

Deteksi Barang Hilang Berbasis Deteksi Visual dengan Metode You Only Look Once

Mohamad Rafi Adidama¹, Aji Slamet al faid², Muhammad Rizki Fadhil³, Riza Samsinar⁴

^{1,2,3,4} Teknik Elektro, Fakultas Teknik, Universitas Muhammadiyah Jakarta

E-mail: ¹23040250003@student.umj.ac.id ²2018420002@student.umj.ac.id,

³20210410200010@student.umj.ac.id, ⁴riza.samsinar@umj.ac.id*

Abstrak – Kehilangan barang berharga seperti tas laptop dan tas ransel di tempat umum menjadi salah satu permasalahan yang sering dihadapi masyarakat modern. Penelitian ini mengusulkan sistem deteksi tas ransel hilang secara real-time berbasis teknologi pengenalan objek menggunakan YOLO (You Only Look Once). Sistem ini dirancang untuk mendeteksi keberadaan tas ransel dalam jangkauan kamera dan memberikan peringatan apabila tas ransel tidak terdeteksi selama periode tertentu. Model YOLO yang digunakan dilatih dengan dataset COCO, yang mencakup kategori "backpack" sebagai representasi tas ransel. Implementasi sistem dilakukan dengan mengintegrasikan library OpenCV untuk pengolahan video dan deteksi objek. Hasil pengujian menunjukkan bahwa sistem mampu mendeteksi tas ransel dengan akurasi tinggi dalam waktu nyata, sehingga dapat digunakan sebagai solusi inovatif untuk meningkatkan keamanan di tempat umum seperti perkantoran, sekolah, atau transportasi umum. Dengan pendekatan yang efisien ini, sistem ini berpotensi untuk diadaptasi lebih lanjut dalam berbagai aplikasi keamanan lainnya. Selain itu, sistem ini memiliki fleksibilitas untuk diterapkan dalam berbagai lingkungan, baik indoor maupun outdoor, dengan performa yang tetap optimal dalam kondisi pencahayaan yang berbeda. Penggunaan teknologi YOLO memungkinkan proses deteksi berlangsung dengan kecepatan tinggi, sehingga sistem dapat merespons kehilangan tas ransel secara instan tanpa jeda signifikan. Peringatan yang dihasilkan dapat dikonfigurasi untuk dikirim melalui berbagai saluran, seperti notifikasi pada perangkat mobile atau alarm lokal, guna meningkatkan kewaspadaan pengguna maupun petugas keamanan. Dengan fitur ini, sistem tidak hanya mendeteksi kehilangan tas ransel, tetapi juga membantu mencegah potensi pencurian melalui pengawasan proaktif. Hasil penelitian ini diharapkan dapat menjadi landasan bagi pengembangan lebih lanjut dalam bidang teknologi keamanan berbasis visi komputer.

Kata Kunci — Deteksi tas ransel, deteksi objek, YOLO

1. PENDAHULUAN

Setiap tahun banyak sekali barang-barang bernilai miliaran dolar dicuri secara global banyak barang-barang berharga yang biasanya dapat ditemukan pada tempat umum ini merupakan permasalahan yang sering terjadi di kehidupan sehari-hari, terutama di lokasi dengan tingkat mobilitas tinggi seperti stasiun, terminal, perkantoran, dan institusi pendidikan. Salah satu barang yang sering dibawa oleh masyarakat modern adalah tas ransel, yang berfungsi sebagai penyimpanan barang penting, termasuk perangkat elektronik, dokumen, dan barang pribadi lainnya. Kehilangan tas ransel tidak hanya menimbulkan kerugian finansial, tetapi juga dapat menyebabkan hilangnya dokumen atau data penting yang memiliki dampak signifikan bagi kehidupan pribadi maupun profesional [1].

Teknologi yang semakin berkembang, dan adanya pengaruh sistem dalam pengawasan cerdas yang memerlukan manajemen dan integrasi pengetahuan perkembangan teknologi, solusi berbasis kecerdasan buatan (Artificial Intelligence, AI) dan visi komputer (Computer Vision) menciptakan sistem keamanan yang lebih cerdas dan efisien. Teknologi deteksi objek berbasis real-time menjadi salah satu pendekatan yang sangat menjanjikan, di mana objek tersebut dapat dikenali dan dapat dilacak secara langsung melalui kamera. Teknologi ini memberikan peluang untuk meningkatkan sistem keamanan dengan cara mendeteksi keberadaan tas ransel, memberikan peringatan jika tas tersebut hilang dari jangkauan kamera, serta membantu mencegah insiden kehilangan barang [2].

YOLO (You Only Look Once) telah menjadi salah satu metode deteksi objek yang sangat populer dalam beberapa tahun terakhir, karena keunggulannya dalam mendeteksi objek secara real-time dengan tingkat akurasinya sangat tinggi. Metode untuk mengembangkan model ini bersifat kualitatif dan menggunakan konvolusional untuk deteksi gambar dan YOLO itu sendiri digunakan pendekatan deteksi satu tahap (single-shot detection). Di mana pada proses klasifikasi dan lokalisasi objek dilakukan secara simultan, sehingga sangat efisien untuk aplikasi yang memerlukan kecepatan tinggi. Dengan menggunakan model yang telah dilatih pada dataset publik seperti COCO, sistem ini mampu mengenali berbagai jenis objek, termasuk tas ransel, yang relevan dengan studi ini [1].

Penelitian ini bertujuan untuk merancang dan mengimplementasikan sistem deteksi tas ransel hilang secara real-time menggunakan metode YOLO. Sistem yang diusulkan dirancang untuk mendeteksi keberadaan tas ransel

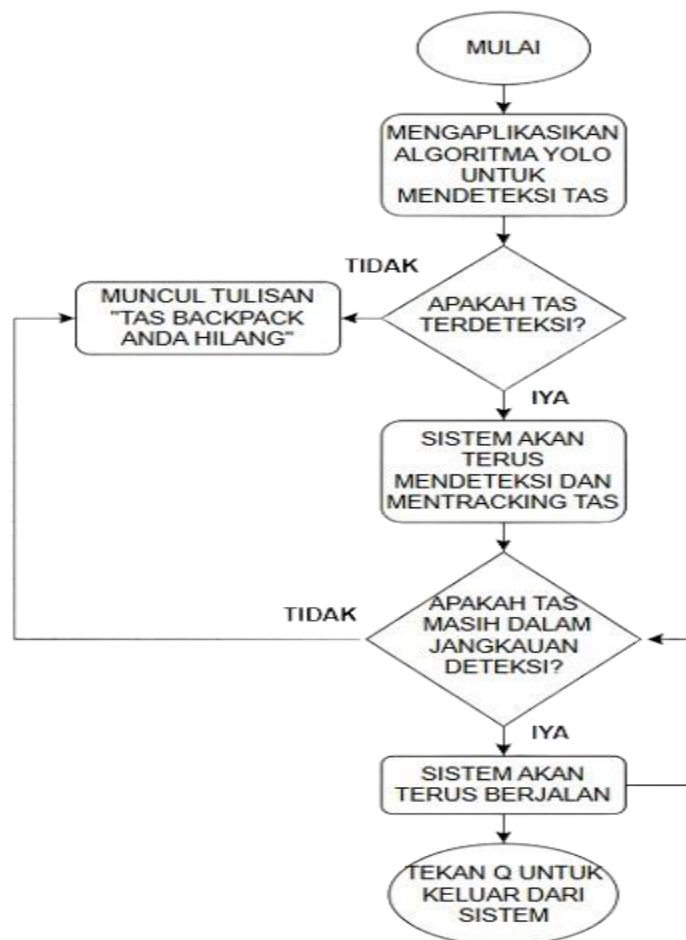
dalam jangkauan kamera dan memberikan notifikasi secara visual jika tas tidak terdeteksi dalam waktu tertentu. Sistem ini juga diintegrasikan dengan antarmuka visual yang sederhana, sehingga dapat digunakan secara praktis oleh pengguna di berbagai lingkungan. Fokus penelitian ini adalah pada efisiensi dan keandalan sistem untuk menjawab kebutuhan keamanan yang terus meningkat [2].

Dengan studi ini, dan penelitian hal ini sebagai cara yang dapat diskalakan untuk memungkinkan deteksi yang efisien dari sejumlah besar kelas objek diharapkan dapat memberikan kontribusi yang signifikan dalam pengembangan model sistem keamanan berbasis deteksi objek. Selain itu, pada penelitian ini dapat membuka peluang untuk pengembangan lebih lanjut, seperti integrasi dengan sistem berbasis IoT (Internet of Things) atau aplikasi dalam pengawasan skala besar, seperti pengawasan di kota pintar (smart city). Implementasi sistem ini bertujuan untuk menciptakan lingkungan yang lebih aman, nyaman, dan mampu mencegah kehilangan barang dengan menggunakan teknologi modern secara efektif [3].

Penelitian ini diharapkan dapat memberikan kontribusi signifikan dalam meminimalisir tindak kejahatan kriminal khususnya kasus pencurian dalam area yang tidak sepenuhnya dapat terawasi oleh manusia..

2. METODE PENELITIAN

2.1 Alur Proses



Gambar 1. Alur Proses

Flowchart ini menggambarkan sebuah sistem deteksi dan *tracking* tas ransel yang menggunakan algoritma YOLO sebagai metode utamanya. Sistem dimulai dengan mengaplikasikan algoritma YOLO untuk mendeteksi keberadaan tas ransel. Setelah proses deteksi awal, sistem akan melakukan pengecekan apakah tas ransel berhasil terdeteksi atau tidak. Dalam hal tas ransel tidak terdeteksi, sistem akan menampilkan pesan peringatan bahwa tas *backpack* anda telah hilang.

Jika tas ransel berhasil terdeteksi, sistem akan melanjutkan ke tahap berikutnya yaitu proses deteksi dan *tracking* tas ransel secara berkelanjutan. Pada tahap ini, sistem secara terus menerus memantau apakah tas ransel masih berada dalam jangkauan deteksi. Selama tas ransel masih terdeteksi dalam jangkauan, sistem akan terus beroperasi dalam sebuah *Loop* yang mengulang proses deteksi dan *tracking*. *Loop* ini merupakan bagian integral dari sistem yang memastikan pemantauan berkelanjutan[10].

Sistem ini dirancang dengan fleksibilitas untuk dapat dihentikan secara manual oleh pengguna dengan menekan tombol Q. Desain *Flowchart* ini menunjukkan bahwa sistem dioptimalkan untuk penggunaan *real-time*

dan memiliki mekanisme yang jelas untuk penanganan kasus ketika tas ransel hilang dari jangkauan atau tidak terdeteksi.

Dataset

Deteksi dengan menggunakan dataset COCO (*Common Objects in Context*) yang berisi berbagai kategori objek untuk menguji kedua metode. Dataset ini mencakup gambar dengan berbagai kondisi pencahayaan dan latar belakang.

Pre Trained YOLO Model

Langkah pertama adalah memilih model YOLO yang telah dilatih sebelumnya. Model YOLO yang telah dilatih sebelumnya dan pembelajaran mendalam memiliki karakteristik yang sama dalam hal kumpulan data pelatihan, yang biasanya besar dan kompleks. Model YOLO yang telah dilatih sebelumnya dengan jaringan seri memiliki arsitektur lapisan berurutan dengan lebih sedikit lapisan tetapi sejumlah besar parameter yang dapat dilatih [4].

Tabel 1. Model Pre-Trained YOLO

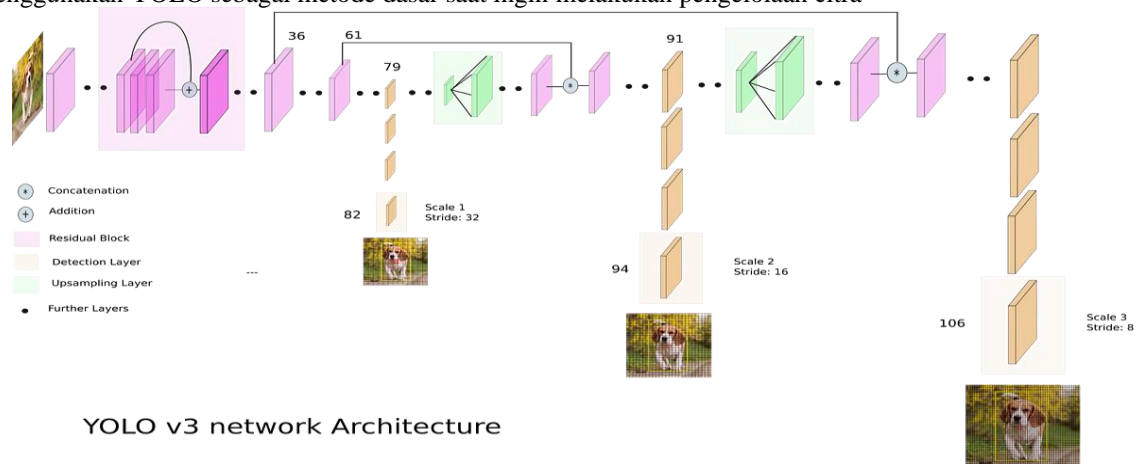
Network	Deep Layers	Total Layers	Layers Connection
Series			
AlexNet	8	25	25
VGG16	16	41	41
VGG19	19	47	47
DAG			
InceptionV3	22	144	170
GoogleNet	42	316	350
ResNet-18	18	72	79
ResNet-50	50	177	192
ResNet-101	101	347	379

Konstruksi lapisan grafik lebih rumit karena variabel jaringan tidak terbatas pada menghubungkan lapisan satu demi satu. Lapisan tersebut dapat memiliki banyak variabel sebagai koneksi *input* dan *output*. Dalam studi ini, kedua jenis jaringan pra-latihan digunakan untuk melatih kumpulan data normal dan abnormal. Tiga jaringan Seri pra-latihan, yaitu Alex Net, VGG16, dan VGG19, dibangun kembali, sementara lima jaringan pra-latihan, yaitu Google Net, Inception-v3, ResNet-18, ResNet-50, dan ResNet-101, didesain ulang sebagai jaringan DAG. Pendekatan pembelajaran transfer menghasilkan hasil yang sebanding untuk semua jaringan pra-latihan.

2.2 Implementasi YOLO

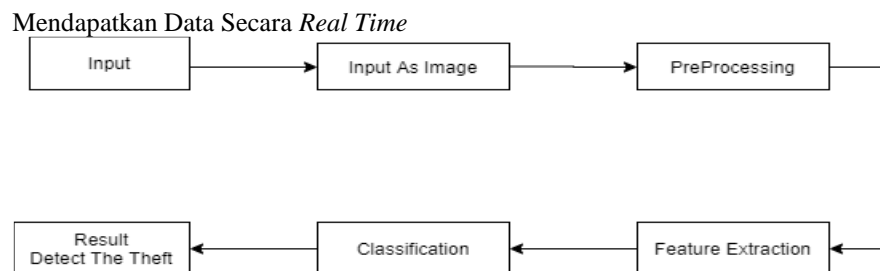
Pembelajaran mendalam telah menunjukkan bahwa jaringan saraf dalam mengungguli metode tradisional secara signifikan. Misalnya, algoritma YOLO digunakan dalam aplikasi Big Data waktu nyata. Ada banyak skenario di mana data perlu dianalisis secara waktu nyata, karena menganalisis data secara *offline* tidak akan menghasilkan hasil atau menyebabkan kerugian yang signifikan. Dalam kebanyakan kasus, skenario ini menangani sejumlah besar data yang berubah dengan cepat dalam konteks yang kompleks. Algoritma pembelajaran mendalam ideal untuk tujuan ini karena skalabilitasnya, terutama untuk deteksi waktu nyata [5].

YOLO merupakan salah satu algoritma dalam *deep learning* yang banyak digunakan untuk menyelesaikan permasalahan yang berkaitan dengan *Computer Vision*. Sejauh ini YOLO telah menunjukkan performa dan hasil yang baik dalam pengolahan data berupa *image* atau citra pada penelitian, sehingga banyak peneliti yang menggunakan YOLO sebagai metode dasar saat ingin melakukan pengelolaan citra



Gambar 2. YOLO V3 Network Architecture

Algoritma YOLO tersusun dari beberapa blok penyusun atau lapisan (*layer*) yang terdiri dari lapisan konvolusional, lapisan penggabungan, dan lapisan yang terhubung sepenuhnya. Pada tahapan awal, gambar yang di-*input* akan mengalami ekstraksi fitur pada dua lapisan pertama yakni *convolutional* dan *pooling layer*. Selanjutnya, pada lapisan ketiga (lapisan yang terhubung sepenuhnya), hasil dari fitur yang sudah diekstraksi akan dipetakan (peta) menjadi keluaran akhir [6].



Gambar 2. Diagram Alir

Implementasi sistem deteksi *real-time* menggunakan Python, OpenCV dengan dukungan CUDA, dan YOLO melibatkan berbagai langkah yang terintegrasi dengan perangkat keras dan perangkat lunak. Langkah pertama adalah instalasi perangkat lunak, termasuk Python, OpenCV, dan dependensi YOLO di lingkungan pengembangan seperti PyCharm. OpenCV menyediakan pustaka pengolahan gambar yang powerful, sementara YOLO (*You Only Look Once*) adalah algoritma deteksi objek yang cepat dan akurat. YOLO termasuk *Bounding Box*, suatu teknik yang digunakan untuk menyempurnakan lokasi objek yang terdeteksi. Ini membantu secara akurat melokalisasi objek dalam gambar [7].

Langkah kedua adalah konfigurasi YOLO, di mana *file-file* penting seperti *yolov4.cfg*, *yolov4.weights*, dan *coco.names* diatur untuk digunakan dalam mendeteksi objek, seperti tas ransel dalam aplikasi. CUDA, teknologi paralel computing dari Nvidia, memungkinkan GPU GeForce RTX 3050 untuk memproses *frame* video secara cepat dibandingkan hanya menggunakan CPU. Dengan menggunakan OpenCV, setiap *frame* yang diambil dari webcam diunggah ke GPU menggunakan *cv2.cuda_GpuMat()* untuk diproses, seperti konversi ke *grayscale* atau pengolahan lainnya. Hasilnya kemudian diunduh kembali ke CPU untuk ditampilkan secara *real-time*. Kombinasi CPU Intel i5-12450H yang andal dan GPU yang mendukung CUDA memberikan performa optimal untuk aplikasi pengolahan gambar [8].

Setelah konfigurasi selesai, langkah terakhir adalah integrasi YOLO dengan OpenCV. Kode Python ditulis untuk memproses gambar atau video secara *real-time* dengan menangkap *frame* dari webcam. *Frame-frame* ini diproses menggunakan YOLO untuk mendeteksi objek target, seperti tas ransel, dan hasilnya divisualisasikan dalam bentuk kotak deteksi dan label yang muncul pada *frame* video. CUDA berperan penting dalam mempercepat pengolahan data besar dengan memindahkan sebagian proses ke GPU, sehingga memungkinkan deteksi *real-time* yang responsif. Sistem ini mendemonstrasikan efisiensi pemrosesan video, mulai dari instalasi perangkat lunak hingga deteksi dan visualisasi yang optimal dengan memanfaatkan kekuatan perangkat keras modern [9].

3. HASIL DAN PEMBAHASAN

Untuk percobaan ini kita menggunakan berbagai variasi Tas dengan warna gelap dan cerah dengan kondisi ruangan yang terang.

3.1 Hasil Pengujian terhadap 4 Tas dan 1 kondisi tidak ada Tas

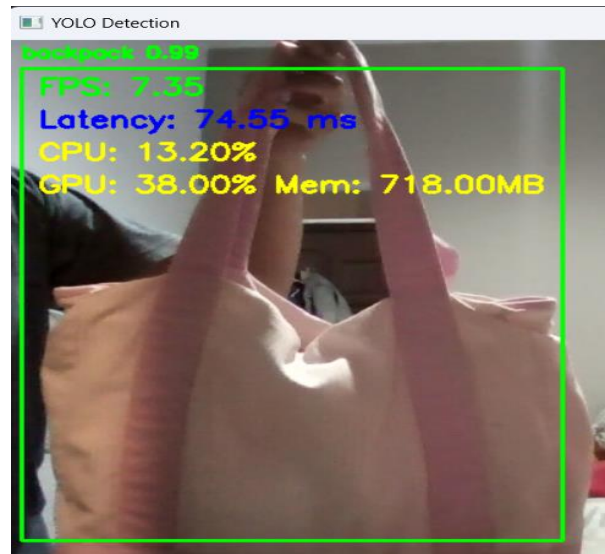
Untuk gambar dari hasil percobaan ini, kami mengambil tangkapan layar dari output ketika program sedang berjalan yaitu berupa jendela dengan tampilan antarmuka webcam yang dilengkapi dengan bounding box apabila program berhasil mendeteksi Tas disertai dengan keterangan indikator pada layar jendela webcam tersebut seperti keterangan FPS, Latency, CPU, GPU, dan Memory



Gambar 4. Hasil pengujian Tas Pertama

Untuk kondisi normal program dapat mendeteksi tas dengan baik. Terlihat dari *confidence score* yang menjadi tingkat keyakinan untuk mendeteksi objek relatif tinggi yaitu 0.96. dengan skala lebih mendekati angka 1 lebih baik. Dari nilai tersebut menunjukkan bahwa metode YOLO sudah cukup baik untuk mendeteksi objek secara akurat.

3.2 Pengujian Tas Kedua



Gambar 5. Hasil pengujian Tas Kedua

Untuk uji coba kedua ini, kami menggunakan tas yang memiliki warna hampir identik dengan tas pada pengujian pertama, tetapi hanya berbeda jenisnya. kali ini kami melakukan uji coba dengan tas selempang Wanita dengan ukuran sedang. Walaupun jenisnya berbeda dengan tas pertama, dan juga bukan merupakan tas ransel, Tetapi program masih dapat mendeteksi objek tersebut sebagai “Tas” atau “Backpack” dengan Tingkat keyakinan atau *confidence score* mendekati nilai sempurna yaitu 0.99.

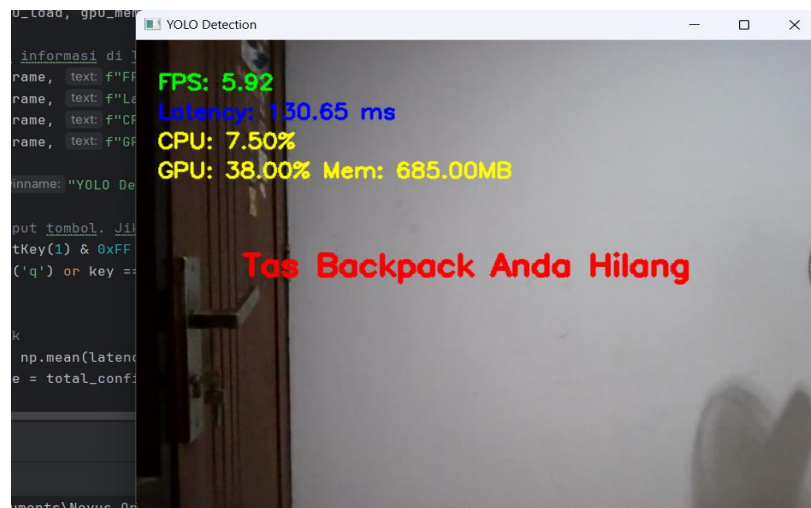
3.3 Pengujian Tas Keempat



Gambar 6. Hasil pengujian Tas Keempat

Pada pengujian yang keempat ini, kami menggunakan tas koper. Yang mana, untuk mendeteksi objek koper ini, terdapat sedikit kesulitan atau kendala pada saat proses pendeteksian. Objek koper sedikit sulit terdeteksi oleh program sehingga penulis harus merubah posisi objek koper sedikit lebih dekat terhadap *Webcam* sampai objek tersebut dapat terdeteksi dengan baik oleh program. Hal ini kemungkinan dipengaruhi oleh ukuran koper yang relatif lebih besar dibandingkan dengan ukuran tas yang lainnya. Untuk *confidence score* pada deteksi koper ini didapatkan hasil yang cukup tinggi yaitu 0.91. yang berarti walaupun ukuranya besar tetapi program masih mampu mendeteksinya dengan baik.

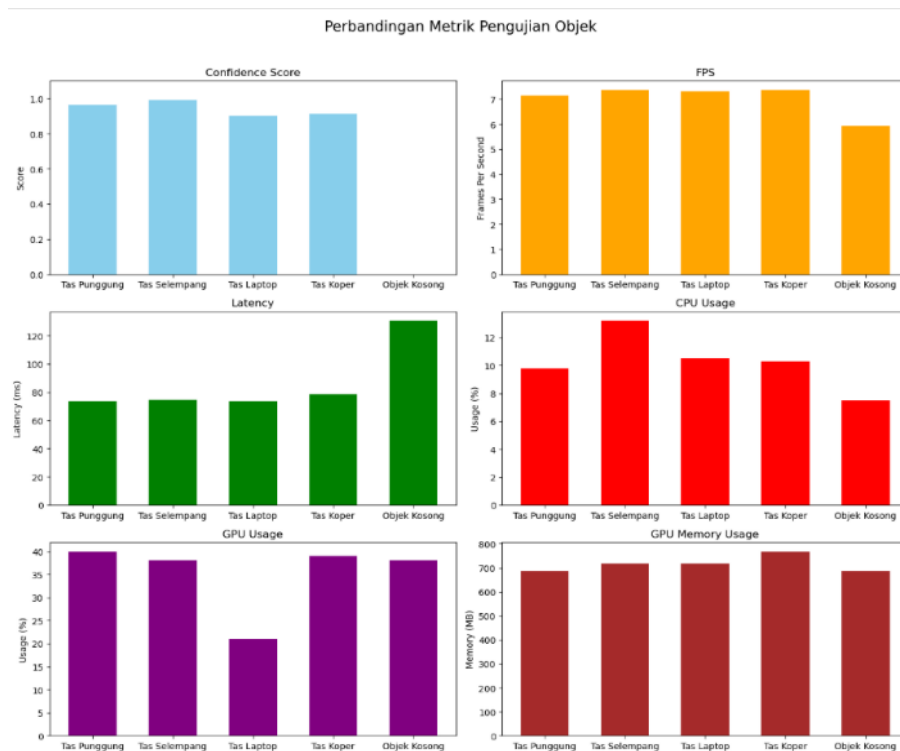
3.4 Pengujian terhadap kondisi tidak ada objek Tas yang terdeteksi



Gambar 7. Hasil pengujian kondisi tidak ada Objek Tas yang terdeteksi

Seperti terlihat pada gambar apabila program tidak dapat menemukan objek apa apa untuk di deteksi dalam kasus di sini adalah tas ransel, maka akan muncul pemberitahuan "Tas Backpack Anda Hilang" pada layar jendela Webcam.

3.5 Grafik Perbandingan Hasil Pengujian



Gambar 8. Gambar Grafik Perbandingan Hasil Pengujian

Berdasarkan data yang diberikan, sistem deteksi objek menunjukkan performa yang baik dalam mendeteksi tas dengan confidence score tinggi, terutama pada Tas Selempang (0.99) dan Tas Punggung (0.96). Namun, deteksi Tas Laptop memiliki confidence score terendah (0.90), menunjukkan bahwa model mungkin perlu perbaikan untuk mengidentifikasi fitur visual dari objek tersebut. Latency relatif konsisten di bawah 80 ms untuk semua tas, kecuali dalam kasus Objek Kosong, di mana latency melonjak hingga 130.65 ms. Hal ini mengindikasikan bahwa sistem membutuhkan waktu lebih lama untuk memverifikasi ketidakhadiran objek, yang dapat dioptimalkan untuk meningkatkan efisiensi.

Dalam hal sumber daya, penggunaan GPU dan memori cenderung lebih tinggi untuk objek dengan fitur kompleks seperti Tas Koper (GPU Memory Usage mencapai 768 MB). Namun, penggunaan CPU dan GPU relatif stabil untuk sebagian besar objek, kecuali pada Tas Laptop, yang memiliki GPU Usage jauh lebih rendah (21%). FPS yang menurun pada Objek Kosong (5.92) menunjukkan bahwa kinerja sistem lebih efisien saat objek terdeteksi.

dibandingkan saat objek tidak ada. Secara keseluruhan, sistem cukup andal, tetapi dapat dioptimalkan untuk meningkatkan performa dalam skenario tanpa objek serta mendeteksi objek dengan fitur visual yang kurang mencolok seperti Tas Laptop.

4. SIMPULAN

Berdasarkan penelitian yang telah dilaksanakan maka diperoleh kesimpulan sebagai berikut:

- Sistem mendeteksi objek dengan confidence score yang tinggi, terutama untuk Tas Selempang (0.99) dan Tas Punggung (0.96). Namun, performa sedikit menurun pada Tas Laptop (0.90), menunjukkan kebutuhan untuk peningkatan model pada jenis tas tersebut.
- Latency rata-rata cukup rendah (<80 ms) saat objek terdeteksi, namun meningkat signifikan pada kasus Objek Kosong (130.65 ms). Ini mengindikasikan bahwa proses verifikasi ketidakhadiran objek memerlukan optimisasi.
- OpenCV GPU Usage dan GPU Memory Usage bervariasi tergantung pada kompleksitas fitur objek. Tas Koper menunjukkan konsumsi memori GPU tertinggi (768 MB), sedangkan Tas Laptop memiliki GPU Usage terendah (21%), yang mungkin terkait dengan pola fitur visualnya.
- FPS relatif stabil untuk semua objek (7.14–7.36), kecuali pada Objek Kosong (5.92), yang menandakan bahwa kinerja sistem lebih efisien saat mendeteksi objek dibandingkan saat tidak ada objek.
- Sistem dapat ditingkatkan untuk menangani skenario tanpa objek dengan lebih efisien dan memperbaiki deteksi objek dengan fitur yang kurang mencolok, seperti Tas Laptop, untuk meningkatkan keandalan secara keseluruhan.

5. SARAN

Berikut adalah beberapa saran untuk penelitian berikutnya berdasarkan simulasi yang sudah dilakukan:

- Menambahkan jumlah dataset dan anotasi pada setiap jenis kelas untuk meningkatkan tingkat mAP saat training.
- Perlu dikembangkan sebuah metode khusus agar mampu mengatasi durasi proses training yang cukup lama.
- Meningkatkan kecepatan pendeteksian objek dalam pengembangan autonomous vehicle sebagai upaya mengurangi angka kecelakaan lalu lintas.

DAFTAR PUSTAKA

- [1] D. Trung Pham and F. Lukowski, *BobDontRob: Hybrid neural network for shoplifting detection: Detecting theft from camera footage in grocery retail stores*, 2024.
- [2] A. V. Rusli, *Sistem deteksi driver drowsiness menggunakan algoritma convolutional neural network (CNN)*, Doctoral dissertation, Politeknik Negeri Ujung Pandang, 2022.
- [3] D. Erhan, C. Szegedy, A. Toshev, and D. Anguelov, "Scalable object detection using deep neural networks," *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pp. 2147–2154, 2014.
- [4] N. F. M. Zamri, N. M. Tahir, M. S. M. Ali, N. D. K. Ashar, and A. Abd Almisreb, "Real-time snatch theft detection using deep learning networks," *J. Adv. Res. Appl. Sci. Eng. Technol.*, vol. 31, no. 1, pp. 79–89, 2023.
- [5] M. Rafly, "Rancang bangun sistem pendeteksi dan penghitung jumlah pengunjung di Museum Lampung menggunakan webcam berbasis algoritma YOLOv5 (You Only Look Once Versi Lima)," 2023.
- [6] F. Loekman, "Sistem manajemen inventori dengan pengenalan barang secara otomatis menggunakan metode convolutional neural network," *Teknika*, vol. 12, no. 1, pp. 47–56, 2023.
- [7] S. Shirole, "Theft detection using deep learning," 2023.
- [8] M. K. Ramadan, A. A. Youssif, and W. H. El-Behaidy, "Detection and classification of human-carrying baggage using DenseNet-161 and Fit One Cycle," *Big Data Cogn. Comput.*, vol. 6, no. 4, p. 108, 2022.
- [9] Mrs. Smita, A. Takalkar, A. Awale, and J. Yewale, "System to detect theft events using Raspberry PI," pp. 868–874, 2022.
- [10] I. Muneer, M. Saddique, Z. Habib, and H. G. Mohamed, "Shoplifting detection using hybrid neural network CNN-BiLSTM and development of benchmark dataset," *Appl. Sci.*, vol. 13, no. 14, p. 8341, 2023.