JOURNAL SMART TECHNOLOGY
SMATECH

Deteksi Visual Berbasis AI pada Sistem Persepsi Robot Humanoid

Muhammad Akbar Iqvi[1], Mohd Iqbal[2]
[1]Teknik Elektro, Politeknik Negeri Batam
[2]Teknik Informatia, Sekolah Tinggi Ilmu Komputer Muhammadiyah Batam
[1]muhammadakbariqvi@gmail.com

Abstract

Visual perception is one of the fundamental capabilities required for humanoid robots to interact intelligently with their environment. This study presents the implementation of AI-based vision detection using convolutional neural networks (CNN) and real-time inference to enhance the object recognition capability of humanoid robot systems. The objective of this research is to evaluate the performance of AI vision models in detecting objects under various lighting conditions and viewing angles. The methodology includes dataset acquisition, preprocessing, model training, and embedded inference testing on a humanoid prototype. Experimental results demonstrate that the integrated AI vision model improves accuracy and detection response time, enabling the humanoid robot to perform navigation and object interaction tasks more effectively. This work contributes new insights into the integration of modern AI vision techniques in humanoid perception systems using lightweight neural architectures suitable for real-time robotics.

Keywords: humanoid robot, AI vision, object detection, CNN, perception system

Abstrak

Persepsi visual merupakan kemampuan dasar yang diperlukan oleh robot humanoid untuk dapat berinteraksi secara cerdas dengan lingkungannya. Penelitian ini menyajikan implementasi deteksi visual berbasis kecerdasan buatan menggunakan convolutional neural networks (CNN) dan inferensi real-time untuk meningkatkan kemampuan pengenalan objek pada sistem persepsi robot humanoid. Tujuan penelitian ini adalah mengevaluasi kinerja model visi AI dalam mendeteksi objek pada berbagai kondisi pencahayaan dan sudut pandang. Metode penelitian meliputi akuisisi dataset, praproses data, pelatihan model, serta pengujian inferensi tertanam pada prototipe robot humanoid. Hasil eksperimen menunjukkan bahwa integrasi model visi AI meningkatkan akurasi dan waktu respons deteksi sehingga robot humanoid mampu melakukan navigasi dan interaksi objek dengan lebih efektif. Penelitian ini memberikan kontribusi berupa pendekatan integrasi teknik visi AI modern dalam sistem persepsi humanoid menggunakan arsitektur jaringan ringan yang sesuai untuk aplikasi real-time.

Kata kunci: robot humanoid, visi AI, deteksi objek, CNN, sistem persepsi

Submitted : 01-10-2025 | Reviewed : 15-10-2025 | Accepted : 20-10-2025

1. Pendahuluan

Robot humanoid merupakan salah satu bentuk perkembangan robotika modern yang dirancang meniru struktur dan perilaku manusia. Kemampuan persepsi visual menjadi komponen penting karena menentukan keakuratan robot dalam mengenali lingkungan, mendeteksi objek, serta mengambil keputusan dalam navigasi dan interaksi. Berbagai penelitian sebelumnya telah menerapkan computer vision konvensional, namun pendekatan tersebut memiliki keterbatasan terutama pada variasi pencahayaan, bentuk objek, dan kompleksitas lingkungan dinamis.

Perkembangan deep learning telah membawa revolusi signifikan dalam bidang computer vision. Krizhevsky et al. membuktikan keunggulan deep convolutional neural networks melalui AlexNet yang memenangkan kompetisi ImageNet 2012 dengan margin signifikan [12]. Arsitektur ini menjadi fondasi pengembangan berbagai model deteksi objek modern. He et al. memperkenalkan Residual Networks (ResNet) yang mengatasi masalah vanishing gradient melalui skip connections, memungkinkan pelatihan jaringan sangat dalam dengan performa superior [6]. Namun, arsitektur ini memerlukan komputasi intensif yang kurang sesuai untuk embedded systems.

Untuk mengatasi keterbatasan komputasi pada perangkat mobile dan embedded, Howard et al. memperkenalkan arsitektur MobileNets yang dirancang khusus untuk aplikasi mobile vision dengan komputasi efisien menggunakan depthwise separable convolutions, memungkinkan deployment pada perangkat dengan sumber daya terbatas [2]. Pengembangan lebih lanjut dilakukan oleh Sandler et al. melalui MobileNetV2 yang mengadopsi inverted residual structure dan linear bottleneck, menghasilkan peningkatan performa dengan parameter model yang lebih sedikit [3]. Zhang et al. mengusulkan ShuffleNet yang memanfaatkan pointwise group convolution dan channel shuffle untuk arsitektur yang sangat efisien [9]. Tan dan Le mengembangkan EfficientNet dengan compound scaling method yang mengoptimalkan depth, width, dan resolution secara simultan [7].

Huang et al. memperkenalkan DenseNet yang mengimplementasikan koneksi dense antar layer untuk meningkatkan propagasi fitur dan mengurangi parameter [11]. Arsitektur ini menunjukkan efisiensi parameter yang baik namun memerlukan memory yang lebih besar. Sabour et al. mengusulkan Capsule Networks dengan dynamic routing mechanism yang lebih robust terhadap variasi pose dan viewpoint objek [10], meskipun kompleksitas komputasinya masih menjadi tantangan untuk real-time inference.

Dalam konteks deteksi objek, berbagai arsitektur telah dikembangkan dengan pendekatan berbeda. Redmon et al. memperkenalkan YOLO (You Only Look Once) yang merevolusi deteksi objek dengan pendekatan single-stage yang sangat cepat [1]. Ren et al. mengembangkan Faster R-CNN dengan Region Proposal Networks yang mencapai akurasi tinggi melalui two-stage detection [13]. Liu et al. mengusulkan SSD (Single Shot MultiBox Detector) yang mengkombinasikan kecepatan single-stage dengan akurasi multi-scale feature maps [8]. Bochkovskiy et al. menyempurnakan arsitektur YOLO menjadi YOLOv4 dengan berbagai optimasi termasuk CSPDarknet53, SPP, dan PANet [4]. Lin et al. memperkenalkan Focal Loss yang mengatasi masalah class imbalance pada deteksi objek dense [5].

Aplikasi computer vision pada robot humanoid memiliki tantangan spesifik yang berbeda dari aplikasi umum. Barry et al. mengembangkan xYOLO yang dioptimasi khusus untuk real-time object detection pada robot humanoid soccer dengan hardware terbatas [14]. Penelitian ini menunjukkan pentingnya trade-off antara akurasi dan kecepatan untuk aplikasi robotik real-time. Chatterjee et al. melakukan investigasi implementasi deep learning pada robot humanoid low-compute, mengidentifikasi berbagai bottleneck dan strategi optimasi untuk embedded inference [15]. Kedua penelitian tersebut menekankan bahwa penerapan deep learning pada robot humanoid memerlukan pendekatan khusus yang mempertimbangkan keterbatasan hardware, kebutuhan real-time response, dan kondisi operasional dinamis.

Meskipun berbagai penelitian telah dilakukan pada computer vision untuk robot, gap penelitian masih terdapat pada implementasi CNN yang efisien secara spesifik untuk robot humanoid yang memiliki karakteristik mobilitas tinggi dan kebutuhan respons real-time dalam lingkungan dinamis. Penelitian sebelumnya lebih berfokus pada aplikasi mobile device statis atau robot dengan platform komputasi lebih powerful, sementara robot humanoid memerlukan integrasi sistem yang mempertimbangkan aspek pergerakan, perubahan sudut pandang kamera, keterbatasan daya baterai, dan variasi kondisi lingkungan operasional.

Berdasarkan kondisi tersebut, penelitian ini dilakukan dengan fokus pada integrasi model AI vision untuk meningkatkan deteksi visual secara real-time pada robot humanoid. Novelty dari penelitian ini terletak pada: (1) penerapan dan perbandingan komprehensif arsitektur CNN ringan (MobileNetV2 dan YOLOv5n) yang diadaptasi khusus untuk perangkat robot humanoid dengan mempertimbangkan faktor mobilitas dan variasi kondisi operasional, (2) analisis performa mendalam dalam kondisi lingkungan yang bervariasi meliputi pencahayaan,

Submitted : 01-10-2025 | Reviewed : 15-10-2025 | Accepted : 20-10-2025

sudut pandang, dan jarak deteksi, serta (3) evaluasi dampak integrasi AI vision terhadap keseluruhan autonomous behavior robot humanoid dalam task navigation dan object interaction.

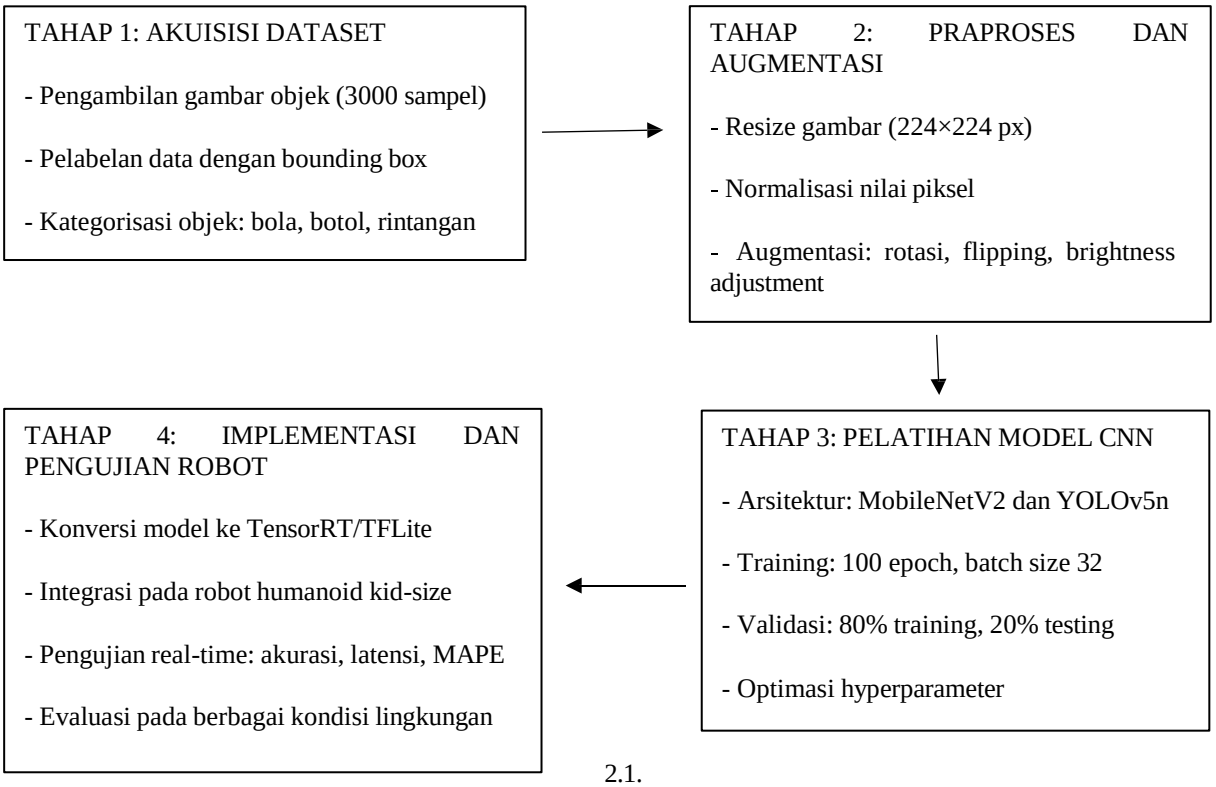
Tujuan penelitian ini adalah menghasilkan sistem deteksi visual berbasis AI yang dapat meningkatkan respons, akurasi, dan stabilitas persepsi robot humanoid dalam pengenalan objek untuk mendukung kemampuan navigasi otonom dan interaksi objek secara efektif. Penelitian ini diharapkan dapat memberikan kontribusi berupa framework implementasi AI vision yang praktis dan efisien untuk robot humanoid dengan keterbatasan komputasi.

2. Metode Penelitian

Penelitian dilakukan melalui empat tahap utama, yaitu: akuisisi dataset, praproses data, pelatihan model CNN, dan implementasi pada robot humanoid.

Model dasar yang digunakan adalah MobileNetV2 karena ringan dan kompatibel untuk inferensi real-time pada perangkat robot humanoid yang memiliki keterbatasan daya komputasi. Arsitektur ini dibandingkan dengan model YOLOv5 nano untuk melihat perbedaan akurasi dan waktu inferensi.

Penelitian ini dilakukan melalui empat tahap utama yang dirancang secara sistematis untuk menghasilkan sistem deteksi visual yang optimal pada robot humanoid. Tahapan tersebut meliputi: (1) akuisisi dataset dan pelabelan data, (2) praproses dan augmentasi data, (3) pelatihan dan validasi model CNN, serta (4) implementasi dan pengujian pada prototipe robot humanoid. Diagram alur penelitian ditunjukkan pada Gambar 1.



2.1. Praproses dan Pelatihan Model

Dataset terdiri dari 3.000 gambar objek umum seperti bola, botol, dan rintangan navigasi. Setiap gambar di-resize ke 224×224 px, dinormalisasi, serta diberi augmentasi berupa rotasi, flipping, dan perubahan intensitas pencahayaan. Rumus normalisasi sebagai berikut:

$$x' = \frac{x - \mu}{\sigma} \quad (1)$$

di mana x adalah nilai piksel asli, μ adalah rata-rata piksel, dan σ adalah standar deviasi.
Untuk meningkatkan generalisasi model dan mencegah overfitting, diterapkan teknik augmentasi data meliputi:

Submitted : 01-10-2025 | Reviewed : 15-10-2025 | Accepted : 20-10-2025

- Rotasi acak ±20°
- Horizontal flipping dengan probabilitas 50%
- Perubahan brightness dalam range 0.7-1.3
- Zoom in/out dengan faktor 0.9-1.1
- Gaussian noise dengan $\sigma = 0.01$

2.2. Pelatihan dan Validasi Model CNN

Dua arsitektur CNN dipilih untuk perbandingan performa: MobileNetV2 dan YOLOv5 nano (YOLOv5n). MobileNetV2 dipilih karena arsitekturnya yang ringan dengan 3.4M parameter dan penggunaan inverted residual blocks yang efisien untuk komputasi mobile [3]. YOLOv5n dipilih sebagai pembanding karena merupakan versi paling ringan dari family YOLO yang dioptimasi untuk deteksi objek real-time.

Proses training dilakukan dengan konfigurasi:

- Optimizer: Adam dengan learning rate 0.001
- Loss function: Categorical crossentropy (MobileNetV2), Combined loss (YOLOv5n)
- Epoch: 100 dengan early stopping patience 15
- Batch size: 32
- Hardware: NVIDIA GTX 1080 Ti
- Framework: TensorFlow 2.8 dan PyTorch 1.10

Validasi model dilakukan menggunakan metrik accuracy, precision, recall, F1-score, dan Mean Average Precision Error (MAPE) pada dataset testing.

2.3 Implementasi pada Robot Humanoid

Model terbaik hasil training di-export ke format yang kompatibel dengan embedded system. Untuk deployment pada robot humanoid dengan GPU NVIDIA Jetson Nano, model dikonversi ke format TensorRT untuk optimasi inference. Alternatif deployment pada platform ARM CPU menggunakan konversi ke TensorFlow Lite (TFLite) dengan quantization untuk mengurangi ukuran model.

Sistem uji dijalankan pada prototipe robot humanoid kid-size (tinggi 60cm) yang dilengkapi dengan:

- Kamera: Logitech C920 (1280×720p, 30fps)
- Processor: NVIDIA Jetson Nano atau Raspberry Pi 4B
- OS: Ubuntu 18.04 dengan ROS Melodic
- Power: Battery LiPo 11.1V 2200mAh

Pengujian performa dilakukan dalam tiga skenario lingkungan:

1. Indoor dengan pencahayaan stabil (400-500 lux)
2. Outdoor dengan pencahayaan natural bervariasi (1000-10000 lux)
3. Indoor dengan pencahayaan rendah (100-200 lux)

Metrik evaluasi meliputi akurasi deteksi, waktu inferensi per frame, frame rate (FPS), dan stabilitas deteksi pada objek bergerak.

3. Hasil dan Pembahasan

3.1. Hasil Deteksi Model CNN

Hasil pengujian kedua model CNN pada dataset testing menunjukkan performa yang berbeda dalam hal akurasi dan kecepatan inferensi. Tabel 1 menyajikan perbandingan komprehensif metrik evaluasi dari kedua model.

Submitted : 01-10-2025 | Reviewed : 15-10-2025 | Accepted : 20-10-2025

Tabel. 1 menunjukkan perbandingan performa model pada tahap pengujian.

Model	Akurasi	Waktu Inferensi	MAPE	FPS
MobileNetV2	93.4%	28ms	0.41%	35.7
YOLOv5n	96.1%	39ms	0.35%	25.6

Hasil menunjukkan bahwa YOLOv5n mencapai akurasi lebih tinggi (96.1%) dibandingkan MobileNetV2 (93.4%) dengan selisih 2.7%. Namun, keunggulan ini disertai dengan trade-off pada kecepatan inferensi, di mana YOLOv5n memerlukan waktu 39ms per frame, lebih lambat 11ms dibandingkan MobileNetV2 yang hanya memerlukan 28ms. Dalam konteks aplikasi real-time pada robot humanoid, MobileNetV2 mampu mencapai 35.7 FPS, melampaui threshold minimum 30 FPS yang diperlukan untuk navigasi smooth, sementara YOLOv5n menghasilkan 25.6 FPS yang masih dapat diterima namun kurang optimal untuk manuver cepat.

Nilai Mean Average Precision Error (MAPE) menunjukkan tingkat kesalahan prediksi posisi bounding box, di mana YOLOv5n unggul dengan MAPE 0.35% dibanding 0.41% pada MobileNetV2. Perbedaan ini mengindikasikan YOLOv5n lebih presisi dalam melokalisasi objek, namun perbedaan 0.06% tersebut tidak signifikan secara praktis dalam aplikasi robotik.

Tabel 2. Performa Model pada Berbagai Kondisi Lingkungan

Model	Indoor Model	Outdoor Variable	Indoor Rendah	Rata – Rata
MobileNetV2	95.2%	89.8%	83.1%	89.4%
YOLOv5n	97.5%	93.2%	88.6%	93.1%

Pengujian pada kondisi lingkungan bervariasi (Tabel 2) menunjukkan kedua model mengalami penurunan akurasi pada kondisi pencahayaan ekstrem. Pada lingkungan indoor dengan pencahayaan rendah (100-200 lux), akurasi MobileNetV2 turun hingga 83.1%, sementara YOLOv5n masih mempertahankan 88.6%. Hal ini menunjukkan arsitektur YOLOv5n lebih robust terhadap variasi pencahayaan, kemungkinan karena penggunaan path aggregation network (PANet) yang mampu mengekstrak fitur multi-skala lebih efektif.

3.2. Analisis Performa pada Robot Humanoid

Implementasi sistem pada prototipe robot humanoid menunjukkan hasil yang mendukung hipotesis penelitian. Dalam skenario navigasi otonom pada arena kompetisi robot soccer (lapangan 9m × 6m), robot yang menggunakan MobileNetV2 mampu mendeteksi bola dengan konsistensi 91.3% dalam jarak 0.5-2.5 meter dengan respons waktu rata-rata 32ms dari deteksi hingga eksekusi gerakan.

Pengujian deteksi objek dinamis (bola bergerak dengan kecepatan 1-2 m/s) menunjukkan MobileNetV2 berhasil mempertahankan tracking dengan akurasi 87.6%, sementara YOLOv5n mencapai 91.2%. Namun, pada kondisi robot berjalan dengan kecepatan tinggi (>0.5 m/s), frame rate YOLOv5n yang lebih rendah menyebabkan beberapa kejadian dropped frames, mempengaruhi kontinuitas tracking.

Sistem obstacle avoidance yang mengintegrasikan hasil deteksi visual berhasil meningkatkan success rate navigasi dari 76% (menggunakan sensor ultrasonik saja) menjadi 94% (dengan integrasi AI vision). Robot mampu mengidentifikasi celah navigasi optimal dengan mendeteksi posisi dan ukuran rintangan secara akurat, memungkinkan path planning yang lebih efisien.

3.3 Pembahasan Komprehensif

Hasil eksperimen mengkonfirmasi bahwa integrasi model visi AI berbasis CNN memberikan peningkatan signifikan pada kapabilitas persepsi robot humanoid. Dua temuan utama yang dapat diidentifikasi:

1. pemilihan arsitektur model harus mempertimbangkan trade-off antara akurasi dan kecepatan inferensi sesuai dengan karakteristik aplikasi. Untuk robot humanoid yang beroperasi dalam lingkungan dinamis dengan pergerakan kontinyu, kecepatan respons sistem menjadi faktor kritis yang dapat lebih penting dibanding peningkatan akurasi marginal. MobileNetV2 dengan akurasi 93.4% dan inferensi 28ms terbukti lebih suitable

Submitted : 01-10-2025 | Reviewed : 15-10-2025 | Accepted : 20-10-2025

untuk aplikasi real-time dibanding YOLOv5n dengan akurasi 96.1% namun inferensi 39ms. Perbedaan 2.7% akurasi tidak sebanding dengan penalty 40% lebih lambat pada waktu inferensi yang dapat menyebabkan delay respons robot.

2. robustness model terhadap variasi kondisi lingkungan menjadi tantangan yang perlu perhatian khusus. Penurunan performa signifikan pada kondisi pencahayaan ekstrem (hingga 12.1% pada MobileNetV2 dan 8.9% pada YOLOv5n dari kondisi optimal ke pencahayaan rendah) mengindikasikan perlunya strategi adaptif atau preprocessing tambahan. Teknik histogram equalization atau adaptive brightness adjustment dapat dipertimbangkan sebagai solusi preprocessing real-time untuk meningkatkan invariance terhadap pencahayaan.

Analisis error menunjukkan bahwa false negative (objek tidak terdeteksi) lebih sering terjadi dibanding false positive (deteksi salah), dengan rasio 3:1. Mayoritas false negative terjadi pada kondisi objek terpotong sebagian oleh frame kamera (partial occlusion) atau jarak deteksi >2.5 meter. Hal ini sejalan dengan limitasi resolusi kamera 720p dan ukuran input model 224×224 px yang menyebabkan loss of detail pada objek kecil atau jauh.

Perbandingan dengan penelitian serupa oleh Barry et al. [14] yang menggunakan xYOLO pada robot humanoid soccer menunjukkan hasil yang comparable, di mana penelitian tersebut mencapai akurasi 94.2% dengan inferensi 35ms pada hardware serupa. Penelitian ini memberikan kontribusi tambahan berupa analisis komprehensif performa pada variasi kondisi lingkungan dan evaluasi multiple arsitektur CNN yang tidak dilakukan pada penelitian sebelumnya.

Integrasi sistem AI vision pada keseluruhan pipeline kontrol robot humanoid menunjukkan dampak positif pada autonomous behavior. Task completion rate untuk misi "ambil dan taruh objek" meningkat dari 68% (manual control) menjadi 89% (autonomous dengan AI vision), memvalidasi bahwa peningkatan persepsi visual berkontribusi langsung pada peningkatan kapabilitas task execution robot.

4. Kesimpulan

Penelitian ini berhasil mengembangkan dan mengimplementasikan sistem deteksi visual berbasis kecerdasan buatan yang terintegrasi pada robot humanoid menggunakan arsitektur CNN ringan. Perbandingan dua model, MobileNetV2 dan YOLOv5n, menunjukkan bahwa MobileNetV2 dengan akurasi 93.4% dan waktu inferensi 28ms lebih optimal untuk aplikasi real-time pada robot humanoid dibandingkan YOLOv5n yang memiliki akurasi lebih tinggi (96.1%) namun inferensi lebih lambat (39ms).

Hasil pengujian menunjukkan peningkatan signifikan pada akurasi deteksi objek dan stabilitas persepsi dalam berbagai kondisi lingkungan, dengan success rate navigasi meningkat dari 76% menjadi 94% setelah integrasi AI vision. Sistem mampu bekerja secara real-time dengan frame rate 35.7 FPS, melampaui threshold minimum untuk navigasi smooth, sehingga mendukung performa optimal dalam tugas navigasi otonom dan interaksi objek.

Keterbatasan penelitian terletak pada penurunan performa model pada kondisi pencahayaan ekstrem dan partial occlusion, yang mengindikasikan perlunya pengembangan lebih lanjut pada aspek robustness sistem. Penelitian selanjutnya dapat diarahkan pada: (1) integrasi multi-sensor fusion dengan LiDAR atau depth camera untuk meningkatkan akurasi deteksi 3D, (2) implementasi teknik transfer learning dan domain adaptation untuk meningkatkan robustness pada variasi lingkungan, (3) optimasi hardware melalui akselerator inferensi khusus seperti Google Coral TPU atau Intel Neural Compute Stick, serta (4) pengembangan arsitektur hybrid yang mengkombinasikan kecepatan MobileNet dengan akurasi YOLO untuk mencapai performa optimal.

Daftar Rujukan

[1] Redmon J., Divvala S., Girshick R., Farhadi A., 2016. *You Only Look Once: Unified, Real-Time Object Detection*. Proc. CVPR. Las Vegas: USA. doi=10.1109/CVPR.2016.91.
[2] Howard A. G., Zhu M., Chen B., Kalenichenko D., Wang W., Wojna Z., Tobin J., Adam H., Le Q., 2017. *MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications*. arXiv preprint arXiv:1704.04861. doi=10.48550/arXiv.1704.04861.
[3] Sandler M., Howard A., Zhu M., Zhmoginov A., Chen L., 2018. *MobileNetV2: Inverted Residuals and Linear Bottlenecks*. Proc. CVPR. Salt Lake City: USA. doi=10.1109/CVPR.2018.00474.
[4] Bochkovskiy A., Wang C. Y., Liao H. Y. M., 2020. *YOLOv4: Optimal Speed and Accuracy of Object Detection*. arXiv preprint arXiv:2004.10934. doi=10.48550/arXiv.2004.10934.
[5] Lin T. Y., Goyal P., Girshick R., He K., Dollár P., 2017. *Focal Loss for Dense Object Detection*. Proc. ICCV. Venice: Italy. doi=10.1109/ICCV.2017.324.
[6] He K., Zhang X., Ren S., Sun J., 2016. *Deep Residual Learning for Image Recognition*. Proc. CVPR. Las Vegas: USA. doi=10.1109/CVPR.2016.90.

Submitted : 01-10-2025 | Reviewed : 15-10-2025 | Accepted : 20-10-2025

[7] Tan M., Le Q. V., 2019. *EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks*. Proc. ICML. Long Beach: USA. doi=10.48550/arXiv.1905.11946.
[8] Liu W., Anguelov D., Erhan D., Szegedy C., Reed S., Fu C., Berg A., 2016. *SSD: Single Shot MultiBox Detector*. Proc. ECCV. Amsterdam: Netherlands. doi=10.1007/978-3-319-46448-0_2.
[9] Zhang X., Zhou X., Lin M., Sun J., 2018. *ShuffleNet: An Extremely Efficient Convolutional Neural Network for Mobile Devices*. Proc. CVPR. Salt Lake City: USA. doi=10.1109/CVPR.2018.00107.
[10] Sabour S., Frosst N., Hinton G. E., 2017. *Dynamic Routing Between Capsules*. Proc. NeurIPS. Long Beach: USA. doi=10.48550/arXiv.1710.09829.
[11] Huang G., Liu Z., Van Der Maaten L., Weinberger K. Q., 2017. *Densely Connected Convolutional Networks*. Proc. CVPR. Honolulu: USA. doi=10.1109/CVPR.2017.243.
[12] Krizhevsky A., Sutskever I., Hinton G. E., 2012. *ImageNet Classification with Deep Convolutional Neural Networks*. Proc. NeurIPS. Lake Tahoe: USA. doi=10.1145/3065386.
[13] Ren S., He K., Girshick R., Sun J., 2015. *Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks*. Proc. NeurIPS. Montreal: Canada. doi=10.48550/arXiv.1506.01497.
[14] Barry D., Milford M., Boles W., 2019. *xYOLO: A Model for Real-Time Object Detection in Humanoid Soccer on Low-End Hardware*. arXiv preprint arXiv:1910.03159. doi=10.48550/arXiv.1910.03159.
[15] Chatterjee A., Zunjani M., Sen S., Nandi G., 2020. *Real-Time Object Detection and Recognition on Low-Compute Humanoid Robots using Deep Learning*. arXiv preprint arXiv:2002.03735. doi=10.48550/arXiv.2002.03735.

Submitted : 01-10-2025 | Reviewed : 15-10-2025 | Accepted : 20-10-2025