No.1, Gn.
Anyar, Kec. Gn. Anyar, Kota SBY, Jawa Timur 60294

Email: wahyu.s.j.saputra.if@upnjatim.ac.id

ABSTRAK

Berdasarkan *Electronic Registration Identification* (ERI) Korlantas Polri tercatat 146.165.956 unit jumlah kendaraan bermotor di Indonesia pada tahun 2022. Hal tersebut berpotensi menyebabkan terjadinya kemacetan hingga kecelakaan lalu lintas, khususnya pada daerah perkotaan padat penduduk. Maka dari itu, telah banyak ahli yang mengusulkan penerapan pendeteksian objek untuk pemantauan lalu lintas sehingga kemacetan dapat diminimalisir dan kecelakaan dapat dihindari. *You Only Look Once Version 3* (YOLOv3) merupakan satu diantara algoritma pendeteksian objek yang cepat, tepat, dan *real time*. Bahkan, banyak sumber mengatakan bahwa YOLOv3 memiliki tingkat akurasi pendeteksian obyek mencapai 90%. Pada paper ini, kami menggunakan dataset coco dan data uji berupa video mobil di jalan raya. Nantinya, YOLOv3 akan menghasilkan output berupa bounding box untuk pendeteksian objek.

Kata kunci: kecerdasan buatan, deep learning, object detection, algoritma yolo, neural network.

1 Pendahuluan

Berdasarkan *Electronic Registration Identification* (ERI) Korlantas Polri tercatat 146.165.956 unit jumlah kendaraan bermotor di Indonesia pada tahun 2022 [1]. Dari data tersebut, terdapat peluang jumlah kendaraan di Indonesia akan bertambah. Hal ini dapat menimbulkan kemacetan bahkan kecelakaan lalu lintas jika tidak ada pemantauan dan pengendalian sistem lalu lintas yang baik. Maka dari itu, YOLOv3 dapat diterapkan untuk pendeteksian objek khususnya mobil di jalan raya sehingga pihak yang berwenang dapat melakukan pemantauan dan pengendalian sistem lalu lintas berdasarkan sumber data lapangan yang telah didapatkan dari pendeteksian objek menggunakan YOLOv3, terlebih, banyak sumber mengatakan bahwa performa YOLOv3 yang sangat cepat dan tingkat akurasi hingga mencapai

90% sehingga YOLOv3 merupakan satu diantara metode pendeteksian objek yang cocok untuk diterapkan dalam kasus ini.

YOLOv3 sebagai metode pendeteksian objek akan melakukan klasifikasi serta *locating* (menentukan lokasi) suatu objek yang ada di dalam citra yang telah diinputkan. Klasifikasi dilakukan dengan cara memindai data berupa format warna merah, hijau, dan biru pada setiap pixel dari citra yang akan yang diuj dan localization dilakukan dengan menghasilkan bounding box.

2 Metode penelitian

Pada bab ini akan dibahas mengenai dataset yang digunakan, data uji yang digunakan, serta penjelasan mengenai algoritma YOLOv3.

2.1 Dataset COCO

Coco merupakan dataset yang sudah dilatih oleh Joseph Redmon sehingga komputer dapat melakukan pendeteksian objek menggunakan YOLOv3. Adapun isi dari dataset ini ialah 80 kelas objek, diantaranya ialah, *person*, *bicycle*, *car*, *motorbike*, *bus*, dan 75 kelas lainnya. [4]-[5]

2.2 Data uji yang digunakan

Penulis menggunakan data uji berupa video yang diambil dari youtube yang berjudul Traffic IP Camera video dengan durasi 29 detik dan kualitas 240p, dengan ukuran lebar 1280 x 720 dengan frame rate ~25fps. dengan jumlah total frame ~746



Gambar 1. Gambaran data uji. Tangkapan Layar dari Video yang Menjadi Data Uji

2.3 Algoritma YOLOv3

secara garis besar, yang dilakukan algoritma YOLO dalam mendeteksi object adalah [6]:

1. Menerima inputan berupa video mobil di jalan raya
2. Membaca data pixel dari sebuah frame atau gambar.
3. Mengubah ukuran data dan mengelompokkannya dalam beberapa grid dengan ukuran SxS, menyesuaikan dengan data inputan.
4. Memetakan setiap grid dengan fully connected layer dan fungsi softmax.
5. Pendeteksian object akan dilakukan pada masing-masing sel grid, deteksi diperjelas dengan bounding box agar bisa dipahami oleh pengguna.
6. jika kurang jelas, akan dilakukan fungsi non max supression untuk memperoleh confidence score yang optimal. ini akan mempengaruhi seberapa tepat mesin dalam memberikan bounding box untuk sebuah objek pada gambar.

3 Hasil dan Pembahasan

Pada bab ini akan dibahas mengenai output yang dihasilkan serta analisis akurasi pada YOLOv3.

3.1 Output

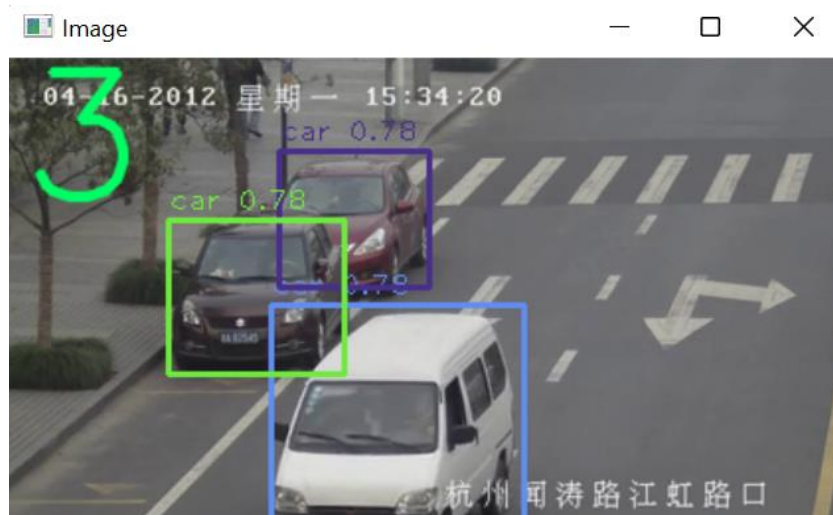
YOLOv3 akan menghasilkan output berupa bounding box pada objek yang akan di deteksi, agar lebih jelas, perhatikan gambar dibawah ini,



Gambar 2. Bounding box. YOLOv3 Menghasilkan Bounding box Untuk Pendeteksian Objek

Berdasarkan gambar 2, dapat kita perhatikan bahwa:

- Terdapat tiga object mobil dan tiga bounding box
- Tulisan *car* di atas bounding merupakan penanda yang diambil dari *class* dataset coco untuk menunjukkan klasifikasi dari objek yang
- Angka *0,31* merupakan *confidence score* yang menunjukkan seberapa besar kemungkinan objek tersebut merupakan sebuah mobil, dari gambar tersebut terlihat bahwa sistem yakin bahwa objek tersebut adalah mobil sebesar 0,31 atau 31%. Nilai ini akan terus berubah tergantung dari objek yang dideteksi, lihat gambar dibawah ini,



Gambar 3. Confidence score lebih besar. Objek yang dideteksi lebih jelas maka nilai dari confidence score juga bertambah

Berdasarkan gambar tersebut, objek yang dideteksi posisinya lebih dekat dengan kamera, sehingga sistem dapat melakukan tugasnya dengan lebih baik dan menghasilkan confidence score yang lebih besar.

```

data.txt - 2. Deteksi Obyek dengan YOLO dan Op
yolo.py data.txt x coco.names
data.txt
8661
8662 object confidence top left righth bottom ukuran bounding box
8663
8664 car 8 0.3315 {x: 409 y: 152} {x: 645 y: 365} {236 x 213}
8665
8666 car 11 0.3315 {x: 242 y: 247} {x: 518 y: 480} {276 x 233}
8667 bounding box terkecil : 30 x 23
8668 -----
8669 frame ke : 743
8670
8671 object confidence top left righth bottom ukuran bounding box
8672
8673 car 8 0.3498 {x: 408 y: 153} {x: 645 y: 365} {237 x 212}
8674
8675 car 11 0.3498 {x: 242 y: 246} {x: 518 y: 479} {276 x 233}
8676 bounding box terkecil : 30 x 23
8677 -----
8678 frame ke : 744
8679
8680 object confidence top left righth bottom ukuran bounding box
8681
8682 car 8 0.3311 {x: 409 y: 153} {x: 645 y: 365} {236 x 212}
8683
8684 car 11 0.3311 {x: 242 y: 247} {x: 518 y: 480} {276 x 233}
8685 bounding box terkecil : 30 x 23
8686 -----
8687 frame ke : 745
8688
8689 object confidence top left righth bottom ukuran bounding box
8690
8691 car 8 0.3322 {x: 409 y: 152} {x: 645 y: 365} {236 x 213}
8692
8693 car 11 0.3322 {x: 242 y: 246} {x: 518 y: 479} {276 x 233}
8694 bounding box terkecil : 30 x 23
8695 -----
8696 frame ke : 746
8697
8698 object confidence top left righth bottom ukuran bounding box
8699
8700 car 8 0.3321 {x: 409 y: 152} {x: 645 y: 365} {236 x 213}
8701
8702 car 11 0.3321 {x: 242 y: 246} {x: 518 y: 479} {276 x 233}
8703 bounding box terkecil : 30 x 23
8704 -----
8705

```

Gambar 4. Log deteksi objek. Melakukan Uji Percobaan Agar Diketahui Objek Terkecil Yang Dapat Dideteksi

berdasarkan percobaan yang dilakukan dengan video tersebut, objek terkecil pada frame yang dapat diamati dan diberi label pada algoritma ini adalah 30 x 23.

3.2 Analisis akurasi

Data uji yang digunakan memiliki total 746 frame gambar, dari jumlah tersebut, rata-rata confidence yang diperoleh dari pendeteksian objek pada video adalah 0.7888687. Ada beberapa faktor yang dapat mempengaruhi nilai confidence seperti pola pixel dari object yang terbaca tidak begitu jelas dan agak melenceng dari data training awal, jika polanya makin mendekati dengan data yang sebelumnya, nilai confidence-nya mungkin akan tinggi. seperti pada contoh ketika saat mobil jauh dari kamera, confidence-nya kecil karena sistem membaca dari banyak kemungkinan dari data set. Jika objek semakin dekat dengan kamera dan sistem dapat membacanya dengan jelas karena pola yang ia peroleh lebih dekat dengan gambar mobil dari data training sebelumnya.

Jarak dan sudut pandang juga mempengaruhi akan seperti apa objek itu dideteksi. Hal ini dapat dibuktikan dari log pada beberapa frame yang menunjukkan adanya 'truck' padahal pada video tidak terdapat truck yang tampil. objek 'truck' ini kemudian dideteksi kembali di frame berikutnya sebagai 'car' dengan confidence value yang tidak begitu jauh. Perhatikan gambar berikut

```

data.txt
754 car 10 0.3640 (x: 407 y: 152) (x: 644 y: 363) (237 x 211)
755
756 car 13 0.3640 (x: 242 y: 247) (x: 518 y: 479) (276 x 232)
757
758 person 15 0.3640 (x: 281 y: 46) (x: 328 y: 223) (47 x 177)
759 bounding box terkecil : 45 x 66
760 -----
761 frame ke : 75
762
763 object confidence top left righth bottom ukuran bounding box
764
765 truck 4 0.3443 (x: 1036 y: 0) (x: 1263 y: 127) (226 x 127)
766
767 car 11 0.3443 (x: 408 y: 152) (x: 644 y: 363) (236 x 211)
768
769 car 14 0.3443 (x: 241 y: 248) (x: 519 y: 479) (278 x 231)
770
771 person 16 0.3443 (x: 282 y: 49) (x: 328 y: 220) (46 x 171)
772 bounding box terkecil : 45 x 66
773 -----
774 frame ke : 76
775
776 object confidence top left righth bottom ukuran bounding box
777
778 truck 4 0.3022 (x: 1034 y: -1) (x: 1263 y: 133) (229 x 134)
779
780 car 10 0.3022 (x: 409 y: 152) (x: 644 y: 364) (235 x 212)
781
782 car 13 0.3022 (x: 241 y: 248) (x: 519 y: 480) (278 x 232)
783
784 person 16 0.3022 (x: 281 y: 51) (x: 329 y: 217) (48 x 166)
785 bounding box terkecil : 45 x 66
786 -----
787 frame ke : 77
788
789 object confidence top left righth bottom ukuran bounding box
790
791 car 5 0.3478 (x: 1051 y: -1) (x: 1259 y: 132) (200 x 133)
792
793 car 12 0.3478 (x: 409 y: 152) (x: 645 y: 365) (236 x 213)
794
795 car 15 0.3478 (x: 241 y: 248) (x: 518 y: 480) (277 x 232)
796
797 person 17 0.3478 (x: 280 y: 40) (x: 327 y: 219) (47 x 170)
798 bounding box terkecil : 45 x 66

```

Gambar 5. Log kesalahan deteksi. Tangkapan layar tentang kesalahan deteksi yang dilakukan pada frame ke 75 dan 76

4 Kesimpulan

Berdasarkan percobaan dengan data uji berupa sebuah video, YOLO3 memiliki kemampuan mendeteksi objek dengan cukup baik dan akurat. Algoritma ini bisa mendeteksi objek dengan ukuran pixel yang kecil meskipun memiliki nilai confidence yang kecil.

Objek akan terdeteksi setidaknya ~10px hingga ~20px, oleh karena itu, jika objek lebih kecil dari rasio frame yg dimaksud, maka bounding box tidak akan diberikan. yolo3 tidak dapat memberikan nilai confidence dengan objek yg dimaksud karena informasi dari nilai pixel yg dibutuhkan masih kurang.

YOLO3 dapat digunakan untuk mengawasi jalan karena penempatan kamera yang juga mendukung untuk algoritma ini. Karena keakuratannya, YOLO3 juga bisa digunakan pada jalan yang sering dilalui kendaraan atau kondisi lalu lintas yang padat.

Daftar pustaka

- [1] A. F. Fandisyah, N. Iriawan, and W. S. Winahju, “Deteksi kapal di laut Indonesia Menggunakan yolov3,” *Jurnal Sains dan Seni ITS*, vol. 10, no. 1, 2021.
- [2] Adit, “Ternyata Segini total Populasi Kendaraan di Indonesia, Terbanyak Bukan Jakarta,” *Berita Mobil AutoFuns*, 21-Jan-2022. [Online]. Available: <https://www.autofun.co.id/berita/ternyata-segini-total-populasi-kendaraan-di-indonesia-terbanyak-bukan-jakarta-39922>. [Accessed: 26-Jun-2022].
- [3] B. Putra, B. Nugroho, and F. Anggraeny, “Penggunaan lift pada gedung-Gedu deteksi Dan Menghitung manusia Menggunakan Yolo-CNN,” *Jurnal Informatika dan Sistem Informasi*, vol. 2, no. 1, pp. 67–76, 2021.
- [4] J. Redmon and A. Farhadi, “Yolov3: An incremental improvement,” *arXiv.org*, 08-Apr-2018. [Online]. Available: <https://doi.org/10.48550/arXiv.1804.02767>. [Accessed: 26-Jun-2022].
- [5] J. Redmon, “YOLO: Real-Time Object Detection,” *Darknet: Open source neural networks in C*. [Online]. Available: <https://pjreddie.com/darknet/>. [Accessed: 26-Jun-2022].
- [6] N. Bansal, “Object detection using yolov3 and opencv,” *Medium*, 03-Dec-2021. [Online]. Available: <https://towardsdatascience.com/object-detection-using-yolov3-and-opencv-19ee0792a420>. [Accessed: 26-Jun-2022].
- [7] *Traffic IP Camera video*. [Film]. Tech Channel00001, 2013.