

Deteksi Helm pada Pengguna Sepeda Motor dengan Metode Convolutional Neural Network

Albert, Kartika Gunadi, Endang Setyati

Program Studi Informatika Fakultas Teknologi Industri Universitas Kristen Petra

Jl. Siwalankerto 121 – 131 Surabaya 60236

Telp. (031) – 2983455, Fax. (031) – 8417658

E-Mail: vinsentiusalbert@gmail.com, kgunadi@petra.ac.id, endang@stts.edu

ABSTRAK

Dalam rangka memastikan langkah-langkah keamanan, pelanggaran lalu lintas adalah suatu hal yang penting. Salah satu pelanggaran lalu lintas terbanyak adalah penggunaan helm pada pengendara sepeda motor. Oleh karena itu dibuatlah program yang dapat membantu dalam mengidentifikasi pengguna helm pada pengendara sepeda motor. Pada proses identifikasi data masalah yang sering dialami adalah pengenalan karakteristik helm. Dalam penelitian ini akan dilakukan percobaan filter agar dapat mengenali karakteristik dari helm.

Penelitian ini menggunakan 2 metode yaitu *You Only Look Once* (YOLO) dan *Convolutional Neural Network* (CNN). Metode YOLO digunakan untuk mencari region dari sepeda motor dan pengendara sepeda motor. Metode CNN digunakan untuk mengklasifikasi pengguna helm pada pengendara sepeda motor. Hasil dari klasifikasi CNN akan dihitung menggunakan confusion matrix agar mendapatkan akurasi dari prediksi yang benar.

Hasil program dari penelitian ini akan mengidentifikasi pengguna helm pada pengendara motor dalam video. Akurasi yang didapatkan antara pengendara sepeda motor yang mengemudi dengan helm dan tanpa helm adalah 70.49%.

Kata Kunci: *Machine Learning, Artificial Neural Network, Convolutional Neural Network, You Only Look Once, Image*

ABSTRACT

In order to ensure security measures, traffic violations are an important matter. One of the most traffic violations is the use of helmets on motorcycle riders. Therefore, a program was created that could help in identifying helmet users for motorcycle riders. In the process of identifying data, a problem that is often experienced is helmet characteristics. In this study a filter experiment will be conducted in order to recognize the characteristics of the helmet.

This study uses 2 methods, You Only Look Once (YOLO) and Convolutional Neural Network (CNN). The YOLO method is used to find regions of motorbikes and motorbike riders. The CNN method is used to classify helmet users in motorcycle riders. The results of the CNN classification will be calculated using a confusion matrix in order to get the accuracy of the correct prediction.

The program results from this study will identify helmet users on motorcyclists in the video. Accuracy obtained between motorcycle riders driving with helmets and without helmets is 70.49%.

Keywords: *Machine Learning, Artificial Neural Network, Convolutional Neural Network, You Only Look Once, Image*

1. PENDAHULUAN

Tingkat kecelakaan di Indonesia dalam 6 bulan terakhir di dominasi oleh jenis kendaraan sepeda motor [8]. Dengan adanya tingkat kecelakaan yang tinggi setiap tahunnya para pengguna motor diwajibkan untuk menggunakan helm demi meningkatkan keselamatan dari pengguna sepeda motor. Kewajiban menggunakan helm standar nasional Indonesia bagi pengendara sepeda motor diatur dalam Pasal 57 ayat (1) dan ayat (2) UU No. 22 tahun 2009. Ada juga kewajiban mengenai penumpang sepeda motor juga harus menggunakan helm yang memenuhi standar nasional Indonesia sesuai dengan Pasal 106 ayat (8) No. 22/2009. Permasalahan yang didapat adalah pengenalan objek pada video yang digunakan untuk pengenalan helm. Deteksi objek pada gambar beregerak dapat digunakan pula pada kamera pengintai di tempat umum oleh karena itu penelitian ini lebih diarahkan pada gambar bergerak atau video. Penelitian ini juga diarahkan pada multi deteksi yang dilakukan pada penumpang sepeda motor (jumlah orang yang ada diatas sepeda motor). Permasalahan yang biasa ditemui dalam mengenali helm adalah dari bentuk dan ukuran. Helm memiliki bentuk dan ukuran yang lebih besar dibanding penutup kepala lain (topi, jilbab, atau songkok).

Berkaitan dengan sistem keamanan untuk pengendara bermotor, banyak negara yang masih mengadopsi sistem manual dan membutuhkan bantuan manusia. Secara umum sistem tersebut tidak layak karena keterlibatan manusia yang efisiensinya menurun dalam durasi yang lama [2]. Otomatisasi sangat diinginkan untuk memantau pelanggar dan juga mengurangi jumlah manusia sebagai sumber daya yang dibutuhkan. Solusi yang tepat untuk menghadapi masalah ini adalah menggunakan *object detection* pada video kamera pengintai di tempat umum.

Komputer harus mampu mengenali area yang merupakan objek motor pada citra agar dapat mempermudah deteksi helm pada pengguna motor. Proses pengenalan objek dari sebuah citra dikenal dengan istilah YOLO (*You Only Look Once*). Metode untuk pengenalan objek helm pada pengguna sepeda motor tersebut adalah CNN (*Convolutional Neural Network*). Diharapkan 2 metode ini dapat menghasilkan akurasi dan kecepatan yang baik dikarenakan metode YOLO merupakan *state-of-the-art* dalam deteksi objek *real-time* [10] dan metode CNN yang sangat baik dalam memproses data spasial dan mengklasifikasi objek [3]

2. DASAR TEORI

2.1 Helm

Helm merupakan komponen wajib pakai untuk pengendara sepeda motor, di Indonesia helm memiliki standart desain dan pembuatan sesuai dengan Standart Nasional Indonesia (SNI). Berdasarkan SNI helm harus memenuhi persyaratan sebagai berikut [7] :

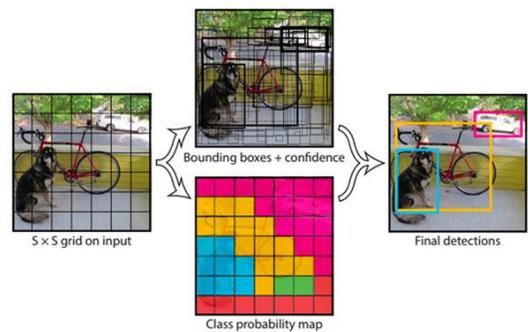
1. Helm harus terdiri dari tempurung yang keras dan memiliki permukaan yang halus, helm juga memiliki tali pengikat ke dagu pengendara.
2. Tinggi helm tidak kurang dari 114 mm yang diukur dari puncak helm ke bidang utamanya, yaitu bagian bawah dari dudukan bola mata dan bidang horizontal melalui lubang telinga.
3. Keliling lingkaran bagian dalam helm adalah S (antara 500 mm– 540 mm, M (540 mm – 580 mm), L (580 mm – 620 mm), XL (lebih dari 620 mm).
4. Tempurung memiliki tekstur keras dan tebal, pelindung tidak menyatu dengan pelindung muka dan mata (khusus helm *half face* bukan untuk helm *full face*).
5. Memiliki peredam beturan yang terdiri dari lapisan peredam kejutan yang dipasang pada bagian dalam tempurung, dengan tebal sekurang-kurangnya 10mm.
6. Memiliki tali pengikat dagu yang lebarnya minimal 20 mm dan harus benar-benar berfungsi sebagai pengikat helm ketika dikenakan di kepala dan dilengkapi dengan penutup telinga dan tengkuk, sesuai dengan konstruksi helm *half face*.
7. Tempurung helm tidak boleh memiliki tonjolan keluar yang tingginya melebihi 5 mm dari permukaan luar tempurung dan setiap tonjolan harus ditutupi dengan bahan lunak. Helm tidak boleh memiliki bagian tepi yang tajam.
8. Lebar sudut pandang sekeliling sekurang-kurangnya 105 derajat pada tiap sisi dan sudut pandang vertikal sekurang-kurangnya 30 derajat di atas dan 45 derajat di bawah bidang utama.
9. Helm harus dilengkapi dengan pelindung telinga, penutup leher, pet yang bisa dipindahkan, tameng atau tutup dagu.

2.2 You Only Look Once (YOLO)

YOLO menerapkan *single neural network* jaringan ini menggunakan fitur dari seluruh gambar untuk memprediksi setiap *bounding box*. YOLO juga dapat memprediksi semua *bounding box* untuk gambar secara bersamaan. YOLO bertugas untuk membagi gambar input menjadi *grid* berukuran $S \times S$. Jika pusat dari suatu objek jatuh ke dalam sel *grid*, maka sel *grid* itu bertanggung jawab untuk mendeteksi objek tersebut. Setiap sel memprediksi B *bounding box* yang di dalamnya berisi *confidence score* untuk menghitung probabilitas objek yang ada di dalam *bounding box* tersebut. Setiap *bounding box* berisi 5 prediksi yaitu: $x, y, w, h, confidence\ score$. Koordinat (x, y) mewakili pusat dari *box relative* menuju batas dari sel *grid*. Lebar dan tinggi memperkirakan kerelatifitasan terhadap seluruh gambar. *Confidence score* bertugas mewakili IOU (*Intersection Over Union*). Setiap sel *grid* juga akan memprediksi 1 set *class* probabilitas $Pr(Class_i | Object)$. *Confidence score* akan dihitung menggunakan persamaan dibawah ini

$$Pr(Class_i | Object) * Pr(Object) * IOU_{truth\ pred} = Pr(Class_i) * IOU_{truth\ pred} \quad (1)$$

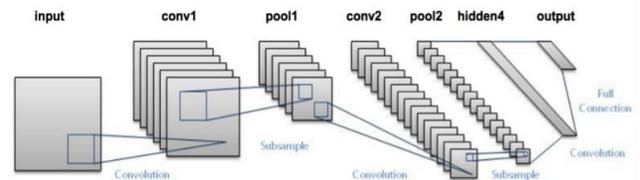
Skor yang dihasilkan mengkodekan kedua probabilitas *class* tersebut seberapa baik kotak yang diprediksi dan cocok dengan objek [10]. Gambar 1. adalah contoh diagram algoritma pengambilan metode *You Only Look Once* (YOLO). Hasil dari YOLO ini yang akan digunakan sebagai pendeteksi region dari sepeda motor. Hasil dari region juga akan didapatkan nilai *confidence* disetiap objek agar dapat digunakan sebagai penentu objek yang digunakan.



Gambar 1. Diagram Algoritma YOLO [11]

2.3 Convolutional Neural Network (CNN)

CNN adalah jaringan saraf yang berisi beberapa *layer* didalamnya diantaranya adalah *convolutional layer*, *pooling layer*, *activation layer*. *Convolutional layer* bertugas untuk memeriksa pola dari suatu gambar dengan cara memberi filter lalu menghasilkan bobot berupa vektor dan bobot tersebut akan bernilai tinggi jika dianggap melihat pola sebelumnya. Kombinasi bobot tinggi dari berbagai filter memungkinkan jaringan memprediksi konten suatu gambar. Lalu setelah memasuki tahap *convolutional layer* jaringan akan memasuki *pooling layer* yang berguna untuk mengurangi spatial dimension tetapi tidak untuk kedalamannya. Cara kerja *pooling layer* adalah dengan cara membagi pooling menjadi *max-pooling* dengan output maksimum nilai dari sub-region dan *average-pooling* dengan output rata-rata nilai dari sub-region. Gambar 2. adalah contoh layer pada convolutional neural network.



Gambar 2. Convolutional Neural Network Layer [4]

Activation Layer berguna untuk menentukan apakah pola tersebut harus dieliminasi atau tidak [12]. Dengan begitu inti dari CNN adalah memfokuskan parameter inputan dengan cara dilatih.

2.4 Confusion Matrix

Confusion matrix adalah suatu informasi mengenai prediksi dan aktual dari suatu sistem klasifikasi yang memiliki data jawaban benar (*supervise*). *Confusion matrix* menggunakan data dalam bentuk matriks [5]. Dari table matriks yang didapatkan akan dihitung akurasi, presisi, F-Score, dan *recall* berdasarkan kondisi data yang diprediksi atau diklasifikasi [1]

Isi dari tabel *confusion matrix* ada 4, yaitu

1. *True Positive* (TP) adalah kondisi dimana model memprediksi data sebagai ya (*TRUE*) dan jawaban aktualnya adalah ya (*TRUE*).
2. *True Negative* (TN) adalah kondisi dimana model memprediksi data sebagai tidak (*FALSE*) dan jawaban aktualnya adalah tidak (*FALSE*).
3. *False Positive* (FP) adalah kondisi dimana model memprediksi data sebagai ya (*TRUE*) dan jawaban aktualnya adalah tidak (*FALSE*).
4. *False Negative* (FN), kondisi dimana model memprediksi data sebagai tidak (*FALSE*) dan jawaban aktualnya adalah ya (*TRUE*).

		Actual Values	
		Positive (1)	Negative (0)
Predicted Values	Positive (1)	TP	FP
	Negative (0)	FN	TN

Gambar 3. Tabel Confusion Matrix [6]

Gambar adalah hubungan antara *True Positive*, *True Negative*, *False Negative*, dan *False Positive* dalam bentuk tabel. Akurasi merupakan suatu tingkat pengukuran dari rasio benar (positif dan negatif) dengan keseluruhan data. Akurasi dapat dihitung menggunakan persamaan (2.2)

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

Presisi merupakan perbandingan jumlah data yang di prediksi benar positif dengan keseluruhan hasil yang diprediksi positif. Presisi dapat dihitung menggunakan persamaan (2.3)

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

Recall merupakan pengukuran pada data dengan klasifikasi positif yang benar. *Recall* dapat dihitung menggunakan persamaan (2.4).

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

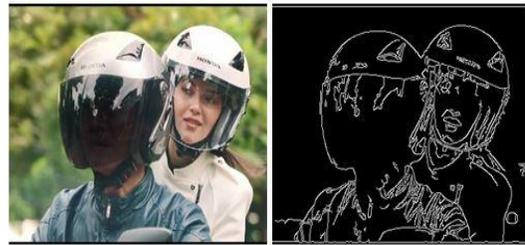
Confusion matrix juga dapat digunakan untuk menghitung *F1 Score* yang bertujuan untuk melakukan perbandingan rata-rata presisi dan *recall*. *F1 Score* dapat dihitung menggunakan persamaan (2.5).

$$F-Score = 2 \cdot \frac{precision \cdot recall}{precision + recall} \quad (5)$$

3. DESAIN SISTEM

3.1 Analisis Helm

Deteksi yang dilakukan mengarah pada objek helm SNI (Standart Nasional Indonesia). Karakteristik dari helm ini adalah terlihat pada warna yang berbeda dibanding kepala manusia yang tidak menggunakan helm. Dari sini intensitas cahaya juga diperhitungkan karena helm dapat memantulkan cahaya dan kepala manusia tidak. Helm juga memiliki bentuk setengah bundar dan menutupi telinga (Hanya wajah yang terlihat). Oleh karena itu ada beberapa filter yang akan digunakan sebagai penunjang dalam pengenalan helm.



Gambar 4. Filter Canny

Filter Canny adalah filter untuk mendapatkan tepi dari gambar agar dapat mengenali bentuk helm yang unik atau setengah lingkaran. Filter Canny dapat dilihat pada Gambar 4.

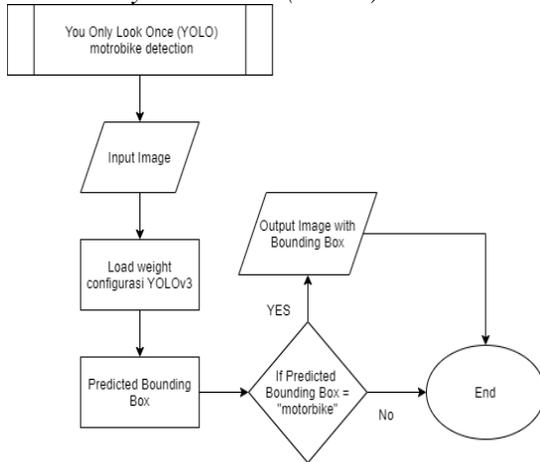
3.2 Analisis Sistem

Flowchart pada Gambar Sistem akan menerima inputan berupa gambar lalu akan diterapkan metode YOLO (*You Only Look Once*) untuk mengetahui lokasi sepeda motor pada video. YOLO disini hanya membantu untuk mengenal lokasi dari sepeda motor, lalu untuk mendapatkan posisi dari pengendara sepeda motor tersebut harus ada penambahan tinggi box dengan cara persentase dari tinggi box (kurang lebih menggunakan 50%-60% dari tinggi sepeda motor). Setelah itu akan digunakan metode CNN (*Convolutional Neural Network*) untuk mengklasifikasi pengguna helm pada pengendara sepeda motor.



Gambar 5. Flowchart Garis Besar Sistem

3.2.1 You Only Look Once (YOLO)



Gambar 6. alur penggunaan yolo

You Only Look Once (YOLO) digunakan untuk mendeteksi objek sepeda motor pada gambar. Dataset yang digunakan adalah YOLOv3 yang diambil dari <https://pjreddie.com/>. *weight* dari dataset yang digunakan sudah mampu untuk mengenali objek sepeda motor dengan akurasi yang cukup tinggi. Dalam hal ini YOLOv3 akan mendapatkan nilai keberadaan *bounding box* dan *confidence score*-nya. Setelah itu YOLO akan mengklasifikasi dan menandai sepeda motor pada *output*. Gambar 6. merupakan *flowchart* dari penggunaan yolo pada penelitian ini. YOLOv3 yang digunakan merupakan hasil train dari COCO trainval yang memiliki mAP (*mean Average Precision*) yang cukup tinggi yaitu 51.5. YOLOv3 menggunakan arsitektur dari Darknet-53 yang artinya memiliki 53 konvolusional layer. Gambar7. adalah layer Darknet-53 yang digunakan sebagai arsitektur YOLOv3

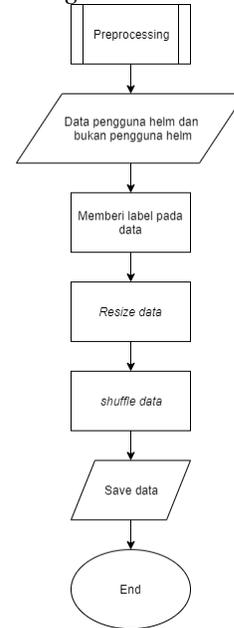
Type	Filters	Size	Output
Convolutional	32	3 × 3	256 × 256
Convolutional	64	3 × 3 / 2	128 × 128
1x Convolutional	32	1 × 1	
Convolutional	64	3 × 3	
Residual			128 × 128
Convolutional	128	3 × 3 / 2	64 × 64
2x Convolutional	64	1 × 1	
Convolutional	128	3 × 3	
Residual			64 × 64
Convolutional	256	3 × 3 / 2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

Gambar 7. Darknet-53 [9]

3.2.2 Convolutional Neural Network (CNN)

Di dalam metode *Convolutional Neural Network* ini akan ada 3 tahap penting yaitu *preprocessing*, *training*, dan *testing* untuk mencapai hasil. 3 tahap ini dibutuhkan untuk mendapatkan *weights* dan model yang tepat agar dapat digunakan untuk mengklasifikasi pengguna helm dan bukan pengguna helm. 3 tahap tersebut akan dibahas pada subbab dibawah.

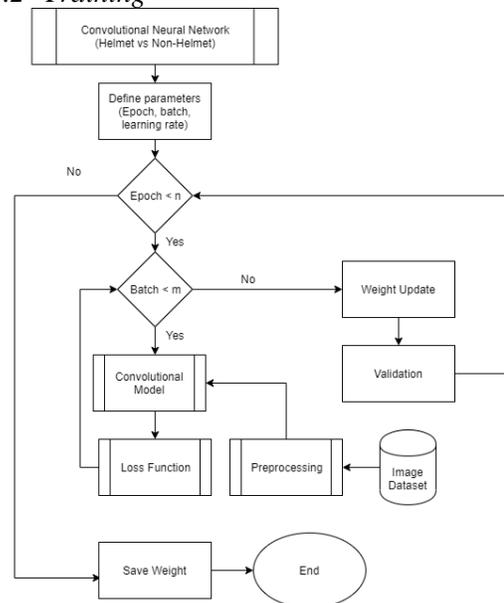
3.2.2.1 Preprocessing



Gambar 8. Alur Preprocessing

Preprocessing digunakan agar data yang masuk sebelum CNN dapat menerima inputan yang benar dan berguna untuk mempermudah proses CNN. Awal dari *Preprocessing* sendiri adalah pemberian label pada data agar dapat di ketahui data mana yang merupakan pengguna helm dan data mana yang merupakan bukan pengguna helm. Setelah itu data akan di lakukan *resize* yang bertujuan untuk memberi ukuran yang sama pada setiap data gambar. Setelah data telah memiliki ukuran yang sama data akan di *shuffle* dengan tujuan agar CNN dapat belajar lebih baik karena dengan adanya *shuffle* data dapat memberi inputan yang berbeda-beda terhadap neuron. Jika semua *preprocessing* selesai maka data akan disimpan terlebih dahulu agar mempermudah *train* CNN. Gambar 8. menunjukan alur dari *preprocessing*.

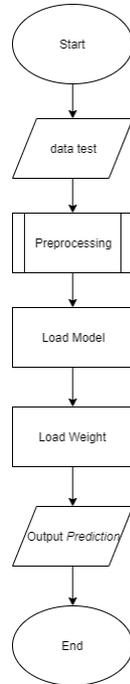
3.2.2.2 Training



Gambar 9. Alur Training CNN

Pada Gambar 9. diawali dengan memberikan parameter *epoch*, *batch size*, dan *learning rate*. Setiap 1 *epoch* terdiri dari beberapa *batch* dan setiap *batch* menyelesaikan tugasnya *weight* akan di-*update*. *Weight* akan di-*update* berdasarkan dari jumlah *loss* yang dihitung. *Loss function* dihitung dari hasil model *convolutional* yang berisi layer-layer CNN. Setelah *epoch* selesai *weight* akan disimpan untuk digunakan sebagai *testing*.

3.2.2.3 Testing



Gambar 10. Alur Testing CNN

Pada Gambar 10. proses *testing* CNN diawali dengan input data *testing* berupa *image*. *Image* akan di-*preprocessing* agar memiliki ukuran yang sama dengan hasil data *training*. Untuk dapat memprediksi hal yang harus dilakukan adalah memberi model yang sama dengan data *training* lalu model akan membaca *weight* data dari hasil *training*.

4. IMPLEMENTASI SISTEM

Implementasi sistem dilakukan pada computer dengan spesifikasi:

- RAM: 8GB, DDR4
- Memory: 512 GB SDD + 1T HDD
- CPU: Intel Core i7
- GPU: NVIDIA GeForce GTX 1650
- OS Windows 10 HOME

Implementasi pengkodean sistem, menggunakan Bahasa pemrograman Python dengan versi 3.7 64bit. Adapun beberapa *library* yang mendukung sistem ini adalah:

- OpenCV
- Tensorflow
- Flask
- Pickle
- Numpy

Aplikasi ini menggunakan tampilan pada web ini dibuat dengan menggunakan arsitektur *Service Oriented*. Data-data yang di-*input* dan dihasilkan diakses melalui berbagai *HTTP Method*, antara lain:

- GET (Menerima data olahan dalam bentuk *array* ataupun *object*)
- POST (Memasukkan data dari web ke program untuk diolah)

5. ANALISA DAN PENGUJIAN

5.1 Pengujian untuk Lokasi Helm

Pengujian ini bertujuan untuk mencari lokasi helm agar lokasi yang didapatkan tidak terpotong. Pada percobaan ini lokasi sepeda motor harus dapat ditemukan terlebih dahulu oleh YOLO. Contoh dari perpotongan YOLO seperti pada Gambar



Gambar 11. Perpotongan YOLO

Dalam percobaan ini diperlukan bagian atas dari sepeda motor, cara untuk mendapatkan bagian atas tersebut adalah menambahkan 50%-60% tinggi dari *bounding box* sepeda motor. Table adalah hasil dari setiap perpotongan tinggi motor dari 50%-60%. Hasil persentase yang ditentukan tidak lebih dari 60% agar gambar tidak memiliki banyak bagian *background* dan posisi helm tetap masuk didalam gambar.

Table 1. Hasil Pengambilan Lokasi *motorcyclist*

n% * h _{bounding box}	Image
50%	
55%	
60%	

Dari hasil percobaan ini dapat ditentukan bahwa 60% adalah angka yg paling bagus dibandingkan angka yang lain. 60% dari tinggi motor ini telah mendapatkan posisi helm secara keseluruhan tanpa terpotong. Akan tetapi metode pengambilan lokasi ini kurang bagus karena perbedaan terhadap tinggi pengendara sepeda motor dapat mempengaruhi *bounding box* yang didapatkan.

5.2 Pengujian Program

Pada pengujian program ini akan dibagi menjadi dua bagian yaitu pengujian malam hari dan siang hari.

5.2.1 Pengujian Siang Hari

Pengujian program siang hari dilakukan dengan mencoba melakukan prediksi terhadap video dengan durasi 1 menit. Dalam pengujian ini gambar yang terlalu kecil (dibawah 80x80 pixel) akan dibuang dan tidak dilakukan prediksi karena sulitnya untuk mengenali helm atau bukan. Pada pengujian ini dilakukan metode *confusion matrix* sebagai penentu *F1-score*, *recall*, dan *precision*. Format yang dilakukan adalah sebagai berikut :

Perincian *confusion matrix* pada video berukuran 720x480:

1. Bukan pengguna helm

- *True Positive* : 6
- *True Negative* : 37
- *False Positive* : 10
- *False Negative* : 8

2. Pengguna helm

- *True Positive* : 37
- *True Negative* : 6
- *False Positive* : 8
- *False Negative* : 10

Dalam uji coba ini ada 77 gambar yang didapatkan dan diklasifikasi. Dalam pengklasifikasiannya, jumlah data yang tidak dapat dipakai adalah 16 hal ini dikarenakan YOLO tidak mampu mencari posisi pengendara sepeda motor dan ada beberapa data yang rusak akibat *motion blur* atau kecepatan dari sepeda motor membuat video tidak mampu untuk menangkap gambar dengan sempurna.

Table 2. Hasil F1-score 720x480

Jenis	<i>Precision</i>	<i>Recall</i>	<i>F1-score</i>	<i>Accuracy</i>
Non_Helm	37.5%	42.85%	40%	70.49%
Helm	82.22%	78.72%	80.43%	70.49%

Table 2. menunjukkan bahwa model lebih cenderung dapat mengklasifikasi Helm akan tetapi kurang dalam mengklasifikasi non helm. Dalam hasil coba ini gambar yang didapatkan kurang maksimal sehingga untuk percobaan berikutnya akan digunakan data video yang sama dengan ukuran 1280x720.

Perincian *confusion matrix* pada video berukuran 1280x720:

1. Bukan pengguna helm

- *True Positive* : 24
- *True Negative* : 113
- *False Positive* : 42

- *False Negative* : 17
- #### 2. Pengguna helm
- *True Positive* : 113
 - *True Negative* : 24
 - *False Positive* : 17
 - *False Negative* : 42

Uji coba video dengan ukuran lebih besar didapatkan 214 gambar. Dalam pengklasifikasiannya 18 gambar tidak dapat digunakan karena YOLO tidak mampu mencari posisi pengendara motor. Pengujian kali ini memiliki hasil gambar yang lebih jelas dari sebelumnya dan jumlah gambar yang didapatkanpun lebih banyak dari uji coba sebelumnya.

Table 3. Hasil F1-score 1280x720

Jenis	<i>Precision</i>	<i>Recall</i>	<i>F1-score</i>	<i>Accuracy</i>
Non_Helm	36.36%	58.53%	44.85%	69.89%
Helm	86.92%	72.90%	79.29%	69.89%

Table 3. Hasil F1-score 1280x720 menunjukkan bahwa hasil dari ukuran video tidak banyak mempengaruhi akurasi.

5.2.2 Pengujian Malam Hari

Pengujian ini dilakukan saat malam hari dengan tempat yang sama dan format yang sama dengan pengujian siang hari. Untuk pengujian pada video berukuran 720x480 tidak dapat dilakukan karena YOLO tidak mampu mengambil gambar yang memiliki ukuran 80x80 pixel. Karena video 720x480 tidak dapat dilakukan maka video yg digunakan selanjutnya adalah 1280x720.

Perincian *confusion matrix* pada video berukuran 1280x720:

1. Bukan pengguna helm

- *True Positive* : 1
- *True Negative* : 23
- *False Positive* : 12
- *False Negative* : 8

2. Pengguna helm

- *True Positive* : 23
- *True Negative* : 1
- *False Positive* : 12
- *False Negative* : 8

Pada pengujian ini YOLO tidak mampu mendapatkan keseluruhan pengendara motor dikarenakan gangguan dari *flare* dan *shutter speed* yang menyebabkan gambar terlihat lebih *blur* dan terjadi pantulan cahaya yang sangat keras dari lampu sepeda motor.

Table 4. Hasil F1-score 1280x720 malam hari

Jenis	<i>Precision</i>	<i>Recall</i>	<i>F1-score</i>	<i>Accuracy</i>
Non_Helm	7.69%	11.11%	9.09%	54.54%
Helm	74.19%	65.71%	69.69%	54.54%

Table 4. menunjukkan bahwa hasil dari pengujian malam hari lebih buruk dibanding pengujian siang hari. Pada pengujian ini tetap memiliki akurasi pengenalan terhadap pengguna helm yang lebih tinggi daripada bukan pengguna helm.

6. KESIMPULAN

Setelah dilakukan perancangan sistem, pengimplementasian, dan pengujian terhadap aplikasi yang telah dibuat, dapat ditarik kesimpulan sebagai berikut:

- *You Only Look Once (YOLO)* memiliki hasil yang bagus akan tetapi *YOLOv3* yang digunakan masih mengalami kesalahan dalam mengenal sepeda atau sepeda motor. *YOLOv3* juga mengalami kesalahan ketika 2 objek motor berdekatan karena akan dianggap satu region sepeda motor.
- Convolutional Neural Network (CNN) memiliki precision yang lebih rendah pada pengendara sepeda motor yang tidak menggunakan helm yaitu sebesar 37.5% dan hasil untuk pengendara sepeda motor yang menggunakan helm sebesar 82.22% (yang dilakukan pada siang hari dengan ukuran video 720x480). Metode CNN kurang tepat untuk digunakan sebagai multi detection (contoh : dalam satu sepeda motor memiliki 2 penumpang, dimana penumpang pertama menggunakan helm dan penumpang kedua tidak menggunakan helm) dikarenakan banyaknya informasi yang diterima oleh metode ini sehingga menyebabkan kebingungan.
- Penambahan persentase tinggi sepeda motor memiliki hasil yang baik dengan parameter 60% dari tinggi sepeda motor karena region yang diambil sudah pas tidak memiliki banyak background yang diambil dan tinggi pengendara dapat diambil seluruhnya.

Saran untuk pengembangan kedepannya adalah:

- Penambahan *dataset* yang lebih jelas dan bervariasi.
- Penambahan metode untuk mendapatkan posisi helm satu per satu agar mendapatkan hasil yang lebih maksimal.
- Penambahan atau penggunaan *layer* lain pada model CNN agar dapat menambah akurasi dari klasifikasi.
- Mengganti *device* untuk pengambilan video dengan pengaturan shutter speed (digunakan untuk dan flare (efek yang timbul saat berhadapan dengan sumber cahaya yang keras) yang lebih baik.

7. DAFTAR PUSTAKA

- [1] *dataschool*. 2014, March 25. Retrieved November 19, 2019, from *dataschool*: <https://www.dataschool.io/simple-guide-to-confusion-matrix-terminology/>
- [2] Adam, A., Rivlin, E., Shimshoni, I., & Reinitz, D. 2008. IEEE Transactions on Pattern Analysis and Machine Intelligence. *Robust Real-Time Unusual Event Detection using Multiple Fixed-Location Monitors*, 30(3), 555-560.
- [3] Chen, Y., Jiang, H., Li, C., Jia, X., & Ghamisi, P. 2016. IEEE Transactions on Geoscience and Remote Sensing. *Deep Feature Extraction and Classification of Hyperspectral Images Based on Convolutional Neural Networks*, 54(10), 6232-6251.
- [4] Gupta, D. 2017, June 29. *Architecture of Convolutional Neural Networks (CNNs) demystified*. Retrieved Oktober 22, 2019, from Architecture of Convolutional Neural Networks (CNNs) demystified: <https://www.analyticsvidhya.com/blog/2017/06/architecture-of-convolutional-neural-networks-simplified-demystified/>
- [5] Hamilton, H. 2018, July 9. *Computer Science*. Retrieved November 1, 2019, from Knowledge Discovery in Databases: <http://www2.cs.uregina.ca/~dbd/cs831/index.html>
- [6] Narkhede, S. 2018, May 9. *Understanding Confusion Matrix*. Retrieved November 2019, 20, from Understanding Confusion Matrix: <https://towardsdatascience.com/understanding-confusion-matrix-a9ad42dcfd62>
- [7] Nayazri, G. M. 2018, February 4. *Home : News*. Retrieved November 10, 2019, from Kompas Web site: <https://otomotif.kompas.com/read/2018/02/04/090200615/in-i-standar-helm-sepeda-motor-sesuai-sni?page=all>
- [8] POLRI, K. 2013. *ACCIDENT DATA: VEHICLE TYPE*. Retrieved September 11, 2019, from KORLANTAS POLRI Web Site: <http://korlantas-irsms.info/graph/vehicleTypeData>
- [9] Redmon, J., & Farhadi, A. 2018. *YOLOv3: An Incremental Improvement*.
- [10] Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. 2016. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). *You Only Look Once: Unified, Real-Time Object Detection*.
- [11] Rosebrock, A. 2018, November 12. *pyimagesearch*. Retrieved Oktober 20, 2019, from *pyimagesearch*: <https://www.pyimagesearch.com/2018/11/12/yolo-object-detection-with-opencv/>
- [12] Shubham, J. 2018, July 15. *Medium Web site*. Retrieved Oktober 20, 2019, from What exactly does CNN see?: <https://becominghuman.ai/what-exactly-does-cnn-see-4d436d8e6e52>