Sistem Deteksi Pelanggaran Zebra Cross Pada Kendaraan Sepeda Motor Menggunakan Algoritma YOLOv4

Cross Zebra Violation Detection System On Motorcycle Vehicles Using The YOLOv4 Algorithm

1st Chianyung
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia
chianyung@student.telkomuniversity.ac.id

2nd Casi Setianingsih
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia
casiesetianingsih@telkomuniversity.ac.id

3rd Marisa W. Paryasto
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia
mariparyasto@telkomuniversity.ac.id

Abstrak—Kecelakaan lalu lintas akhir-akhir ini harus menjadi perhatian khusus bagi kita semua. Kecelakaan terjadi karena kurangnya kesadaran diri dari masyarakat akan peraturan lalu lintas itu sendiri. Polisi sebagai institusi yang membantu masyarakat dalam mengedukasi pentingnya mematuhi peraturan lalu lintas harus tidak boleh lelah dalam menjalankan tugasnya. Salah satu cara edukasi yang dilakukan kepolisian adalah dengan cara pengawasan lalu lintas. Mereka turun ke lapangan untuk menindak pelanggar-pelanggar yang tidak mematuhi aturan, tentunya cara seperti ini memiliki beberapa kekurangan seperti SDM yang terbatas dan juga waktu yang terbatas. Dengan itu teknologi sebenarnya dapat dimanfaatkan untuk melakukan fungsi pengawasan yang lebih fleksibel. Sistem berbasis object detection dapat digunakan sebagai solusi untuk menindak pelanggaran khususnya kendaraan sepeda motor yang seringkali diam di area zebra cross saat lampu merah sedang menyala. Hasil dari penelitian yang telah dilakukan pada tugas akhir ini menunjukan bahwa sistem pendeteksi pelanggar kendaraan sepeda motor pada zebra cross berbasis deteksi objek menggunakan algoritma YOLO ini mendapatkan hasil Precision 100%, Recall 100%, F1-Score 100%, Average IoU 82.23%, Average Loss 2.31%, mAP 99.99% serta akurasi yang didapatkan mencapai 99.66% dengan parameter yang digunakan adalah rasio data latih 70% : 30% data uji, Batchsize 64, Learning Rate 0.004, dan Max Batches 4000.

Kata kunci — deep learning, lalu lintas, deteksi objek, YOLO

Abstract —Traffic accidents these days should be a special concern for all of us. Accidents occur because of a lack of self-awareness from the public about the traffic rules themselves. The police as an institution that helps the public in educating the importance of obeying traffic rules must not be tired in carrying out their duties. One way of education carried out by the police is by means of traffic control. They go to the field to act against violators who do not comply with the rules, of course this method has several drawbacks, such as limited human resources and limited time. With that technology can be used to perform a more flexible supervisory function. Object detection-based systems can be used as a solution to act against violations, especially motorcycle vehicles that are often silent in the zebra cross area when the red light is on. The results of the research that has been carried out in this final project show that the detection system for motorcycle violators at the zebra cross based on object detection using the YOLO algorithm gets 100% Precision, 100% Recall, F1-Score 100%, Average IoU 82.23%, Average Loss 2.31%, mAP 99.99% and the accuracy obtained reaches 99.66% with the parameters used are training data ratio 70%:30% test data, Batchsize 64, Learning Rate 0.004, and Max Batches 4000.

Keywords— deep learning, traffic, object detection, YOLO

I. PENDAHULUAN

Kecelakaan lalu lintas sering terjadi akibat pelanggaran lalu lintas yang disebabkan oleh kelalalian manusia itu sendiri, seperti tidak mematuhi rambu lalu lintas, bertindak semena-mena selama berkendara dan hal lainnya. Dari data sendiri diketahui populasi kendaraan bermotor pada tahun 2018 adalah sebanyak 141.428.052 unit dan 81,58% populasi kendaraan bermotor adalah sepeda motor, dominasi sepeda motor itu sendiri meningkatkan faktor resiko keterlibatan sepeda motor pada kejadian kecelakaan lalu lintas. Dari dominasi ini pada tahun 2018 jumlah kecelakaan lalu lintas jalan yang melibatkan sepeda motor mencapai angka 73,49% yang artinya ada 196.457 kejadian kasus kecelekaan. [1]Untuk itu diperlukan kesadaran masyarakat untuk mengurangi kecelekaan yang terjadi, salah satunya dengan pengawasan lalu lintas yang dilakukan oleh kepolisian. Pengawasan lalu lintas ini menjadi salah satu faktor kunci dalam mengurangi kecelakaan lalu lintas yang sering terjadi, dengan pengawasan masyarakat menjadi lebih patuh terhadap aturan lalu lintas yang ada. Umumnya masyarakat lebih patuh karena adanya denda jika mereka tidak mematuhi aturan lalu lintas.

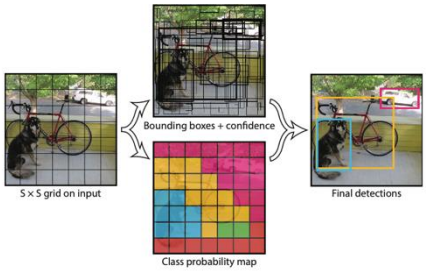
Dengan ini teknologi tentu menjadi suatu yang dapat diaplikasikan dalam meminimalisir pelanggaran lalu lintas yang tengah marak terjadi saat ini. Penggunaan *image processing* dalam pendeteksian objek dapat menjadi solusi yang dapat membantu pelaksanaan pengawasan pelanggaran lalu lintas [2]. Polisi tidak perlu lagi menjaga dengan terus melihat pengguna sepeda motor dijalan, dengan adanya sistem maka secara otomatis pelanggaran akan dideteksi dan diberikan pemberitahuan jika melanggar.

Pada Tugas Akhir ini dibuat sistem deteksi objek pelanggaran lalu lintas pada *zebra cross* untuk sepeda motor. Penggunaan algoritma *You Look Only Once* (YOLO) dianggap cocok untuk membantu pelaksanaan pengawasan ini dikarenakan cara kerja YOLO bisa dikatakan paling optimal di banding algoritma lainnya[2]. Algoritma YOLO dapat bekerja secara *real time* dan cepat tentunya dikarena hanya melalui *convolutional network* sekali saja. Cara kerja dari sistem ini adalah dengan mendeteksi kendaraan sepeda motor dan menghitung waktu sepeda motor tersebut berhenti di area *zebra cross*, lalu jika terbukti melanggar sistem akan mencetak informasi pelanggar seperti titik koordinat sepeda motor tersebut pada video dan mengirimkannya melalui notifikasi Telegram.

II. KAJIAN TEORI /MATERIAL DAN METODE/PERANCANGAN

A. You Look Only Once

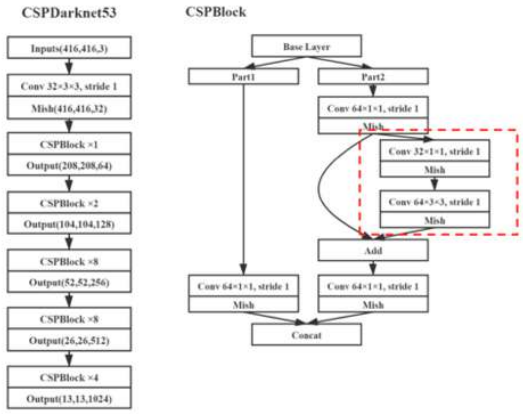
You Look Only Once (YOLO) adalah jaringan syaraf pintar untuk melakukan deteksi objek secara *real-time* . YOLO merupakan jaringan syaraf tunggal pada keseluruhan gambar. YOLO memiliki beberapa kelebihan diantara algoritma lain sejenis seperti lebih sederhana , lebih cepat, presisi yang baik, dapat mempelajari representasi objek yang dapat di generalisasikan[3]. Saat ini YOLO memiliki beberapa versi antara lain YOLO, YOLOv2, YOLO9000, YOLOv3, YOLOv4 dan YOLOv5.



GAMBAR 1
PROSES DETEKSI PADA YOLO [3]

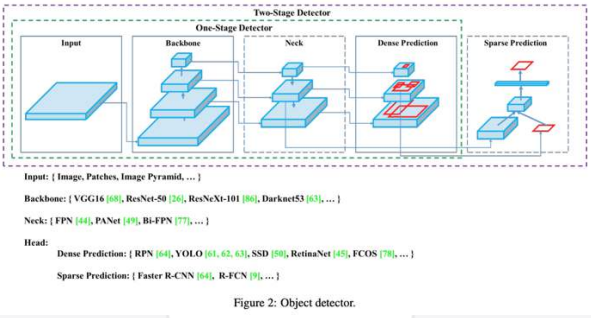
YOLO membagi *input* gambar menjadi *grid* berukuran $S \times S$ [3]. Grid-grid tersebut bertanggung jawab dalam menentukan deteksi objek. *Bounding box* dan nilai *confidence* akan diprediksi disetiap grid. Nilai *confidence* adalah nilai yang menyatakan seberapa yakin *bounding box* tersebut berisi objek dan keakuratan prediksinya[4] .YOLOv4 menggunakan *feature extractor* CSPDarknet53 yang yang merupakan pengembangan dari Darknet53,

dibawah ini adalah gambar dari CSPDarknet53[5]



GAMBAR 2
LAYER CSPDARKNET53[5]

Algoritma YOLOv4 utamanya terdiri dari 3 bagian yaitu *Backbone,Neck* dan *Head*. Untuk *Backbone* dan *Neck* sendiri memiliki fungsi untuk ekstraksi fitur dan agregasi dan untuk Head memiliki fungsi untuk mendeteksi atau memprediksi. Dalam algoritma YOLOv4 modul yang digunakan yaitu CSPDarknet53 (*Backbone*), PANet (*Neck*) dan YOLOv3 (*Head*) [6].



GAMBAR 3
NETWORK ARCHITECTURE OF YOLOV4 [6]

B. Parameter Performansi

Parameter performansi adalah parameter tolak ukur untuk menghitung bagaimana model yang telah dibuat berhasil mendeteksi objek yang diinginkan, semakin nilai-nilai parameter dalam performansi baik maka model yang telah dibuat dianggap berhasil. Berikut ini adalah beberapa parameter performansi :

1. *Confusion Matrix*

Confusion Matrix adalah tabel uji yang digunakan dalam mendeskripsikan kinerja model klasifikasi pada beberapa data uji yang diketahui nilainya[7]

2. *Accuracy*

Akurasi adalah rasio prediksi benar (positif) dengan keseluruhan data [7]. Untuk nilai Akurasi memiliki rumus sebagai berikut :

$$Accuracy = \frac{TP}{TP+FN+FP} \times 100\% \quad (1)$$

3. *Precision*

Precision adalah rasio prediksi benar positif dibandingkan dengan keseluruhan hasil yang diprediksi positif [7]. Untuk nilai *Precision* memiliki rumus sebagai berikut :

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

4. *Recall*

Recall adalah rasio prediksi benar positif dibandingkan dengan keseluruhan data yang benar positif dan salah negatif[7]. Untuk nilai *recall* memiliki rumus sebagai berikut :

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

5. *F1-Score*

F1-Score adalah nilai yang didapat untuk menunjukkan performa algorima yang diterapkan [7]. Untuk nilai *F1-score* memiliki rumus sebagai berikut :

$$F1 - Score = 2 \times \frac{Precision \times Recall}{Precision+Recall} \quad (4)$$

6. *Intersection Over Union*

IOU digunakan untuk menentukan kesamaan *bounding box* yang diprediksi dengan luas dari objek yang sebenarnya pada citra (*ground truth*). IOU memiliki persamaan perhitungan sebagai berikut [8]:

$$\text{IOU} = \frac{Area\ Overlap}{Area\ Union} \quad (5)$$

7. *mean Average Precision*

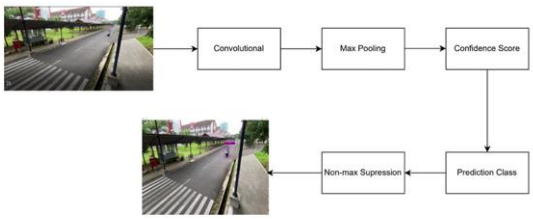
Mean Average Precision adalah nilai rata-rata yang membentuk metrik evaluasi untuk mengukur kinerja deteksi objek. Untuk *mAP* sendiri memiliki rumus sebagai berikut[8] :

$$mAP = \sum_{i=1}^{N} \frac{APi}{N} \times 100\% \quad (6)$$

III. METODE

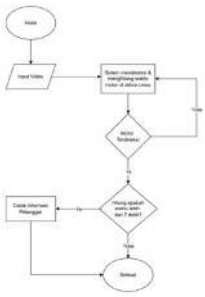
A. Proses algoritma YOLO

Perancangan model pelatihan mesin untuk algoritma YOLO dimulai dari menginput dataset kemudian dilakukan proses konvolusi selanjutnya dilakukan proses *max pooling* untuk mereduksi dimensi citra. Selanjutnya adalah proses dimana pemberian skor kepercayaan dan pelabelan kelas pada *bounding box*. Proses terakhir adalah proses *Non-Max Supression* yaitu mengeliminasi semua *bounding box* yang memiliki skor kepercayaan rendah dan membiarkan 1 dengan skor kepercayaan tertinggi.



GAMBAR 4
PROSES ALGORITMA YOLO

3.2 *Flowchart* Pendeteksi Pelanggaran



GAMBAR 5
FLOWCHART PENDETEKSI PELANGGARAN

Seperti terlihat pada gambar *flowchart*, awalnya akan diinputkan video untuk dideteksi apakah objek dengan *class* motor terdeteksi. Jika motor terdeteksi maka akan dihitung berapa lama motor tersebut berada pada zebra cross dengan ketentuan waktu motor di zebra cross hanya boleh sama dengan 7 detik. Jika motor melewati batas waktu ketentuan maka akan dianggap sebagai pelanggaran lalu sistem akan mencetak informasi pelanggar dan mengirimkannya melalui Telegram.

C. Anotasi Citra

Dataset yang telah dimiliki pada tahap ini dilakukan anotasi citra yaitu menggambarkan setiap *bounding box* kepada objek yang terdeteksi pada gambar kemudian setelah digambar akan disimpan file output berupa titik koordinat objek pada gambar.

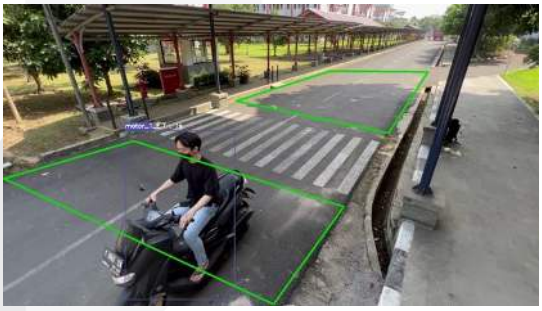
GAMBAR 6
PROSES PELABELAN

D. Deteksi Motor dan Penghitungan Waktu

Setelah proses *Training* dilakukan maka langkah selanjutnya adalah proses bagaimana model dapat mendeteksi waktu saat sepeda motor berada di area zebra cross, pertama-tama akan digambar garis berbentuk persegi panjang untuk membuat area 1 dan area 2, garis dibuat dengan memanfaatkan *library* OpenCV dengan menentukan titik koordinat tiap ujung persegi. Area 1 merupakan area dimana sistem mendeteksi objek kemudian ketika objek meninggalkan area 1 maka otomatis waktu penghitungan akan berjalan, kemudian waktu akan berhenti ketika objek telah sampai di area 2. *Bounding Box* pada objek juga akan menampilkan keterangan *Tracker ID* seperti terlihat pada gambar yaitu motor_1 dan juga estimasi waktu yang terlihat pada gambar yaitu 1s (1 detik)

```
          #x2          x1          x3          x4
area_1 = [(1169, 238), (832, 345), (1377, 460), (1518, 241)]
area_2 = [(714, 495), (63, 736), (823, 1070), (1182,793)]
area_3 = [(465, 488), (2, 623), (966, 1072), (1217,723)]
```

Gambar 7. Penggambaran Area berbentuk Persegi Panjang



Gambar 8. Skenario Pendeteksi Pelanggaran

IV. HASIL DAN PEMBAHASAN

A. Pengujian *Hyperparameter Training*

1. Pengujian Rasio Dataset

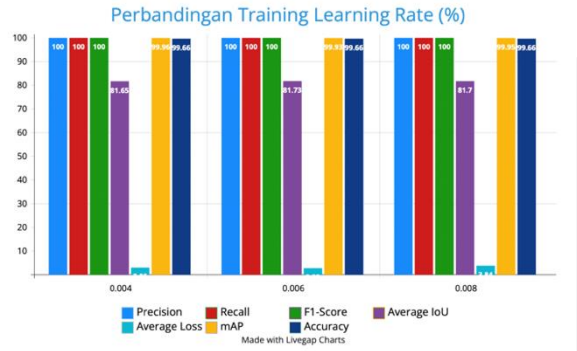
Pengujian Rasio Dataset Setelah dilakukan pelatihan dengan membagi dataset rasio mulai dari 70% data latih : 30% data uji, 80% data latih : 20% data uji, dan 90% data latih : 10% data uji maka dapat diketahui bahwa rasio 90 % data pelatihan: 10% data pengujian menghasilkan model dengan performa terbaik. Ini karena semakin banyak

data yang dilatih, semakin baik model dalam pembelajaran mesin. Konfigurasi awal yang dipakai adalah *Batchsize* = 64, *Learning rate* = 0.001, *Max Batches* = 4000. Setelah dilakukan *training* didapati hasil terbaik berada pada rasio dataset 70:30. Hasil visual dapat dilihat pada grafik dibawah ini.

GAMBAR 10
GRAFIK PERBANDINGAN RASIO DATASET

2. Pengujian *Learning Rate*

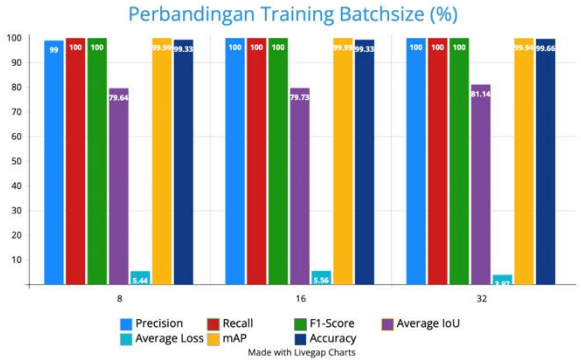
Pengujian l*earning rate* dilakukan menggunakan 3 nilai *learning rate* berbeda yaitu 0.004, 0.006 dan 0.008 . Konfigurasi yang digunakan adalah *Batchsize* = 64, rasio data latih 70%:30% data uji, *Max Batches* = 4000. Setelah dilakukan *training* didapatkan nilai *learning rate* terbaik berada pada nilai 0.004. Hasil visual dapat dilihat pada grafik dibawah ini.



GAMBAR 11
GRAFIK PERBANDINGAN *LEARNING RATE*

3. Pengujian *Batchsize*

Pengujian *Batchsize* dilakukan menggunakan 3 nilai *Batchsize* berbeda yaitu 8,16 dan 32. Konfigurasi yang digunakan adalah *Learning rate* = 0.004, rasio data latih 70%:30% data uji, *Max Batches* = 4000. Setelah dilakukan *training* didapatkan nilai *Batchsize* terbaik berada pada nilai 64. Hasil visual dapat dilihat pada grafik dibawah ini.



GAMBAR 12
GRAFIK PERBANDINGAN *BATCHSIZE*

4. Pengujian *Max Batches*

Pengujian *Max Batches* dilakukan menggunakan 3 nilai *Max Batches* berbeda yaitu 1000, 2000 dan 3000. Konfigurasi yang digunakan adalah *Learning rate* = 0.004, rasio data latih 70%:30% data uji, *Batchsize* = 64. Setelah dilakukan *training* didapatkan nilai *Max Batches* terbaik berada pada nilai 4000 yaitu konfigurasi awal yang digunakan. Hasil visual dapat dilihat pada grafik dibawah ini.

GAMBAR 13
GRAFIK PERBANDINGAN *MAX BATCHES*

B. Pengujian Sistem

Pengujian Sistem dilakukan untuk mengetahui pada kondisi sebenarnya bagaimana sistem dapat mendeteksi objek dengan benar berdasarkan beberapa parameter pengujian seperti kecepatan, jarak dan lux.

TABEL 1
PENGUJIAN JARAK, KECEPATAN DAN LUX

Parameter	Nilai	Akurasi
Jarak (Meter)	0 - 6	100%
	7 - 12	100%
	13 - 18	100%
Kecepatan (Km/h)	10	100%
	20	100%
	30	100%
	30 (percepatan video 5x)	100%
Lux	7	66.7%
	294	100%

C. Pengujian Dataset Negatif & Cross Validation

Pengujian ini dilakukan dengan penambahan dataset negatif dengan pembagian dataset adalah sebanyak 600 dataset negatif dan 600 dataset positif. Selain itu dilakukan juga pengujian *cross validation* untuk mengetahui akurasi maksimal yang bisa didapatkan model. Untuk pengujian *cross validation* digunakan nilai k=3.Setelah dilakukan pengujian didapatkan akurasi terbaik pada 3-fold adalah sebesar 79.61%. Berikut ini adalah detail pengujian dataset negatif dan *cross validation*:

TABEL 2
PENGUJIAN *CROSS VALIDATION*

K-Fold	*Precision*	*Recall*	*F1-Score*	*Average IoU*	*Average Loss*	*mAP*	*Accuracy*
1	96%	90%	93%	78.98%	3.86%	92.32%	87.08%
2	81%	82%	82%	66.07%	3.36%	83.91%	69.04%
3	93%	88%	91%	76.95%	4.31%	88.56%	82.73%

V. KESIMPULAN & SARAN

A. Kesimpulan

Setelah pengujian parameter pada algoritma YOLO selesai dilakukan, didapatkan performansi model terbaik ada pada rasio 70% data latih : 30% data uji, *Batchsize* 64, *Learning Rate* 0.004, dan *Max Batches* 4000 dengan hasil *Precision* 100%, *Recall* 100%, *F1-Score* 100%, *Average IoU* 82.23%, *Average Loss* 2.31% dan *mAP* 99.99% serta akurasi yang dihasilkan mencapai 99.66%.

B. Saran

Untuk penelitian kedepannya diharapkan pengembangan dapat mendeteksi plat nomor kendaraan untuk disimpan di *database* juga sistem dapat memotret objek yang melanggar untuk disimpan didalam *database* juga.

REFERENSI

[1] "Sepeda Motor Penyumbang Kecelakaan Terbesar di Jalan Raya." https://bisnisnews.id/detail/berita/sepeda-motor-penyumbang-kecelakaan-terbesar-di-jalan-raya (accessed Dec. 17, 2021).

[2] J. Du, "Understanding of Object Detection Based on CNN Family and YOLO," in *Journal of Physics: Conference Series*, Apr. 2018, vol. 1004, no. 1. doi: 10.1088/1742-6596/1004/1/012029.

[3] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," Jun. 2015, [Online]. Available: http://arxiv.org/abs/1506.02640

[4] "What is a Raspberry Pi?" https://www.raspberrypi.org/help/what-%20is-a-raspberry-pi/ (accessed Dec. 17, 2021).

[5] P. Xu *et al.*, "On-board real-time ship detection in hisea-1 sar images based on cfar and lightweight deep learning," *Remote Sens (Basel)*, vol. 13, no. 10, May 2021, doi: 10.3390/rs13101995.

[6] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "YOLOv4: Optimal Speed and Accuracy of Object Detection," Apr. 2020, [Online]. Available: http://arxiv.org/abs/2004.10934

[7] M. Vakili, M. Ghamsari, and M. Rezaei, "Performance Analysis and Comparison of Machine and Deep Learning Algorithms for IoT Data Classification."

[8] M. Ahmed *et al.*, "Survey and Performance Analysis of Object Detection in Challenging Environments," 2021, doi: 10.20944/preprints202106.0590.v1.