

Sistem Deteksi Pengendara Sepeda Motor Tanpa Helm Menggunakan Algoritma SSD

Motorcycle Riders Detection System Without Helmet Using Ssd Alghorithms

1st Farhan Fuadi

Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia

farhanfuadi@student.telkomu
niversity.ac.id

2nd Casi Setianingsih

Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia

setiacasie@telkomuniversity.
ac.id

3rd Marisa W. Paryasto

Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia

marisaparyasto@telkomunive
rsity.ac.id

Abstrak—Pelanggaran lalu lintas sudah banyak terjadi untuk saat ini. Salah satu pelanggaran yang terjadi, disebabkan oleh pengendara sepeda motor. Banyak dari pengendara sepeda motor tidak menggunakan helm saat bepergian, sehingga dapat meningkatkan risiko kematian jika terjadi kecelakaan. Salah satu penyebab banyaknya pelanggaran peraturan lalu lintas yaitu tidak adanya pengawasan dari polisi lalu lintas secara real-time. Oleh karena itu, pada Tugas Akhir ini dibuat sebuah sistem deteksi pelanggaran helm pada kendaraan roda dua menggunakan algoritma SSD yang dapat mempermudah pendeteksian pelanggaran tidak menggunakan helm pada saat mengendarai sepeda motor. Cara kerja sistem ini yaitu kamera yang telah dipasang di tempat yang ditentukan, akan mendeteksi motor yang lewat. Jika kamera mendeteksi pelanggaran pada tempat yang telah dipasang kamera, maka nantinya akan mengirimkan pesan notifikasi ke pihak kepolisian. Dari hasil penelitian tugas akhir sistem deteksi pengendara tanpa helm pada kendaraan roda dua menggunakan algoritma SSD memperoleh nilai mAP@50IOU 79,2% dan AR@100 61,4% dengan variabel konfigurasi yang digunakan adalah rasio data train 90% dan data test 10%, learning rate 0.004, epochs 1, dan batch size 24.

Kata kunci — deteksi sepeda motor, deteksi helm, SSD

Abstract—Traffic violations have happened a lot for the time being. One of the violations that occurred, caused by motorcyclists. Many motorcyclists do not use helmets when traveling, so it can increase the risk of death in the event of

an accident. One of the reasons many motorcyclists violate traffic rules is the absence of supervision from the traffic police in real-time. Therefore, in this Final Project proposal, it is proposed to create a helmet violation detection system on two-wheeled vehicles using an SSD algorithm that can make it easier to detect violations of not using a helmet when riding a motorcycle. The way this system works is that the camera that has been installed in the designated place, will detect a passing motor. If the camera detects a violation in the place where the camera has been installed, it will later send a notification message to the police. From the results of the final project study of the helmetless rider detection system on two-wheeled vehicles this time using the SSD algorithm obtained a value of mAP@50IOU 79.2% and AR@100 61.4% with the configuration variables used were a data ratio of 90% train and 10% test, learning rate 0.004, epochs 1, and batch size.

Keywords— motorcycle detection, helmet detection, SSD

I. PENDAHULUAN

Lalu lintas menurut Undang-Undang No. 22 tahun 2009 didefinisikan sebagai gerak kendaraan dan orang di ruang lalu lintas jalan, sedangkan yang dimaksud dengan ruang lalu lintas jalan adalah prasarana yang diperuntukkan bagi gerak pindah kendaraan, orang, dan atau barang yang berupa jalan dan fasilitas pendukung [1].

Dalam berlalu lintas, umumnya ada dua jenis kendaraan yang sering melewatinya, yaitu kendaraan roda empat dan juga kendaraan roda dua. Bagi kendaraan

roda dua, tidak menggunakan helm adalah salah satu bentuk pelanggaran yang dapat dikenai hukum pidana, tidak menggunakan helm juga dapat menyebabkan kecelakaan dalam berlalu lintas. Kecelakaan lalu lintas menurut pasal 1 angka 22 UU Nomor 22 tahun 2009 adalah suatu peristiwa di jalan yang tidak diduga dan tidak disengaja yang melibatkan kendaraan dengan atau tanpa pengguna jalan lain yang mengakibatkan korban manusia dan/atau kerugian harta benda[1]. Keterlibatan polisi dalam menindak lanjut kasus pelanggaran kendaraan roda dua adalah hal yang penting, namun tidak jarang ada beberapa pengguna kendaraan roda dua yang lolos dari pengawasan polisi, hal ini dapat terjadi dikarenakan polisi yang bertugas untuk mengawasi juga memiliki keterbatasan dalam memantau gerak gerik dari pengguna jalan.

Oleh karena itu diperlukan sebuah sistem yang dapat mempermudah aktivitas pihak kepolisian dalam menindak lanjut pelanggaran tidak menggunakan helm. Pada penelitian ini membuat sebuah sistem menggunakan algoritma SSD yang bertujuan untuk mempermudah pihak kepolisian dalam menindak lanjut para pelanggar kendaraan roda dua yang tidak menggunakan helm, jika terbukti melanggar maka sistem akan memberikan sebuah notifikasi kepada pihak berwajib.

II. KAJIAN TEORI / PERANCANGAN SISTEM

A. UU Lalu Lintas dan Angkutan Jalan

UU Lalu lintas dan Angkutan Jalan diterbitkan oleh Dewan Perwakilan Rakyat pada tahun 2009 yang memiliki tujuan yang tercantum pada pasal 3 UU. No. 22/2009 Tentang Lalu lintas dan Angkutan Umum sebagai berikut:

1. Terwujudnya pelayanan Lalu Lintas dan Angkutan Jalan yang aman, selamat, tertib, lancar, dan terpadu dengan modal angkutan lain untuk mendorong perekonomian nasional, memajukan kesejahteraan umum, memperkukuh persatuan dan kesatuan bangsa, serta mampu menjunjung tinggi martabat bangsa.
2. Terwujudnya etika berlalu lintas dan budaya bangsa; dan
3. Terwujudnya penegakan hukum dan kepastian hukum bagi masyarakat[1].

B. UU Penggunaan Helm

UU Penggunaan helm standar nasional Indonesia bagi pengendara kendaraan roda dua diatur dalam Pasal 57

ayat (1) dan ayat (2) UU No.22 Tahun 2009 tentang lalu lintas dan angkutan jalan yang berbunyi :

1. Setiap kendaraan bermotor yang dioperasikan di jalan wajib dilengkapi dengan perlengkapan kendaraan bermotor.
2. Perlengkapan sebagaimana dimaksud pada ayat (1) bagi sepeda motor berupa helm standar nasional Indonesia[1].

C. Machine Learning

Machine Learning merupakan bagian dari *artificial intelligence* (AI) yang bertujuan untuk mengoptimalkan kriteria kinerja dengan cara menganalisis sampel data yang sudah ada yang telah disimpan untuk menghasilkan sebuah prediksi. Model dapat bersifat prediktif untuk membuat prediksi di masa depan, atau deskriptif untuk memperoleh pengetahuan dari data, atau keduanya[2].

Machine learning juga dapat diartikan sebagai teknik untuk melakukan inferensi terhadap data dengan pendekatan matematis. Inti machine learning adalah untuk membuat model (matematis) yang merefleksikan pola-pola data. Pada abad ke-21 ini, *machine learning* banyak memanfaatkan statistika dan aljabar linier[3].

D. Image Processing

Image processing adalah representasi matriks dari citra dua dimensi menggunakan sejumlah elemen sel titik yang terbatas, biasanya disebut piksel (elemen gambar, atau pel). Setiap piksel diwakili oleh nilai numerik: untuk gambar *grayscale* (abu-abu), satu nilai yang mewakili intensitas piksel (biasanya dalam rentang [0,255]) sudah cukup.

Dalam operasi *image processing*, sebagian besar operasi dilakukan dalam gambar *grayscale*. Untuk aplikasi pengolahan *color image*, *color image* dapat didekomposisi menjadi komponen merah (R), hijau (G), dan biru (B) dan masing-masing komponen diproses secara independen sebagai gambar *grayscale*. Dalam *image processing* memiliki tiga tingkatan :

1. Low-Level Image Processing

Operasi sederhana pada gambar (misalnya peningkatan kontras, pengurangan *noise*, dll.) dimana *input* dan *output* nya berupa gambar.

2. Mid-Level Image Processing

Dalam tingkatan ini, melibatkan operasi ekstraksi atribut(misalnya tepi, kontur, wilayah, dll.).

3. High-Level Image Processing

Dalam tingkatan ini melibatkan operasi pemrosesan gambar kompleks yang terkait dengan analisis dan interpretasi konten untuk beberapa pengambilan keputusan[4].

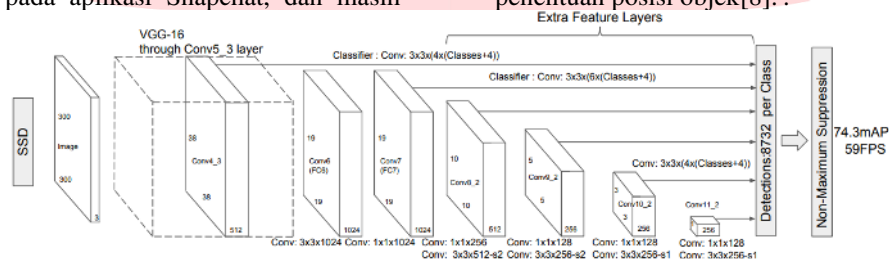
E. Deep Learning

Deep learning adalah sub bidang dalam *machine learning* yang berhubungan dengan algoritma yang mirip dengan sebuah versi *over-simplified* dari kecerdasan manusia . banyak contoh umum yang bisa ditemukan di dalam ekosistem *smartphone* (IOS dan Android) seperti : *face detection* pada kamera *smartphone*, koreksi otomatis pada *keyboard*, *AI-enhanced beautification apps*, asisten pintar seperti Siri/Alexa/Google Assistant, *face-ID (Face unlock* pada iPhone), fitur saran video pada Youtube, rekomendasi pertemanan di Facebook, filter kucing pada aplikasi Snapchat, dan masih

banyak lagi . pada dasarnya *deep learning* sudah banyak digunakan dan ditemukan pada *digital life's* saat ini[5]. Untuk sekarang ini, *Deep learning* juga banyak digunakan untuk penelitian *Machine Learning*[6].

F. Single Shot Detector (SSD)

Single Shot Detector (SSD) adalah algoritma pendeteksi objek *deep learning*. Algoritma SSD ini dapat meningkatkan kecepatan deteksi dan akurasi deteksi[7]. Sebagai salah satu algoritma *object detection* berbasis *deep learning*, SSD memiliki performa tinggi baik dalam akurasi pendeteksian maupun kecepatan pendeteksian. Algoritma SSD diusulkan oleh Liu W dkk. pada tahun 2016 untuk mengatasi masalah akurasi deteksi yang kurang memadai dari algoritma YOLO dalam penentuan posisi objek[8].



GAMBAR 1
ARSITEKTUR ALEXNET MODIFIKASI

Pada gambar 2.1, merupakan struktur arsitektur dari algoritma SSD yang dibangun menggunakan VGG 16. Tetapi tidak menggunakan *fully connected layer*. Alasan VGG-15 digunakan pada arsitektur SSD adalah karena memiliki performa yang bagus dan menghasilkan kualitas *image classification* yang bagus.

1. Multi-Scale Feature Maps

Multi-Scale Feature Maps adalah fitur yang digunakan untuk mendeteksi fitur objek dari satu layer. Sistem memproses fitur ini dengan cara menambahkan layer fitur *convolutional* di akhir *base network*. Layer ini kemudian berkurang secara bertahap, memungkinkan prediksi dengan skala yang berbeda-beda[9].

2. Convolutional Predictor

Convolutional Predictor adalah fitur yang digunakan untuk mendapatkan *predicted detection* menggunakan seperangkat filter konvolusional. Setiap fitur yang ditambahkan dapat menghasilkan *predicted detection* yang akurat menggunakan filter konvolusi. Cara kerjanya

adalah dengan mencari nilai parameter untuk *potential detection*. Lalu, *output* dari *potential detection* adalah skor kategori terhadap koordinat *default*. Di setiap kernel yang diterapkan, menghasilkan nilai *output* dari *bounding box* yang diukur dengan cara relatif terhadap kotak pembatas default[9].

3. Default Box dan Aspect Rasio

Default Box dan *Aspect Ratio* adalah fitur yang mengaitkan serangkaian *default bounding box* dengan *feature map* tertentu yang diperoleh di *network* atas. Cara kerjanya adalah representasi *default bounding box* dilakukan dengan menggunakan fitur *convolutional* untuk mendapatkan lokasi *bounding box* yang tetap. Kemudian, setiap sel fitur memperkirakan *offset relative* terhadap bentuk *default bounding box* setiap sel dan menghitung skor untuk setiap kotak. Selanjutnya, hasil tersebut diterapkan ke beberapa lokasi fitur dengan resolusi yang berbeda-beda[9].

G. Object Detection

Object detection merupakan salah satu topik penelitian penting dalam bidang *computer vision*. Tugas utama dari *object detection* adalah untuk menemukan objek pada suatu gambar. Dalam beberapa tahun terakhir, *object detection* telah banyak digunakan dalam *video monitoring*, deteksi kesalahan, perawatan medis dan bidang lainnya. Tujuan dari penggunaan *object detection* adalah untuk mendeteksi satu atau beberapa class yang diketahui, seperti orang, mobil, atau wajah dalam sebuah gambar [10].

H. Variabel Konfigurasi SSD Inception V2

Dalam variabel konfigurasi SSD terdapat beberapa *hyperparameter* yang digunakan yaitu *learning rate*, *epochs*, dan *batch size*.

1. Learning Rate

Learning rate adalah salah satu *hyperparameter* yang mengontrol seberapa banyak model harus diubah sebagai respons terhadap estimasi *error* setiap kali bobot model diperbarui. *Learning rate* digunakan untuk mengetahui efek kecepatan pembelajaran pada performa model, perilaku model serta mengetahui *loss* diminimalkan atau tidak [11].

2. Epochs

Epochs adalah ketika seluruh dataset sudah melalui proses *training* pada *neural network* sampai dikembalikan ke awal untuk satu kali putaran. Dalam melakukan *training* diperlukan lebih dari satu *epochs* untuk memperoleh nilai bobot yang akurat.

3. Batch Size

Batch size adalah istilah yang digunakan dalam *machine learning* yang mengacu pada jumlah *training* yang digunakan dalam satu iterasi dan merupakan salah satu *hyperparameter* terpenting dalam sistem *deep learning*. Ukuran *batch* yang terlalu besar dapat mempercepat proses komputasi namun akan memberikan hasil yang kurang optimal. Semakin besar ukuran *batch size* semakin tidak teliti hasilnya [12].

J. Confusion Matrix

Confusion matrix merupakan salah satu metode yang dapat digunakan untuk mengukur kinerja suatu metode klasifikasi. Pada dasarnya *confusion matrix* mengandung informasi yang membandingkan hasil klasifikasi yang dilakukan oleh sistem dengan hasil klasifikasi yang seharusnya.

		Actual Values	
		1 (Positive)	0 (Negative)
Predicted Values	1 (Positive)	TP (True Positive)	FP (False Positive) <i>Type I Error</i>
	0 (Negative)	FN (False Negative) <i>Type II Error</i>	TN (True Negative)

Confusion Matrix

GAMBAR 2
TABEL CONFUSION MATRIX 2X2 [13]

Pada pengukuran kinerja menggunakan *confusion matrix*, terdapat 4 (empat) istilah sebagai representasi hasil proses klasifikasi. Keempat istilah tersebut adalah *True Positive* (TP), *True Negative* (TN), *False Positive* (FP) dan *False Negative* (FN). Nilai *True Negative* (TN) merupakan jumlah data negatif yang terdeteksi dengan benar, sedangkan *False Positive* (FP) merupakan data negatif namun terdeteksi sebagai data positif [14].

1. Recall

Recall digunakan dalam mengevaluasi kemampuan sistem untuk menemukan semua item yang relevan dari koleksi dokumen dan didefinisikan sebagai

persentase dokumen yang relevan terhadap *query* [15].

2. Precision

Precision digunakan dalam mengevaluasi kemampuan sistem untuk menemukan peringkat yang paling relevan, dan didefinisikan sebagai persentase dokumen yang di retrieve dan benar-benar relevan terhadap *query* [15].

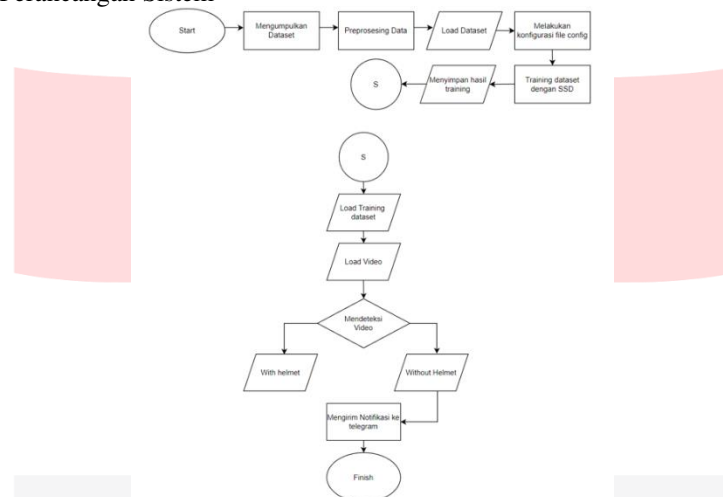
3. Accuracy

Accuracy merupakan perbandingan kasus yang diidentifikasi benar dengan jumlah seluruh kasus dan *error rate* merupakan kasus yang diidentifikasi salah dengan jumlah seluruh kasus [15].

K. Mean Average Precision

Mean Average Precision (mAP) adalah metrik untuk mengukur performa model yang nilainya merupakan rata-rata dari *Average Precision* (AP). *Average Precision* sendiri adalah menemukan area yang beririsan pada kurva *Precision* dan *recall*. Untuk mendapatkan nilai mAP adalah menghitung semua nilai rata-rata pada *Average Precision*[16].

L. Desain dan Perancangan Sistem



GAMBAR 3
FLOWCHART SYSTEM

Dari gambar 3. di atas menunjukkan rancangan sistem yang akan digunakan mulai dari pengumpulan dataset, *preprocessing* dataset, *load* dataset, konfigurasi *file config*, melakukan *train* dataset, dan menyimpan hasil dari *training* yang dilakukan. Setelah melakukan *load train* dataset dan *load video*, maka sistem akan mendeteksi 2 class yang telah ditentukan sebelumnya, yaitu *with helmet* dan *without helmet*. Jika terdeteksi kelas melanggar maka sistem akan mengirim notifikasi ke telegram.

M. Desain Rancangan Sistem

Desain rancangan sistem ini akan menjelaskan bagaimana cara penulis memberikan anotasi label pada gambar yang digunakan sebagai dataset nantinya, Lalu menjelaskan cara melakukan proses *training*, dan akan menampilkan hasil *output* dari hasil *training* yang dilakukan.

1. Mengumpulkan Dataset

Pengumpulan data akan mengambil *file* berupa gambar dan *video* menggunakan kamera iPhone. Pengambilan gambar dan *video* dilakukan di lingkungan kampus Telkom University. Jumlah data yang dikumpulkan berjumlah 1056 gambar dengan format .jpg yang terbagi menjadi 528 gambar *with helmet* dan 528 gambar *without helmet*.

Desain sistem merupakan gambaran sistematis pada deteksi pelanggaran helm yang akan di implementasikan pada pembuatan tugas akhir. Input video yang telah di load nantinya akan dideteksi oleh model apakah terdapat pelanggaran tidak menggunakan helm atau *without helmet*. Jika terjadi pelanggaran maka sistem akan memberikan notifikasi melalui bot telegram.

2. Pembuatan Model

Pembuatan model ini akan membahas cara pemberian anotasi label menggunakan *software* Labellmg untuk nantinya data yang telah diberi label akan siap digunakan untuk melakukan proses *training*. Pada bagian labeling *image* ini semua data yang telah dikumpulkan sebelumnya dalam format .jpg akan diberi anotasi agar sistem dapat dengan mudah membedakan mana data yang diberi label *with helmet* dan *without helmet*.

Setelah melakukan pemberian anotasi label pada gambar, tahap selanjutnya yang harus dilakukan adalah melakukan konversi *file* dengan tipe .XML yang telah dibuat agar menjadi file dengan tipe .CSV. lalu dilanjutkan dengan proses konversi file .CSV menjadi file Tfrecored yang berisikan *test.record* dan *train.record* serta dibuat file *labelmap* yang bertujuan untuk memetakan anotasi label, ide, kelas, dan objek yang sudah diberi label sebelumnya.

3. Proses Training

Sebelum melakukan proses *training*, diperlukan penyesuaian *hyperparameter* yang berada pada *file* model yang akan digunakan. setelah melakukan

proses konfigurasi variabel seperti *steps*, *learning rate*, *epochs*, dan *batch size*. Maka hal yang dilakukan selanjutnya yaitu melakukan proses *training*. dalam melakukan setiap proses *training*, memakan waktu kurang lebih selama 7 jam. Setiap beberapa *steps*, hasil dari *training* data yang disebut *checkpoint* akan disimpan pada *folder training* yang telah dibuat.

Setelah proses *training* selesai, langkah selanjutnya yang harus dilakukan adalah mengubah *file checkpoint* menjadi *file inference graph* yang telah di *training* sebelumnya dengan format *.pb*.

III. HASIL DAN PEMBAHASAN

A. Pengujian Partisi Data

Pengujian partisi data kali ini akan menggunakan metode *holdout*. Dataset yang telah dikumpulkan akan dibagi menjadi dua bagian yang akan disimpan di dalam folder *train* dan *test*. Pembagian partisi data yang akan diuji akan dibagi menjadi lima skenario *training*.

TABEL 1
PENGUJIAN PARTISI DATA

Partisi Data	mAP@.50IoU	AR@100
50 : 50	27%	45.8%
60 : 40	38%	46%
70 : 30	36.2%	46.4%
80 : 20	35.5%	61.5%
90 : 10	73.7%	70.1%

Dapat dilihat pada tabel 1, diperoleh nilai partisi data terbaik setelah melakukan lima pengujian untuk mencari nilai *Mean Average Precision* (mAP) dan *Average Recall* (AR) tertinggi. Diperoleh nilai tertinggi pada partisi data *train* 90% dan *test* 10% dengan nilai *Mean Average Precision* (mAP) 73,7% dan *Average Recall* (AR) 70,1%. Dari hasil ini dapat disimpulkan bahwa jumlah data yang lebih besar di folder *train* dapat menghasilkan nilai mAP dan AR yang tinggi dibandingkan dengan partisi data lainnya.

B. Pengujian Training Variabel Konfigurasi

Pengujian variabel konfigurasi bertujuan untuk menemukan nilai terbaik dari *learning rate*, *epochs*, dan *batch size*. Dengan menggunakan partisi data terbaik yang telah diperoleh dari pengujian sebelumnya, yaitu *train* 90% dan *test* 10%.

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

Hyperparameter	Nilai	mAP@.50IoU	AR@100
<i>Learning Rate</i>	0.001	59.5%	61.3%
	0.004	73.7%	63.2%
	0.008	67.5%	63.1%
<i>Epochs</i>	1	79.2%	61.4%
	150	65.3%	65.6%
	300	77.2%	67.3%
<i>Batch Size</i>	24	79.2%	61.4%
	26	60.2%	63.1%
	30	0%	0%

Dapat dilihat dari tabel 2, diperoleh nilai *learning rate* terbaik yaitu 0.004, dengan nilai mAP@.50IoU 73.7% dan AR@100 63.2%. Nilai *epochs* terbaik yaitu 1, dengan nilai mAP@.50IoU 79.2% dan AR@100 61.4%. Nilai *batch size* terbaik dengan nilai 24, dengan nilai mAP@.50IoU 79.2% dan AR@100 61.4%.

C. Pengujian Sistem Berdasarkan Beberapa Parameter

Pada tahap ini akan dilakukan pengujian sistem berdasarkan beberapa parameter yaitu jarak yang dapat terbaca oleh sistem, sudut kamera, kecepatan, *lux* dan jumlah orang.

TABEL 3
PENGUJIAN JARAK SUDUT KAMERA, KECEPATAN, LUX, DAN JUMLAH ORANG.

Parameter	Nilai	Akurasi
Jarak (Meter)	0 - 6	66.7%
	7 - 12	100%
	13 - 18	16.6%
Kecepatan (Km/h)	0 - 5	83.3%
	6 - 10	66.7%
	11 - 15	66.7%
	16 - 20	50%
	21 - 30	33.3%
Sudut Kamera (°)	40	100%
	90	66.7%
	140	66.7%
Lux	3.6	50%
	294	100%
Jumlah Orang (Orang)	1	100%
	2	0%
	3	0%

Pada tabel 3 diatas , dari lima parameter yang digunakan yaitu jarak, kecepatan, sudut kamera, lux, dan jumlah orang. Sistem dapat mendeteksi objek dengan akurat pada jarak 7 – 12 meter, sudut kamera 40°, lux 294, dan jumlah orang adalah 1. Dapat disimpulkan sistem yang dibuat belum dapat memprediksi dengan benar pada beberapa percobaan.

D. Pengujian K-Fold

Pada pengujian k-fold ini dilakukan menggunakan tambahan dataset baru berjumlah 326 gambar. Selama proses *training* dataset baru ini akan menggunakan k-fold atau *cross validation* dengan nilai k=3.

TABEL 4
PENGUJIAN K-FOLD.

Pengujian	Nilai	mAP@.50IoU	AR@100
K-Fold	3	44.9%	52.5%

Dari hasil pengujian *cross validation* menggunakan K-Fold bernilai 3 diperoleh nilai *Mean Average Precision* (mAP) sebesar 44.9% dan nilai *Average Precision* (AP) sebesar 52.5%.

IV. KESIMPULAN

Dari rangkaian pengujian yang telah dilakukan menggunakan algoritma SSD, diperoleh hasil performansi model pada rasio dataset *train* 90% dan *test* 10%, *learning rate* 0,004, *epochs* 1, dan *batch size* 24 dengan nilai mAP 79,2% dan AR 61,4%. Sistem yang telah dibuat dapat berjalan baik pada jarak 7

– 12 meter, kecepatan 0 – 5 km/h, sudut kamera 40°, lux 294, dan jumlah orang 1 dengan akurasi tertinggi yaitu 100% serta dapat memberikan notifikasi melalui telegram dengan FPS rata-rata yang diperoleh adalah 2.

REFERENSI

- [1] “J.D.I.H. - Dewan Perwakilan Rakyat.”
[\(https://www.dpr.go.id/jdih/index/id/539%20\(accessed%20Dec.%2019,%202021\)\)](https://www.dpr.go.id/jdih/index/id/539%20(accessed%20Dec.%2019,%202021)) (accessed Aug. 23, 2022).

- [2] "Ethem Alpaydin - Introduction to Machine Learning-The MIT Press (2014)".
- [3] J. Wira and G. Putra, "Pengenalan Konsep Pembelajaran Mesin dan Deep Learning Edisi 1.4 (17 Agustus 2020)."
- [4] V. Tyagi, *Understanding Digital Image Processing*. CRC Press, 2018. doi: 10.1201/9781315123905.
- [5] N. Ketkar and J. Moolayil, *Deep Learning with Python*. Apress, 2021. doi: 10.1007/978-1-4842-5364-9.
- [6] R. Vargas, A. Mosavi, and L. Ruiz, "DEEP LEARNING: A REVIEW," 2017.
- [7] Q. Shuai and X. Wu, "Object detection system based on SSD algorithm," in *Proceedings - 2020 International Conference on Culture-Oriented Science and Technology, ICCST 2020*, Oct. 2020, pp. 141–144. doi: 10.1109/ICCST50977.2020.00033.
- [8] S. Zhai, D. Shang, S. Wang, and S. Dong, "DF-SSD: An Improved SSD Object Detection Algorithm Based on DenseNet and Feature Fusion," *IEEE Access*, vol. 8, pp. 24344–24357, 2020, doi: 10.1109/ACCESS.2020.2971026.
- [9] W. Liu *et al.*, "SSD: Single shot multibox detector," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2016, vol. 9905 LNCS, pp. 21–37. doi: 10.1007/978-3-319-46448-0_2.
- [10] Y. Amit, P. Felzenszwalb, and R. Girshick, "Object Detection," in *Computer Vision*, Springer International Publishing, 2020, pp. 1–9. doi: 10.1007/978-3-030-03243-2_660-1.
- [11] C. Gde Wahyu Pramana, D. Care Khrisne, and N. Putra Sastra, "Rancang Bangun Object Detection Pada Robot Soccer Menggunakan Metode Single Shot Multibox Detector (SSD MobileNetV2)," 2021.
- [12] N. Rochmawati *et al.*, "Analisa Learning rate dan Batch size Pada Klasifikasi Covid Menggunakan Deep learning dengan Optimizer Adam."
- [13] "Confusion Matrix untuk Evaluasi Model pada Supervised Learning | by Kuncahyo Setyo Nugroho | Medium." <https://ksnugroho.medium.com/confusion-matrix-untuk-evaluasi-model-pada-unsupervised-machine-learning-bc4b1ae9ae3f> (accessed Aug. 02, 2022).
- [14] "confusion matrix".
- [15] A.- Arini, L. K. Wardhani, and D.-Octaviano, "Perbandingan Seleksi Fitur Term Frequency & Tri-Gram Character Menggunakan Algoritma Naïve Bayes Classifier (Nbc) Pada Tweet Hashtag #2019gantipresiden," *KILAT*, vol. 9, no. 1, pp. 103–114, Apr. 2020, doi: 10.33322/kilat.v9i1.878.
- [16] "What is Mean Average Precision (MAP) and how does it work." <https://xailient.com/blog/what-is-mean-average-precision-and-how-does-it-work/> (accessed Aug. 02, 2022).