

Perbandingan Algoritma YOLOv4 dan *Scaled* YOLOv4 untuk Deteksi Objek pada Citra Termal

AZIZAH AULIA RAHMAN, SISLY DESTRI AGUSTIN, NUR IBRAHIM, NOR CAECAR KUMALASARI

Program Studi S1 Teknik Telekomunikasi, Universitas Telkom
Email: sislydestri.a@gmail.com

Received 3 Maret 2022 | Revised 12 Mei 2022 | Accepted 23 Juni 2022

ABSTRAK

Minimnya visibilitas pejalan kaki dan pengendara pada malam hari karena kurangnya pencahayaan pada lampu jalan menyebabkan kecelakaan rentan terjadi pada rentang waktu tersebut. Sistem penglihatan komputer berbeda dengan manusia, semua objek dengan suhu di atas nol dapat memancarkan radiasi inframerah jika direkam menggunakan kamera termal. Dalam penelitian ini penulis mengidentifikasi citra termal dalam bentuk citra RGB dengan algoritma YOLOv4 dan Scaled YOLOv4 sebagai deteksi objek. Performa sistem diukur berdasarkan nilai presisi, recall, f1-score, dan mAP. Eksperimen dilakukan pada dataset citra termal dengan objek manusia. Skenario yang digunakan adalah mendeteksi objek dengan jarak 5m, 10m, 15m, dan 20m. Hasil deteksi didapatkan algoritma Scaled YOLOv4 CSP lebih unggul dengan nilai pengujian precision 94,3%, recall 83,8%, f1-Score 88,7%, dan mAP 86,9%. Hasil tersebut dipengaruhi oleh ukuran citra dan jumlah dataset dari citra training, citra validation, dan citra uji.

Kata kunci: Citra Termal, YOLO, YOLOv4, Scaled-YOLOv4, Deteksi Objek

ABSTRACT

The lack of visibility of pedestrians and drivers at night due to lack of lighting in street lights makes accidents prone to occur during this time. The computer vision system is different from the humans, any object with a temperature above zero can emit infrared radiation when using a thermal camera. In this study, the authors identify thermal images in RGB using YOLOv4 and Scaled YOLOv4 as object detection algorithms. System performance is measured based on the value of precision, recall, f1-score, and mAP. Experiments were carried out on a thermal image dataset with human objects. The scenario used was to detect objects at a distance of 5m, 10m, 15m, and 20m. The detection results show that Scaled YOLOv4 CSP algorithms is the best, based on the test value of 94.3% precision, 83.8% recall, 88.7% f1-Score, and 86.9% mAP. These results are influenced by the size of the image and the number of datasets from training images, validation images, and test images.

Keywords: Thermal Image, YOLO, YOLOv4, Scaled-YOLOv4, Object Detection

1. PENDAHULUAN

Berkendara dengan kecepatan tinggi meningkatkan risiko terjadinya kecelakaan lalu lintas, terutama bagi pengendara lain dan pejalan kaki. Selain itu, peningkatan jumlah kendaraan bermotor tanpa diikuti oleh peningkatan infrastruktur transportasi seperti lampu jalan dan trotoar juga turut meningkatkan risiko terjadinya kecelakaan lalu lintas. Minimnya visibilitas pejalan kaki dan pengendara pada malam hari karena kurangnya pencahayaan pada lampu jalan menyebabkan kecelakaan rentan terjadi pada rentang waktu tersebut. Organisasi Kesehatan Dunia atau biasa dikenal dengan *World Health Organization* (WHO) mengumumkan angka kematian lalu lintas jalan pada tahun 2018 mencapai 1,35 juta jiwa, pada tahun 2016 WHO menempatkan Indonesia sebagai negara kedua dengan korban kecelakaan jalan terbanyak, tercatat 5.005 orang meninggal atau setidaknya 14 pejalan kaki meninggal setiap harinya **(Tran, 2018)**.

Dalam kondisi pencahayaan yang buruk, mata manusia dan sistem penglihatan komputer yang beroperasi dalam spektrum tampak rentan gagal, sensor yang dikalibrasi untuk menerima cahaya redup dapat dengan mudah disilaukan oleh kendaraan yang mendekat dari arah yang berlawanan. *Computer vision* merupakan sebuah kemampuan komputer atau mesin yang didesain agar mampu melihat, memahami, serta menganalisis suatu objek selayaknya manusia **(Sonka, dkk., 2013)**. Kinerja *computer vision* berbeda dengan mata manusia, semua objek dengan suhu di atas nol mutlak memancarkan radiasi inframerah yang dapat direkam dengan kamera termal. Sensor termal dapat digunakan untuk melakukan deteksi tubuh manusia tanpa perlu menerangi subjek dengan benar.

Berdasarkan penelitian sebelumnya yang berjudul "*Thermal Object Detection in Difficult Weather Conditions Using YOLO*" berhasil membuat sistem deteksi objek dalam citra COCO RGB dengan menggunakan kamera termal **(Kristo, dkk., 2020)**. Skenario yang digunakan adalah mendeteksi objek dengan jarak mulai dari 30m hingga 215m pada kondisi cuaca yang berbeda. Beberapa algoritma yang diuji yakni Faster R-CNN, SSD, FCOS, Cascade R-CNN, dan YOLOv3. Namun, kinerja YOLOv3 secara signifikan lebih cepat dan akurat. Penelitian lainnya menggunakan metode Deblur-SRRGAN dan deteksi objek berbasis R-CNN dan dataset citra termal abu-abu yang direkonstruksi menjadi citra RGB **(Batchuluun, dkk., 2021)**. Hasil penelitian ini hanya mendeteksi 1 objek dengan nilai total kedua metode 10.78 *fps*, dimana tergolong kurang cepat untuk memprosesnya. Penelitian berikutnya, berfokus pada deteksi pejalan kaki dalam kondisi malam yang parah **(Nowosielski, dkk., 2020)**. Sistem yang diusulkan ialah dengan memproses citra termal menggunakan komputer mikro ODROID XU4. Penelitian ini melakukan evaluasi pada algoritma detektor AdaBoost+ Haar dan algoritma CNN yaitu YOLOv2. Hasilnya menunjukkan algoritma CNN memiliki kinerja system yang rendah namun tingkat akurasi *false detection* lebih baik daripada detektor AdaBoost+ Haar. Algoritma detektor AdaBoost+ Haar kurang mampu belajar mendeteksi objek yang menunjukkan variasi pose kompleks. Meskipun demikian penulis percaya bahwa lebih dari 9 *fps* yang diperoleh dapat lebih dipercepat dengan beberapa pengoptimalan kode. More Rahul Tanaji melakukan penelitian **(Tanaji, 2021)**, tingkat akurasi untuk mendeteksi manusia dengan kamera termal mencapai 85.6%. Skenario untuk dataset yang digunakan adalah mendeteksi objek terjauh, terdekat, tertutup, dan grup dikondisi minim cahaya. Metode yang digunakan yakni meng-ekstraksi gambar termal untuk mendapatkan klasifikasi terbaik. Setelah *filtering* dan *clustering*, dataset memiliki informasi berupa pola gerak, yang akan diidentifikasi pola gerak ini merupakan manusia atau makhluk hidup yang lain. Penelitian lainnya, **(Tumas, dkk., 2020)** melakukan penelitian tentang deteksi pejalan kaki pada cuaca yang buruk. Penelitian ini berfokus pada pengenalan *benchmark database* baru yang lebih unggul, metode yang digunakan ialah YOLOv3 dan TINYv3, dalam prosesnya penelitian ini menguji dan

menyediakan modifikasi kedalaman 16-bit untuk detektor berbasis DNN YOLOv3, dan hasil yang didapatkan mencapai *mean Average Precision* (mAP) hingga 89,1%.

Pada penelitian ini algoritma detektor yang diusulkan ialah YOLOv4 dan *Scaled* YOLOv4. Algoritma *You Only Look Once* (YOLO) memiliki beberapa versi, YOLOv4 merupakan versi yang ditingkatkan dari YOLOv3. YOLOv4 dapat meng-identifikasi objek secara cepat, efisien, dan akurat terbukti dengan nilai *Average Precision* (AP) yang mencapai angka 65.7% (**Bochkovskiy, dkk., 2020**). *Scaled* YOLOv4 merupakan YOLOv4 yang ditingkatkan dan diskalakan, menjadi yang paling akurat (55,8% AP *Microsoft COCO test-dev*) diantara *neural network* yang diterbitkan (**C. Y. Wang, dkk., 2021**). Berdasarkan uraian di atas, penulis merancang sistem deteksi objek pada kamera termal menggunakan dua algoritma berbeda yaitu YOLOv4 dan *Scaled* YOLOv4 untuk melihat dan menemukan nilai terbaik diantara keduanya. Diharapkan sistem dapat bekerja secara efisien serta memiliki tingkat akurasi tinggi sehingga meningkatkan visibilitas pengendara terhadap pejalan kaki pada pencahayaan yang buruk dan meminimalisir terjadinya kecelakaan di jalan.

2. METODE PENELITIAN

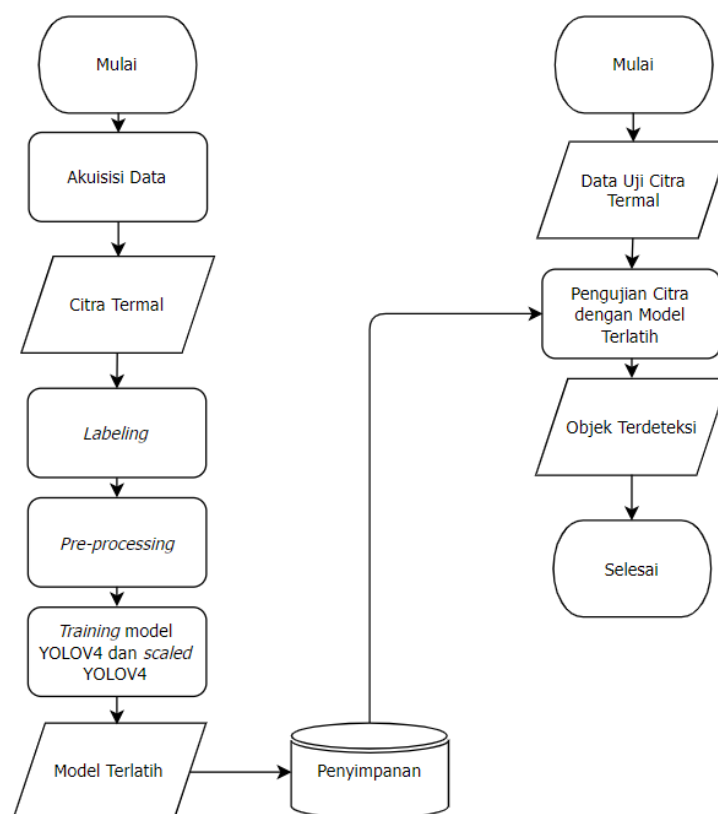
Sistem deteksi secara umum terdiri dari beberapa tahapan. Tahap pertama mengakuisisi dataset lalu didapatkan citra termal. Citra yang didapat akan diolah untuk meningkatkan kualitas citra agar sesuai dengan kebutuhan. Setelah *pre-processing* dataset akan dilatih dan diuji pada sistem yang dirancang seperti pada Gambar 1.



Gambar 1. Diagram Blok Sistem

2.1 Desain Sistem

Sistem deteksi manusia ini didesain menggunakan algoritma YOLOv4 dan *Scaled* YOLOv4 dan diproses dengan 2 tahapan. Tahap pertama proses pelatihan, yakni proses citra dari dataset akan melalui *labeling* dan *pre-processing* lalu akan dilatih untuk mendapatkan nilai bobot setiap kelas yang akan disimpan dan diproses dengan model YOLOv4 dan *Scaled* YOLOv4. Tahap kedua ialah proses pengujian, model akan menerima input baru yaitu data pengujian nilai bobot setiap kelas yang dihasilkan akan dibandingkan dengan nilai bobot pada saat proses pelatihan untuk mendapatkan dan memvalidasi nilai akurasi sistem. Untuk memudahkan pemahaman setiap proses dalam sistem ini, maka dibuatlah *flowchart* (diagram alur) yang merepresentasikan alur kinerja setiap proses seperti pada Gambar 2 berikut.

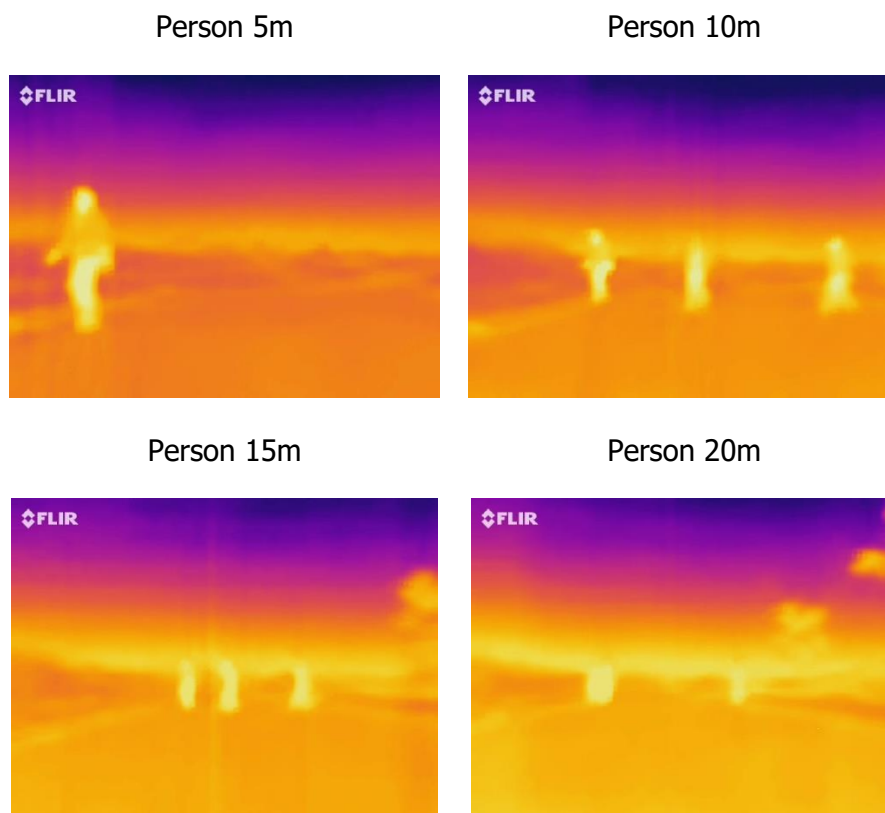


Gambar 2. Flowchart Sistem

2.2 Dataset Citra Termal

Citra termal atau biasa dikenal dengan termogram adalah tampilan visual dari jumlah energi inframerah yang dipancarkan, ditransmisikan, dan dipantulkan oleh suatu objek. Gelombang inframerah meliputi semua radiasi elektromagnetik yang lebih dari gelombang cahaya tampak, tetapi lebih pendek dari *microwaves* (Arif, dkk., 2017). Pencitraan inframerah atau *infrared* (IR) disebut sebagai pencitraan termal karena panas yang dihasilkan atau dipantulkan oleh suatu objek sebanding dengan radiasi elektromagnetik yang dipancarkan IR. Inframerah tidak bisa terlihat oleh manusia karena memiliki panjang gelombang lebih panjang daripada cahaya tampak, berkisar antara 400nm hingga 1400nm (Kristo & Ivasic-Kos, 2018). Model kamera FLIR (*Forward Looking Infrared*) adalah salah satu pencitraan termal yang cukup handal. Menurut penelitian (Kirimtat & Krejcar, 2018) kamera FLIR memiliki kualitas dan visibilitas citra yang cukup tinggi, karena memiliki detektor termal dan foton sehingga percobaan lebih baik dilakukan ketika tidak ada cahaya dominan atau cahaya langsung pada objek. FLIR juga memiliki *converter* untuk mengubah sudut kamera.

Dataset yang digunakan diambil pada malam hari jam 19.00-20.30 WIB dengan cuaca sedikit gerimis menggunakan kamera FLIR ONE Gen 3. Skenario pengambilan dataset ialah 3 pejalan kaki yang berjalan secara bergantian, berpapasan, dan berdampingan dengan skenario jarak yang berbeda. Sehingga dataset terdiri dari empat kelas, yakni berdasarkan skenario jarak pengambilan citra dari objek mulai dari jarak 5m, 10m, 15m, hingga 20m seperti pada Gambar 3. Meskipun terdiri dari empat kelas, objek yang dideteksi jenisnya hanya satu, yaitu *person*. Citra objek yang terdeteksi ialah format JPG berukuran 640×480 piksel. Citra yang terkumpul sebanyak 2535 citra dengan uraian, jarak 5m sebanyak 681 citra, jarak 10m sebanyak 703 citra, jarak 15m sebanyak 672 citra, dan jarak 20m sebanyak 479 citra.



Gambar 3. Citra Berdasarkan Kelas

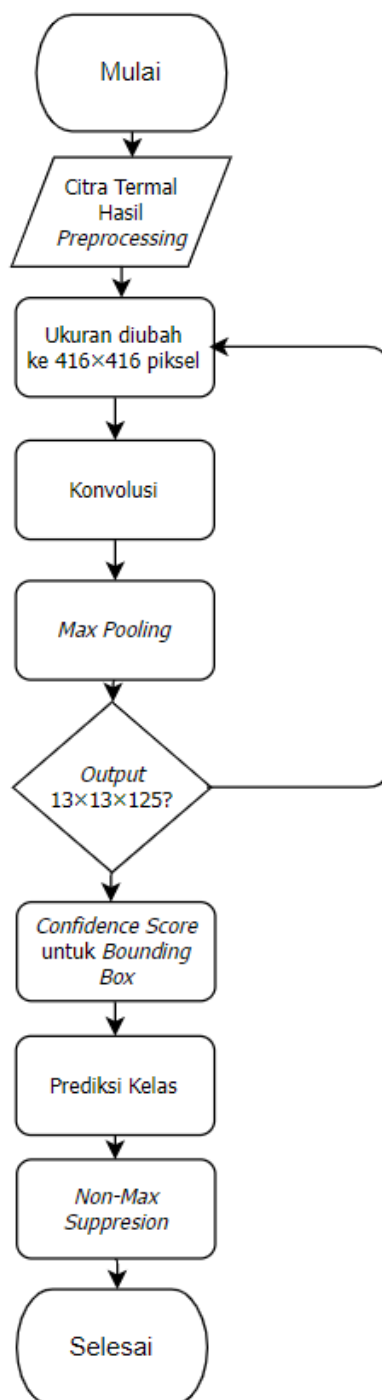
2.3 Pre-processing

Tidak semua citra yang didapatkan berada dalam kondisi yang ideal. Beberapa kondisi citra tidak dianggap ideal, seperti masih banyak noise, ukuran citra yang belum sesuai, hingga kurang atau kelebihan kontras pada suatu citra. Agar citra dapat diolah secara maksimal membutuhkan tahapan *pre-processing*. *Pre-processing* adalah mengubah citra awal menjadi citra lain dengan cara menambahkan atau mengurangi bagian bagian hal yang dianggap penting maupun tidak untuk dapat diproses pada tahapan selanjutnya (**Rasyid, dkk., 2019**). Dataset yang telah di dapatkan sebelumnya perlu diberi label kelas pada setiap citranya, proses ini dapat disebut sebagai anotasi data. Proses pelabelan dilakukan secara manual menggunakan *tools labeling images* dikarenakan *tools* tersebut sederhana dan mudah untuk digunakan, hanya dengan membuat kotak pembatas dan memberi kelas pada setiap objek yang akan dideteksi pada citra. Aplikasi yang digunakan untuk pelabelan adalah LabelImg dengan file output dapat dalam format PascalVOC atau XML. Pada penelitian ini, citra akan melalui jenis *pre-processing* *resize*. *Resize* adalah proses memperbesar atau memperkecil ukuran suatu citra dari ukuran citra asli dalam satuan piksel.

2.4 Pelatihan Model

Setelah melakukan *pre-processing* pada dataset, maka selanjutnya adalah melakukan pelatihan model pada dataset. Implementasi dari penelitian ini adalah membuat model yang dapat mendeteksi pergerakan sebuah objek. Tahapan ini berfungsi untuk mendapatkan pola atau ciri khusus dari citra yang dimasukkan dan dapat mendeteksi objek berdasarkan kelas. Pola khusus pada citra akan digunakan untuk membedakan berbagai macam gerakan. Model

dari penelitian ini akan menggunakan algoritma YOLOv4 dan *Scaled* YOLOv4. Tahap pelatihan model pada algoritma YOLO ditunjukkan pada Gambar 4.

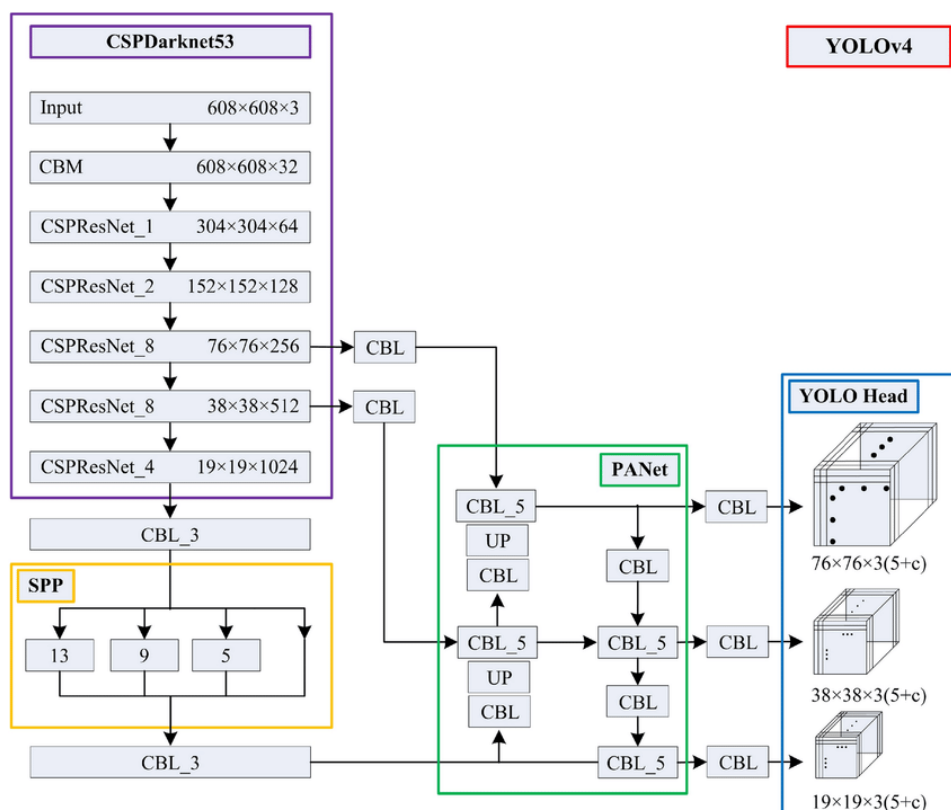


Gambar 4. Flowchart Pelatihan Model dengan Algoritma YOLO

2.4.1 Arsitektur YOLOv4

Dalam penelitian yang dilakukan oleh Redmon, YOLO dapat melakukan pengenalan objek secara *real-time* dengan kecepatan 45 *fps* karena memiliki arsitektur yang sederhana, selain itu akurasi rata-rata yang didapat mencapai 88% dalam ImageNet 2012 Validation. Cara kerja YOLO dengan melihat citra secara keseluruhan, kemudian melewati *neural network* sekali dan otomatis mendeteksi objek yang ada (Redmon, dkk., 2016). Arsitektur YOLOv4

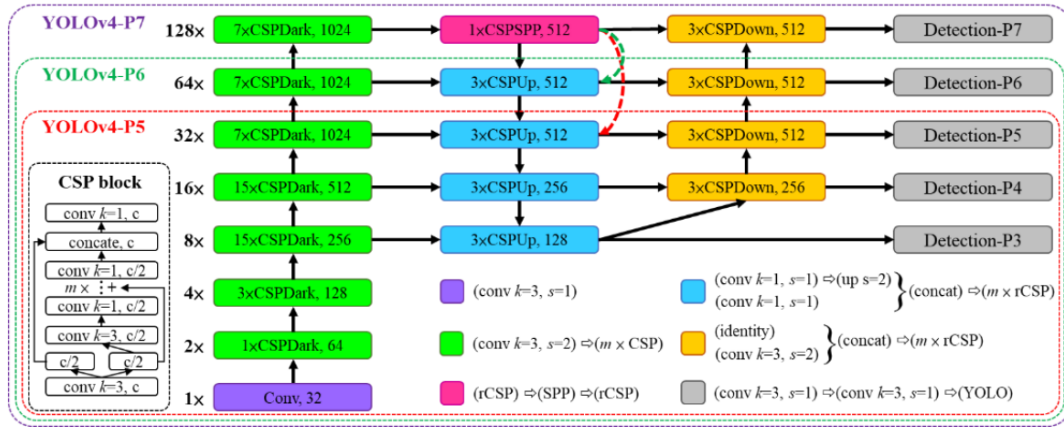
menggunakan *backbone*: CSPDarknet53, *neck*: SPP dan PAN, dan *head*: YOLOv3 seperti pada Gambar 5. *Feature extraction* yang bernama CSPDarknet53 ini memiliki 29 *convolutional layers* 3×3 , 725×725 pada *receptive field* dan 27.6 M parameter. Berdasarkan nilai ini, CSPDarknet53 merupakan model yang optimal sebagai *feature extraction* atau *backbone* dari detektor. Beberapa pengaruh dari kemampuannya saat citra di-input yakni bisa melihat objek secara detail, dan meningkatkan jumlah kombinasi setiap titik di suatu citra. SPP pada *neck* membantu memisahkan fitur konteks secara signifikan dan tidak mengurangi kecepatan operasi jaringan. Sedangkan, PANet digunakan sebagai parameter penyatuan dari beberapa *level backbone* (Bochkovskiy, dkk., 2020).



Gambar 5. Arsitektur YOLOv4 (S. Wang, dkk., 2021)

2.4.2 Arsitektur *Scaled* YOLOv4

Scaled YOLOv4 menggunakan teknik penskalaan jaringan yang optimal untuk mendapatkan jaringan yaitu YOLOv4-CSP (P5, P6, dan P7). Pada jaringan arsitektur dilakukan peningkatan dengan *backbone* dan dioptimalkan dengan menggunakan koneksi *Cross Stage Partial* (CSP) dan aktivasi *Mish* seperti pada Gambar 6. Untuk setiap resolusi jaringan, jaringan saraf dilatih terpisah tidak seperti di YOLOv4, hanya satu jaringan saraf yang dilatih untuk semua resolusi. Dalam *Scaled* YOLOv4, dilakukan perbaikan pada *normalizer* lapisan. pelatihan jaringan kemungkinan akan lebih cepat karena pada aktivasi, tinggi dan lebar diubah.



Gambar 6. Arsitektur *Scaled*YOLOv4 (C. Y. Wang, dkk., 2021)

2.5 Parameter Kinerja

Adapun parameter uji yang digunakan untuk menentukan tingkat keberhasilan dari kinerja sistem menggunakan model yang telah dilatih untuk mendeteksi objek. Beberapa parameter uji yang digunakan adalah mAP, presisi, *f1-score*, dan *recall*.

2.5.1 Presisi

Presisi adalah nilai yang menggambarkan akurasi antara data yang diinginkan dengan hasil prediksi yang keluar. Nilai presisi dapat dihitung menggunakan Persamaan (1) berikut.

$$Presisi = \frac{TP}{TP + FP} \times 100 \quad (1)$$

Dimana presisi dinyatakan dalam bentuk persen (%) TP sebagai nilai *true positive*, FP sebagai nilai *false positive*.

2.5.2 Recall

Recall atau sensitivitas adalah nilai rata-rata dari seluruh data yang benar dan menggambarkan keberhasilan model dalam menemukan kembali sebuah informasi yang ada. Nilai *recall* dapat dihitung menggunakan Persamaan (2) berikut.

$$Recall = \frac{TP}{TP + FN} \times 100 \quad (2)$$

Dimana *recall* dinyatakan dalam bentuk persen (%) TP sebagai nilai *true positive*, FN sebagai nilai *false negative*.

2.5.3 F1-Score

F1-Score adalah nilai rata-rata yang menggambarkan perbandingan presisi dan recall. Nilai *f1-Score* dapat dihitung menggunakan Persamaan (3) berikut.

$$F1 - Score = 2 \times \frac{Presisi \times Recall}{Presisi + Recall} \quad (3)$$

2.5.4 Mean Average Precision (mAP)

Mean Average Precision (mAP) merupakan metrik populer untuk mengukur akurasi deteksi suatu objek. *Mean Average Precision* digunakan pada algoritma deteksi objek seperti R-CNN, Faster R-CNN, SSD, dll. *Mean average precision* merata-ratakan nilai AP dari setiap kelas yang ada pada model yang telah diuji. Untuk mengetahui nilai AP suatu kelas, perlu diketahui nilai presisi dan *recall* (Hamman, dkk., 2020). Setelah nilai presisi dan *recall* diketahui, nilai

presisi diplot terhadap nilai *recall* untuk mendapatkan pola yang berbentuk zig-zag. Untuk menghitung nilai AP, hasil plot berbentuk zig-zag tersebut akan dihaluskan menggunakan Persamaan (4).

$$P_{inter}(r_{n+1} - r_n) = \max p(r') ; r' \geq r_{n+1} \quad (4)$$

Selanjutnya, nilai AP dapat dihitung menggunakan Persamaan (5).

$$AP = \sum (r_{n+1} - r_n) P_{inter}(r_{n+1}) \quad (5)$$

Setelah nilai AP diketahui, maka nilai *mean average precision* dapat dihitung menggunakan Persamaan (6).

$$mAP = \sum_{i=1}^N \frac{AP(i)}{N} \times 100\% \quad (6)$$

Keterangan :

- P : Presisi
- r : *Recall*
- r' : *Recall* selanjutnya
- N : Banyaknya kelas yang terdapat pada suatu model yang telah dilatih

3. HASIL DAN PEMBAHASAN

Pada penelitian ini dilakukan skenario pengujian dimana hasil dari tiap skenario yang dilakukan akan dianalisis sehingga didapatkan model sistem terbaik dalam mendeteksi objek menggunakan algoritma YOLOv4 dan *Scaled* YOLOv4. Skenario pengujian yang akan dilakukan yaitu dengan membagi jumlah data latih dan data uji dengan rasio 80:20, untuk tahap *pre-processing* dilakukan *resizing* citra sebanyak 3 kali, yakni pada ukuran 416×416, 448×448, dan 512×512, lalu *optimizer* yang digunakan ialah *optimizer Stochastic Gradient Descent* (SGD), dan nilai *batch size* yang digunakan adalah 32, sedangkan untuk nilai *learning rate* yang digunakan adalah 0.01. Hasil pengujian menggunakan algoritma YOLOv4 dapat dilihat pada Tabel 1, sedangkan untuk hasil pengujian menggunakan algoritma *Scaled* YOLOv4 dapat dilihat pada Tabel 2.

Tabel 1. Hasil Testing Menggunakan Algoritma YOLOv4

<i>Resizing</i>	<i>Learning Rate</i>	<i>Average</i>			mAP
		<i>Precision</i>	<i>Recall</i>	<i>F1-Score</i>	
416×416	0.01	0.96	0.76	0.85	84.46 %
448×448	0.01	0.98	0.77	0.86	84.34 %
512×512	0.01	0.98	0.77	0.86	85.44 %

Berdasarkan hasil pengujian pada Tabel 1 dalam melakukan deteksi objek pada citra termal dengan algoritma YOLOv4, ukuran citra mempengaruhi perubahan nilai *precision*, *recall*, dan *f1-score* namun perubahan nilai tidak signifikan. Kinerja sistem terbaik pada ukuran citra 512×512 piksel mendapatkan nilai rata-rata *precision* 98%, *recall* 77%, dan *f1-score* 86% dengan nilai mAP 85.44%.

Scaled YOLOv4 diskalakan dari YOLOv4 menjadi CSP, P5, P6, dan P7. *Scaled* YOLOv4 CSP memiliki kinerja sistem yang cukup baik, serta waktu pelatihan dan pengujian lebih cepat dibandingkan dengan P5, P6, dan P7. Maka dari itu pengujian menggunakan algoritma *Scaled* YOLOv4 CSP.

Tabel 2. Hasil Testing Menggunakan Algoritma *Scaled*YOLOv4 CSP

Resizing (piksel)	Learning Rate	Average			mAP
		Precision	Recall	F1-Score	
416×416	0.01	0.943	0.838	0.88	86.90 %
448×448	0.01	0.957	0.81	0.87	81.00 %
512×512	0.01	0.967	0.816	0.88	81.40 %

Berdasarkan hasil pengujian pada Tabel 2 dalam melakukan deteksi objek pada citra termal dengan algoritma *Scaled*YOLOv4 CSP, ukuran citra mempengaruhi perubahan nilai *precision*, *recall*, dan *f1-score*. Kinerja sistem terbaik pada ukuran citra 416×416 piksel mendapatkan nilai rata-rata *precision* 94.3%, *recall* 83.8%, dan *f1-score* 88% dengan nilai mAP 86.90%.

Berdasarkan hasil pengujian kedua algoritma, didapatkan *Scaled* YOLOv4 CSP sebagai algoritma terbaik diantara keduanya dengan nilai mAP yang cukup tinggi yakni 86.9%. Pada kedua algoritma hasil kinerja sistem cukup buruk pada ukuran citra masukan 448×448.

4. KESIMPULAN

Penelitian ini telah membandingkan YOLOv4 dan *Scaled* YOLOv4 untuk deteksi objek pada citra termal. Dapat dilihat dari kedua algoritma tersebut, *Scaled* YOLOv4 memiliki performansi lebih baik dari YOLOv4. Hasil tersebut ditunjukkan pada nilai rata-rata *precision* 94.3%, *recall* 83.8%, dan *f1-score* 88% dengan nilai mAP 86.90%. Pengujian dilakukan dengan menggunakan nilai learning rate 0.01, dan nilai batch size 32. Hasil pengujian dipengaruhi oleh ukuran citra yang diperbesar sebanyak 3 kali dan jumlah dataset dari data latih sebesar 2028 citra, dan data uji sebesar 507 citra. Berdasarkan jarak pengambilan citra dari 5m hingga 20m, sistem tetap dapat mengenali objek. Dengan hasil kinerja sistem yang cukup baik, diharapkan sistem dapat bekerja secara efisien serta memiliki mAP tinggi sehingga meningkatkan visibilitas pengendara terhadap pejalan kaki pada pencahayaan yang buruk dan meminimalisir kecelakaan di jalan.

DAFTAR RUJUKAN

- Arif, M., Alandani, R., Arya, M., & Thermal, A. R. (2017). Thermal Vision pada Manusia dengan Pengaruh Terhadap Warna Pakaian. *5th Indonesian Symposium on Robotic Systems and Control*, (pp. 243–248).
- Batchuluun, G., Kang, J. K., Nguyen, D. T., Pham, T. D., Arsalan, M., & Park, K. R. (2021). Deep Learning-Based Thermal Image Reconstruction and Object Detection. *IEEE Access*, *9*, 5951–5971.
- Bochkovskiy, A., Wang, C.-Y., & Liao, H.-Y. M. (2020). *YOLOv4: Optimal Speed and Accuracy of Object Detection*.
- Hammam, H., Asyhar, A., Wibowo, S. A., & Budiman, G. (2020). Implementasi Dan Analisis Performansi Metode You Only Look Once (Yolo) Sebagai Sensor Pornografi Pada Video Implementation and Performance Analysis of You Only Look Once (YOLO) Method As

- Porn Censorship in Video. *e-Proceeding of Engineering*, 7(2), 3631–3638.
- Kirimtat, A., & Krejcar, O. (2018). FLIR vs SEEK in Biomedical Applications of Infrared Thermography. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics): Vol. 10814 LNBI* (Nomor September). Springer International Publishing.
- Kristo, M., & Ivasic-Kos, M. (2018). An overview of thermal face recognition methods. *2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics*, (pp. 1098–1103).
- Kristo, M., Ivasic-Kos, M., & Pobar, M. (2020). Thermal Object Detection in Difficult Weather Conditions Using YOLO. *IEEE Access*, 8, 125459–125476.
- Nowosielski, A., Malecki, K., Forczmanski, P., Smolinski, A., & Krzywicki, K. (2020). Embedded Night-Vision System for Pedestrian Detection. *IEEE Sensors Journal*, 20(16), 9293–9304.
- Rasyid, M. R., Tahir, Z., & Syafaruddin, N. (2019). Digital Image Processing for Detecting Industrial Machine Work Failure with Quantization Vector Learning Method. *Journal Pekommas*, 4(2), 131.
- Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You Only Look Once: Unified, Real-Time Object Detection. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Sonka, M., Hlavac, V., & Boyle, R. (2013). *Image processing, analysis and machine vision* (4th ed.). USA: Cengage Learning.
- Tanaji, M. R. (2021). Detection of Human Targets from Thermal Images. *International Journal of Engineering Research & Technology (IJERT)*, 10(02), 590–593.
- Tran, N. (2018). *GLOBAL STATUS REPORT ON ROAD SAFETY 2018*. World Health Organization.
- Tumas, P., Nowosielski, A., & Serackis, A. (2020). Pedestrian Detection in Severe Weather Conditions. *IEEE Access*, 8, 62775–62784.
- Wang, C. Y., Bochkovskiy, A., & Liao, H. Y. M. (2021). Scaled-yolov4: Scaling cross stage partial network. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, (pp. 13024–13033).
- Wang, S., Zhao, J., Ta, N., Zhao, X., Xiao, M., & Wei, H. (2021). A real-time deep learning forest fire monitoring algorithm based on an improved Pruned + KD model. *Journal of Real-Time Image Processing*, 18.