

# Sistem Deteksi Pelanggaran Kelebihan Penumpang Pada Kendaraan Sepeda Motor Roda Dua Menggunakan Algoritma *Faster RCNN*

## *Passenger Excess Violation Detection System On Two Wheel Vehicles Using Faster R-CNN Algorithm*

1<sup>st</sup> Erwan Maulana  
Fakultas Teknik Elektro  
Universitas Telkom  
Bandung, Indonesia

lerwanm@student.telkomuniversity.ac.id

2<sup>nd</sup> Casi Setianingsih  
Fakultas Teknik Elektro  
Universitas Telkom  
Bandung, Indonesia

setiacasie@telkomuniversity.ac.id

3<sup>rd</sup> Marisa W. Paryasto  
Fakultas Teknik Elektro  
Universitas Telkom  
Bandung, Indonesia

marisaparyasto@telkomuniversity.ac.id

**Abstrak**—Kesadaran masyarakat mengenai pentingnya menaati peraturan lalu lintas terkadang masih dipandang sebelah mata. Akibatnya tingkat kecelakaan lalu lintas di Indonesia terus bertambah setiap tahunnya. Salah satu pelanggaran yang sering dijumpai yaitu kelebihan penumpang pada kendaraan sepeda motor. Hal ini dapat mengurangi tingkat keamanan dari pengendara sepeda motor dan pengguna jalan lainnya. Oleh karena itu, dibutuhkan sebuah sistem yang dapat mendeteksi pelanggaran kelebihan penumpang pada kendaraan sepeda motor. Sebuah sistem untuk mendeteksi pelanggaran kelebihan penumpang dirancang menggunakan algoritma Faster R-CNN yang menggunakan bahasa pemrograman Python, beserta Library pembelajaran mesin seperti, Tensorflow, dan OpenCV. Dataset yang digunakan merupakan dataset custom yang terdiri dari 3 kategori. Masing-masing kategori berisikan 300 gambar, sehingga total dari dataset yang digunakan berjumlah 900 gambar. Model terbaik yang digunakan didapat menggunakan perhitungan Confusion Matrix pada dataset. Model menggunakan rasio train 95% dan test 5% hasilnya

memiliki nilai akurasi sebesar 92%, nilai recall sebesar 88%, nilai presisi sebesar 86%, steps sebesar 15000, batch size sebesar 1, epochs sebesar 10 dan learning Rate sebesar 0.002.

**Kata kunci** — deteksi pelanggaran, kelebihan penumpang sepeda motor, Faster R-CNN.

*Abstract*—Public awareness about the importance of obeying traffic rules is sometimes still underestimated. As a result, the rate of traffic accidents in Indonesia continues to increase every year. One of the violations that are often encountered is excess passengers on motorbikes. This can reduce the level of safety of motorcyclists and other road users. Therefore, we need a system that can detect violations of excess passengers on motorcycle vehicles. A system to detect excess passenger violations is designed using the Faster R-CNN algorithm that uses the Python programming language, along with machine learning libraries such as, Tensorflow, and OpenCV. The dataset used is a custom dataset which consists of 3 categories. Each category contains 300 images, the total of the dataset used is 900 images. The best model used is obtained using the Confusion Matrix calculation on the

*dataset. The model using 95% train and 5% test, results has an accuracy value of 92%, recall value of 88%, precision value of 86%, steps of 150000, batch size of 1, epochs of 10 and Learning Rate of 0.002*

**Keywords:** *detection o violations, excess motorcycle passengers, Faster R-CNN.*

## I. PENDAHULUAN

Menurut Undang-undang No. 22 tahun 2009 Lalu Lintas memiliki definisi sebagai gerak kendaraan dan orang di ruang lalu lintas, sedangkan yang dimaksud dengan ruang lalu lintas jalan adalah prasarana yang diperuntukkan bagi gerak pindah kendaraan, orang, dan atau barang yang berupa jalan dan fasilitas pendukung.

Pelanggaran lalu lintas merupakan salah satu permasalahan yang sering terjadi di jalan raya. Akibatnya sering terjadi kecelakaan lalu lintas yang disebabkan oleh pelanggaran lalu lintas tersebut. Jumlah kecelakaan lalu lintas oleh sepeda motor yang terjadi di Indonesia berdasarkan Badan Pusat Statistik sebanyak 116.441 kasus pada tahun 2019. Pertumbuhan kecelakaan lalu lintas mengalami kenaikan 4,87 persen tiap tahunnya [1]. Salah satu penyebab tingginya kecelakaan lalu lintas yaitu kurangnya kesadaran masyarakat dalam menaati peraturan lalu lintas, salah satunya pengguna kendaraan sepeda motor yang membawa penumpang lebih dari satu seperti yang sudah di tetapkan di peraturan penumpang pada sepeda motor pasal 106 ayat (9) UU. No. 22 Tahun 2009 Tentang Lalu Lintas dan Angkutan Umum.

Upaya telah dilakukan dengan tujuan menekan jumlah pelanggaran lalu lintas yaitu mengadakan sosialisasi dan razia pada jalan raya. Akan tetapi solusi tersebut memiliki kekurangan yaitu keterbatasan Sumber Daya Manusia untuk disebar di seluruh titik jalan raya guna menindak lanjuti pelanggaran lalu lintas. Salah satu cara untuk mengatasi permasalahan tersebut dengan menerapkan sistem deteksi pelanggaran kelebihan penumpang pada sepeda motor yang bertujuan untuk membantu kinerja pihak tertentu dalam menindak lanjuti pelanggaran yang terjadi di jalan raya.

## II. KAJIAN TEORI / MATERIAL DAN

## METODE

### A. Aturan Penumpang Sepeda Motor

Peraturan penumpang pada sepeda motor diatur pada pasal 106 ayat (9) UU. No. 22 Tahun 2009 Tentang Lalu Lintas dan Angkutan Umum yang berbunyi “*Setiap orang yang mengemudikan Sepeda Motor tanpa kereta samping dilarang membawa Penumpang lebih dari 1 (satu) orang*”. Jika melanggar peraturan tersebut, terdapat hukuman atas pelanggaran tersebut yang diatur dalam pasal 292 UU. No. 22 tahun 2009 Tentang Lalu Lintas dan Angkutan Umum yang berbunyi [2]: “*Setiap orang yang mengemudikan Sepeda Motor tanpa kereta samping yang mengangkut Penumpang lebih dari 1 (satu) dipidana dengan pidana kurungan paling lama 1(satu) bulan atau denda paling banyak Rp250.000,00 (dua ratus lima puluh ribu rupiah).*”

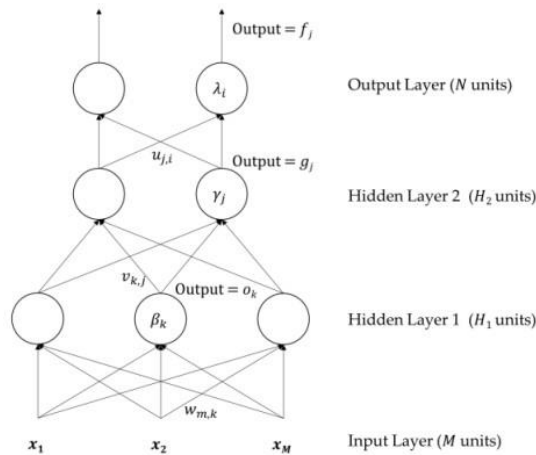
### B. Machine Learning

*Machine Learning* adalah metode analisis yang membantu menangani data besar dengan cara mengembangkan algoritma komputer. Dengan menggunakan data, pembelajaran mesin memungkinkan komputer menemukan wawasan tersembunyi tanpa diprogram secara eksplisit saat mencarinya. *Machine Learning* merupakan cabang aplikasi dari *Artificial Intelligence* (Kecerdasan Buatan) yang fokus pada pengembangan sebuah sistem yang mampu belajar “sendiri” tanpa harus berulang kali di program oleh manusia. Fokus besar penelitian *Machine Learning* adalah bagaimana mengenali secara otomatis pola kompleks dan membuat keputusan cerdas berdasarkan data[3]. Pembagian *machine learning* menurut [4] yaitu *Supervised Learning*, *Unsupervised Learning*, *Semi-Supervised Learning*, *Trancedutive Inference*, *On-line learning*, *Reinforcement Learning*, *Active Learning*.

### C. Deep Learning

*Deep Learning* merupakan bagian dari *machine learning*. Sebuah model *machine learning* yang perlu ‘diberitahu’ untuk menciptakan prediksi akurat, dengan terus diberikan data. Untuk mencapai kemampuan itu, *deep learning* menggunakan struktur algoritma berlapis yang disebut *artificial neural network* (ANN). Keuntungan utama *deep learning* yaitu dapat mengubah data dari *non-linearly separable* menjadi *linearly separable*

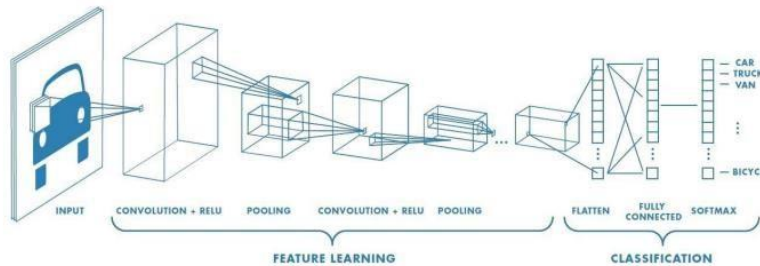
melalui serangkaian transformasi [5].



GAMBAR 1  
DEEP LEARNING DENGAN 4 LAYER

D. *Convolutional Neural Network*  
*Convolutional Neural Network* adalah salah satu metode *machine learning* dari pengembangan *MultiLayer Perceptron* (MLP) yang di desain untuk

mengolah data dua dimensi. CNN memiliki beberapa *hidden Layer* yang terdapat di proses ekstraksi fitur, yaitu *Convolutional Layer*, *Rectified Linear Unit* (ReLU), dan *Pooling* [6].



GAMBAR 2  
LAPISAN CONVOLUTIONAL NEURAL NETWORK

Hasil dari *Convolutional Layer* pertama akan digunakan untuk *Convolutional Layer* selanjutnya. *Output* berupa hasil klasifikasi yang dihasilkan melalui proses klasifikasi

*fully -connected* dan *Softmax* [3]. Penggunaan *pooling layer* pada CNN memiliki tujuan untuk mereduksi ukuran citra sehingga mudah digantikan dengan sebuah *convolutional layer* dengan *ide* yang sama dengan *pooling layer* [7].

1	1	2	4
5	6	7	8
3	2	1	0
1	2	3	4

Max Pool with 2x2 filters



6	8
3	4

GAMBAR 3  
ILUSTRASI POOLING LAYER

### E. Image Processing

*Image Processing* adalah teknik pemrosesan gambar dengan *input* berupa citra dua dimensi. Tujuan *image processing* untuk menyempurnakan citra dan mendapatkan informasi yang berguna untuk diolah menjadi beberapa keputusan. Operasi yang sering dilakukan pada pemrosesan citra yaitu gambar *grayscale*. Warna untuk penerapan *image processing* bisa didekomposisi menjadi komponen merah (R), hijau (G) dan biru (B) masing-masing diproses secara independen sebagai gambar *grayscale* [8].

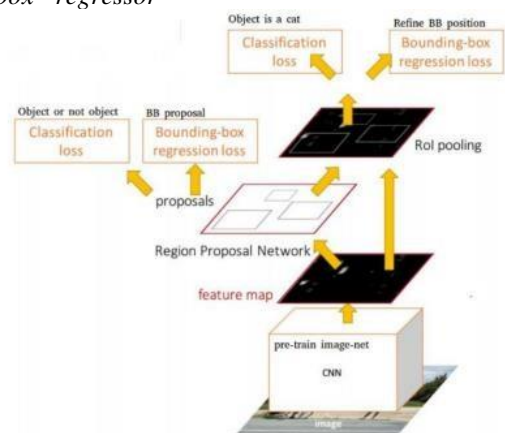
### F. Faster Region based Convolutional Neural Network (Faster R-CNN)

*Faster R-CNN* merupakan pengembangan dari *Convolutional Neural Network (CNN)* yang memiliki perbedaan dengan terdapatnya *Region Proposal Network (RPN)*. RPN terdapat di akhir konvolusi pada model, dan memiliki *output* berupa *Bounding Box* dan *Classifier* [9]. *Faster R-CNN* menyingkirkan SVMs yang digunakan dalam klasifikasi gambar *bounding box images* dan *bounding box regressor*

untuk memperbaiki pembuatan *bounding box* [10].

Modul RPN untuk jaringan generasi regional, RPN adalah saraf konvolusi penuh jaringan yang memiliki perbedaan internalnya dari jaringan saraf konvolusi biasa [11]. *Ground truth* merupakan *bounding box* yang dialokasikan terhadap objek yang akan dituju. *Anchor box* adalah *box* yang berfungsi untuk mencari dimana lokasi objek tersebut dengan melakukan pergeseran terhadap citra [12].

Tujuan RPN yaitu dapat mengurangi kebutuhan komputasi yang berlebihan secara signifikan dan membuat keseluruhan objek dapat dilatih. Kemungkinan terdapatnya Batasan lokasi objek pada gambar dari wilayah yang diketahui disebut dengan *Region of Interest (ROI)* [10]. Tujuan dari *region of interest* mengurangi area deteksi pada video sehingga program dapat mengetahui objek yang akan dideteksi terkhusus hanya pada area yang ditentukan oleh *Region of Interest* [12]



GAMBAR 4  
FASTER R-CNN

Lapisan *pooling* RoI menggunakan *Max pooling* untuk konversi fitur di dalam wilayah yang diinginkan ke dalam fitur kecil dengan tingkat spasial tetap  $H \times W$ , di mana  $H$  dan  $W$  adalah parameter lapisan [13]. *Inception v2* merupakan sebuah modul yang di desain untuk mengurangi kompleksitas dari konvolusi, dengan cara membuat jaringan konvolusi menjadi lebih lebar, bukan lebih Panjang. Terdapat tiga jenis tipe yang berbeda [14].

### G. Confusion Matrix

*Confusion Matrix* merupakan suatu metode untuk mengetahui

informasi mengenai prediksi suatu sistem klasifikasi yang memiliki label kebenaran. Evaluasi terhadap hasil tes dengan menggunakan *confusion matrix* dengan 4 kondisi aktual yaitu : *True Positive (TP)*, *False Negative (FN)*, *False Positive (FP)*, dan *True Negatif (TN)*. Dalam melakukan evaluasi menggunakan hasil dari rumus *Recall*, *Precision*, *F-Score*, dan *Accuracy* [15].

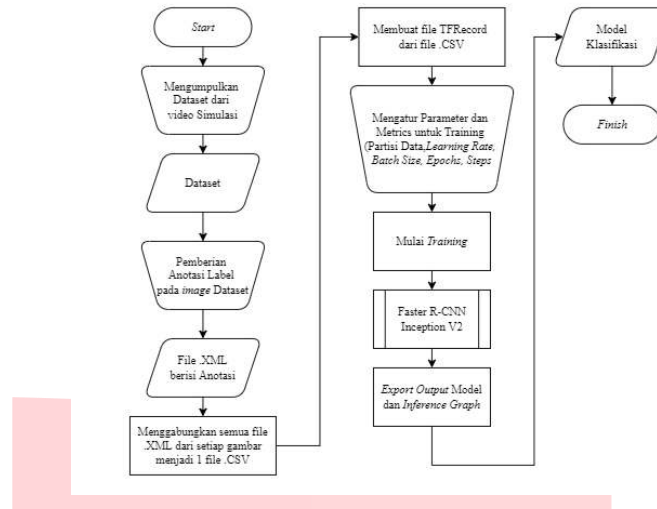
### H. Perolehan Dataset

*Dataset* yang digunakan merupakan *dataset* yang diambil secara manual menggunakan kamera *Handphone* yang diambil dengan cara membuat simulasi di Lingkungan

Telkom University. Jumlah *dataset* yang digunakan pada penelitian kali ini berjumlah 900 gambar. Gambar tersebut merupakan gambar pengendara sepeda motor serta penumpang yang diambil

nantinya akan digunakan untuk *training model* klasifikasi.

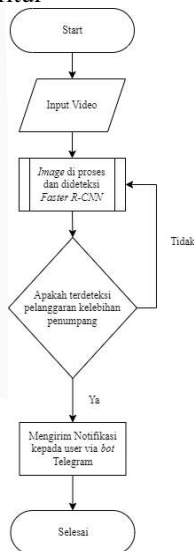
I. Desain Perancangan Sistem



GAMBAR 5 FLOWCHART PROSES TRAINING

*Flowchart* dibawah merupakan proses yang akan dilakukan pada saat *training*. Setelah mengumpulkan *dataset* dari simulasi yang telah dibuat di Lingkungan Telkom University, *dataset* akan menjadi *Input* dan melalui beberapa tahap, seperti pelabelan citra yang fungsinya untuk memilih fitur

penting dari setiap citra yang akan di *training*. Proses *training* akan menghasilkan sebuah *model* berdasarkan *dataset* yang sudah dilatih, yang akan digunakan pada klasifikasi deteksi objek.



GAMBAR 6 ILUSTRASI ALUR KERJA SISTEM

Gambar 6 merupakan desain sistem deteksi pelanggaran kelebihan penumpang pada sepeda motor. *Input* yang berupa rekaman video simulasi akan di proses oleh algoritma *Faster R-CNN*, lalu setelah berhasil melakukan deteksi pada objek, sistem akan melakukan cek jika objek tersebut

melakukan pelanggaran kelebihan penumpang. Jika terdeteksi melakukan pelanggaran kelebihan penumpang, maka secara otomatis sistem akan mengirim notifikasi menggunakan *bot Telegram* menuju *user*.



### III. HASIL DAN PEMBAHASAN

#### A. Pengujian Partisi Data

Tahap pengujian Partisi *dataset* kali ini, telah dibuat 6 *model* dari variasi Partisi yang berbeda. Dimulai dari pembagian *dataset* 50% data latih dan 50% data uji sampai 95% data latih dan

5% data uji. Tahap ini bertujuan untuk menemukan nilai Akurasi tertinggi dari setiap partisi data, dengan menghitung nilai *F-1 Score*, *Recall*, *Precision*, dan *Accuracy*. Berikut merupakan tabel hasil pengujian Partisi *dataset*.

TABEL 1  
PENGUJIAN PARTISI DATA

No	Pengujian				
	Partisi Data (%)	F1-Score (%)	Precision (%)	Recall (%)	Accuracy (%)
1	50:50	48	57	49	70
2	60:40	44	43	45	69
3	70:30	46	47	46	67
4	80:20	54	54	55	72
5	90:10	56	63	61	73
6	95:5	68	68	68	81

Dari tabel di atas dapat kita perhatikan bahwa partisi data terbaik setelah melakukan enam skema pengujian partisi *dataset*, didapatkan *model* terbaik dengan partisi data *train* 95% dan *test* 5% dengan nilai *F-1 Score* 68%, *Precision* 68%, *Recall* 68%, dan *Accuracy* 81%. Hal ini dapat disimpulkan karena semakin banyak data *train*, maka akan semakin baik bagi *model*.

#### B. Pengujian Training Variabel Konfigurasi

Tahap pengujian variabel konfigurasi ini bertujuan untuk mencari nilai terbaik dari *model* yang akan di *training*. Variabel konfigurasi yang digunakan yaitu *learning rate*, *epochs*, *batch size*, dan *steps*.

TABEL 2  
PENGUJIAN LEARNING RATE, EPOCHS, BATCH SIZE, DAN STEPS

Hyperparameter	Pengujian				
	Nilai	F1-Score (%)	Precision (%)	Recall (%)	Accuracy (%)
Learning Rate	0.00002	62	62	62	78
	0.0002	64	65	66	78
	0.002	79	79	79	88
	0.02	0	0	33	47
Epochs	1	79	79	79	88
	5	68	68	69	82
	10	82	82	83	90
	50	73	74	76	84
	100	75	75	77	85
Batch Size	1	82	82	83	90
	2	73	73	73	84
	4	72	72	72	84
Steps	40000	82	82	83	90
	80000	81	81	83	88
	100000	77	77	79	87
	150000	87	86	88	92
	200000	78	78	79	88

Berdasarkan hasil dari *F1-Score*, *Precision*, *Recall*, dan *Accuracy* dari proses pengujian partisi *dataset* yang telah dilakukan, telah didapatkan nilai

tertinggi yaitu *F-1 Score* 87%, *Precision* 86%, *Recall* 88%, dan *Accuracy* 92%. Pada pengujian kali ini partisi *dataset* yang digunakan dan

mendapatkan nilai terbaik yaitu 95% *train* dan 5% *test*. Model terbaik ini akan digunakan dalam proses selanjutnya yaitu pengujian sistem deteksi objek menggunakan beberapa parameter – parameter.

#### A. Pengujian Sistem Berdasarkan Parameter

Pada tahap ini akan dilakukan pengujian berdasarkan parameter seperti jarak, intensitas cahaya, sudut kamera, kecepatan, dan k-fold cross validation. Akurasi yang digunakan ketika melakukan pengujian sistem berdasarkan parameter jarak yaitu 0 – 30 meter, kemudian intensitas yang digunakan dalam pengujian sistem ini yaitu 3,6 *lux* dan 294 *lux*. Sudut kamera yang digunakan pada proses pengujian sistem kali ini yaitu 40 °, 90 °, dan 140 °. Kecepatan yang diuji dalam proses

pengujian kali ini yaitu 0 – 30 km/jam. Dari hasil pengujian K-Fold *Cross Validation* rata – rata akurasi yang berhasil didapat menggunakan K yang bernilai 3 yaitu 75%.

Hasil dari pengujian sistem ini terdapat pada tabel 3, ketika objek berjarak diatas 13 – 18 meter akurasi yang didapatkan akan jauh menurun dibandingkan dengan akurasi jarak 0 – 12 meter. Sedangkan pengujian intensitas cahaya mendapatkan akurasi 100% dikedua parameter. Pengujian sudut kamera mendapatkan akurasi yang sama baiknya disetiap parameter. Sedangkan pada pengujian kecepatan, semakin tinggi kecepatan yang dilakukan oleh objek, maka akurasi yang didapat akan semakin menurun.

TABEL 3  
PENGUJIAN JARAK, INTENSITAS CAHAYA, SUDUT KAMERA, KECEPATAN, DAN CROSS VALIDATION

Parameter	Nilai	Akurasi (%)
Jarak (Meter)	0 - 6	100
	7 - 12	100
	13 - 18	33.3
Intensitas Cahaya (Lux)	3,6	91.7
	294	91.7
Sudut Kamera (Derajat)	40	100
	90	100
	140	100
Kecepatan (Km/jam)	0 - 5	100
	6 - 10	100
	11 - 15	100
	16 - 20	91.7
	21 - 30	77.7
K-Fold Cross Validation	K = 3	75

#### IV. KESIMPULAN

Sistem yang menggunakan algoritma Faster R-CNN dapat mengidentifikasi terjadinya pelanggaran kelebihan penumpang pada sepeda motor, melalui skema pengujian partisi data dan beberapa pengujian parameter. Hasil terbaik didapat dengan nilai *F1-Score* sebesar 87%, *Precision* sebesar 86%, *Recall* sebesar 88%, dan *Accuracy* sebesar 92%. Partisi *dataset* menggunakan perbandingan 95% *train* dan 5% *test* serta menggunakan *Learning Rate* bernilai 0.002, *Epochs* bernilai 10, *Batch Size* bernilai 1, dan menggunakan *Steps* sebanyak 150000.

Sistem dapat mendeteksi pada saat melakukan pengujian menggunakan parameter seperti Jarak, Intensitas

Cahaya, Sudut Kamera, dan Kecepatan. Pada saat objek terdeteksi sistem akan memberikan notifikasi dengan menggunakan *bot* Telegram yang berfungsi hanya saat sistem mendeteksi adanya pelanggaran kelebihan penumpang atau kelas *violated* dan dapat berjalan dengan baik.

#### REFERENSI

- [1] BPS-Statistik Indonesia, “Statistik Transportasi Darat 2020”.
- [2] DPR RI, “J.D.I.H. - Dewan Perwakilan Rakyat.” <https://www.dpr.go.id/jdih/index/id/539>(accessed Dec. 19, 2021)..
- [3] E. C. Putro, R. M. Awangga, and R. Andarsyah, *Tutorial Object Detection People With Faster*

- region-Based Convolutional Neural Network(Faster R-CNN)*. 2020.
- [4] M. Mohri, A. Rostamizadeh, and A. Talwalkar, "Foundations of Machine Learning," 2012.
- [5] Y. U. Hanafi *et al.*, "DETEKSI PENGGUNAAN HELM PADA PENGENDARA BERMOTOR BERBASIS DEEP LEARNING."
- [6] M. T. Audina, F. Utamingrum, and D. Syauqi, "Sistem Deteksi dan Klasifikasi Jenis Kendaraan berbasis Citra dengan menggunakan Metode Faster-RCNN pada Raspberry Pi 4B," 2021. [Online]. Available: <http://j-ptiik.ub.ac.id>.
- [7] J. T. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller, "Striving for Simplicity: The All Convolutional Net," Dec. 2014, [Online]. Available: <http://arxiv.org/abs/1412.6806>.
- [8] V. Tyagi, *Understanding Digital Image Processing*. CRC Press, 2018. doi: 10.1201/9781315123905.
- [9] A. Arinaldi, J. A. Pradana, and A. A. Gurusinga, "Detection and classification of vehicles for traffic video analytics," in *Procedia Computer Science*, 2018, vol. 144, pp. 259–268. doi: 10.1016/j.procs.2018.10.527.
- [10] D. Alamsyah and M. Fachrurrozi, "Faster R-CNN with inception v2 for fingertip detection in homogenous background image," in *Journal of Physics: Conference Series*, Apr. 2019, vol. 1196, no. 1. doi: 10.1088/1742-6596/1196/1/012017.
- [11] J. Wang, T. Zhang, Y. Cheng, and N. Al-Nabhan, "Deep learning for object detection: A survey," *Computer Systems Science and Engineering*, vol. 38, no. 2. Tech Science Press, pp. 165–182, Apr. 23, 2021. doi: 10.32604/CSSE.2021.017016..
- [12] F. Yaghmaee and M. Ebadi, "ROI detection in images using annotation output," 2014.
- [13] R. Girshick, "Fast R-CNN," in *2015 IEEE International Conference on Computer Vision (ICCV)*, Dec. 2015, pp. 1440–1448. doi: 10.1109/ICCV.2015.169.
- [14] S. Ruuska, W. Hämmäläinen, S. Kajava, M. Mughal, P. Matilainen, and J. Mononen, "Evaluation of the confusion matrix method in the validation of an automated system for measuring feeding behaviour of cattle," *Behavioural Processes*, vol. 148, pp. 56–62, Mar. 2018, doi: 10.1016/j.beproc.2018.01.004.
- [15] D. Alamsyah and M. Fachrurrozi, "Faster R-CNN with Inception V2 for Fingertip Detection in Homogenous Background Image," *Journal of Physics: Conference Series*, vol. 1196, p. 012017, Mar. 2019, doi: 10.1088/1742-6596/1196/1/012017.