

Implementasi Pengenalan Rambu Berdasarkan *Deep Neural Network* Untuk Lajur Pada *Quadcopter*

Implementation Of Sign Recognition Based On Deep Neural Network For Quadcopter Trajectory

1st Mohammad Zachary
Abidin
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia
mhdzacharyabidin@student.t
elkomuniversity.ac.id

2nd Suryo Adhi Wibowo
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia
suryoadhiwibowo@telkomun
iversity.ac.id

3rd Fityanul Akhyar
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia
fityanul@telkomuniversity.ac
.id

Abstrak—*Object detection* merupakan salah satu bidang yang diminati dalam perkembangan *Deep Neural Network* (DNN) pada *computer vision*. Dengan adanya perkembangan *Artificial Intelligent* pada bidang *computer vision*, pekerjaan manusia tentunya dapat terbantu dengan implementasi dari sistem *object detection* menggunakan *Unmanned Aerial Vehicle* (UAV) berjenis *quadcopter*. *Quadcopter* dipilih karena memiliki kemampuan untuk lepas landas dan terbang secara vertikal, sehingga tidak memerlukan area yang luas untuk penggunaannya. Berdasarkan permasalahan tersebut, penelitian ini, telah dirancang sistem *object detection* menggunakan algoritma YOLOv5 Nano yang akan disandingkan pada *quadcopter* dan dilakukan penambahan teknik regularisasi *dropout* sebesar 10%, 20%, dan 30%. Hasil dari penelitian ini yaitu *quadcopter* dapat mengenali rambu yang diberikan menggunakan model objek detektor YOLOv5 Nano dan mengetahui nilai *confidence score* dari setiap jarak yang ditentukan pada saat pengujian. Dengan mengenali rambu yang diberikan, *quadcopter* seharusnya dapat membuat keputusan dengan sendirinya saat mengenali rambu dengan tepat dan bermanuver sesuai instruksi rambu yang dilihatnya. Dari penelitian ini, didapatkan nilai mAP pada YOLOv5 Nano sebesar 0.994, dengan teknik regularisasi *dropout* 10% didapat mAP sebesar 0.993, teknik regularisasi *dropout* 20% didapat mAP sebesar 0.994, teknik regularisasi *dropout* 30% didapat mAP sebesar 0.994.

Kata Kunci — yolo, yolov5, quadcopter, object detection, sign recognition

Abstract—*Object detection* is one of the areas of interest in the development of *Deep Neural Network* (DNN) in *computer vision*. With the development of *Artificial Intelligence* in the field of *computer vision*, human work can certainly be helped by the implementation of an *object detection* system using an *Unmanned Aerial Vehicle* (UAV) of the *quadcopter* type. The *quadcopter* was chosen because it could take off and fly vertically, so it does not require a large area for its use. Based on these problems, in this study, an *object detection* system has been designed using the YOLOv5 Nano algorithm which will be paired with a *quadcopter* and an additional 10%, 20%, and 30% *dropout* regularization techniques have been added. The results of this study are that the *quadcopter* can recognize the given sign using the YOLOv5 Nano detector object model and find out the *confidence score* for each distance determined during the test. By recognizing the signs given, the *quadcopter* should be able to make decisions on its own when it comes to recognizing the signs correctly and maneuvering according to the instructions of the signs it sees. From this research, the mAP value of YOLOv5 Nano is 0.994, with 10% *dropout* regularization technique, mAP is 0.993, 20% *dropout* regularization technique is 0.994, 30% *dropout* regularization technique is 0.994.

Keywords— internet of things (iot), heart rate, gps, and temperature

I. PENDAHULUAN

Artificial Intelligence telah menjadi penggerak revolusi industri 4.0 yang memberikan banyak kemudahan dalam melakukan pekerjaan baik dalam bidang pemerintahan maupun industri. Hal ini digerakkan oleh banyak data dan diimplementasikan kedalam teknologi Artificial Intelligence (A.I). Untuk pekerjaan yang sifatnya mengulang, dapat digantikan oleh mesin. Salah satu jenis pekerjaan tersebut adalah membawa barang atau alat dari suatu tempat ke tempat lainnya menggunakan *Unmanned Aerial Vehicle* (UAV) berjenis *Quadcopter* dengan menerapkan sistem automasi dengan cara mendeteksi dan mengenali objek berupa rambu-rambu perintah yang diberi [4]. *Object detection* adalah salah satu yang menggunakan konsep *deep learning* dalam *computer vision*. Pengenalan rambu adalah salah satu bagian terpenting dari teknologi kendaraan otonom berbasis *deep learning* agar dapat beroperasi dengan aman [3]. Penelitian kali ini UAV berjenis *Quadcopter* dipilih untuk mendeteksi rambu-rambu perintah otomasi dari UAV itu sendiri.

Quadcopter akan diimplementasikan sistem kecerdasan buatan. Kecerdasan Buatan ini yang akan mendeteksi rambu-rambu perintah dengan kamera yang terpasang. Deteksi objek berbasis *deep learning* memang telah berkembang pesat, tetapi masih menjadi sebuah tantangan dalam perspektif *Unmanned Aerial Vehicle* (UAV). Untuk mendeteksi rambu, akan digunakan metode *You Only Look Once* (YOLO) versi 5. YOLOv5 adalah salah satu model terbaik yang ada untuk deteksi objek saat ini dengan performa yang lebih tinggi dan kinerja yang lebih ringan dari versi sebelumnya [19].

YOLO menerapkan neural network pada sebuah citra, kemudian membagi citra menjadi daerah dan memprediksi *bounding box* serta probabilitas untuk masing-masing daerah. Probabilitas untuk setiap *bounding box* kemudian dihitung untuk mengklasifikasikan sebagai objek atau bukan. Metode *You Only Look Once* dapat melakukan pengenalan objek secara *real time* dengan kecepatan 45 *frame per second* [1]. Karena akan dilakukan deteksi rambu secara *real time* menggunakan *quadcopter* yang berukuran kecil, maka akan digunakan model YOLOv5 *nano* yang merupakan model paling kecil diantara varian YOLOv5 lainnya dengan tujuan untuk mengurangi waktu *delay* pada saat mendeteksi rambu.

Pada penelitian ini juga ditambahkan teknik regularisasi *dropout* sebesar 10%, 20%, dan 30% pada model YOLOv5 *nano* yang digunakan.

II. KAJIAN TEORI

A. Object Detection (Deteksi Objek)

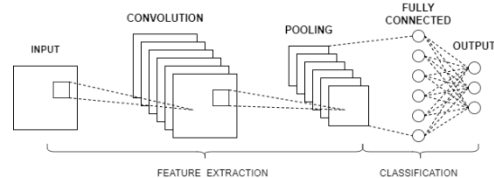
Deteksi objek adalah sebuah proses untuk menentukan di mana objek berada dalam gambar yang diberikan (lokalisasi objek) dan menentukan kategori mana saja yang dimiliki objek itu (klasifikasi objek) [8]. Pada umumnya, deteksi objek dapat dibagi menjadi tiga tahapan: pemilihan wilayah objek yang di deteksi, ekstraksi fitur, dan klasifikasi dari objek yang ditangkap. Dalam penerapannya, objek deteksi banyak digunakan dalam berbagai hal seperti klasifikasi objek pada gambar, deteksi rambu untuk *autonomous car*, dan lain sebagainya.

B. Sign Recognition (Pengenalan Rambu)

Traffic Sign Recognition atau pengenalan pada rambu lalu lintas adalah komponen utama dalam *traffic sign assist system* untuk memberikan instruksi dan peringatan kepada pengemudi di waktu yang tepat mengenai pembatas dan informasi rambu lalu lintas. Untuk sistem mobil otonom yang dapat berjalan sendiri, input dari pengenalan rambu akan digunakan mobil untuk membuat keputusan sendiri. Menggunakan konsep yang sama seperti pengenalan rambu, konsep ini juga yang akan digunakan pada penelitian kali ini. Dengan mengenali rambu, *quadcopter* akan membuat keputusan sendiri sesuai dengan rambu yang telah dikenalnya.

C. Convolutional Neural Network (CNN)

Istilah *Deep Learning* atau *Deep Neural Network* (DNN) mengacu pada jaringan syaraf tiruan dengan multi-layer [10]. Salah satu DNN yang paling populer adalah *Convolutional Neural Network* (CNN). CNN memiliki beberapa lapisan yang diantaranya adalah *convolutional layer*, *pooling layer*, dan *fully-connected layer* [20].



GAMBAR 1
ARSITEKTUR CNN

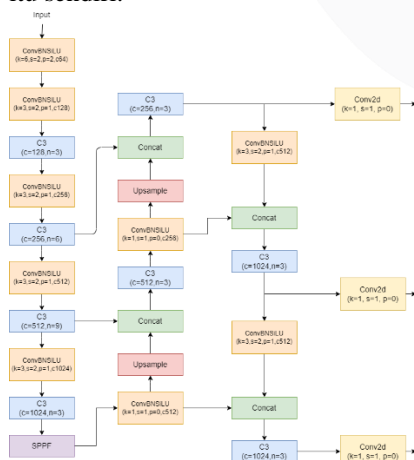
D. You Only Look Once (YOLO)

YOLO bekerja secara *real time* dengan mempertimbangkan deteksi sebagai

masalah regresi daripada klasifikasi. Seluruh proses deteksi dilakukan dalam satu evaluasi gambar, itulah mengapa algoritma ini disebut *you only look once*. Cara kerja YOLO yaitu membagi gambar *input* ke dalam *grid* $S \times S$ dan memprediksi *bounding box*, *confidence score* dari *bounding box* tersebut, dan menentukan probabilitas kelas objek yang di deteksi secara serentak. Untuk *bounding box* itu sendiri memiliki beberapa atribut, diantaranya x dan y untuk koordinat kotak, w untuk lebar kotak, h untuk tinggi kotak, serta *confidence score*.

1. YOLOv5

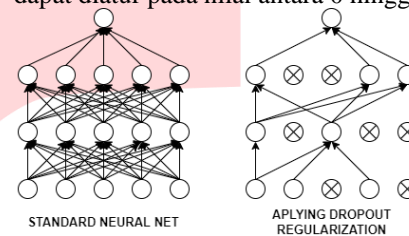
YOLOv5 memiliki beberapa layer seperti Conv, C3, Concat, Upsample, SPPF, dan memiliki 3 keluaran feature map dan juga menggunakan CSPDarknet53 sebagai backbone. YOLOv5 menggunakan Path Aggregation Network (PANet) sebagai jaringan pada neck untuk mempercepat aliran informasi. PANet menggunakan Feature Pyramid Network (FPN) yang mencakup beberapa lapisan bottom up dan top down sehingga dapat meningkatkan propagasi level rendah fitur dalam model. Conv menunjukkan convolution layer. C3 terdiri dari tiga convolution layer dan modul cascade dari berbagai bottleneck. Spatial Pyramid Pooling - Fast (SPPF) adalah pyramid pooling layer untuk menghilangkan ukuran pembatas pada jaringan. Upsample digunakan dalam upsampling fusi lapisan sebelumnya di node terdekat. Concat adalah lapisan pengiris dan digunakan untuk mengiris lapisan sebelumnya. Tiga Conv2d terakhir adalah modul deteksi yang digunakan di kepala dari jaringan YOLOv5 itu sendiri.



GAMBAR 2 STRUKTUR JARINGAN YOLOV5

E. Dropout

Dropout merupakan proses mencegah terjadinya *overfitting* dan juga mempercepat proses learning. Dropout mengacu kepada menghilangkan neuron yang berupa *hidden* maupun *layer* yang *visible* di dalam jaringan. Neuron yang akan dihilangkan akan dipilih secara acak. Setiap neuron akan diberikan probabilitas yang bernilai antara 0 dan 1. Dengan memutuskan unit, maksudnya adalah untuk sementara menghapusnya dari jaringan, bersamaan dengan semua koneksi masuk dan keluarnya, seperti yang ditunjukkan pada Gambar 2.9. Pilihan unit mana yang akan dijatuhkan adalah acak. Dalam kasus yang paling sederhana, setiap unit dipertahankan dengan probabilitas tetap p dari unit lain, di mana p dapat dipilih menggunakan validasi atau dapat diatur pada nilai antara 0 hingga 1.



GAMBAR 3 ILUSTRASI DROPOUT

F. Intersection over Union (IoU)

Salah satu cara untuk mengevaluasi model deteksi objek adalah dengan menggunakan *Intersection over Union* (IoU). *Intersection over Union* (IoU) adalah nilai evaluasi antara dua *Bounding Box* (BB), yaitu *Bounding Box prediction* dengan *Bounding Box ground truth*. *Intersection Over Union* (IoU) didapat dari perhitungan antara irisan *Bounding Box prediction* dengan *Bounding Box ground truth* yang dibagi dengan gabungan antara *Bounding Box prediction* dengan *Bounding Box ground truth*.

$$IoU = \frac{area(box(Pred) \cap box(Truth))}{area(box(Pred) \cup box(Truth))}$$

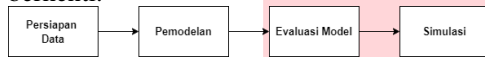
G. Quadcopter

Quadcopter, atau juga bisa dikenal dengan quadrotor, adalah sejenis helikopter dengan empat lengan dan satu rotor di tiap lengannya. Quadcopter adalah salah satu jenis Unmannad Aerial Vshicles (UAV) karena dapat dikendalikan tanpa operator. Quadcopter dapat digunakan dalam search and rescue, inspeksi konstruksi, dan beberapa pengaplikasian lainnya termasuk membawa barang dari suatu tempat ke tempat lainnya.

III. METODE

A. Desain Sistem

Pada tugas akhir ini, sistem dirancang agar dapat mendeteksi objek berupa rambu yang telah diberikan menggunakan YOLOv5 Nano. Data akan di *training* menggunakan YOLOv5 Nano dan ditambah dengan teknik regularisasi *dropout* sebesar 10%, 20%, dan 30%. Hasil deteksi tersebut akan membuat quadcopter dapat mendeteksi objek berupa rambu secara real time dan menentukan bounding box berdasarkan rambu yang dideteksi bersamaan dengan confidence score-nya. Dataset rambu yang diberikan berupa rambu belok kanan, rambu belok kiri, dan rambu *stop* atau berhenti.



GAMBAR 4
DIAGRAM ALIR SISTEM

1. Persiapan Data

Persiapan data merupakan langkah paling awal dan akan mempengaruhi proses selanjutnya hingga menghasilkan keluaran akhir dari model yang telah dibuat. Persiapan data dapat berupa akuisisi data atau pembuatan *dataset* secara mandiri, kemudian dilakukan penyeimbangan pada data, pemberian label pada *dataset* sesuai dengan kelasnya.

2. Pemodelan

Dalam proses pelatihan data YOLOv5 diberi gambar *input* sebesar 416 x 416. Proses *training* dilakukan dengan *batch default* sebesar 16 *batch* dan proses pelatihan dilakukan sebanyak 300 *epoch*. Proses pelatihan juga diberi *dropout* yang masing-masing sebesar 10%, 20%, dan 30% sehingga terdapat empat jenis model YOLOv5 Nano yang berbeda. *Weight* terbaik dipilih dari setiap model dan dijalankan pada *device* untuk menguji hasil pengenalan objek rambu pada *quadcopter*.

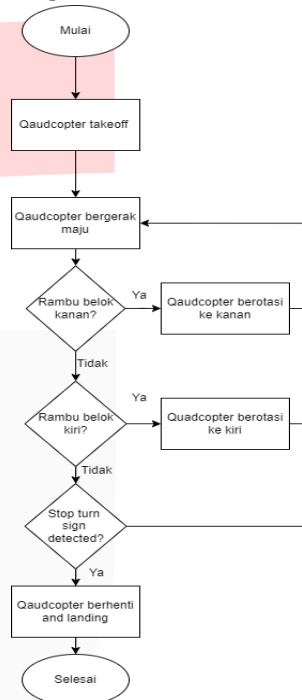
3. Evaluasi Model

Evaluasi model diperlukan untuk melihat kemampuan seberapa baik model yang dibuat dalam mendeteksi objek. *Mean average precision* (mAP) digunakan sebagai metrik untuk menganalisis keakuratan model deteksi objek. *Precision* digunakan untuk mengukur akurasi prediksi, sedangkan *recall*

digunakan untuk mengukur jumlah prediksi dengan *ground truth*.

4. Implementasi Sistem

Dengan *weight* terbaik dari proses *training* dan model yang ditambahkan teknik regularisasi *dropout* sebelumnya, *weight* tersebut akan dijalankan pada *device* berupa laptop atau komputer. Dari laptop atau komputer akan dikoneksikan ke kamera yang sudah ada pada *quadcopter* melalui sambungan *wi-fi*. *Quadcopter* akan melakukan keputusan sendiri sesuai dengan perintah pada rambu yang tertangkap pada kamera *quadcopter*.



GAMBAR 4
DIAGRAM ALIR IMPLEMENTASI PADA ALAT

B. Desain Perangkat

Terdapat beberapa perangkat baik perangkat lunak maupun perangkat keras yang digunakan guna mendukung penelitian dan memberikan hasil yang baik. Perangkat keras yang digunakan pada penelitian ini berupa laptop dan *Dji tello drone*. Laptop digunakan untuk memproses *dataset* dan melakukan *training*. Laptop juga digunakan sebagai alat untuk menjalankan sistem yang sudah dibuat.

C. Parameter Performansi

Pada tugas akhir ini digunakan beberapa parameter performansi seperti *precision*, *recall*, dan *mean Average Precision* (mAP). Parameter ini digunakan untuk melihat

seberapa baik model dalam mendeteksi gambar.

1. *Precision*

Precision menggambarkan tingkat keakuratan antara data yang diminta dengan hasil prediksi yang diberikan oleh model.

2. *Recall*

Recall menggambarkan keberhasilan model dalam menemukan kembali sebuah informasi. *Recall* adalah peluang benar positif dibagi dengan seluruh hasil data yang diprediksi benar positif.

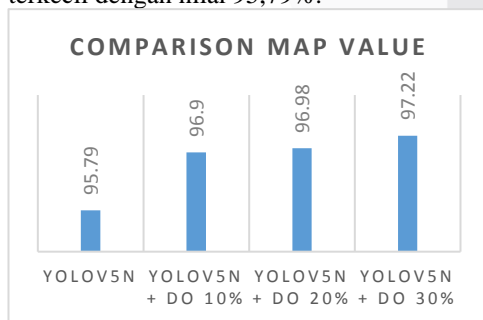
3. *Mean Average Precision (mAP)*

Masing-masing kelas objek gambar dapat memiliki nilai *precision*. Untuk mengetahui performa model deteksi secara keseluruhan, maka nilai *precision* pada masing-masing kelas diambil nilai rata-ratanya. Nilai ini disebut *Average Precision (AP)*. Nilai AP terdiri dari beberapa kriteria sesuai dengan IoU yang ditentukan.

IV. HASIL DAN PEMBAHASAN

A. Analisis mAP

Dapat dilihat pada Gambar 4.1 bahwa mAP@.95 tertinggi didapat pada model YOLOv5n ditambahkan *dropout* 30% dengan nilai sebesar 97,22%. Model YOLOv5n dengan *dropout* 20% mendapat nilai mAP@.95 sebesar 96,98%. Model YOLOv5n dengan *dropout* 10% mendapat nilai mAP@.95 sebesar 96.9%. Model YOLOv5 orisinal mendapat nilai mAP@.95 terkecil dengan nilai 95,79%.



GAMBAR 8
PERBANDINGAN NILAI MAP

B. Skenario Pengujian Sistem

Tugas Akhir ini diuji menggunakan empat *weight* dari hasil training dengan berbagai macam skenario pengujian yang berbeda. *Weight* yang digunakan merupakan satu hasil model YOLOv5 nano dan tiga model lainnya ditambahkan teknik

regularisasi *dropout* 10%, 20% dan 30%. Data dilatih menggunakan dataset rambu belok kanan, belok kiri, dan rambu berhenti. Jarak antara *quadcopter* dengan rambu diatur sejauh 25cm, 50cm, 75cm, dan 100cm. Berikut ini adalah skenario yang diujikan:

1. Skenario 1: Pengujian pada jam 7.30 – 8.30
2. Skenario 1: Pengujian pada jam 11.30 – 12.30
3. Skenario 1: Pengujian pada jam 17.30 – 18.30

C. Hasil dan Analisis Pengujian Sistem

1. Skenario 1: Pengujian pada jam 7.30-8.30

TABEL 1
HASIL PENGUJIAN SKENARIO 1

		25 cm	50 cm	75 cm	100 cm
Stop Sign	Normal	0.95	0.94	0	0
	Dropout 10%	0.86	0.94	0.8	0.42
	Dropout 20%	0.8	0.93	0.85	0.45
	Dropout 30%	0.78	0.91	0.82	0.45
Average		0.84	0.93	0.61	0.33
Turn Left Sign	Normal	0.87	0.64	0	0
	Dropout 10%	0.89	0.93	0.83	0
	Dropout 20%	0.85	0.88	0.79	0.48
	Dropout 30%	0.79	0.78	0.61	0
Average		0.85	0.8	0.55	0.12
Turn Right Sign	Normal	0.68	0.88	0	0
	Dropout 10%	0.82	0.85	0.76	0
	Dropout 20%	0.82	0.76	0.83	0.42
	Dropout 30%	0.72	0.85	0.78	0
Average		0.705	0.83	0.59	0.12

Tabel 1 merupakan hasil *confidence score* masing-masing objek rambu berdasarkan empat jarak yang berbeda di pagi hari. Empat *weight* dari hasil *training* diimplementasikan pada *quadcopter* sehingga mendeteksi rambu dan menunjukkan seberapa yakin model dalam mendeteksi rambu yang diberi.

a. Analisis Skenario 1

Dari rata-rata *confidence score* yang didapat pada tabel 1, model paling baik mendeteksi rambu *stop* pada jarak 50cm. Untuk rambu *right*, model juga paling baik mendeteksi rambu pada jarak 50cm. Untuk rambu *left*, model paling baik mendeteksi pada jarak 25cm. Pada jarak 100cm, model mendapatkan *confidence score* yang rendah di semua kelas.

2. Skenario 2: Pengujian pada jam 11.30-12.30

TABEL 2
HASIL PENGUJIAN SKENARIO 2

		25 cm	50 cm	75 cm	100 cm
Stop Sign	Normal	0.85	0.91	0.86	0.6
	Dropout 10%	0.74	0.83	0.8	0.47
	Dropout 20%	0.8	0.95	0.79	0.42
	Dropout 30%	0.72	0.82	0.76	0
	Average	0.77	0.87	0.80	0.37
Turn Left Sign	Normal	0.9	0.83	0.79	0.26
	Dropout 10%	0.7	0.9	0.72	0.43
	Dropout 20%	0.82	0.87	0.86	0.5
	Dropout 30%	0.66	0.87	0.73	0
	Average	0.77	0.86	0.775	0.29
Turn Right Sign	Normal	0.9	0.92	0.83	0.6
	Dropout 10%	0.78	0.87	0.81	0.45
	Dropout 10%	0.74	0.88	0.8	0.42
	Dropout 10%	0.62	0.72	0.64	0
	Average	0.76	0.84	0.77	0.36

Tabel 2 merupakan hasil *confidence score* masing-masing objek rambu berdasarkan empat jarak yang berbeda di pagi hari. Empat *weight* dari hasil *training* diimplementasikan pada *quadcopter* sehingga mendeteksi rambu dan menunjukkan seberapa yakin model dalam mendeteksi rambu yang diberi.

a. Analisis Skenario 2

Dari rata-rata *confidence score* yang didapat pada tabel 2, model paling baik

mendeteksi rambu *stop* pada jarak 50cm. Untuk rambu *left*, model juga paling baik mendeteksi rambu pada jarak 50cm. Untuk rambu *right*, model paling baik mendeteksi pada jarak 25cm. Pada jarak 100cm, model mendapatkan *confidence score* lebih tinggi daripada skenario 1.

3. Skenario 3: Pengujian pada jam 17.30-18.30

TABEL 3
HASIL PENGUJIAN SKENARIO 3

		25c m	50c m	75c m	100c m
Stop Sign	Normal	0.5	0.8	0	0
	Dropou t 10%	0.8	0.87	0	0
	Dropou t 20%	0.82	0.91	0.6	0
	Dropou t 30%	0.8	0.92	0.6	0
	Average	0.73	0.875	0.3	0
Turn Left Sign	Normal	0.57	0.8	0	0
	Dropou t 10%	0.7	0.76	0	0
	Dropou t 20%	0.73	0.82	0.7	0
	Dropou t 30%	0.71	0.91	0.5	0
	Average	0.67	0.82	0.3	0
Turn Right Sign	Normal	0.61	0.79	0	0
	Dropou t 10%	0.6	0.86	0	0
	Dropou t 20%	0.8	0.78	0.68	0
	Dropou t 30%	0.74	0.8	0.5	0
	Average	0.68	0.80	0.29	0

Tabel 3 merupakan hasil *confidence score* masing-masing objek rambu berdasarkan empat jarak yang berbeda di pagi hari. Empat *weight* dari hasil *training* diimplementasikan pada *quadcopter* sehingga mendeteksi rambu dan menunjukkan seberapa yakin model dalam mendeteksi rambu yang diberi.

a. Analisis Skenario 3

Dari rata-rata *confidence score* yang didapat pada tabel 3, model paling baik mendeteksi rambu *stop* pada jarak 50cm. Untuk rambu *left*, model juga paling baik mendeteksi rambu pada jarak 50cm. Untuk rambu *right*, model paling baik mendeteksi pada jarak 50cm. Pada jarak 100cm, model mendapatkan *confidence score* paling rendah daripada skenario 1 dan 2.

V. KESIMPULAN

A. Kesimpulan

YOLOv5 *nano* original dan model yang ditambah teknik regularisasi *dropout* dapat diimplementasikan kedalam *quadcopter* untuk mendeteksi rambu berhenti, belok kiri, dan belok kanan beserta *confidence score* dalam mendeteksi objek. Dari hasil pengujian, dapat diketahui bahwa sebagian besar model memiliki jarak optimal deteksi dengan jarak antara 50 cm hingga 75 cm di pagi hari dengan keadaan ruangan cukup pencahayaan, namun sulit untuk mendeteksi rambu pada jarak 100cm. Pada keadaan siang hari, sebagian besar model juga memiliki jarak optimum deteksi sebesar 50cm hingga 75 cm, dan dapat mendeteksi rambu pada jarak 100cm dengan keadaan ruangan terang. Pada sore menjelang malam hari, model memiliki jarak optimum deteksi objek yang semakin pendek, yaitu diantara 25 cm hingga 50 cm dengan keadaan ruangan yang kurang pencahayaan. Pengujian juga dipengaruhi oleh tingkat pencahayaan pada ruang pengujian. Semakin terang tingkat pencahayaan pada ruangan, deteksi objek rambu akan semakin baik. Jika ruangan semakin gelap, maka model akan sulit untuk mendeteksi objek rambu sehingga tingkat *confidence score*-nya rendah. Objek rambu yang paling mudah dideteksi adalah rambu *stop*. Model paling baik selama masa pengujian ini adalah model YOLOv5 *nano* dengan ditambahkan *dropout* 20%, karena model tersebut memiliki rata-rata paling tinggi hampir di setiap deteksi rambu di tiga waktu pengujian yang berbeda.

B. Saran

Berdasarkan kesimpulan, berikut terdapat beberapa saran yang dapat dilakukan untuk penelitian selanjutnya:

1. Menambahkan jenis kelas rambu pada dataset sehingga terdapat lebih banyak jenis perintah yang dapat dideteksi oleh *quadcopter*.
2. Menggunakan jenis *hardware* atau *quadcopter* yang lebih baik untuk mengurangi waktu latensi antara *quadcopter* dengan komputer.
3. Menggunakan model YOLOv7 untuk mempercepat waktu komputasi dan dapat mengurangi waktu latensi.
4. Melakukan eksplorasi fitur dan modifikasi pada arsitektur YOLOv5 daripada menggunakan teknik

regularisasi *dropout* untuk mempermudah analisa.

5. Memakai dataset yang diambil dari sudut pandang UAV atau *quadcopter* sehingga diharapkan diperoleh hasil yang lebih baik.

REFERENSI

- [1] J. Redmon, S. Divvala, R. Girshick and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 779-788, doi: 10.1109/CVPR.2016.91.
- [2] M. Liu, X. Wang, A. Zhou, X. Fu, Y. Ma, and C. Piao, "Uav-yolo: Small object detection on unmanned aerial vehicle perspective," *Multidiscip. Digit. Publ. Inst. Int. J. Mol. Sci. Switz.*, vol. 20, no. 8, pp. 1–12, 2020, doi: 10.3390/s20082238.
- [3] U. Kamal, T. I. Tonmoy, S. Das and M. K. Hasan, "Automatic Traffic Sign Detection and Recognition Using SegU-Net and a Modified Tversky Loss Function With L1-Constraint," in *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 4, pp. 1467-1479, April 2020, doi: 10.1109/TITS.2019.2911727.
- [4] A. You, M. A. Y. Be, and I. In, "Object and lane detection for autonomous vehicle using YOLO V3 algorithm," *AIP Conf. Proc.*, vol. 140009, no. November, 2021.
- [5] J. Du, "Understanding of Object Detection Based on CNN Family and YOLO," *J. Phys. Conf. Ser.*, vol. 1004, no. 1, 2018, doi: 10.1088/1742-6596/1004/1/012029.
- [6] M. Kasper-Eulaers, N. Hahn, P. E. Kummervold, S. Berger, T. Sebulonsen, and Ø. Myrland, "Short communication: Detecting heavy goods vehicles in rest areas in winter conditions using YOLOv5," *Multidiscip. Digit. Publ. Inst. Int. J.*

- Mol. Sci. Switz.*, vol. 14, no. 4, 2021, doi: 10.3390/a14040114.
- [7] Z. Zhao, P. Zheng, S. Xu and X. Wu, "Object Detection With Deep Learning: A Review," in *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 11, pp. 3212-3232, Nov. 2019, doi: 10.1109/TNNLS.2018.2876865
- [8] S. P. Rajendran, L. Shine, R. Pradeep and S. Vijayaraghavan, "Real-Time Traffic Sign Recognition using YOLOv3 based Detector," 2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT), 2019, pp. 1-7, doi: 10.1109/ICCCNT45670.2019.8944890.
- [9] S. Albawi, T. A. Mohammed and S. Al-Zawi, "Understanding of a convolutional neural network," 2017 International Conference on Engineering and Technology (ICET), 2017, pp. 1-6, doi: 10.1109/ICEngTechnol.2017.8308186.
- [10] V. H. Phung and E. J. Rhee, "A Deep Learning Approach for Classification of Cloud Image Patches on Small Datasets," *Journal of Information and Communication Convergence Engineering*, vol. 16, no. 3, pp. 173–178, Sep. 2018.
- [11] S. H. Wang, Y. D. Lv, Y. Sui, S. Liu, S. J. Wang, and Y. D. Zhang, "Alcoholism Detection by Data Augmentation and Convolutional Neural Network with Stochastic Pooling," *J. Med. Syst.*, vol. 42, no. 1, 2018, doi:10.1007/s10916-017-0845-x.
- [12] Q. Lina, Medium, 3 January 2019. [Online]. Available: <https://medium.com/@16611110/apa-itu-convolutional-neural-network-836f70b193a4>. [Accessed 30 November 2021].
- [13] D. Muller, "Morioh," 27 November 2021. [Online]. Available: <https://morioh.com/p/54540ca11fde>. [Accessed 30 November 2021].
- [14] W. Chen, Z. Liqiang, Y. Tianpeng, J. Tao, J. Yijing, and L. Zhihao, "Research on the state detection of the secondary panel of the switchgear based on the YOLOv5 network model," *J. Phys. Conf. Ser.*, vol. 1994, no. 1, 2021, doi: 10.1088/1742-6596/1994/1/012030.
- [15] Kuhlman, Dave. "A Python Book: Beginning Python, Advanced Python, and Python Exercises". Section 1.1. Archived from the original (PDF) on 23 June 2012.
- [16] Python Software Foundation, "About Python." <https://www.python.org/about/>.
- [17] Muhardian, Ahmad. "Belajar Pemrograman Python: Pengenalan Dasar Python dan Persiapan Awal," 2 December 2021. [Online]. Available: <https://www.petanikode.com/python-linux/>. [Accessed 30 November 2021].
- [18] Z. Zou, Z. Shi, Y. Guo, and J. Ye, "Object Detection in 20 Years: A Survey," pp. 1–39, 2019, [Online]. Available: <http://arxiv.org/abs/1905.05055>.
- [19] A. Nemati and M. Kumar, "Modeling and control of a single axis tilting quadcopter," *Proc. Am. Control Conf.*, pp. 3077–3082, 2014, doi: 10.1109/ACC.2014.6859328.
- [20] N. Mittal, A. Vaidya, and A. Shreya Kapoor, "Object Detection and Classification Using Yolo," *Int. J. Sci. Res. Eng. Trends*, vol. 5, no. 2, 2019.
- [21] U. Nepal and H. Eslamiat, "Comparing YOLOv3, YOLOv4 and YOLOv5 for Autonomous Landing Spot Detection in Faulty UAVs," *Sensors*, vol. 22, no. 2, 2022, doi: 10.3390/s22020464.
- [22] G. Jocher, A. Stoken, A. Chaurasia, J. Borovec, NanoCode012, TaoXie, Y. Kwon, K. Michael, L. Changyu, J. Fang, A. V, Laughing, tkianai,

yxNONG, P. Skalski, A. Hogan, J. Nadar, imyhxy, L. Mamma, AlexWang1900, C. Fati, D. Montes, J. Hajek, L. Diaconu, M. T. Minh, Marc, albinxavi, fatih, oleg, and wanghaoyang0106, "ultralytics/YOLOv5: v6.0 - YOLOv5n 'Nano' models, Roboflow integration, TensorFlow export, OpenCV DNN support," Oct. 2021. [Online]. Available: <https://doi.org/10.5281/zenodo.5563715>

