

# Pengembangan Sistem Deteksi Pemakaian Sabuk Pengaman Berbasis Convolutional Neural Network (CNN)

Fahmi Fidyah Cahyadi, Susi Juniastuti, Reza Fuad Rachmadi

<sup>1</sup>Dept. Teknik Komputer Institut Teknologi Sepuluh Nopember Surabaya, Indonesia

Email: fahmifidyancahyadi@gmail.com, susi@ee.its.ac.id, fuad@its.ac.id

## Abstrak

Salah satu penerapan pengolahan citra adalah pendeteksian penggunaan sabuk pengaman. Banyak penelitian telah mengusulkan berbagai solusi berbasis pengolahan citra untuk mendeteksi penggunaan sabuk pengaman. Oleh karena itu, dengan memanfaatkan *Convolutional Neural Network*, dibuat sebuah sistem yang dapat secara otomatis mendeteksi penggunaan sabuk pengaman penumpang kursi depan yang akan diterapkan di jalan raya. Proses *training* dilakukan menggunakan *YOLOv3-SPP* dan *YOLOv3-tiny* dengan menggunakan 1240 dataset. Data tersebut dibagi menjadi 969 citra *train* dan 271 citra uji. Hasil *training* tertinggi yang diperoleh oleh *YOLOv3-SPP* adalah sebesar 90,2% mAP dan hasil *training* tertinggi yang diperoleh oleh *YOLOv3-tiny* adalah sebesar 92% mAP. Hasil pengujian kinerja menunjukkan nilai rata-rata presisi (*mean Average Precision*/mAP) tertinggi pada *YOLOv3-SPP* dengan nilai sebesar 90,2% dibandingkan dengan *YOLOv3-tiny* yang memperoleh nilai sebesar 92,6%. Pada waktu pemrosesan untuk pengujian dataset, *YOLOv3-tiny* memiliki keunggulan waktu pemrosesan untuk mendeteksi 271 dataset, yaitu membutuhkan waktu 7 detik, sedangkan pada *YOLOv3-SPP* membutuhkan waktu 15 detik untuk seluruh citra. Tingkat kesalahan dalam pendeteksian penggunaan sabuk pengaman berada pada nilai terendah yaitu sebesar 5%.

**Keyword:** Deteksi, Deteksi Pelanggaran Sabuk Pengaman, You Only Look Once (YOLO), Convolutional Neural Network (CNN)

---

Diterima Redaksi: 1-Nov-2024 Selesai Revisi: 15-Nov-2024 Diterbitkan Online: 15-Des-2024  
DOI: <https://doi.org/10.59378/jcenim.v2i3.56>

---

## I. PENDAHULUAN

Ilmu dan teknologi *computer vision* yang baru-baru ini diusulkan telah membangkitkan minat yang besar di banyak bidang penelitian lainnya, seperti penegakan hukum lalu lintas secara otomatis, sistem bantuan pengemudi, dan lain-lain [1, 2, 3]. *Computer vision* merupakan suatu disiplin ilmu yang membahas bagaimana komputer dapat memperoleh tingkat pemahaman yang tinggi dari citra digital atau video. Beberapa manfaat dari *computer vision* adalah untuk memperoleh informasi yang lebih jelas dalam bentuk pendeteksian objek yang sebelumnya sulit untuk dilihat secara jelas.

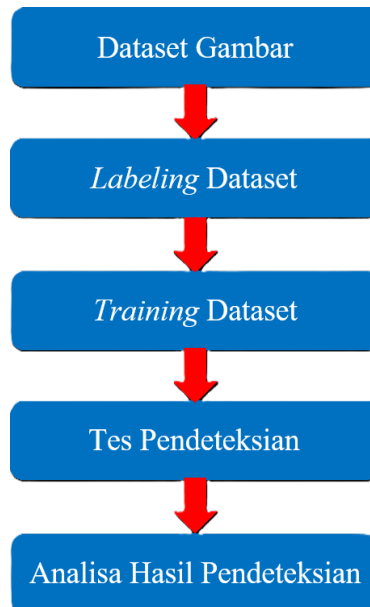
Salah satu penerapan pengolahan citra adalah pendeteksian penggunaan sabuk pengaman. Pemeriksaan penggunaan sabuk pengaman secara manual yang dilakukan oleh pihak berwenang di jalan diketahui tidak efisien, mahal, dan tidak efektif. Pada tahun 2016, National Highway Traffic Safety Administration (NHTSA) melaporkan bahwa sebanyak 10.428 jiwa meninggal di jalan raya Amerika Serikat karena tidak menggunakan sabuk pengaman [4]. Pada penelitian lain, World Health Organization (WHO) melaporkan bahwa penggunaan sabuk pengaman dapat menurunkan angka kematian sebesar 50% bagi penumpang kursi depan [5]. Berbagai penelitian berbasis visi komputer telah dikembangkan untuk mendeteksi penggunaan sabuk pengaman secara otomatis guna meningkatkan keselamatan berkendara [6]. Salah satu faktor yang mendominasi penyebab tingginya angka kecelakaan akibat faktor manusia adalah penumpang yang tidak menggunakan sabuk pengaman saat berkendara. Sering kali, hal ini disebabkan oleh kebiasaan yang rendah dalam menggunakan sabuk pengaman. Pada kondisi pengereman mendadak dengan kecepatan tinggi, hal tersebut dapat menyebabkan kecelakaan yang mengakibatkan penumpang kursi depan mengalami cedera karena terlempar ke arah dashboard atau kaca depan mobil. Hal ini terjadi karena penumpang tidak mengalami perlambatan yang sama seperti kendaraan akibat tidak menggunakan sabuk pengaman.

Berdasarkan uraian tersebut, dibuat suatu sistem yang komprehensif yang dapat mendeteksi penggunaan sabuk pengaman menggunakan *Convolutional Neural Network* (CNN) yang dapat dikembangkan

dan digunakan pada sistem lainnya [6]. CNN merupakan pengembangan dari *Multilayer Perceptron* (MLP) yang dirancang untuk mengolah data dua dimensi. CNN termasuk ke dalam jenis *Deep Neural Network* karena memiliki kedalaman jaringan yang tinggi dan banyak diterapkan pada data citra [7].

## II. DESAIN SISTEM

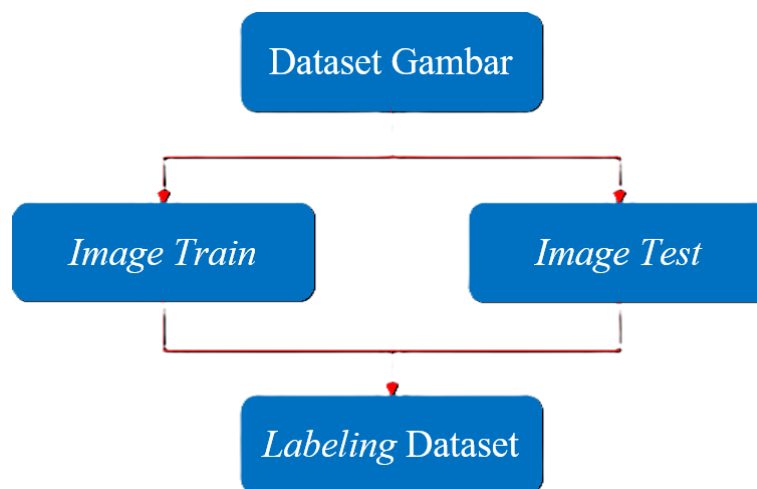
Gambar 1 merupakan diagram blok dari sistem pendeteksian penggunaan sabuk pengaman menggunakan *Convolutional Neural Network* (CNN).



Gambar 1: Diagram Blok Sistem Pendeteksian Penggunaan Sabuk Pengaman

### A. Dataset Citra

Pada tahap dataset citra, citra diperoleh dari internet berupa gambar yang berasal dari jurnal yang dibuat oleh Pan Chen, Xiang Bai, dan Wenyu Liu [8]. Agar dataset dapat dilatih menggunakan YOLO, data tersebut perlu diberi label. Gambar 2 merupakan proses pembuatan dataset citra.



Gambar 2: Diagram Alir Pembuatan Dataset

## B. Pelabelan Dataset

Tahap pelabelan atau anotasi objek merupakan proses pemberian informasi berupa kelas dan posisi penumpang kursi depan mobil yang akan dideteksi. Dari dataset yang diperoleh, dilakukan pelabelan satu per satu sehingga koordinat *ground-truth bounding box* dapat dibandingkan dengan *bounding box* hasil prediksi. Dari dua jenis *bounding box* yang tersedia, akan diperoleh nilai *Intersection over Union* (IoU) berdasarkan perbandingan kedua *bounding box* tersebut.

Proses pelabelan *bounding box* dilakukan secara manual satu per satu pada setiap citra yang terdapat penumpang kursi depan mobil. Informasi hasil pelabelan kemudian disimpan dalam bentuk berkas TXT.

## C. Pra-pemrosesan

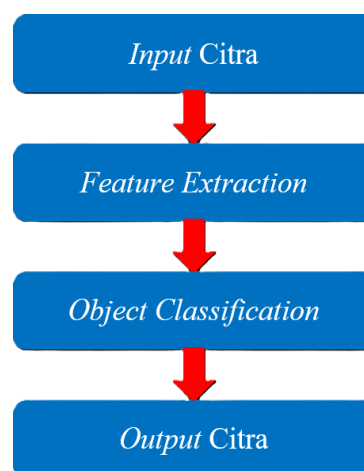
*Pra-pemrosesan* merupakan proses manipulasi citra untuk memperoleh fitur-fitur penting yang dapat digunakan sebagai indikator dalam proses pengenalan citra, yaitu citra penumpang kursi depan mobil yang telah dideteksi menggunakan YOLO. Metode *pra-pemrosesan* yang digunakan adalah *Histogram Equalization*.

### 1) Histogram Equalization

*Histogram* suatu citra dapat dimodifikasi untuk memperoleh citra yang sesuai dengan kebutuhan atau untuk meningkatkan kualitas citra. Salah satu metode yang dapat digunakan untuk memodifikasi *histogram* citra adalah *histogram equalization*. *Histogram equalization* merupakan proses yang mengubah distribusi nilai derajat keabuan pada suatu citra agar menjadi merata. Tujuan dari *histogram equalization* adalah untuk memperoleh distribusi *histogram* yang merata sehingga setiap derajat keabuan memiliki jumlah piksel yang relatif sama. Penyetaraan *histogram* diperoleh dengan mengubah derajat keabuan suatu piksel ( $r$ ) menjadi derajat keabuan baru ( $s$ ) dengan fungsi transformasi  $T$ . Secara matematis dapat dituliskan dengan persamaan:  $s = T(r)$ .

## D. Pelatihan dan Deteksi Menggunakan YOLO

Pada tahap ini dataset yang telah diberi label akan dilatih untuk membentuk suatu pola yang hasilnya berupa *weights*. *Weights* tersebut akan digunakan untuk mengenali objek pada citra. Proses pelatihan yang digunakan berbasis CNN, yaitu *You Only Look Once* (YOLO). Perangkat yang digunakan dalam proses pelatihan adalah Google Collaboratory. YOLO yang digunakan adalah *YOLOv3-SPP* dan *YOLOv3-tiny* dengan total dua kelas.



Gambar 3: Pendeteksian Penggunaan Sabuk Pengaman Menggunakan YOLO

Proses pendeteksian dilakukan menggunakan program yang telah dibuat sebelumnya kemudian dilakukan perubahan parameter serta penambahan kode agar dapat digunakan pada tahap selanjutnya. Pada tahap ini, citra berupa gambar atau video digunakan sebagai masukan yang kemudian diproses melalui sistem berbasis CNN, yaitu YOLO. Keluaran dari proses pendeteksian berupa nilai *confidence*, nama kelas, dan koordinat posisi *bounding box* dari penggunaan sabuk pengaman yang terdeteksi oleh sistem. Ketiga nilai tersebut merupakan nilai yang dibutuhkan dalam proses analisis kinerja metode yang

digunakan. Gambar 3 menunjukkan proses pendeteksian penggunaan sabuk pengaman menggunakan YOLO.

### 1) Pemrosesan Input

YOLO membutuhkan masukan citra dengan ukuran  $416 \times 416$  piksel, yang masih dapat diubah selama ukurannya merupakan kelipatan dari 32. Semakin besar ukuran citra masukan, maka hasil pendeteksian yang diperoleh akan semakin akurat. Namun, ukuran citra masukan yang semakin besar akan menyebabkan proses komputasi menjadi lebih lambat. Sebaliknya, jika ukuran citra masukan semakin kecil maka akurasi hasil pendeteksian akan menurun, tetapi proses komputasi akan berjalan lebih cepat.

### 2) Ekstraksi Fitur dan Klasifikasi Objek

YOLOv3 menggunakan metode *feature extraction* yang disebut *Darknet53*, yang memiliki 53 lapisan dan telah dilatih menggunakan ImageNet. Untuk keperluan pendeteksian, ditambahkan kembali 53 lapisan ke dalam jaringan sehingga total terdapat 106 lapisan yang mendasari arsitektur konvolusional YOLOv3. Hal ini menyebabkan proses pendeteksian menjadi lebih lambat, namun menghasilkan tingkat akurasi yang lebih baik. Gambar 4 menunjukkan susunan lapisan dari *Darknet53*.

	Type	Filters	Size	Output
	Convolutional	32	$3 \times 3$	$256 \times 256$
	Convolutional	64	$3 \times 3 / 2$	$128 \times 128$
1x	Convolutional	32	$1 \times 1$	$128 \times 128$
	Convolutional	64	$3 \times 3$	
	Residual			
	Residual			
	Convolutional	128	$3 \times 3 / 2$	$64 \times 64$
2x	Convolutional	64	$1 \times 1$	$64 \times 64$
	Convolutional	128	$3 \times 3$	
	Residual			
	Residual			
	Convolutional	256	$3 \times 3 / 2$	$32 \times 32$
8x	Convolutional	128	$1 \times 1$	$32 \times 32$
	Convolutional	256	$3 \times 3$	
	Residual			
	Residual			
	Convolutional	512	$3 \times 3 / 2$	$16 \times 16$
8x	Convolutional	256	$1 \times 1$	$16 \times 16$
	Convolutional	512	$3 \times 3$	
	Residual			
	Residual			
	Convolutional	1024	$3 \times 3 / 2$	$8 \times 8$
4x	Convolutional	512	$1 \times 1$	$8 \times 8$
	Convolutional	1024	$3 \times 3$	
	Residual			
	Residual			
	Avgpool		Global	
	Connected		1000	
	Softmax			

Gambar 4: Susunan Lapisan Darknet53

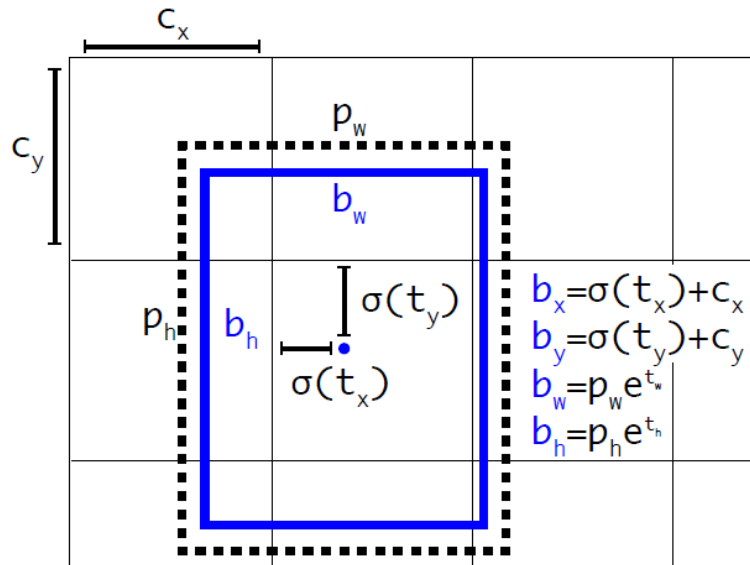
### 3) Pemrosesan Keluaran

YOLO menghasilkan prediksi berupa koordinat *bounding box* ( $t_x, t_y, t_w, t_h$ ), nilai *confidence*, dan probabilitas kelas. Nilai *confidence* merupakan tingkat keyakinan adanya objek pada citra, sedangkan probabilitas kelas menunjukkan jenis kelas objek yang terdeteksi. YOLO membentuk *bounding box* pada lokasi *feature map* menggunakan fungsi aktivasi sigmoid logistic. Proses pembentukan *bounding box* dari hasil pendeteksian ditunjukkan pada Gambar 5.

Hasil pendeteksian kemudian dibandingkan dengan *ground-truth bounding box* yang telah diberikan pada proses pelabelan sebelumnya. Perbandingan ini dilakukan dengan menghitung nilai *Intersection over Union* (IoU) atau nilai tumpang tindih antara *ground-truth bounding box* dan *bounding box* hasil prediksi yang dapat dituliskan pada Persamaan 1.

$$IoU = \frac{A_O}{A_U} \quad (1)$$

Apabila nilai mAP@IoU dari kedua *bounding box* tersebut lebih besar dari ambang batas yang ditetapkan pada penelitian ini, yaitu sebesar 0,5, maka hasil pendeteksian dianggap benar. Sebaliknya, apabila nilai mAP@IoU yang diperoleh kurang dari 0,5, maka hasil pendeteksian dianggap salah. Dari



Gambar 5: Prediksi Lokasi Bounding Box

perhitungan IoU, nilai True Positive, False Positive, False Negative, dan True Negative digunakan untuk memperoleh nilai akurasi, presisi, dan *recall* dari hasil pendeteksian. Nilai akurasi, presisi, dan *recall* dihitung menggunakan Persamaan 2, 3, dan 4.

$$A = \frac{TP + TN}{TP + FP + FN + TN} \quad (2)$$

$$P = \frac{TP}{TP + FP} \quad (3)$$

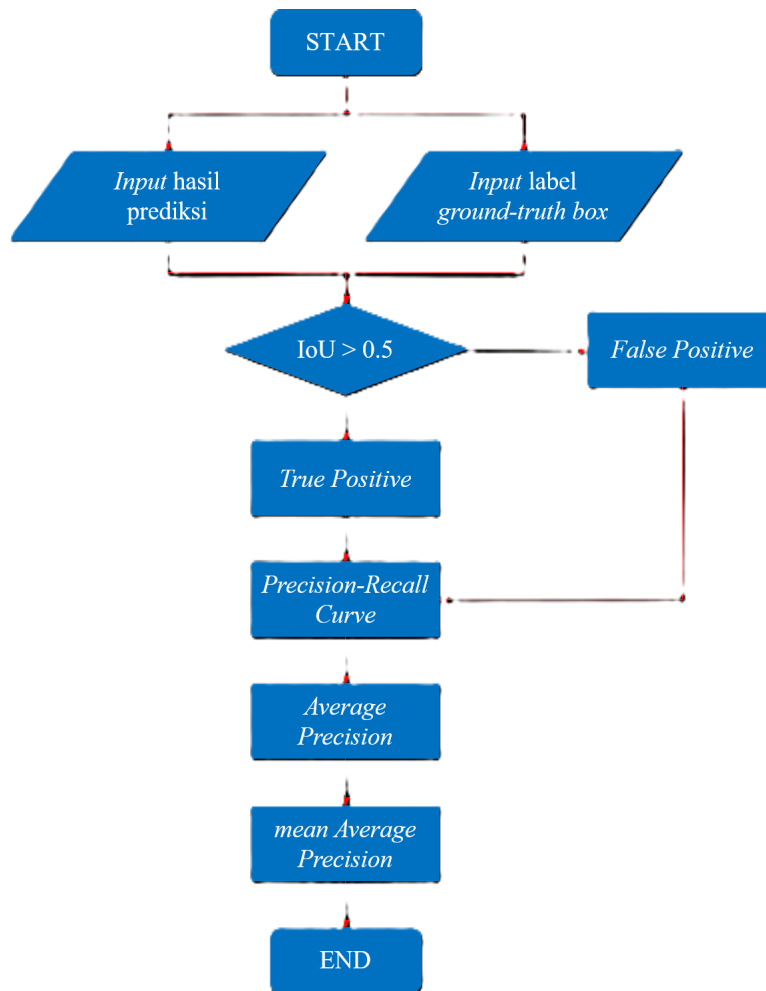
$$R = \frac{TP}{TP + FN} \quad (4)$$

## E. Analisis Hasil Deteksi

Pada tahap ini dilakukan analisis terhadap hasil pendeteksian penggunaan sabuk pengaman oleh sistem. Proses analisis ini ditunjukkan pada Gambar 6. Sistem pendeteksian penggunaan sabuk pengaman akan menghasilkan keluaran berupa nilai *confidence* dan *bounding box* hasil prediksi. Hasil pendeteksian penggunaan sabuk pengaman tersebut kemudian dibandingkan dengan *ground-truth bounding box* yang telah diberikan pada proses pelabelan sebelumnya. Perbandingan ini dilakukan dengan menghitung nilai *Intersection over Union* (IoU) atau nilai tumpang tindih antara *ground-truth bounding box* dan *bounding box* hasil prediksi. Apabila nilai mAP@IoU dari kedua *bounding box* tersebut lebih besar dari ambang batas yang ditetapkan pada penelitian ini, yaitu sebesar 0,5, maka hasil pendeteksian dianggap benar. Sebaliknya, apabila nilai mAP@IoU yang diperoleh kurang dari 0,5, maka hasil pendeteksian dianggap salah. Dari hasil perhitungan IoU, nilai True Positive, False Positive, dan False Negative digunakan untuk memperoleh nilai presisi dan *recall* dari hasil pendeteksian penggunaan sabuk pengaman. Nilai presisi dan *recall* tersebut kemudian digambarkan dalam sebuah kurva yang disebut sebagai *precision-recall curve*. Nilai *Average Precision* (AP) diperoleh dari perhitungan luas area di bawah kurva untuk setiap kelas yang terdeteksi oleh sistem, sedangkan nilai *mean Average Precision* (mAP) diperoleh dengan menghitung rata-rata nilai AP dari seluruh kelas yang terdeteksi.

## III. PENGUJIAN DAN HASIL

Pada penelitian ini disajikan hasil pengujian dan analisis dari perancangan serta implementasi sistem, yaitu sistem pendeteksian penggunaan sabuk pengaman menggunakan *Convolutional Neural Network* (CNN).



Gambar 6: Diagram Alir Proses Analisis

## A. Pembuatan Sistem Pendeteksian Penggunaan Sabuk Pengaman

### 1) Pembuatan Dataset

Pembuatan dataset bertujuan untuk menyiapkan data yang akan digunakan pada proses *training*. Data berupa citra penumpang kursi depan mobil diperoleh dengan pengambilan data secara langsung di lokasi dalam bentuk foto atau video maupun dengan mengambil data dari internet. Gambar 7 merupakan contoh pengambilan data secara langsung di lokasi dan Gambar 8 merupakan contoh pengambilan data dari internet. Jumlah data yang diperoleh sebanyak 1240 data. Data tersebut dibagi menjadi 969 citra *train* dan 271 citra uji.





Gambar 7: Pengambilan Data Secara Langsung di Lokasi



Gambar 8: Pengambilan Data dari Internet

## 2) Hasil Histogram Equalization

Gambar 9 merupakan contoh hasil dari proses *Histogram Equalization*.

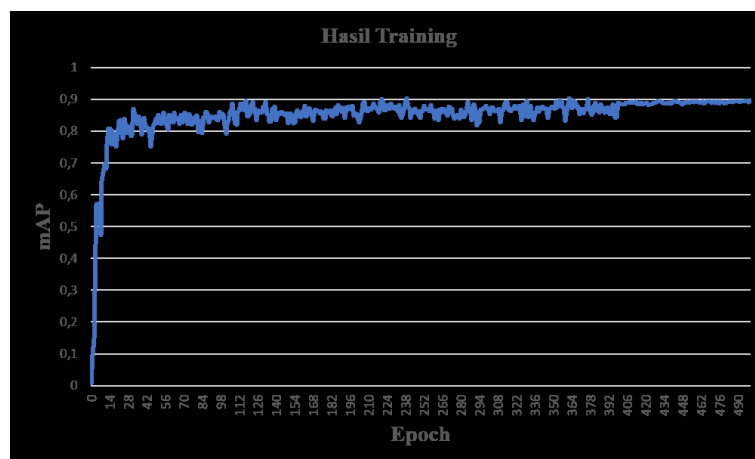
## 3) Data Training

Data *training* bertujuan untuk menghasilkan bobot (*weights*) yang akan digunakan dalam proses pendeteksian penggunaan sabuk pengaman. Proses *training* data dilakukan menggunakan *YOLOv3-SPP* dan *YOLOv3-tiny* dengan *pytorch*. Proses *training* dilakukan menggunakan Google Collaboratory. Proses *training* berlangsung selama 14 jam dengan jumlah *epoch* sebanyak 500, ukuran citra (*img size*) sebesar

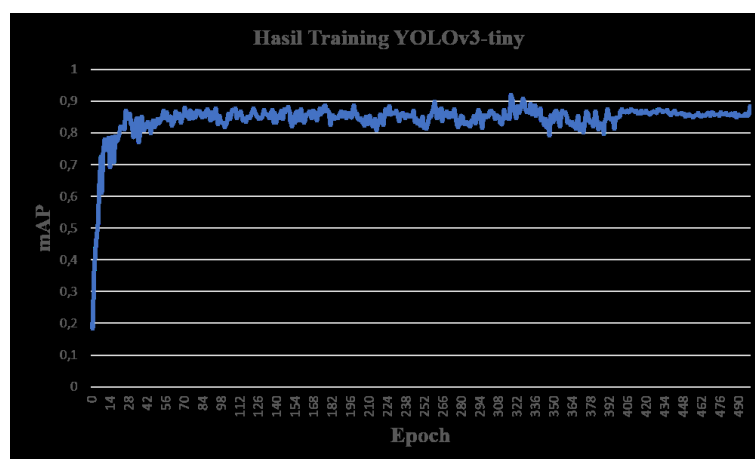


Gambar 9: Hasil Histogram Equalization

416, dan *batch size* sebesar 16 dengan total 2 kelas. Gambar 10 merupakan grafik hasil *training YOLOv3-SPP* dan Gambar 11 merupakan grafik hasil *training YOLOv3-tiny*. Berdasarkan grafik tersebut, nilai mAP terbaik yang diperoleh oleh *YOLOv3-SPP* adalah sebesar 90,2%, sedangkan nilai mAP terbaik yang diperoleh oleh *YOLOv3-tiny* adalah sebesar 92%.



Gambar 10: Grafik Hasil Training YOLOv3-SPP



Gambar 11: Grafik Hasil Training YOLOv3-tiny



#### 4) Hasil Pendeteksian

Gambar 12 merupakan salah satu contoh hasil pendeteksian penggunaan sabuk pengaman menggunakan data yang diambil secara langsung di lokasi dan Gambar 13 merupakan salah satu contoh hasil pendeteksian menggunakan data dari internet.



Gambar 12: Hasil Pendeteksian Data Secara Langsung di Lokasi



Gambar 13: Hasil Pendeteksian Data dari Internet

### B. Pengujian Kinerja YOLOv3-SPP dan YOLOv3-tiny

Pengujian kinerja dilakukan untuk mengetahui tingkat akurasi dari *YOLOv3-SPP* dan *YOLOv3-tiny*. Dalam melakukan pengujian kinerja diperlukan dataset uji yang telah dianotasi, di mana dataset uji tersebut merupakan dataset yang tidak pernah digunakan sebagai dataset *training*. Dataset valid yang dibutuhkan sekitar 10% hingga 20% dari total dataset *training*. Dataset valid yang telah dianotasi dan diberi kelas diproses sebagai *ground-truth box* kemudian dibandingkan dengan *predicted box* sehingga menghasilkan *Confusion Matrix*. Selanjutnya dilakukan perhitungan untuk memperoleh nilai Precision,

Recall, *Average Precision* (AP), F1-score, *Intersection over Union* (IoU), dan *Mean Average Precision* (mAP) sesuai dengan rumus yang telah dijelaskan pada perancangan dan implementasi sistem.

Pada pengujian kinerja, masing-masing model menggunakan Google Collaboratory dengan spesifikasi perangkat keras seperti yang ditunjukkan pada Tabel 1.

Tabel 1: Spesifikasi Perangkat Keras yang Digunakan

Komponen	Spesifikasi
Prosesor	Intel(R) Xeon(R) CPU @ 2.30GHz
RAM	12.72 GB
Penyimpanan	68.40 GB
Kartu Grafis	NVIDIA Tesla P4 8GB

#### IV. KESIMPULAN

1. Pada penelitian ini, setiap model mampu mendeteksi penggunaan sabuk pengaman menggunakan *Convolutional Neural Network* (CNN).
2. Berdasarkan hasil *pengujian* kinerja menggunakan data dari internet, *YOLOv3-tiny* memiliki waktu pemrosesan yang lebih cepat yaitu 7 detik, sedangkan *YOLOv3-SPP* membutuhkan waktu 15 detik pada *pengujian* kinerja dataset uji.
3. Berdasarkan hasil *pengujian* kinerja menggunakan data dari internet yang ditunjukkan oleh nilai *mean Average Precision* (mAP), *YOLOv3-tiny* memiliki tingkat akurasi yang lebih tinggi yaitu sebesar 92,6%, sedangkan *YOLOv3-SPP* sebesar 90,2% pada *pengujian* kinerja dataset uji.
4. Berdasarkan hasil *pengujian* kinerja menggunakan data yang diambil secara langsung di lokasi yang ditunjukkan oleh nilai *mean Average Precision* (mAP), *YOLOv3-SPP* memiliki tingkat akurasi yang lebih tinggi yaitu sebesar 65,8%, sedangkan *YOLOv3-tiny* sebesar 47,5% pada *pengujian* kinerja dataset uji.
5. Berdasarkan hasil *pengujian* kinerja menggunakan data yang diambil secara langsung di lokasi dengan menerapkan *Histogram Equalization* yang ditunjukkan oleh nilai *mean Average Precision* (mAP), *YOLOv3-tiny* memiliki tingkat akurasi yang lebih tinggi yaitu sebesar 61%, sedangkan *YOLOv3-SPP* sebesar 35,7% pada *pengujian* kinerja dataset uji.
6. Berdasarkan hasil *pengujian* pendeteksian, penggunaan data masukan berupa foto dari internet menghasilkan tingkat kesalahan sebesar 5%, penggunaan data masukan berupa foto yang diambil secara langsung di lokasi menghasilkan tingkat kesalahan sebesar 9,1%, dan penggunaan data masukan berupa video yang diambil secara langsung di lokasi menghasilkan tingkat kesalahan sebesar 26,6%.

#### DAFTAR PUSTAKA

- [1] R. D. Kusumanto and A. N. Tompunu, "Pengolahan citra digital untuk mendeteksi obyek menggunakan pengolahan warna model normalisasi rgb," *Semantik*, vol. 1, no. 1, 2011.
- [2] R. Patil, P. Ahire, K. Bamane, A. Patankar, P. D. Patil, S. Badoniya, R. Desai, G. Bhandari, and B. S. Dhami, "Real-time traffic sign detection and recognition system using computer vision and machine learning," *Journal of Electrical Systems*, vol. 20, no. 2, 2024.
- [3] H. Ghahremannezhad, H. Shi, and C. Liu, "Real-time accident detection in traffic surveillance using deep learning," in *2022 IEEE International Conference on Imaging Systems and Techniques (IST)*, Kaohsiung, Taiwan, 2022, pp. 1–6.
- [4] National Highway Traffic Safety Administration, "Usdot releases 2016 fatal traffic crash data," 2017. [Online]. Available: <https://nhtsa.gov/>
- [5] World Health Organization, "Save lives-a road safety technical package," 2017. [Online]. Available: <http://www.who.int/>

- [6] V. R. Sajja, D. Venkatesulu, J. N. Rao, and D. B. N. G. S. L., “Driver’s seat belt detection using cnn,” *Turkish Journal of Computer and Mathematics Education (TURCOMAT)*, vol. 12, no. 5, pp. 776–785, 2021.
- [7] I. W. S. E. Putra, “Klasifikasi citra menggunakan convolutional neural network (cnn) pada caltech 101,” 2016.
- [8] P. Chen, X. Bai, and W. Liu, “Vehicle color recognition on an urban road by feature context,” 2014.