

PEMANFAATAN YOLOV4 UNTUK DETEKSI PELANGGARAN HELM DAN MASKER SERTA IDENTIFIKASI PELAT NOMOR MENGGUNAKAN TESSERACT-OCR

Rohmat Syamsul Huda¹, Resty Wulanningrum², Daniel Swanjaya³

Program Studi Teknik Informatika, Fakultas Teknik, Universitas Nusantara PGRI Kediri
Jl. Ahmad Dahlan No.76, Mojoroto, Kec. Mojoroto, Kota Kediri
Telp. 0813-3525-7409

E-mail: rohmathuda123@gmail.com, resty0601@gmail.com, daniel@unpkediri.ac.id
087880144906

ABSTRAKS

Dunia sedang dilanda pandemi sehingga mengharuskan manusia memakai masker saat berada di luar ruangan. Dalam rangka mencegah persebaran virus dan memastikan ketertiban pengendara sepeda motor, penting bagi pengendara sepeda motor memakai helm dan masker secara bersamaan. Oleh sebab itu dibuatlah program yang dapat mendeteksi pelanggaran helm dan masker serta mendapatkan nomor plat pelanggar secara otomatis. Penelitian ini menggunakan metode transfer learning YOLOv4 dan memanfaatkan Tesseract-OCR. YOLOv4 mampu mendeteksi objek helm, masker, sepeda motor dan plat dalam satu gambar. Dengan menggunakan dataset sejumlah 600 gambar menghasilkan 8 model, model dengan Mean Average Precision (mAP) tertinggi 93.38% dan F1-Score 0.77. model dengan F1-Score tertinggi 0.86 dengan mAP 88.78%. Dapat disimpulkan bahwa model dengan F1-Score tertinggi lebih baik dalam deteksi dan klasifikasi objek. Sementara pelatihan model Tesseract membantu meningkatkan identifikasi karakter pelat nomor.

Kata Kunci: deteksi objek, pengenalan karakter, yolo, deep learning

ABSTRACT

The world is being hit by a pandemic that requires humans to wear masks when outdoors. In order to prevent the spread of the virus and ensure orderliness for motorcyclists, it is important for motorcyclists to wear helmets and masks at the same time. Therefore, a program was created that can detect helmet and mask violations and get the violator's plate number automatically. This study uses the YOLOv4 transfer learning method and utilizes Tesseract-OCR. YOLOv4 is able to detect helmets, masks, motorcycles and license plates in one image. Using a dataset of 600 images resulted in 8 models, the model with the highest Mean Average Precision (mAP) 93.38% and F1-Score 0.77. the model with the highest F1-Score of 0.86 with an mAP of 88.78%. It can be concluded that the model with the highest F1-Score is better at detecting and classifying objects. While training the Tesseract model helps improve the identification of number plate characters.

Keywords: object detection, character recognition, yolo, deep learning

1. PENDAHULUAN

1.1 Latar Belakang

Dunia diguncang oleh COVID-19 sejak akhir Desember 2019 (Yuliana 2020). Kemudian pemerintah menetapkan Kedaruratan Kesehatan Masyarakat *Coronavirus Disease 2019* (COVID-19) disusul himbauan memakai masker dan penyediaan sarana cuci tangan memakai sabun (Keppres 2020)(Kemenkes 2021).

Kewajiban memakai masker berimbas terhadap aktivitas berkendara sepeda motor, sepeda motor sendiri merupakan kendaraan roda dua favorit masyarakat Indonesia (BPS 2022b). Demi menjaga keselamatan bersama, pengendara sepeda motor diwajibkan menggunakan helm sesuai standar guna menekan tingginya jumlah kecelakaan (Republik Indonesia 2009)(BPS 2022a). Oleh sebab itu, penting bagi pengendara sepeda motor memakai masker dan helm secara bersamaan untuk memutus persebaran virus serta menekan angka kecelakaan.

Dalam mengatasi permasalahan tersebut, pemanfaatan teknologi dibutuhkan agar dapat mengawasi pengendara sepeda motor yang melakukan pelanggaran. Cabang ilmu komputer yang mendukung sistem tersebut adalah *computer vision*, yang di dalamnya memuat aspek deteksi objek (Dong, Wang, and Abbas 2021). Deteksi objek sendiri telah mengalami peningkatan yang sangat signifikan setelah lahirnya YOLO (*You Only Look Once*), salah satu detektor satu tahap (Zaidi et al. 2022). Keunggulan detektor satu tahap adalah kecepatan dalam deteksinya, di mana tidak membutuhkan *region proposal* seperti keluarga CNN pada umumnya (Zaidi et al. 2022).

Beragam kajian telah dilakukan menggunakan algoritma YOLO. Modifikasi YOLOv2 digunakan untuk mendeteksi masker dalam (Loey et al. 2021). (Maliye et al. 2021) menggunakan YOLOv3 untuk mendeteksi pelanggaran helm atau masker, di mana pelat motor

pelanggar di deteksi menggunakan deteksi tepi *canny*. (Wu et al. 2019) memodifikasi YOLOv3 untuk mendeteksi helm pelindung pekerja. YOLOv4 digunakan untuk mendeteksi karakter pelat kendaraan Korea dalam (Sung, Yu, and Korea 2020).

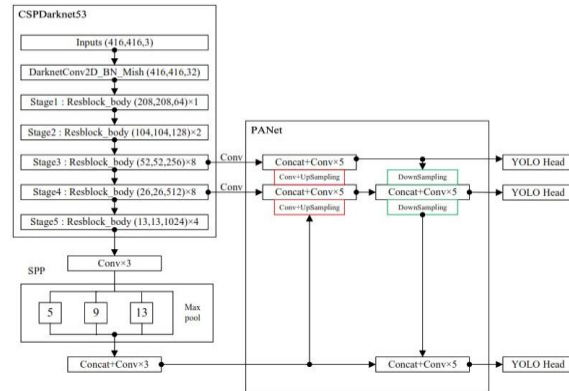
Berdasarkan kajian di atas, peneliti bertujuan untuk mendeteksi empat kelas citra yang terdiri dari sepeda motor, helm, masker dan pelat, kemudian jika masker atau helm tidak terdeteksi maka dilakukan identifikasi nomor pada pelat motor terkait. Kajian ini menggunakan model *transfer learning* YOLOv4 dan proses pengenalan karakter pelat menggunakan Tesseract – OCR.

1.2 Referensi

1.2.1 YOLOv4

Transfer learning merupakan suatu teknik CNN untuk mentransfer parameter jaringan saraf yang sudah dilatih pada suatu *dataset* sehingga dapat digunakan untuk mengekstrak fitur dari *dataset* lain secara otomatis (Raza and Hong 2020). Model *transfer learning* yang digunakan pada penelitian ini adalah YOLOv4 yang sudah dilatih sebelumnya pada *dataset* COCO dengan objek sebanyak 80 (Padilla, Netto, and Da Silva 2020).

YOLOv4 sebagai detektor satu tahap mempunyai 3 komponen penting yakni blok tulang punggung, blok leher dan blok kepala. Blok tulang punggung bertanggung jawab untuk mengekstrak fitur gambar, di mana YOLOv4 menggunakan tulang punggung CSPDarknet53 (Bochkovskiy, Wang, and Liao 2020). Kemudian blok leher digunakan untuk memperoleh fitur yang lebih efektif (menghasilkan piramida fitur) dengan menambahkan lapisan ekstra antara tulang punggung dengan kepala, blok leher menggunakan *Path Aggregation Network* (PANet) yang dimodifikasi dan *Spatial Pyramid Pooling* (SPP) (Bochkovskiy et al. 2020). Terakhir blok kepala menggunakan YOLOv3 yang berfungsi untuk menemukan kotak pembatas (*bounding box*) dan mengklasifikasikan apa yang ada di dalam setiap kotak dengan menerapkan kotak jangkar (*anchor box*) pada peta fitur dan menghasilkan vektor keluaran akhir dengan probabilitas kelas, skor objek, dan kotak pembatas (*bounding box*) (Redmon and Farhadi 2018). Arsitektur YOLOv4 seperti pada Gambar 1.

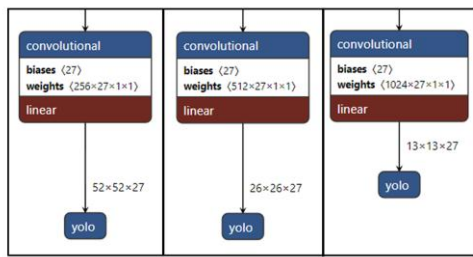


Gambar 1. Arsitektur YOLOv4

Pada blok tulang punggung, Darknet53 merupakan tulang punggung YOLOv3, angka 53 mengacu pada banyaknya layer konvolusi pada jaringan tersebut (Redmon and Farhadi 2018). Sementara CSP (*Cross Stage Partial connections*) adalah ide membagi lapisan layer menjadi 2 bagian. Satu lapisan yang akan melalui blok konvolusi dan satunya tidak yang kemudian akan dibandingkan hasilnya (Wang et al. 2020). Darknet berisi 5 modul residu, dan CSPDarknet53 menggunakan konvolusi 1×1 untuk membagi lapisan saluran peta fitur *input* menjadi dua sebelum ke setiap jaringan residu, dan menambahkan CSP setelah setiap modul residu besar (Chen et al. 2020). Fungsi *aktivasi* utama dalam CSPDarknet53 adalah fungsi *aktivasi Mish* (Bochkovskiy et al. 2020).

Kemudian pada blok leher SSP menggunakan 4 ukuran *kernel* geser yang berbeda yakni (1×1) , (5×5) , (9×9) dan (13×13) untuk mengonvolusikan kandidat gambar, dan kemudian menerapkan *Multiscale Maxpooling* untuk mendapatkan dimensi yang sama dari peta fitur (Bochkovskiy et al. 2020). SPP memungkinkan ukuran spasial dari setiap kandidat peta untuk dipertahankan, dan kemudian menghubungkan peta fitur dengan ukuran inti yang berbeda sebagai *output*, dengan *output* berupa peta fitur ukuran tetap (Chen et al. 2020). PANet mengusulkan ROI Pooling (*Region of Interest Pooling*) yang lebih fleksibel yang dapat mengekstrak dan mengintegrasikan fitur pada berbagai skala (Chen et al. 2020).

Terakhir peta fitur dari keluaran skala yang berbeda oleh PANet disambung, dan setelah operasi konvolusi, 3 kepala skala yang berbeda dapat diperoleh (Chen et al. 2020). Ukuran masukkan kepala saat pelatihan (416×416) seperti pada Gambar 2 adalah sebagai berikut $[(52 \times 52 * 3 * (4 + 1 + (\text{kelas}=4)))$, $(26 \times 26 \times 3 * (4 + 1 + (\text{kelas}=4)))$, $(13 \times 13 \times 3 * (4 + 1 + (\text{kelas}=4)))]$. Di mana 4 adalah nilai koordinat kotak kalibrasi, 1 adalah tingkat kepercayaan kotak kalibrasi, kelas adalah jumlah kategori, setiap kepala berisi 3 kotak pembatas, dan 3 kepala skala yang berbeda dapat digunakan untuk mendeteksi objek dari ukuran yang berbeda (Redmon and Farhadi 2018).



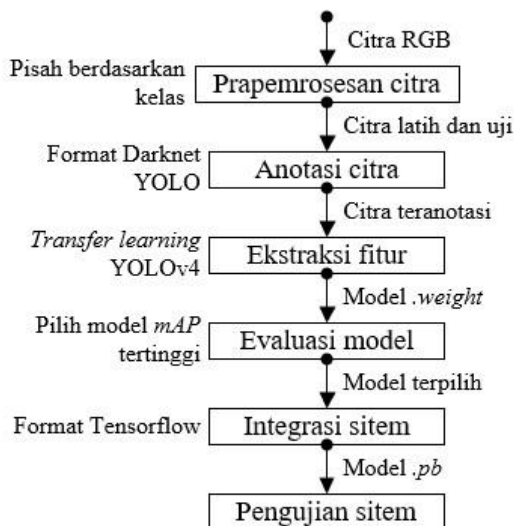
Gambar 2. Ukuran fitur YOLOv4 untuk head

1.2.2 Tesseract-OCR

Optical Character Recognition (OCR) adalah metode paling umum untuk mengubah teks dan gambar menjadi bentuk digital sehingga dapat dimanipulasi oleh mesin (Sabu and Das 2018). Tesseract sendiri sejak 2006 dikembangkan oleh Google dengan lisensi Apache 2.0 yang berarti library OCR tersebut bebas digunakan (*open source*) (Smith 2007).

2. PEMBAHASAN

Tahapan pada penelitian ini terdiri dari tahap *preprocessing* citra, anotasi citra, *transfer learning* YOLOv4, evaluasi model YOLOv4, integrasi sistem dan pengujian sistem. Tahapan dimulai dengan mengumpulkan data citra, memisahkan citra berdasarkan kelas, melakukan pemotongan dan *penskalaan*, membagi data menjadi data latih dan data uji. Menganotasi citra dengan format YOLO. Proses *transfer learning* dengan YOLOv4 dilakukan untuk mendapatkan hasil ekstraksi fitur. Evaluasi model YOLOv4 dilakukan untuk memilih model berdasarkan *mAP* tertinggi. Mengintegrasikan model terpilih dengan mengubah format model YOLO menjadi format model yang mendukung dengan *TensorFlow*. Pengujian sistem dilakukan untuk mengetahui kinerja dari sistem yang dibuat.



Gambar 3. Tahapan penelitian

2.1.1 Prapemrosesan citra

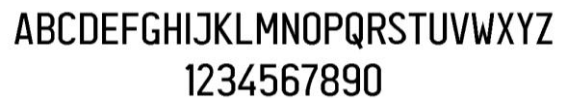
Data menggunakan 600 citra, seperti ditunjukkan pada Gambar 4. Data masker dan pelat diperoleh dari situs (Kaggle 2020)(Kaggle 2018), sementara data sepeda motor dan helm diperoleh dari situs *google images*. Saat pelatihan, ukuran citra tidak boleh terlalu besar sehingga ukuran 416 x 416 *pixels* digunakan agar tidak menghilangkan karakteristik sampel setelah diubah ukurannya (Lu et al. 2018). Kemudian membagi data tersebut menjadi data latih dan data uji untuk pelatihan model YOLOv4. Untuk meningkatkan akurasi pengenalan pelat, model Tesseract dilatih dengan jenis huruf yang diperoleh dari situs (dafont 2020), ditunjukkan pada Gambar 5.

Kemudian data di atas di atas dilakukan pemotongan dan penskalaan sehingga citra menjadi satu ukuran yakni 416 x 416 *pixels*, kemudian membagi 600 data tersebut menjadi data latih dan data uji model YOLOv4 dengan perbandingan 568:32 seperti perincian pada Tabel 1.



Gambar 4 Dataset untuk pelatihan YOLOv4:

a) Sepeda motor, b) helm, c) masker, d) plat



Gambar 5. Jenis huruf untuk pelatihan Tesseract

Tabel 1. Rincian pembagian data

No.	Kelas	Format	Jumlah	Latih	Uji
1	Sepeda motor	Jpg	150	138	12
2	Helm	Jpg	150	138	12
3	Masker	Png	150	137	13
4	Plat	Jpg,Jpeg	150	137	13
Total			600	568	32

Kemudian data uji sistem diperoleh melalui kamera *smartphone* dengan total 3 data seperti ditunjukkan pada Gambar 6. Data tersebut diambil dari pengendara sepeda motor dengan 3 kondisi sebagai berikut :

- Memakai helm dan masker
- Hanya memakai helm
- Hanya memakai masker



Gambar 6. Data uji sistem

2.1.2 Anotasi data

Anotasi data atau pelabelan digunakan untuk membuat kotak pembatas pada objek dan memberi kelas yang sesuai pada objek tersebut, sehingga membuat objek dapat dikenali oleh mesin. Pada format anotasi YOLO, setiap citra akan memiliki anotasi yang berbentuk *file .txt* dengan setiap *file .txt* tersebut memuat [kelas, x, y, lebar, tinggi]. Anotasi data dilakukan dengan aplikasi LabelImg seperti pada Gambar 7.



Gambar 7. Anotasi data dengan LabelImg

2.1.3 Ekstraksi fitur

YOLOv4 dapat di akses pada *github* berikut <https://github.com/AlexeyAB/darknet>, kemudian membangunnya menggunakan aplikasi *Cmake* agar bisa digunakan pada komputer lokal. Ekstraksi fitur memerlukan 3 konfigurasi yakni *obj.names*, *obj.data* dan *yolov4-obj.cfg*. *obj.names* berisi keempat nama kelas, sementara *obj.data* berisi *classes = 4*, *train = data/train.txt*, *valid = data/test.txt*, *names = data/obj.names*. Terakhir *yolov4-obj.cfg* berisi parameter yang digunakan *batch 64*, *subdivisions 16*, *classes 4*, *max_batches 8000 (classes x 2000)*, *steps 6400,7200 (80% dan 90% dari max_batches)*, *width 416*, *height 416* dan *filters 27 ((classes + 5) x 3)*. Total iterasi adalah 8000, setiap 1000 iterasi akan menghasilkan 1 model sehingga total model yakni 8.

2.1.4 Evaluasi model

Untuk memilih model yang akan diintegrasikan dengan sistem maka perlu melakukan pengecekan *mAP (mean Average Precision)* atau *F1-Score*. *mAP* adalah metrik yang digunakan untuk mengukur akurasi detektor objek di semua kelas

dalam basis data, di mana *mAP* merupakan *AP* rata-rata di semua kelas (Padilla et al. 2020). Semakin tinggi skor *mAP* pada suatu model, maka semakin akurat model dalam pendeteksiannya. Sementara *F1-Score* merupakan *harmonic mean* dari *precision* dan *recall* dengan rentang 0. Sampai 1.0 yang berarti jika *F1-Score* punya skor yang baik mengindikasikan bahwa model tersebut mempunyai *precision* dan *recall* yang baik

Evaluasi model dinyatakan dalam Tabel 2. Terlihat bahwasanya *mAP* tertinggi berada pada iterasi 2000 (model 2) 93.38%. Sementara *F1-Score* tertinggi berada pada iterasi 6000 (model 6) 0.86. Kedua model tersebut kemudian dipilih untuk dilakukan pengujian sistem.

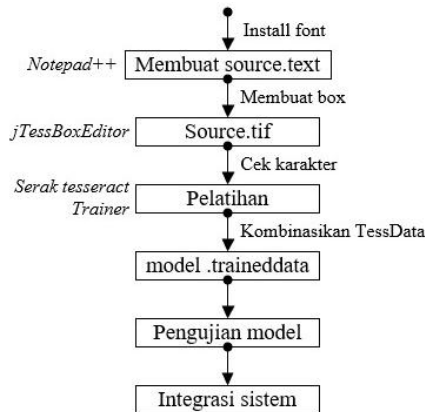
Tabel 2. Evaluasi model pelatihan YOLOv4

Iterasi	mAP	F1-Score	Objek	AP	TP	FP
1000	91,91%	0.78	Helm	90.62	20	6
			Masker	88.63	37	15
			Motor	98.32	22	17
			Plat	90.08	21	9
2000	93.38%	0.77	Helm	96.83	20	5
			Masker	87.89	36	21
			Motor	98.33	26	12
			Plat	90.46	21	14
3000	91.95%	0.85	Helm	93.66	19	3
			Masker	90.56	37	11
			Motor	94.92	26	7
			Plat	88.66	21	7
4000	92.04%	0.83	Helm	93.27	19	2
			Masker	88.34	36	18
			Motor	96.76	26	6
			Plat	89.78	21	6
5000	90.48%	0.85	Helm	88.28	18	2
			Masker	91.44	37	14
			Motor	93.63	26	6
			Plat	88.56	21	4
6000	88.78%	0.86	Helm	80.16	16	1
			Masker	93.96	38	10
			Motor	92.34	26	4
			Plat	88.67	21	7
7000	92.28%	0.85	Helm	87.97	17	3
			Masker	93.13	37	13
			Motor	93.15	26	5
			Plat	94.89	22	6
8000	92.40%	0.85	Helm	88.01	17	3
			Masker	93.20	37	15
			Motor	93.31	26	5
			Plat	95.09	22	3

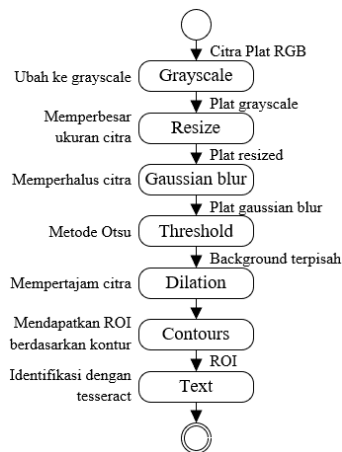
2.1.5 Integrasi sistem

Untuk dapat menggunakan model YOLOv4 menggunakan TensorFlow, maka model terpilih harus diubah terlebih dahulu formatnya menjadi format yang didukung yakni *model .pb*. Selain mengintegrasikan model YOLOv4, pada penelitian ini juga mengintegrasikan model dari pelatihan Tesseract yang sudah dilatih menggunakan huruf seperti pada Gambar 5. Proses identifikasi pelat nomor seperti pada Gambar 9. Hasil evaluasi Tesseract menunjukan bahwasanya identifikasi dengan menggunakan model Tesseract yang dilatih membantu Tesseract mengenali karakter lebih baik

daripada tidak menggunakan model seperti pada Tabel 3.



Gambar 8. Pelatihan model Tesseract



Gambar 9. Identifikasi karakter pelat nomor

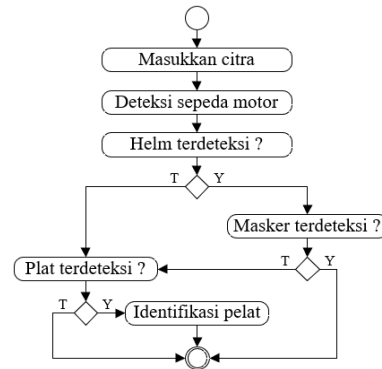
Terdapat beberapa tahapan yang dilakukan dalam proses identifikasi karakter pelat nomor seperti yang ditampilkan pada gambar diatas.

Tabel 3. Evaluasi model Tesseract

Nomor plat	Tanpa model		Dengan model	
	Prediksi	TP	Prediksi	TP
	A?6?53RAX	6	AG6263RAX	8
	A533???BN	2	A6380?YBA	6
	A?48?5?AV	4	AG5?75?AX	6

2.1.6 Pengujian sistem

Pengujian dilakukan terhadap sistem yang telah dibuat dengan memasukkan data uji (Gambar 6) sehingga diperoleh *confidence score* dari objek dan mengidentifikasi nomor pelat apabila terjadi pelanggaran. Tahapan yang dilakukan pada pengujian sistem adalah dengan melakukan pemanggilan pada bobot yang telah diubah ke dalam model TensorFlow kemudian alurnya seperti pada Gambar 10.



Gambar 10. Alur pengujian sistem

a. Pengujian dengan model mAP tertinggi
 Pengujian sistem menggunakan model 2 dinyatakan dalam Tabel 4, terlihat bahwasanya YOLOv4 tidak berhasil mendeteksi masker pada data uji A. Kemudian melakukan identifikasi karakter pelat dinyatakan dalam Tabel 5.

Tabel 4. Hasil pengujian dengan model mAP

No.	Data uji Kondisi	Deteksi objek			
		Motor	Helm	Masker	Plat
1	A	89%	99%	Tidak terdeteksi	100%
2	B	70%	99%	pelanggaran	100%
3	C	86%	pelanggaran	99%	85% dan 100%

Tabel 5. Hasil identifikasi karakter

No.	Data Uji		Nomor plat	Terbaca
	Kondisi			
1	A		AG3804YBA	AG3804YB
2	B			CGO
3	C			G60 dan ?IQI

b. Pengujian dengan model F1-Score tertinggi
 Pengujian sistem menggunakan model 6 dinyatakan dalam Tabel 6, terlihat bahwasanya YOLOv4 berhasil mendeteksi dan memprediksi objek tersebut dengan benar. Kemudian melakukan identifikasi karakter pelat pada citra uji B dan C yang dinyatakan dalam Tabel 7.

Tabel 6. Hasil pengujian dengan model F1-Score







No	Data uji		Deteksi objek			
	Kondisi		Motor	Helm	Masker	Plat
1	A		97%	99%	93%	86%
2	B		80%	99%	Pelanggaran	99%
3	C		90%	pelanggaran	99%	98%

Tabel 7. Hasil identifikasi karakter






No.	Data Uji		Nomor plat	Terbaca
	Kondisi			
1	B		AG3804YBA	CEO
2	C			AG3804IBE

Berikut ini adalah hasil deteksi dari kedua pengujian di atas yang dinyatakan dalam Tabel 8 dan Tabel 9 berisi citra hasil identifikasi pelat nomor.

Tabel 8. Citra hasil deteksi

Citra Uji	Dengan mAP	Dengan F1-Score
A		
B		
C		

Tabel 9. Citra hasil identifikasi pelat nomor

Citra Uji	Dengan mAP	Dengan F1-Score
A		-
B		
C		

3. KESIMPULAN

YOLOv4 sebagai detektor satu tahap berhasil mendeteksi berbagai objek dalam 1 gambar seperti sepeda motor, helm, masker dan pelat. Dengan menggunakan 600 data pelatihan didapatkan model dengan mAP tertinggi berada pada iterasi 2000 yakni 93.38% dan F1-Score 0.77, sementara F1-Score tertinggi berada pada iterasi 6000 yakni 0.86 dengan mAP 88.78%. Dari pengujian sistem yang dilakukan dapat disimpulkan bahwa model dengan F1-Score tertinggi lebih baik daripada model dengan mAP tertinggi. Kemudian pelatihan model Tesseract

cukup meningkatkan pengenalan karakter plat. Sehingga sistem ini berhasil mendeteksi pelanggaran helm atau masker kemudian otomatis mengidentifikasi karakter pelat nomor terkait.

Untuk meningkatkan deteksi diperlukan penambahan variasi *dataset*, sementara untuk meningkatkan pengenalan karakter plat diperlukan metode tambahan berupa *Geometric Transformations* atau mengganti *Tesseract* dengan melatih model karakter pelat sehingga deteksi karakter juga menggunakan YOLOv4.

PUSTAKA

- Bochkovskiy, Alexey, Chien-Yao Wang, and Hong-Yuan Mark Liao. 2020. "YOLOv4: Optimal Speed and Accuracy of Object Detection."
- BPS. 2022a. "Jumlah Kecelakaan, Korban Mati, Luka Berat, Luka Ringan, Dan Kerugian Materi 2017-2019." Retrieved April 19, 2022 (<https://www.bps.go.id/indicator/17/513/1/jumlah-kecelakaan-korban-mati-luka-berat-luka-ringan-dan-kerugian-materi.html>).
- BPS. 2022b. "Perkembangan Jumlah Kendaraan Bermotor Menurut Jenis (Unit), 2018-2020." Retrieved April 19, 2022 (<https://www.bps.go.id/indicator/17/57/1/perkembangan-jumlah-kendaraan-bermotor-menurut-jenis.html>).
- Chen, Qingqiang, Qianghua Xiong, Qingqiang Chen, and Qianghua Xiong. 2020. "Garbage Classification Detection Based on Improved YOLOV4." *Journal of Computer and Communications* 8(12):285-94. doi: 10.4236/JCC.2020.812023.
- dafont. 2020. "Plat Nomor Font." Retrieved December 8, 2021 (<https://www.dafont.com/plat-nomor.font>).
- Dong, Shi, Ping Wang, and Khushnood Abbas. 2021. "A Survey on Deep Learning and Its Applications." *Computer Science Review* 40.
- Kaggle. 2018. "Indonesian Plate Number." Retrieved November 25, 2021 (<https://www.kaggle.com/imamdigmi/indonesian-plate-number>).
- Kaggle. 2020. "Face Mask Detection." Retrieved November 25, 2021 (<https://www.kaggle.com/andrewmvd/face-mask-detection>).
- Kemenkes. 2021. "Surat Edaran Kementerian Kesehatan Republik Indonesia Nomor HK.02.02/I/385/2020."
- Keppres. 2020. "Keputusan Presiden Republik Indonesia Nomor 11 Tahun 2020 Tentang Penetapan Kedaruratan Kesehatan Masyarakat Corona Virus Disease 2019 (COVID-19)."
- Loey, Mohamed, Gunasekaran Manogaran, Mohamed Hamed N. Taha, and Nour Eldeen M. Khalifa. 2021. "Fighting against COVID-19: A Novel Deep Learning Model Based on YOLO-v2 with ResNet-50 for Medical Face

- Mask Detection.” *Sustainable Cities and Society* 65. doi: 10.1016/j.scs.2020.102600.
- Lu, Junyan, Chi Ma, Li Li, Xiaoyan Xing, Yong Zhang, Zhigang Wang, and Jiuwei Xu. 2018. “A Vehicle Detection Method for Aerial Image Based on YOLO.” *Journal of Computer and Communications* 06(11):98–107. doi: 10.4236/jcc.2018.611009.
- Maliye, Shravani, Jayom Oza, Jayesh Rane, and Nileema Pathak. 2021. “Mask and Helmet Detection in Two-Wheelers Using YOLOv3 and Canny Edge Detection.” *Www.Irjet.Net* 08(04).
- Padilla, Rafael, Sergio L. Netto, and Eduardo A. B. Da Silva. 2020. “A Survey on Performance Metrics for Object-Detection Algorithms.” in *International Conference on Systems, Signals, and Image Processing*. Vols. 2020-July.
- Raza, Kazim, and Song Hong. 2020. “Fast and Accurate Fish Detection Design with Improved Yolo-v3 Model and Transfer Learning.” *International Journal of Advanced Computer Science and Applications* (2). doi: 10.14569/ijacsa.2020.0110202.
- Redmon, Joseph, and Ali Farhadi. 2018. “YOLOv3: An Incremental Improvement.”
- Republik Indonesia. 2009. “Undang-Undang Republik Indonesia Nomor 22 Tahun 2009 Tentang Lalu Lintas Dan Angkutan Jalan.”
- Sabu, Abin M., and Anto Sahaya Das. 2018. “A Survey on Various Optical Character Recognition Techniques.” in *Proc. IEEE Conference on Emerging Devices and Smart Systems, ICEDSS 2018*.
- Smith, Ray. 2007. “An Overview of the Tesseract OCR Engine.” in *Proceedings of the International Conference on Document Analysis and Recognition*. IEEE.
- Sung, Ju Yeong, Saet Byeol Yu, and Se Ho Park Korea. 2020. “Real-Time Automatic License Plate Recognition System Using YOLOv4.” in *2020 IEEE International Conference on Consumer Electronics - Asia, ICCE-Asia 2020*.
- Wang, Chien Yao, Hong Yuan Mark Liao, Yueh Hua Wu, Ping Yang Chen, Jun Wei Hsieh, and I. Hau Yeh. 2020. “CSPNet: A New Backbone That Can Enhance Learning Capability of CNN.” in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*. Vols. 2020-June.
- Wu, Fan, Guoqing Jin, Mingyu Gao, Zhiwei He, and Yuxiang Yang. 2019. “Helmet Detection Based on Improved YOLO V3 Deep Model.” in *Proceedings of the 2019 IEEE 16th International Conference on Networking, Sensing and Control, ICNSC 2019*.
- Yuliana, Y. 2020. “Corona Virus Diseases (Covid-19): Sebuah Tinjauan Literatur.” *Wellness And Healthy Magazine* 2(1):187–92. doi: 10.30604/well.95212020.
- Zaidi, Syed Sahil Abbas, Mohammad Samar Ansari, Asra Aslam, Nadia Kanwal, Mamoona Asghar, and Brian Lee. 2022. “A Survey of Modern Deep Learning Based Object Detection Models.” *Digital Signal Processing* 126:103514. doi: 10.1016/J.DSP.2022.103514.