Jurnal Riset Matematika (JRM)
e-ISSN 2798-6306 | p-ISSN 2808-313X
https://journals.unisba.ac.id/index.php/JRM

Tersedia secara online di
Unisba Press
https://publikasi.unisba.ac.id/

Deteksi Plat Nomor Kendaraan Menggunakan Algoritma YOLOv5 dengan Metode *Convolutional Neural Network*

Taufik Maulana, Erwin Harahap*

Prodi Matematika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Islam Bandung, Indonesia.

ARTICLE INFO

Article history :
Received : 29/09/2024
Revised : 20/12/2024
Published : 31/12/2024

Volume : 4
No. : 2
Halaman : 103 - 112
Terbitan : **Desember 2024**

Terakreditasi Sinta Peringkat 5 berdasarkan Ristekdikti No. 177/E/KPT/2024

ABSTRAK

Sistem pengawasan lalu lintas yang efektif sangat dibutuhkan untuk mengelola arus lalu lintas yang semakin kompleks di kota-kota besar. Pemantauan plat nomor kendaraan menggunakan teknologi pengenalan objek berbasis *machine learning* dapat membantu penegakan hukum lalu lintas secara efisien. Penelitian ini mengimplementasikan algoritma YOLOv5 yang dikenal dengan kecepatan dan akurasinya dalam mendeteksi objek secara *real-time*, dikombinasikan dengan metode *Convolutional Neural Network* (CNN) untuk mendeteksi plat nomor kendaraan. CNN digunakan untuk mengekstraksi fitur-fitur penting dari gambar, yang digunakan oleh YOLOv5 untuk mendeteksi dan menentukan lokasi plat nomor kendaraan secara akurat. Penelitian ini bertujuan untuk mengembangkan sistem deteksi plat nomor kendaraan dengan menguji tingkat akurasi dari model yang dibuat, serta melakukan pembacaan karakter menggunakan *Optical Character Recognition* (OCR) berbasis *easyOCR*. Hasil penelitian menunjukkan bahwa kombinasi algoritma YOLOv5 dan CNN mampu mendeteksi plat nomor dengan akurasi yang tinggi, serta pembacaan karakter yang juga akurat, dimana sistem ini diujicobakan pada berbagai kondisi gambar kendaraan yang bergerak dan diam. Implementasi YOLOv5 terbukti efisien dalam memproses gambar, menjadikannya solusi yang handal untuk sistem pengawasan lalu lintas.

Kata Kunci : *YOLOv5, Convolutional Neural Network, Deteksi Plat Nomor.*

ABSTRACT

An effective traffic surveillance system is needed to manage the increasingly complex traffic flow in big cities. Vehicle license plate monitoring using machine learning-based object recognition technology can help traffic law enforcement efficiently. This research implements the YOLOv5 algorithm which is known for its speed and accuracy in detecting objects in real-time, combined with the Convolutional Neural Network (CNN) method to detect vehicle license plates. CNN is used to extract important features from the image, which are used by YOLOv5 to detect and accurately determine the location of the vehicle license plate. This research aims to develop a vehicle license plate detection system by testing the accuracy of the model created, as well as reading characters using Optical Character Recognition (OCR) based on easyOCR. The results showed that the combination of the YOLOv5 algorithm and CNN was able to detect license plates with high accuracy, as well as accurate character reading, where this system was tested on various conditions of moving and stationary vehicle images. The YOLOv5 implementation proved to be efficient in processing images, making it a reliable solution for traffic surveillance systems.

Keywords : *YOLOv5, Convolutional Neural Network, License Plate Detection.*

Corresponding Author : Email : erwin2h@unisba.ac.id
Indexed : Garuda, Crossref, Google Scholar
DOI : https://doi.org/10.29313/jrm.v4i2.5060

A. Pendahuluan

Sistem pengawasan lalu lintas yang efektif dan konsisten merupakan kebutuhan yang mendesak dalam memperluas kendali lalu lintas dan pengelolaannya. Arus lalu lintas dapat menunjukkan keadaan lalu lintas dalam interval waktu yang tetap dan membantu dalam mengatasi dan mengontrol terutama saat terjadi pelanggaran lalu lintas dan ketika kecepatan kendaraan dapat menimbulkan terjadinya kecelakaan [1]. Dengan semakin pesatnya pertumbuhan jumlah kendaraan di jalan, terutama di kota-kota besar, pemantauan dan penegakan hukum lalu lintas menjadi semakin menantang dan kompleks [2]. Sehingga membutuhkan suatu sistem untuk mengenali plat nomor kendaraan, dimana sebuah kamera CCTV *(Closed-Circuit Television)* memiliki peran penting untuk memindai jenis pelanggaran yang terjadi. Namun, untuk memindai pelanggaran lalu lintas di kota besar tentu saja akan dibutuhkan lebih banyak kamera CCTV karena jumlah kendaraan dan jumlah jalan yang lebih kompleks [3]. Pembuatan suatu sistem pengenalan plat nomor kendaraan dapat membantu mengefisiensikan penggunaan sumber daya manusia untuk mengamati dan merekam setiap plat nomor kendaraan yang terdeteksi melakukan suatu pelanggaran [4].

Convolutional neural network (CNN) adalah salah satu teknik dalam *Deep learning* yang terdiri dari beberapa modul penting seperti *convolutional layer* dan *pooling layer* [10]. Modul-modul ini disusun secara berurutan untuk membentuk arsitektur jaringan dan model yang mendalam [13]. Teknologi YOLOv5, yang merupakan pengembangan terbaru dari keluarga YOLO, menawarkan berbagai peningkatan dalam hal arsitektur model, optimasi pelatihan, dan performa deteksi [14]. YOLOv5 dirancang untuk lebih efisien dalam memproses gambar dan video, serta lebih fleksibel dalam beradaptasi dengan berbagai skenario pengawasan lalu lintas.

Berdasarkan permasalahan yang telah dipaparkan di atas, maka akan dilakukan penelitian tentang pengimplementasian *machine learning* untuk mendeteksi plat nomor kendaraan dengan menggunakan metode *Convolutional Neural Network* (CNN) atau algoritma YOLOv5 serta pembacaan karakter pada plat nomor kendaraan menggunakan *easyOCR*, yaitu untuk mengetahui hasil dari implementasi tersebut dan mencari tingkat akurasi terbaik [11].

Berdasarkan latar belakang yang telah diuraikan, maka perumusan masalah dalam penelitian ini adalah ”bagaimana menerapkan pengenalan plat nomor kendaraan menggunakan algoritma CNN/YOLOv5 dan pengenalan karakter menggunakan *easyOCR*?” Serta ”bagaimana cara menguji tingkat akurasi pengenalan plat nomor kendaraan dan karakter pada plat nomor dengan metode yang digunakan?”. Selanjutnya, tujuan dalam penelitian ini pun yaitu menerapkan pengenalan plat nomor kendaraan menggunakan Algoritma CNN/YOLOv5 dengan rekognisi huruf dan angka menggunakan *EasyOCR* [15]. Serta, menguji tingkat akurasi pengenalan plat nomor kendaraan dengan membandingkan plat nomor hasil algoritma YOLOv5 dengan yang asli [12].

B. Metode Penelitian

Penelitian ini menggunakan metode kuantitatif dengan menggunakan data sekunder yang diambil dari Kaggle, sebuah *platform* yang menyediakan berbagai *dataset* untuk keperluan analisis dan *machine learning*. *Dataset* yang digunakan dalam penelitian ini berasal dari Kaggle dengan judul "Plat Nomor Kendaraan Indonesia". *Dataset* ini dipilih karena relevansinya dengan topik penelitian yaitu deteksi plat nomor kendaraan.

Untuk mengidentifikasi dan membaca plat nomor kendaraan secara otomatis. Algoritma YOLOv5 digunakan untuk mendeteksi lokasi plat nomor kendaraan dengan bantuan jaringan saraf tiruan (CNN) dalam mendeteksi gambar atau video secara *real-time* dengan mengukur akurasi dan kecepatan [16]. Setelah plat kendaraan terdeteksi, *Optical Character Recognition* (OCR) digunakan untuk mengenali dan mengekstrak teks dari plat tersebut. CNN, yang merupakan jenis *deep learning* khusus untuk pemrosesan citra, memungkinkan sistem untuk mengenali karakter dengan akurasi tinggi, bahkan dalam kondisi pencahayaan dan sudut pandang yang bervariasi [17].

Plat Nomor Kendaraan

Di indonesia TNKB diberikan kepada setiap kendaraan bermotor yang beroperasi di Indonesia. Plat ini berfungsi sebagai identifikasi resmi yang menghubungkan kendaraan dengan pemiliknya dan memastikan kepatuhan terhadap peraturan lalu lintas. TNKB dibuat dan dikeluarkan oleh Kepolisian Negara Republik Indonesia melalui Satuan Lalu Lintas (Satlantas) dan memiliki berbagai komponen seperti nomor registrasi,

kode wilayah, serta masa berlaku. Peraturan mengenai TNKB diatur dalam berbagai undang-undang dan peraturan pemerintah. Salah satu peraturan utama adalah Undang-Undang Nomor 22 Tahun 2009 tentang Lalu Lintas dan Angkutan Jalan (LLAJ). Pasal 68 undang-undang ini menyatakan bahwa setiap kendaraan bermotor wajib dilengkapi dengan TNKB sebagai identitas resmi. Selain itu, peraturan lebih rinci mengenai bentuk, ukuran, dan spesifikasi teknis TNKB ditetapkan dalam Peraturan Kepala Kepolisian Negara Republik Indonesia Nomor 5 Tahun 2012 tentang Registrasi dan Identifikasi Kendaraan Bermotor [5].

Convolutional Neural Network

Convolutional Neural Network (CNN) adalah pengembangan dari *Multilayer Perceptron* (MLP) yang didesain untuk mengolah data dua dimensi. CNN termasuk dalam jenis *Deep Neural network* karena kedalaman jaringan yang tinggi dan banyak diaplikasikan pada data citra. CNN pertama kali dikembangkan dengan nama *NeoCognitron* oleh Kunihiko Fukushima [6], seorang peneliti dari NHK *Broadcasting Science Research Laboratories*, Kinuta, Setagaya, Tokyo, Jepang 4 CEK Konsep tersebut kemudian dimatangkan oleh Yann LeCunn, seorang peneliti dari *AT&T Bell Laboratories* di Holmdel, *New Jersey*, USA.

Kerangka lapisan jaringan dari *neural network* ke CNN kurang lebih berfungsi sama yang membedakan hanya fungsi dan pembentukan jaringan [18]. Dalam CNN terdapat beberapa lapisan, dimana tiap lapisan memiliki suatu fungsi tersendiri, lapisan CNN terdiri dari *Input Layer* yang berfungsi untuk menginisialisasi data gambar input untuk membuat semua dimensi data berpusat nol. Kemudian, itu menormalkan skala semua *input* data di dalamnya [0, 1] untuk mempercepat kecepatan konvergen, dan memutihkan data untuk mengurangi redundansi. Lalu, *Convolutional Layer* yang merupakan inti dari CNN. Lapisan konvolusional menggunakan *kernel* CONV sebagai filter untuk meluncur di gambar asli. Nilai setiap piksel data berkorelasi lokal di dalam filter akan dikalikan dan ditambahkan sebagai hasil konvolusional. Secara matematis tahapan ini akan memanfaatkan inputan yang di dapatkan sebelumnya yaitu *output image* misalnya dari *Xception* yaitu n x n x 2048. Konvolusi pada tahapan ini menggunakan filter 32 yang artinya lapisan konvolusi akan menghasilkan 32 peta fitur, *kernel* n x n, *padding same (0), activation relu, strides* 1.

$$Y(i,j) = (X * K)(i,j) = \sum_{m=0}^{K-1} \sum_{n=0}^{K-1} X(i+m, j+n) * K(m,n) \tag{1}$$

Setelah melakukan tahapan konvolusi disetiap layer maka di dapatkanlah nilai *red, green* dan *blue.* Setiap nilai dari *channel* yang sudah melalui tahapan perhitungan akan dijumlahkan sehingga menghasilkan output total.

$$Total = red + green + blue \tag{2}$$

Activation Layer membuat hasil dari lapisan CONV non-linear, *activation layer* diatur untuk menyelesaikan lenyapnya masalah *gradien* yang disebabkan oleh *underfitting*.

$$ReLU(x) = \max(0, x) \tag{3}$$

Pooling Layer digunakan untuk mengurangi dimensi hasil dari lapisan CONV, terletak di antara dua lapisan CONV. Terdapat tiga jenis penyatuan: *General Pooling*, *Overlapping Pooling* dan *Spatial Pyramid Pooling* (SPP), pada tahapan ini menggunakan *maxpooling* dengan *kernel* size k x k dengan *stride* 1 dengan *output* dari tahapan konvolusi yaitu n x n x 32.

$$Y(i,j) = \max\{X(i * s + p, j * s + q) | 0 \le p < k, 0 \le q < k\} \tag{4}$$

Flatten Layer merupakan tahapan mengubah matriks ke bentuk *vector/matrix* 1 dimensi dari *output droput layer* sebesar $n \times n \times 32$ akan didapatkan bentuk *vector* baru.

$$j = i_1 \cdot (d_2 \cdot d_3 \cdot \ldots \cdot d_n) + i_2 \cdot (d_3 \cdot d_4 \cdot \ldots \cdot d_n) + \cdots + i_{n-1} \cdot d_n + i_n \tag{5}$$

Fully Connected Layer seringkali merupakan lapisan terakhir di ujung CNN. Mereka mentransmisikan data ke output, serta menyederhanakan dan mempercepat perhitungan data. Tahapan akhir dari model CNN yang dibangun ini adalah *Dense* dengan fungsi aktivasi *softmax.* Berikut perhitungan dari setiap *hidden layer.*

$$\sum_{i=0}^{n} I_i * v_{ij} = J_i \tag{6}$$

$$\sum_{i=0}^{n} J_i * w_{ij} = H_i \tag{7}$$

$$\sum_{i=0}^{n} H_i * x_{ij} = 0_i \tag{8}$$

$$s(o_i) = \frac{e^{o_i}}{\sum_{j=1}^{n} e^{o_j}} \tag{9}$$

YOLO

You Only Look Once (YOLO) adalah sebuah algoritma yang dikembangkan oleh Joseph Redmon, Ali Farhadi dkk untuk mendeteksi sebuah objek secara *real-time*. Sistem pendeteksian yang dilakukan adalah dengan menggunakan *repurpose classifier* atau *localizer* untuk melakukan deteksi. Sebuah model diterapkan pada sebuah citra di beberapa lokasi dan skala.

Daerah dengan citra yang diberi *score* paling tinggi akan dianggap sebagai sebuah pendeteksian [7]. YOLOv5 merupakan versi pengembangan terbaru dari algoritma YOLO. YOLOv5 adalah model deteksi objek *real-time* yang menggunakan jaringan saraf konvolusional (CNN) untuk mendeteksi objek dalam gambar atau video. YOLOv5 merupakan iterasi terbaru dari model YOLO, yang terkenal dengan kecepatan dan akurasinya.

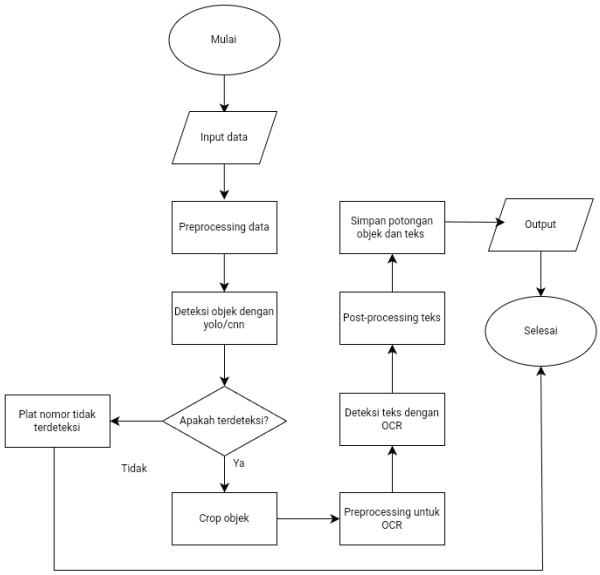
Optical Recognition Character (OCR)

Optical Character Recognition (OCR) adalah sebuah sistem komputer yang dapat membaca huruf, baik yang berasal dari sebuah pencetak (*printer* atau mesin ketik) maupun yang berasal dari tulisan tangan. OCR adalah aplikasi yang menerjemahkan gambar karakter (*image character*) menjadi bentuk teks dengan cara menyesuaikan pola karakter per-baris dengan pola yang telah tersimpan dalam *database* aplikasi [19]. Hasil dari proses OCR adalah berupa teks sesuai dengan gambar *output scanner* dimana tingkat keakuratan penerjemahan karakter tergantung dari tingkat kejelasan gambar dan metode yang digunakan [8].

EasyOCR

EasyOCR merupakan pustaka Python *open-source* yang dirancang untuk menghadirkan kemudahan dan fleksibilitas dalam menjalankan tugas pengenalan karakter optik (OCR). Teknologi OCR memungkinkan konversi teks dari gambar atau dokumen digital menjadi format teks yang dapat dibaca.

Kemampuan utama *easyOCR* terletak pada akurasinya yang tinggi dalam menangani berbagai tugas OCR. Pustaka ini terbukti handal dalam membaca teks dari gambar natural *scene* maupun teks padat dalam dokumen [9].



Gambar 1. Diagram Alir Penelitian

C. Hasil dan Pembahasan

Pre-Processing Data

Pre-processing data bertujuan untuk meningkatkan efisiensi dan efektivitas pelatihan model [20]. Dengan melakukan teknik augmentasi seperti rotasi, perubahan skala, dan transformasi warna, *dataset* dapat diperbesar secara artifisial, membantu model untuk belajar dari lebih banyak variasi data tanpa perlu mengumpulkan gambar tambahan. Langkah pertama diawali dengan Pengumpulan *dataset* serta proses anotasi, didapatkan dari data sekunder, yang diperoleh dari *Kaggle.com.* Diperoleh total dataset sebanyak 358 plat nomor dengan format JPG. Pada penelitian ini akan dilakukan training dengan file ekstensi JPG dan hasil anotasi berupa TXT. Lalu, dilanjutkan dengan Anotasi Data yang merupakan proses memberikan label atau tanda pada data, proses ini sangat penting karena model YOLOv5 memerlukan data yang sudah diberi anotasi untuk dapat belajar mengenali dan mendeteksi objek pada gambar. Hasil anotasi ini disimpan dalam format yang dapat dibaca oleh model, yaitu format YOLO (file .txt dengan koordinat *bounding box* dan label objek) atau format XML.

Augmentasi digunakan untuk meningkatkan jumlah dan variasi data pelatihan melalui transformasi yaitu rotasi, *flipping*, *zooming*, dan perubahan kontras. Teknik ini membantu memperbesar dataset secara artifisial. Terakhir dilakukan Pemisahan Data dimana proporsi umum yang digunakan untuk pemisahan ini adalah 70% untuk data *training*, 20% untuk data *validation*, dan 10% untuk data *testing*.

Tabel 1. Data Sebelum di *Augmentation*

Keterangan	Data *Training*	Data *Validation*	Data *Testing*	Total
Proporsi	70%	20%	10%	100%
Jumlah	251	71	36	358

Tabel 2. Data Sesudah di *Augmentation*

Keterangan	Data *Training*	Data *Validation*	Data *Testing*	Total
Proporsi	88%	8%	4%	100%
Jumlah	748	71	36	855

Training Data Menggunakan YOLOv5

Pre-processing Dataset yang sudah di *labelling* dirubah ukurannya menjadi 640 x 640 piksel. Dataset plat nomor yang sudah dibuat (*anotation)* dan di exsport jalankan di custom training YOLOv5 untuk persiapan *training* data. Model YOLOv5 digunakan sebagai dasar pelatihan untuk mempercepat konvergensi dan meningkatkan akurasi awal. Perintah untuk memulai pelatihan adalah:

!python train.py --img 640 --batch 16 --epochs 150 --data {dataset.location}/data.yaml --weights yolov5s.pt –cache

Parameter yang digunakan dalam perintah ini memastikan bahwa pelatihan dilakukan dengan gambar ukuran 640x640 piksel, *batch size* 16, selama 150 *epoch*. Hasil training data dari YOLOv5 nantinya menghasilkan data *.pt* yaitu terdiri dari *last.pt* dan *best.pt*. namun selain itu ada data hasil *training* YOLOv5 berbentuk *.onnx*. Data inilah yang akan digunakan dalam pendeteksian mennggunakan Python. Cara merubah data *.pt* menjadi *.onnx* terlihat pada program di bawah.
!python export.py --weights /content/yolov5/runs/train/exp/weights/best.pt --include onnx --opset 11

Proses Deteksi Objek Menggunakan YOLOv5

Dalam YOLOv5, *backbone* adalah bagian dari arsitektur jaringan yang bertanggung jawab untuk mengekstraksi fitur dari gambar input. *Backbone* ini berfungsi sebagai pondasi bagi model untuk memahami dan mengidentifikasi berbagai elemen dalam gambar.

Secara matematis tahapan ini akan memanfaatkan inputan yang didapatkan sebelumnya yaitu *output image* misalnya dari *Xception* yaitu 4 x 4 x 2048. Konvolusi pada tahapan ini menggunakan filter 32, *kernel* 3 x 3, *padding same (0), activation ReLU, strides* 1.

Berikut proses konvolusi dengan memberikan nilai filter pada matriks. Pixel citra dengan 3 *channel red, green* dan *blue* diambil data pixelnya masing-masing dan di tambahkan *padding* 0 di setiap pixelnya sehingga di dapatkan pixelnya. Dapat dilihat pada Gambar 2.

Channel red

0	0	0	0	0	0
0	70	35	28	42	0
0	45	25	30	48	0
0	31	28	32	49	0
0	30	30	35	50	0
0	0	0	0	0	0

Channel green

0	0	0	0	0	0
0	107	59	45	58	0
0	76	42	45	65	0
0	51	41	46	65	0
0	42	41	48	64	0
0	0	0	0	0	0

Channel blue

0	0	0	0	0	0
0	112	69	51	59	0
0	79	48	47	63	0
0	56	43	46	60	0
0	47	42	46	59	0
0	0	0	0	0	0

Gambar 2. *Channel Red, Green,* dan *Blue*

Pada percobaan ini digunakan *kernel* 3 x 3 dengan nilai seperti pada Gambar 3.

1	0	-1
1	0	-1
1	0	-1

Gambar 3. Daerah *Kernel* 3x3

Langkah selanjutnya dilakukan proses perhitungan disetiap *channel* dengan di kalikan dengan *kernel* ukuran 3x3 pada gambar 3. Tahapan ini dilakukan berulang dengan pergeseran *kernel* sebanyak 1 *strides* disetiap channelnya dengan menggunakan rumus 1 sehingga didapatkan hasil perhitungan di setiap channelnya, dapat dilihat pada Gambar 4.

channel red

-60	57	-30	58
-88	56	-51	90
-83	9	-64	97
-58	-6	-41	67

channel green

-101	93	-22	90
-142	98	-46	136
-124	30	-70	139
-82	-1	-47	94

channel blue

-117	93	-5	98
-160	103	-22	144
-133	43	-49	139
-85	11	-34	92

Gambar 4. Hasil Perhitungan *Channel red, Green,* dan *Blue*

Setiap nilai dari *channel* yang sudah melalui tahapan perhitungan akan dijumlahkan menggunakan rumus 2 sehingga menghasilkan *output* total, dapat dilihat pada Gambar 5.

-278	243	-57	246
-390	257	-119	370
-340	82	-183	375
-225	4	-122	253

Gambar 5. Total Perhitungan *Red*, *Green*, dan *Blue*

Dengan *activation* ReLU maka setiap nilai yang negatif akan di ubah menjadi angka 0 yaitu dengan menggunajan rumus 3. Sehingga didapatkanlah hasil *convolutional*-nya dapat dilihat pada Gambar 6.

0	243	0	246
0	257	0	370
0	82	0	375
0	4	0	253

Gambar 6. Hasil ReLU

Hasil tahapan konvolusi akan dijadikan sebagai input pada *pooling layer*, pada tahapan ini menggunakan *maxpooling* dengan *kernel* size 2x2 dengan *stride* 2 dengan *output* dari tahapan konvolusi yaitu 4x4x32 dengan menggunakan rumus 4. Dapat dilihat pada Gambar 7.

257	370
82	375

Gambar 7. Hasil *Maxpooling*

Didapatkan *output* pada *layer maxpooling* yaitu 2x2x32 dengan nilai 257, 370, 82, dan 357. Kemudian akan di *flatten* menjadi matriks 1 dimensi yaitu menggunakan rumus 5. Dapat dilihat pada Gambar 8.

257
370
82
375

Gambar 8. Hasil *Flatten* menjadi Vektor 1 Dimensi

Pada tahap ini, YOLOv5 memprediksi *bounding box*, probabilitas kelas, dan skor objek. Dengan menggunakan *treshold 0.2*, objek yang terdeteksi lebih dari ambang batas yang ditentukan yaitu lebih dari 0.2 maka proses akan di deteksi positif (dilanjutkan ke tahap *croping bounding box*). Dapat mengasumsikan bahwa nilai 257 dan 370 adalah koordinat pusat *bounding box*, 82 dan 375 adalah lebar dan tinggi *bounding box.*

Tahapan akhir dari model CNN dalam YOLOv5 yang dibangun adalah *Dense* dengan fungsi aktivasi *softmax.* Berikut hasil perhitungan dari setiap *hidden layer* dengan mengunakan rumus 6, 7, dan 8, dapat dilihat dalam Tabel 3.

Tabel 3. Hasil Perhitungan Setiap *Hidden Layer*

I	J	H	O
$i_1 = 257$	$j_1 = 108.4$	$h_1 = 422.76$	$o_1 = 1973.28$
$i_2 = 370$	$j_2 = 433.6$	$h_2 = 704.06$	
$i_3 = 82$	$j_3 = 542$	$h_3 = 281.84$	$o_2 = 591.864$
$i_4 = 375$	$j_4 = 325.2$	$h_4 = 563.68$	

Langkah selanjutnya adalah perhitungan *softmax* dengan rumus eksponensial o_1 dibagi dengan total eksponensial o_1 dan o_2. Tahapan ini dilakukan 2 kali baik pada o_1 dan o_2 dengan rumus 9.

$$s(o_1) = \frac{e^{1973.28}}{e^{1973.28} + e^{591.864}} = 0.8$$

$$s(o_2) = \frac{e^{591.864}}{e^{1973.28} + e^{591.864}} = 0.2$$

$$s(o_1) + s(o_2) = 1$$

$$0.8 + 0.2 = 1$$

Didapatkan bobot nilai probabilitas yang lebih besar pada o_1 yaitu 0,8 yang berarti input gambar yang dimasukkan diprediksi adalah plat nomor. Adapun hasil matriks evaluasi dari pengujian data gambar sebanyak 36 sampel, dapat dilihat dalam Tabel 4.

Tabel 4. Hasil Matriks Evaluasi

Accuracy	*Precision*	*Recall/Sensitivity*	*F1-score*
97%	100%	97%	98.5%

Dari Tabel 4 di atas menyajikan matriks evaluasi yang diperoleh dari model: *accuracy* sebesar 97%, *precision* sebesar 100%, *recall* sebesar 97%, dan *f1-score* sebesar 98.5%. Akurasi yang tinggi menunjukan keandalan secara keseluruhan data yang di uji, sedangkan recall yang tinggi (97%) sangat penting karena memastikan bahwa hampir semua gambar yang terdeteksi bahwa itu plat nomor teridentifikasi dengan benar.

Analisis Hasil Performa Pengujian

Berdasarkan data yang di *testing* sebanyak 36 gambar didapatkan hasil deteksi plat nomor dan karakternya, dapat dilihat dalam Tabel 5.

Tabel 5. Hasil Akurasi Plat Nomor dan Karakter

Nama gambar	Akurasi plat	Akurasi karakter	Nama gambar	Akurasi plat	Akurasi karakter
Tes1.jpg	0.97	0.75	Tes19.jpg	0.93	0.65
Tes2.jpg	0.96	0.86	Tes20.jpg	0.97	0.69
Tes3.jpg	0.96	0.33	Tes21.jpg	0.98	0.70
Tes4.jpg	0.96	0.72	Tes22.jpg	0.97	0.78
Tes5.jpg	0.96	0.60	Tes23.jpg	0.98	0.84
Tes6.jpg	0.97	0.28	Tes24.jpg	0.96	0.73
Tes7.jpg	0.97	0.64	Tes25.jpg	0.97	0.89
Tes8.jpg	0.96	0.96	Tes26.jpg	0.97	0.67
Tes9.jpg	0.98	0.68	Tes27.jpg	0.97	0.53
Tes10.jpg	0.96	0.83	Tes28.jpg	0.98	0.74
Tes11.jpg	0.97	0.31	Tes29.jpg	0.97	0.63
Tes12.jpg	0.97	0.77	Tes30.jpg	0.90	0.84
Tes13.jpg	0.98	0.92	Tes31.jpg	0.97	0.49
Tes14.jpg	0.92	0.59	Tes32.jpg	0.97	0.76
Tes15.jpg	0.97	0.59	Tes33.jpg	0.97	0.71
Tes16.jpg	0.97	0.76	Tes34.jpg	0. 91	0.57
Tes17.jpg	0.97	0.90	Tes35.jpg	0.96	0.94
Tes18.jpg	0.95	0.71	Tes36.jpg	0.97	0.90

Berdasarkan hasil pengujian yang dilakukan, sistem deteksi plat nomor yang dikembangkan menunjukkan tingkat akurasi yang sangat tinggi, dengan YOLOv5 mencapai akurasi sebesar 98% dengan rata-rata 96,25% dalam mendeteksi plat nomor, menandakan kemampuannya untuk mengidentifikasi lokasi plat

nomor kendaraan dengan baik. Pada tahap pengenalan karakter alfanumerik, *easyOCR* mencapai tingkat akurasi sebesar 96% dengan rata-rata 70,15%, dengan menegenali rata-rata pembacaan sebanyak 6-12 karakter, yang cukup memuaskan namun masih memiliki ruang untuk perbaikan mengingat faktor-faktor seperti kualitas gambar, variasi *font*, dan kondisi fisik plat nomor dapat mempengaruhi hasilnya. Sistem juga diuji untuk kecepatan proses, dengan YOLOv5 mampu mendeteksi plat nomor secara cepat, dan *easyOCR* mengenali karakter dengan cepat setelah deteksi, memungkinkan aplikasi ini untuk digunakan dalam kondisi *real-time*. Adapun hasil pengujian secara *real-time* seperti pada Gambar 9.

Gambar 9. Deteksi Plat secara *Real-time*

D. Kesimpulan

Berdasarkan pembahasan dalam penelitian ini, peneliti menyimpulkan beberapa hasil penelitian yaitu implementasi YOLOv5 menunjukkan hasil yang sangat baik dari model yang telah di *training*, kemudian di uji menggunakan data *testing* hasil akurasi deteksi plat nomor mencapai 98% dalam pengenalan plat nomor. Penggunaan *Optical Character Recognition* (OCR) menggunakan *easyOCR* berhasil mencapai akurasi 96% dalam pengenalan karakter alfanumerik pada plat nomor. Serta, sistem deteksi plat nomor kendaraan dengan menggunakan algoritma YOLOv5 diperoleh rata-rata akurasi pengenalan plat nomor kendaraan adalah sebesar 96,25% dan pengenalan karakter didapatkan rata-rata akurasi sebesar 70,15%, dengan mengenali rata-rata pembacaan sebanyak 6-12 karakter. Sistem ini diuji dengan membandingkan hasil deteksi dan pengenalan plat nomor menggunakan YOLOv5 dengan plat nomor asli, sehingga dapat memvalidasi tingkat akurasi sistem yang dikembangkan.

Daftar Pustaka

[1] M. Zulfikri, K. A. Latif, H. Hairani, A. Ahmad, R. Hammad, and M. Syahrir, “Deteksi dan Estimasi Kecepatan Kendaraan dalam Sistem Pengawasan Lalu Lintas Menggunakan Pengolahan Citra,” Techno. Com, vol. 20, no. 3, pp. 455–467, 2021.

[2] E. Harahap, D. Darmawan, and F. H. Badruzzaman, “Simulation of Traffic T-Junction at Cibiru-Cileunyi Lane Using SimEvents MATLAB,” J Phys Conf Ser, vol. 1613, no. 1, p. 012074, 2020, doi: 10.1088/1742-6596/1613/1/012074.

[3] E. Harahap, F. H. Badruzzaman, Y. Permanasari, M. Y. Fajar, and A. Kudus, “Traffic engineering simulation of campus area transportation using MATLAB SimEvents,” IOP Conf Ser Mater Sci Eng, vol. 830, no. 2, p. 022078, 2020, doi: 10.1088/1757-899X/830/2/022078.

[4] R. Shreyas, B. V. P. Kumar, H. B. Adithya, B. Padmaja, and M. P. Sunil, “Dynamic traffic rule violation monitoring system using automatic number plate recognition with SMS feedback,” in 2017 2nd International Conference on Telecommunication and Networks (TEL-NET), IEEE, Aug. 2017, pp. 1–5. doi: 10.1109/TEL-NET.2017.8343528.

[5] O. Mellolo, “Pengenalan Plat Nomor Polisi Kendaraan Bermotor,” Jurnal ilmiah sains, pp. 35–42,

2012.
[6] K. Fukushima, “Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position,” Biol Cybern, vol. 36, no. 4, pp. 193–202, 1980.
[7] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 779–788.
[8] R. S. Bahri, “Perbandingan algoritma template matching dan feature extraction pada optical character recognition,” KOMPUTA: Jurnal Komputer dan Informatika, vol. 1, no. 1, 2012.
[9] M. F. Haidar and F. Utaminingrum, “Deteksi Plat Nama Ruangan untuk Kendali Kursi Roda Pintar menggunakan YOLOv5 dan EasyOCR berbasis TX2,” Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer, vol. 7, no. 2, pp. 658–662, 2023.
[10] S. Zein and G. Gunawan, “Prediksi Hasil FIFA World Cup Qatar 2022 Menggunakan Machine Learning dengan Python,” *J. Ris. Mat.*, pp. 153–162, Dec. 2022, doi: 10.29313/jrm.v2i2.1382.
[11] D. Pitriyani and Y. Permanasari, “Prediksi Jumlah Penumpang Pesawat dengan Backpropagation Neural Network,” *J. Ris. Mat.*, pp. 129–136, Dec. 2022, doi: 10.29313/jrm.v2i2.1327.
[12] T. Barokah and E. Harahap, “Peramalan Beban Jangka Panjang Sistem Kelistrikan Kota Bandung Menggunakan Artificial Neural Network,” *J. Ris. Mat.*, vol. 4, no. 1, pp. 65–72, Jun. 2024, doi: 10.29313/jrm.v4i1.3603.
[13] G. Achyar and O. Rohaeni, “Penggunaan Hybrid K-Means dan General Regression Neural Network untuk Prediksi Harga Saham Indeks LQ45,” J. Ris. Mat., pp. 111–120, Dec. 2022, doi: 10.29313/jrm.v2i2.1193.
[14] A. Khaerunnisa, “Analisis Tingkat Kelulusan Mahasiswa di Unisba dengan menggunakan Algoritma K-Means Clustering,” J. Ris. Mat., pp. 67–76, Jul. 2022, doi: 10.29313/jrm.v2i1.1018.
[15] S. A. Savitri and D. Suhaedi, “Penerapan Inference Fuzzy Mamdani dalam Seleksi Penerima Bantuan Sosial Tunai Kabupaten Belitung Timur,” J. Ris. Mat., pp. 163–172, Dec. 2022, doi: 10.29313/jrm.v2i2.1383.
[16] S. T. Utami Putri and E. Kurniati, “Prediksi Harga Saham Menggunakan Jump Diffusion Model dan Analisis Value at Risk,” J. Ris. Mat., pp. 131–140, Dec. 2023, doi: 10.29313/jrm.v3i2.2832.
[17] I. Putri Fatimah and D. Suhaedi, “Sistem Pendukung Keputusan Pemilihan Tingkat Prestasi Siswa Menggunakan Metode PROMETHEE,” J. Ris. Mat., pp. 141–148, Dec. 2023, doi: 10.29313/jrm.v3i2.2833.
[18] S. Sofiyani and Y. Permanasari, “Penerapan Metode Cubic Spline Interpolation untuk Menentukan Peluang Kematian pada Tabel Mortalita,” J. Ris. Mat., pp. 29–36, Jul. 2023, doi: 10.29313/jrm.v3i1.1735.
[19] W. Ismarnita and Respitawulan, “Penerapan Logika Fuzzy dalam Menentukan Tingkat Kerawanan Longsor di Suatu Wilayah,” J. Ris. Mat., pp. 45–54, Jul. 2023, doi: 10.29313/jrm.v3i1.1737.
[20] N. N. Layla, E. Kurniati, and D. Suhaedi, “Peramalan Indeks Harga Saham dengan Autoregressive Moving Average Generelized Autoregressive Conditional Heteroscedasticity (ARMA-GARCH),” J. Ris. Mat., vol. 1, no. 1, pp. 7–12, 2021, doi: 10.29313/jrm.v1i1.103.