

Deteksi Pelanggaran Pada Bahu Jalan Tol Dengan Intelligent Transportation System Menggunakan Algoritma YOLOv5

Violation Detection On The Roadside Of The Toll Roads With Intelligent Transportation System Using YOLOv5 Algorithm

1st Alfian Imran
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia

alfianimran@student.telkomuniversity.
ac.id

2nd Casi Setianingsih
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia

setiacasie@telkomuniversity.ac.id

3rd Randy Erfa Saputra
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia

resaputra@telkomuniversity.ac.id

Abstrak—Pemerintah melalui Peraturan Pemerintah No. 15 Tahun 2005 Pasal 41 ayat (2) memberlakukan penggunaan bahu jalan tol serta larangan pada bahu jalan tol. Bahu jalan tol dapat digunakan untuk arus lalu lintas pada keadaan darurat; bagi kendaraan yang berhenti darurat; tidak digunakan untuk menarik/menderek/mendorong kendaraan; tidak digunakan untuk keperluan menaikkan atau menurunkan penumpang dan/ atau barang dan/ atau hewan; dan tidak digunakan untuk mendahului kendaraan. Intelligent transportation sistem yang dibuat diharapkan dapat mengurangi pelanggaran yang terjadi pada bahu jalan tol. Hasil akhir dari pengembangan model *intelligent transportation system* dengan YOLOv5 ini sudah sangat baik. Dengan konfigurasi dataset 240 *data test*, 5 *data validation*, dan 5 *data test*. Konfigurasi *learning-rate* = 0,01; *batch-size* = 64; dan *epochs* = 100. Didapatkan hasil akhir yang sangat memuaskan dengan *mAP* = 97,9 %; *precision* = 93,7 %; dan *recall* = 97,1 %. Penulis berharap sistem yang telah dibuat ini dapat terintegrasi dengan sistem *e-tilang*.

Kata kunci—*bahu jalan tol, YOLOv5, intelligent transportation system*

Abstract—Indonesia's government through Peraturan pemerintah no. 15 of 2005 Article 41 paragraph (2) stipulates the use of roadside of the toll roads and a ban on the roadside of the toll roads. Roadside of the toll roads can be used for traffic flow in an emergency; for vehicles that have an emergency stop; not used for pulling/towing/pushing vehicles; not used for the purpose of raising or lowering passengers and/or things and/or animals; and not used to slip the vehicle. The intelligent transportation system is expected to be able to reduce the number of violations on the roadside of the toll roads. The result of the development of the

intelligent transportation system model with YOLOv5 is good. With a dataset configuration of 240 test data, 5 validation data, and 5 test data. Learning-rate configuration = 0.01; batch-size = 64; and epochs = 100. Very satisfactory results were obtained with mAP = 97.9 %; precision = 93.7 %; and recall = 97.1%. The author hopes that the system that has been created can be integrated with the e-ticket system.

Keywords—*Roadside of the toll roads, YOLOv5, intelligent transportation system*

I. PENDAHULUAN

Berdasarkan Peraturan Pemerintah No. 15 Tahun 2005 Pasal 1 ayat (2), jalan tol adalah jalan umum yang merupakan bagian dari sistem jaringan jalan dan sebagai jalan nasional yang penggunaannya diwajibkan membayar tol [1]. Jalan tol juga dapat disebut sebagai jalan bebas hambatan. Namun, dari banyaknya kelebihan dari penggunaan jalan tol sebagai jalur transportasi, masih banyak angka kecelakaan yang terjadi pada ruas jalan tol. Berdasarkan data Badan pengatur Jalan Tol (BPT), jumlah kasus kecelakaan yang terjadi pada tahun 2020 terdapat 3.907 kasus dan meningkat pada tahun 2021 menjadi 3.988 kasus [2].

Tingginya angka kecelakaan di ruas jalan tol banyak disebabkan oleh kelalaian manusia. Seorang pakar Peneliti Pusat Studi Transportasi dan Logistik (PUSTRAL) UGM menyebutkan beberapa faktor penyebab kecelakaan di jalan tol di antaranya adalah faktor kelalaian pengemudi, kendaraan, lingkungan dan jalan, serta cuaca [3]. Beberapa hal yang dapat menjadi penyebab kecelakaan dari faktor pengemudi di antaranya adalah kondisi pengemudi yang mengantuk, sengaja melanggar peraturan lalu lintas, dan tidak fokus. Salah satu peraturan lalu lintas yang sering dilanggar adalah mendahului melalui bahu jalan tol.

Dalam Peraturan Pemerintah No. 15 Tahun 2005 Pasal 41 ayat (2) telah dijelaskan bahwa bahu jalan tol dapat digunakan untuk arus lalu lintas pada keadaan darurat; bagi kendaraan yang berhenti darurat; tidak digunakan untuk menarik/menderek/mendorong kendaraan; tidak digunakan untuk keperluan menaikkan atau menurunkan penumpang dan/ atau barang dan/ atau hewan; dan tidak digunakan untuk mendahului kendaraan. Berdasarkan peraturan tersebut, telah jelas bahwa menggunakan bahu jalan tol untuk memarkir maupun mendahului merupakan suatu pelanggaran yang dapat diberi sanksi.

Namun, untuk mendeteksi pelanggaran-pelanggaran tersebut diperlukan tenaga manusia yang banyak untuk berjaga di setiap sudut ruas jalan tol. Jika hal ini dilakukan akan memakan banyak biaya dan membutuhkan banyak sumber daya manusia yang kompeten untuk berjaga terus menerus. Oleh karena itu, diperlukan sebuah sistem untuk dapat mendeteksi pelanggaran pada bahu jalan tol secara otomatis.

Intelligent Transportation System adalah sebuah sistem yang dibuat untuk mendeteksi pelanggaran-pelanggaran pada jalan tol khususnya pada bahu jalan tol. Sistem ini bertujuan untuk mempermudah petugas jalan tol untuk mendeteksi pelanggaran lalu lintas. Dengan adanya sistem ini diharapkan dapat menurunkan angka kecelakaan lalu lintas pada ruas jalan tol.

Untuk membuat sistem tersebut dapat mendeteksi pelanggaran-pelanggaran pada bahu jalan tol, maka diperlukan algoritma *Deep Learning* yang dapat bekerja dengan baik untuk mendeteksi objek pelanggar pada CCTV tol. Sebelumnya telah dilakukan beberapa penelitian untuk mendeteksi pelanggaran-pelanggaran pada bahu jalan tol dengan menggunakan algoritma YOLOv4, Faster R-CNN, dan SSD (*Single Shot MultiBox Detector*).

Pada penelitian yang telah dilakukan sebelumnya oleh Maulana Headiy Yusfian, Casi Setianingsih, dan Ratna Astuti (2022) yang menggunakan algoritma YOLOv4 didapatkan hasil *training* dengan *learning rate* sebesar 0.06 dengan *max batches* 4000 serta nilai mAP 97.96% dan akurasi yang dihasilkan sebesar 80% [4]. Sistem yang dihasilkan pada penelitian tersebut dapat mendeteksi dan mengirimkan pemberitahuan kepada petugas jalan tol yang sedang bertugas dengan baik.

Pada penelitian ini penulis akan membuat *Intelligent Transportation system* dengan menggunakan algoritma YOLO (*You Only Look Once*) versi 5. Diharapkan agar sistem ini dapat bekerja dengan lebih baik dibandingkan penelitian yang telah dilakukan sebelumnya.

II. DASAR TEORI

A. Bahu Jalan Tol

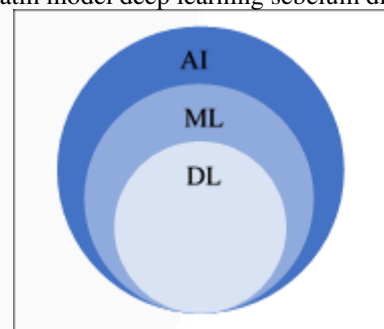
Jalan tol merupakan jalan umum yang hanya dapat dilalui oleh kendaraan beroda empat atau lebih. Penggunaan jalan tol diatur melalui Peraturan Pemerintah No. 15 Tahun 2005 [1]. Peraturan Pemerintah tersebut mengatur penggunaan jalan tol, spesifikasi jalan tol, larangan-larangan apa saja yang patut dipatuhi oleh para pengguna jalan tol, dan lain-lain. Salah satu larangan yang paling sering dilanggar oleh para pengguna jalan tol adalah menggunakan bahu jalan tol baik untuk memarkir kendaraan maupun mendahului.

Bahu jalan adalah jalur pinggir yang terletak berdampingan dengan jalur lalu lintas utama yang berfungsi untuk memarkir kendaraan untuk sementara waktu dengan kondisi tertentu; jalur untuk kendaraan *emergency* jika diperlukan; sebagai jalur drainase air yang menggenang pada ruas jalan tol; dan lain-lain.

B. Deep Learning

Deep learning merupakan salah satu sub tema dari machine learning (gambar 2.1) yang terinspirasi dari konsep *neural networks* yang mencoba untuk meniru cara kerja otak manusia dalam memproses sebuah informasi. Istilah “deep” pada *deep learning* dapat dikotakan pada proses analisis terhadap data yang dilakukan melalui beberapa layer. Deep learning banyak digunakan untuk aplikasi yang memerlukan otomatisasi, melakukan analisis terhadap data, dan tugas fisik tanpa adanya intervensi langsung dari manusia [5]. Deep learning akan sangat bermanfaat pada bermacam-macam kondisi yang kinerjanya akan setara dengan kemampuan manusia atau bahkan lebih baik dibandingkan manusia pada beberapa kasus.

Dalam membuat model deep learning, diperlukan data yang banyak untuk memastikan *metrics* dari model yang dibuat memiliki persentase yang baik sehingga rekomendasi atau pengambilan keputusan yang dibuat oleh model tersebut memiliki tingkat keberhasilan yang tinggi serta terhindar dari kesalahan. Data tersebut kemudian akan dibagi menjadi *data train*, *data test*, dan *data validation* untuk melatih model deep learning sebelum diaplikasikan



Gambar 2.1
Deep Learning

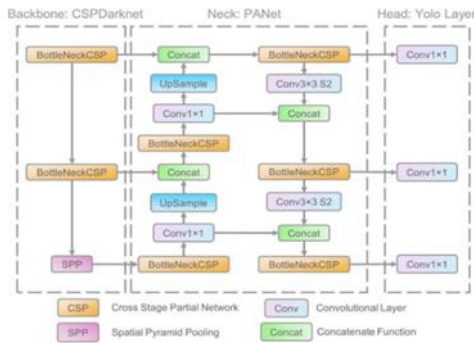
Perkembangan deep learning berbanding lurus dengan semakin tingginya penggunaan teknologi *Big Data*. Pengembangan dan penelitian teknologi deep learning oleh para ahli dipastikan akan terus berlanjut sejalan dengan berkembangnya penggunaan deep learning pada *object detection*, *image recognition*, dan lain-lain dengan semakin menyederhanakan pengimplementasian deep learning pada aplikasi-aplikasi tersebut [6].

C. You Only Look Once Versi 5 (YOLOv5)

Dewasa ini, perkembangan YOLO (*You Only Look Once*) telah berkembang cukup pesat. Setelah YOLO pertama kali *publish* oleh Redmond et al, YOLOv1 kemudian berkembang menjadi YOLOv2, YOLOv3, YOLOv4, dan YOLOv5.

Pengembangan YOLOv5 banyak mengadopsi konsep yang diusung pada YOLOv1-YOLOv3. YOLOv5 berkembang lumayan pesat dibanding YOLOv4 dengan

waktu *runtime* yang semakin meningkat yang mencapai 140 *frames per second* [13]. Ukuran file YOLOv5 yang jauh lebih kecil dibandingkan YOLOv4 dan versi lainnya memungkinkan YOLOv5 untuk diaplikasikan ke perangkat tertanam (*embedded devices*). Dibandingkan dengan YOLO versi sebelumnya YOLOv5 memiliki tingkat akurasi yang paling tinggi dan kemampuan yang lebih baik untuk mendeteksi objek-objek yang lebih kecil.



Gambar 2.2
Arsitektur Network YOLOv5

Arsitektur network YOLOv5 tersusun atas 3 bagian, yaitu *backbone*: CSPDarknet; *neck*: PANet; dan *head*: YOLO layer [16]. Ketika data diinput pertama kali, maka data tersebut akan diproses terlebih dahulu oleh CSPDarknet untuk ekstraksi fitur, kemudian dilanjutkan ke PANet untuk *feature fusion* kemudian data tersebut akan diproses oleh layer YOLO untuk mendeteksi objek pada data tersebut.

III. PERANCANGAN

A. Desain Sistem

Proses deteksi dimulai pada CCTV yang terpasang pada setiap ruas sudut jalan tol. CCTV tersebut akan terhubung ke perangkat komputer yang kemudian akan mendeteksi objek-objek pada CCTV tersebut.

Sebelum mengaplikasikan *Intelligent Transportation System*, perlu dirancang sebuah sistem agar deteksi-deteksi yang dilakukan oleh sistem tidak terjadi error. Oleh karena itu, yang akan dilakukan pertama kali adalah membuat model dengan algoritma YOLOv5 untuk mendeteksi objek-objek yang ditangkap oleh kamera.

Langkah pembuatan model tersebut dimulai dengan mengumpulkan dataset yang dibutuhkan, kemudian memberikan label pada masing-masing gambar tersebut agar sistem dapat mendeteksi ketika terjadi sebuah pelanggaran, label/klasifikasi pada objek yang dibuat adalah “Melanggar” dan “Tidak Melanggar”. Setelah itu, menginstall *dependency* YOLOv5, kemudian konfigurasi model dan arsitektur YOLOv5, dan langkah terakhir yang dilakukan adalah train model tersebut.

Setelah melakukan pelatihan/train pada model yang telah dibuat, hal yang tak kalah penting untuk dilakukan adalah men-*deploy* model yang telah dibuat agar dapat diaplikasikan ke aplikasi *Intelligent Transportation System*.

B. Analisis Kebutuhan Dataset

Dataset yang akan digunakan pada penelitian ini akan sama dengan penelitian yang dilakukan sebelumnya oleh Maulana Heady Yusfian, Casi Setianingsih, dan Ratna

Astuti [4] yaitu di ruas tol JORR-S KM 30+300 dengan tambahan 150 dataset baru. Data tersebut dikumpulkan di website resmi binamarga dengan alamat web binamarga.pu.go.id (Gambar 3.2). Dataset tersebut kemudian dibagi menjadi data train, data validation, dan data test.

C. Menambahkan Label pada Dataset dengan Roboflow

Setelah memiliki dataset yang diperlukan, selanjutnya adalah memberi label dan *bounding box* pada objek yang akan dideteksi dengan label “Melanggar” dan “Tidak Melanggar” seperti yang ditunjukkan pada gambar 3.4. Roboflow adalah salah satu platform pada *object detection* yang dapat digunakan untuk memberi label atau melakukan anotasi pada gambar. Selain itu, pada Roboflow juga banyak tersedia dataset publik yang mudah diakses. Selain itu, Roboflow juga memberikan akses kepada *user* agar dapat meng-*upload* dataset sesuai dengan kebutuhan.

Tahap selanjutnya yang akan dilakukan adalah melakukan *generate* pada roboflow agar dataset yang telah dikumpulkan dan dianotasi dapat dipartisi menjadi data *train*, *validation*, dan *test*. Selain itu, pada tahap *generate* juga perlu melakukan *preprocessing* dan *augmentasi*.

D. Training Custom YOLOv5 model

Sebelum melakukan pelatihan model pada YOLOv5 hal pertama yang akan dilakukan adalah menentukan versi YOLOv5 yang akan digunakan. Versi tersebut antara lain adalah YOLOv5n, YOLOv5s, YOLOv5m, YOLOv5l, YOLOv5x. Perbedaan kelima versi YOLOv5 tersebut dapat dilihat pada bagian lampiran. Versi yang dipilih pada penelitian ini adalah YOLOv5s karena versi ini berukuran lebih kecil dibandingkan versi lainnya serta paling cocok untuk diimplementasikan pada *embedded system* atau IoT (Internet of Things) bila dibandingkan dengan YOLOv5n.

Setelah menentukan versi, selanjutnya adalah melakukan pelatihan model pada dataset tollRoads, dengan menggunakan dataset yang telah digenerate di Roboflow. Hal yang perlu diperhatikan pada tahap ini adalah penentuan *batch-size*, *image size*, dan *epochs*.

E. Analisis Performa Deteksi YOLOv5

Ketika proses training model telah selesai dilakukan, Langkah selanjutnya yang akan dilakukan adalah melakukan analisis performansi pada model YOLOv5 yang telah dibuat. Analisis ini bertujuan agar model yang telah dikembangkan sesuai dengan yang telah diharapkan dengan melihat metrik validasi. Parameter-parameter yang akan menjadi acuan dalam menganalisis performa model adalah sebagai berikut.

1. mAP.5 dihitung dengan nilai rata-rata (*mean*) dari *Average Precision* (AP) dan *IoU thresholds*.
2. mAP.5:.95
3. Precision digunakan untuk menganalisa tingkat keakuratan dari prediksi yang dilakukan.
4. Recall adalah persentase kecenderungan sistem untuk menemukan *True Positive* pada keseluruhan prediksi yang dilakukan.
5. Objectness loss merepresentasikan kemampuan algoritma dalam memastikan kemungkinan adanya objek pada *Region of Interest* (RoI).

6. Box loss merepresentasikan kemampuan algoritma dalam memposisikan *bounding box* pada objek
7. Classification loss merepresentasikan kemampuan algoritma dalam memprediksi "1" atau "0" pada masing-masing kelas.

IV. HASIL DAN ANALISIS

A. Pengujian Partisi Data

Tabel 4.1
Perbandingan Hasil Training Partisi Dataset

	mAP.5	mAP.5:95	Precision	Recall
Kondisi I	91,1 %	44,9 %	92,6 %	87,5 %
Kondisi II	91,1 %	46,1 %	88,6 %	88,2 %
Kondisi III	94,7 %	48,3 %	93,2 %	87,8 %
Kondisi IV	97 %	48,4 %	89,9 %	97,1 %

Setelah melakukan *training* dan menganalisis performa dari masing-masing model yang telah dibuat. Melalui analisis pada *metrics evaluation* dengan mAP.5, mAP.5:95, *precision*, dan *recall*, dapat disimpulkan bahwa partisi data terbaik untuk kasus "Deteksi Pelanggaran pada Bahu Jalan Tol" adalah pada kondisi IV dengan 96% *data train* atau 240 gambar, 2% *data validation* dan *data test* dengan masing-masing 5 gambar.

B. Pengujian Batch Size

Tabel 4.2
Perbandingan Hasil Training Batch-Size

Batch-Size	mAP.5	mAP.5:95	Precision	Recall
16	97 %	48,4 %	89,9 %	97,1 %
32	94,7 %	51,4 %	84,9 %	97,1 %
64	97,9 %	51,9 %	93,7 %	97,1 %
128	87,1 %	46,4 %	90,3 %	97,1 %

Dari 4 *batch-size* yang diuji, dapat disimpulkan bahwa model YOLOv5 terbaik didapatkan pada *batch-size* = 64. Setelah mendapatkan model terbaik dengan partisi dataset dan konfigurasi *batch-size* dengan beberapa kondisi, masih harus dilakukan pengujian dengan *epoch*.

C. Pengujian Learning Rate

Tabel 4.3
Perbandingan Hasil Training Learning Rate

Learning Rate	mAP.5	mAP.5:95	Precision	Recall
0,01	97,9 %	51,4 %	93,7 %	97,1 %
0,001	43,6 %	17,9 %	87 %	30,4 %
0,0001	0,968 %	0,19 %	0,8 %	3 %
0,00001	1,61 %	0,213 %	0,446 %	19,1 %

Dari pengujian Learning Rate di atas, untuk kasus ini dapat disimpulkan bahwa *Learning Rate* default yang disediakan YOLOv5 adalah *learning rate* terbaik sehingga untuk pembuatan model selanjutnya dapat disarankan

untuk tak perlu melakukan pengujian pada *learning rate* lagi. Penulis mengasumsikan bahwa untuk konfigurasi *Learning Rate*, *momentum*, dan *optimizer* tidak perlu diuji lagi karena konfigurasi default hyperparameter tersebut sudah diuji terlebih dahulu. Namun, karena tidak adanya *paper/journal* dari organisasi yang pertama kali memperkenalkan YOLOv5, hal tersebut tidak dapat dipastikan.

D. Pengujian Epochs

Tabel 4.4
Tabel Perbandingan Hasil Training Epoch

Epoch	mAP.5	mAP.5:95	Precision	Recall
100	97,9 %	51,4 %	93,7 %	97,1 %
300	95,9 %	48,6 %	81,6 %	96,5 %
450	93,3 %	51,7 %	93,3 %	87,8 %
600	95,7 %	53,2 %	88,4 %	91,7 %

Dari pengujian *epochs* yang ditunjukkan pada tabel 4.4 di atas, dapat disimpulkan bahwa model terbaik dihasilkan dengan *epochs* = 100. Hal ini menunjukkan dalam melatih model *deep learning* yang perlu diperhatikan adalah bukan seberapa banyak *epochs* yang diinput tapi sesuai dengan yang dibutuhkan algoritma.

V. KESIMPULAN DAN SARAN

A. Simpulan

Berdasarkan pengujian dan analisis yang telah dilakukan, maka dapat ditarik beberapa kesimpulan, yaitu sebagai berikut.

1. Berdasarkan penelitian yang telah dilakukan, konfigurasi terbaik agar proses *training* menghasilkan model terbaik adalah dengan partisi dataset 96:2:2, *batch-size* = 64, dan *epochs* = 100.
2. Hasil *training* dengan model terbaik tersebut menghasilkan model dengan mAP (*mean Average Precision*) = 0,979; *precision* = 0,937; dan *recall* = 0,971.
3. Model belum dapat diaplikasikan ke CCTV yang berbeda dengan yang ada pada dataset.

B. Saran

Berdasarkan analisis dan kesimpulan yang telah dikemukakan, maka saran dari penelitian ini adalah :

1. Jumlah dataset yang dibutuhkan harus cukup agar menghasilkan model yang dapat bekerja dengan baik.
2. Jumlah dataset yang digunakan seharusnya lebih banyak dari yang digunakan pada penelitian ini.
3. Ketelitian sangat dibutuhkan pada tahap *annotate* atau memberi label pada objek-objek tertentu agar model yang dihasilkan memiliki performa yang baik.
4. Untuk dapat diaplikasikan ke lapangan dapat menggunakan perangkat keras tambahan seperti *Jetson Nano* atau dapat langsung mendeteksi objek dengan *streaming live* CCTV Jalan tol Jasa Marga.
5. Menambahkan algoritma *object tracking* agar sistem dapat bekerja pada CCTV yang berbeda dengan dataset.

REFERENSI

- [1] Pemerintah Republik Indonesia. Peraturan Pemerintah No. 15 Tahun 2005 Tentang Jalan Tol. Pemerintah Pusat. Jakarta
- [2] Sarnita Sadya (2022, Sept 19). “Ada 3.988 Kecelakaan di Jalan Tol pada 2021,” [Online]. Available: <https://dataindonesia.id/sektor-riil/detail/ada-3988-kecelakaan-di-jalan-tol-pada-2021> [Diakses 23 Desember 2022, 20:00:00 WIB]
- [3] Gloria (2021, Nov 07). “Pakar UGM Sebut Empat Faltor Penyebab Kecelakaan di Jalan Tol,” [Online]. Available: <https://ugm.ac.id/id/berita/21920-pakar-ugm-sebut-empat-faktor-penyebab-kecelakaan-di-jalan-tol> [Diakses 23 Desember 2022, 20:05:00 WIB]
- [4] Maulana Headiy Yusfian, Casi Setianingsih, Ratna Astuti, “Deteksi Pelanggaran Parkir pada Bahu Jalan Tol dengan Intelligent Transportation System Menggunakan Algoritma YOLO,” *e-proceeding of Eng.*, vol 9, no.3, pp. 1064-1069, Juny 2022
- [5] IBM Cloud Education. (2020, May 1). “*Deep Learning*,” [Online]. Available : <https://www.ibm.com/cloud/learn/deep-learning> [Diakses 3 Oktober 2022, 08:34:30 WIB]
- [6] Laith Alzubaidi, et al., “*Review of Deep Learning: Concepts, CNN, Architectures, Challenges, Applications, Future Directions*,” *Journal of Big Data*, 2021
- [7] 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
- [8] Do Thuan, “*Evolution of Yolo Algorithm and YoloV5: The State-of-the-Art Object Detection Algorithm*,” Bachelor’s Thesis, Oulu University of Applied Sciences, Oulu, Finland, 2021