

Pengembangan Real-Time Object Detection System pada Perangkat Single-Board Computer

Riyadhul Fajri*, Sriwinar, Firza Fitria

Fakultas Ilmu Komputer, Program Studi Informatika, Universitas Almuslim, Bireuen, Indonesia

Email: ¹*riyadhul1812@gmail.com, ²sriwinarafifa@gmail.com, ³f.fitria.f@gmail.com

Email Penulis Korespondensi: riyadhul1812@gmail.com

Abstrak—Perkembangan teknologi di bidang kecerdasan buatan telah membuka pintu bagi aplikasi yang semakin canggih dalam analisis visual dan deteksi objek. Penelitian ini bertujuan untuk mengimplementasikan sistem real-time object detection menggunakan perangkat Raspberry Pi, sebuah perangkat berbasis Single Board Computer (SBC) yang berdaya rendah dengan sumber daya perangkat keras terbatas. Kami menggunakan TensorFlow Lite, versi ringan dari framework deep learning TensorFlow, untuk menjalankan model deteksi objek di Raspberry Pi. Sistem kami didukung oleh USB web cam sebagai sumber gambar. Hasil penelitian menunjukkan bahwa Raspberry Pi mampu menjalankan deteksi objek secara real-time dengan baik, dengan efisiensi penggunaan sumber daya yang memadai. Sistem yang dibangun mampu melakukan klasifikasi yang lebih baik, yang menunjukkan kecepatan deteksi yang lebih baik dari penggunaan TensorFlow Lite, dibandingkan dengan penggunaan TensorFlow. Meskipun ada kendala dalam hal kecepatan deteksi tergantung pada kompleksitas model dan jumlah objek, hasil ini membuka peluang besar untuk penggunaan perangkat embedded dalam berbagai aplikasi seperti pemantauan keamanan dan analisis gambar. Studi ini juga menekankan pentingnya optimasi model untuk mencapai performa terbaik pada perangkat berdaya rendah.

Kata Kunci: Real-Time Object Detection; Raspberry Pi; TensorFlow Lite; Embedded Systems; Computer Vision.

Abstract—Technological developments in the field of artificial intelligence have opened the door to increasingly sophisticated applications in visual analysis and object detection. This research aims to implement a real-time object detection system using a Raspberry Pi device, a low-power Single Board Computer (SBC) based device with limited hardware resources. We used TensorFlow Lite, a lightweight version of the TensorFlow deep learning framework, to run object detection models on a Raspberry Pi. Our system is powered by a USB web cam as an image source. The research results show that the Raspberry Pi is able to carry out real-time object detection well, with adequate resource usage efficiency. The system built is able to perform better classification, which shows a better detection speed from using TensorFlow Lite, compared to using TensorFlow. Although there are constraints in terms of detection speed depending on the complexity of the model and the number of objects, these results open up great opportunities for the use of embedded devices in various applications such as security monitoring and image analysis. This study also emphasizes the importance of model optimization to achieve the best performance on low-power devices.

Keywords: Real-Time Object Detection; Raspberry Pi; TensorFlow Lite; Embedded Systems; Computer Vision.

1. PENDAHULUAN

Seiring dengan perjalanan waktu, perkembangan teknologi komputer telah mengalami transformasi luar biasa, membawa kita dari era awal komputer besar dan berat hingga era perangkat komputasi ringkas dan kuat seperti Raspberry Pi [1]. Perjalanan ini dimulai dari mesin komputer kamar yang membutuhkan ruangan besar dan konsumsi daya yang tinggi, hingga mencapai titik di mana kecerdasan komputasi yang dahulu hanya bisa diakses oleh beberapa ahli teknologi, kini telah menjadi lebih terjangkau dan mudah digunakan oleh berbagai lapisan masyarakat. Pentingnya komputer dalam berbagai aspek kehidupan modern telah memicu inovasi berkelanjutan dalam bidang ini. Kemajuan semakin cepat dengan munculnya mikroprosesor yang semakin kecil dan kuat, memungkinkan kita untuk membawa kekuatan komputasi dalam genggaman tangan kita [2], [3]. Namun, transformasi yang paling menarik adalah lahirnya perangkat single board computer (SBC) seperti Raspberry Pi [4]. Raspberry Pi, dengan ukurannya yang kecil sebesar kartu kredit, dan kemampuan yang mengesankan, telah membuka pintu bagi para pengembang, pembelajar, dan inovator untuk mengakses teknologi komputer dan mengimplementasikannya dalam berbagai proyek dan aplikasi yang tidak pernah terpikirkan sebelumnya [5]–[7].

Beberapa penelitian telah dilakukan untuk mengembangkan real-time object detection, diantaranya adalah pengembangan real-time object detection pada perangkat embedded device menggunakan YOLOv4 [8], [9], real time object detection menggunakan Deep SCNN [10], real time object detection yang diaplikasikan pada perangkat UAV [11], pengembangan model real time object detection berbasis CNN yang dinamakan WAVE [12], pengembangan Tiny YOLOv3 untuk real time object detection [13] dan pengembangan TF-YOLO (Tiny Fast – YOLO) untuk real time object detection yang memiliki kecepatan tinggi dalam hal klasifikasi dan deteksi object citra secara real time [14].

Penggunaan perangkat Single Board Computer (SBC) seperti Raspberry Pi 3 telah menjadi semakin umum dalam berbagai aplikasi, mulai dari proyek hobi hingga implementasi industri [15]–[21]. Keberhasilan dan fleksibilitas Raspberry Pi 3 dalam menjalankan berbagai tugas telah memicu minat untuk mengaplikasikan teknologi pengenalan objek secara real-time pada perangkat ini [22]. Implementasi Real-Time Object Detection pada Raspberry Pi 3 memiliki potensi besar untuk berbagai aplikasi seperti pemantauan keamanan, deteksi objek dalam lingkungan industri, dan sistem navigasi otonom pada kendaraan [23], [24].

Namun, Raspberry Pi 3 memiliki keterbatasan dalam hal daya komputasi dan sumber daya perangkat keras dibandingkan dengan komputer yang lebih canggih [25][26]. Oleh karena itu, implementasi real-time object detection pada perangkat ini memerlukan optimisasi yang cermat agar tetap mampu menjalankan tugas tersebut dengan kinerja

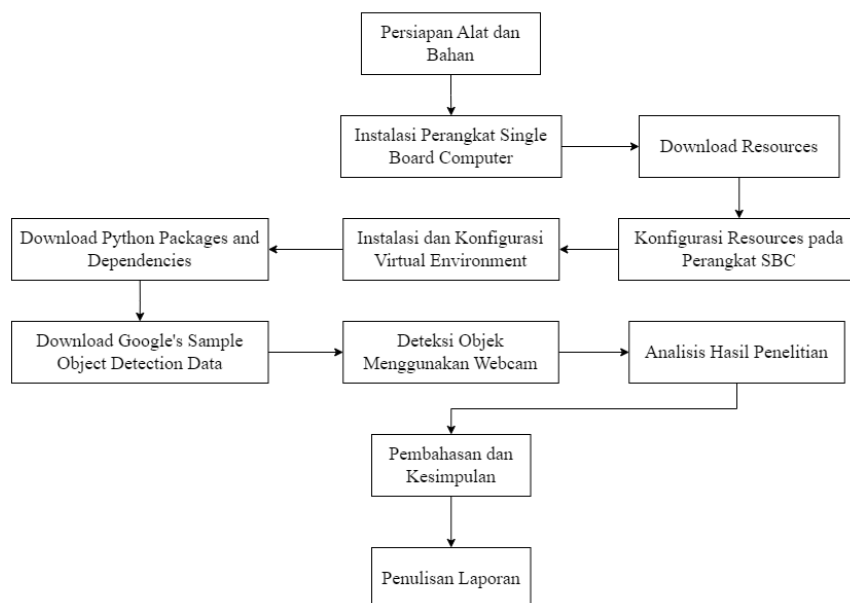
yang memadai. Masalah utama dalam penelitian ini adalah bagaimana merancang dan mengimplementasikan sistem deteksi objek yang efisien dan akurat pada Raspberry Pi 3 dengan mempertimbangkan keterbatasan perangkat keras [27]–[29].

Berdasarkan uraian tersebut, penelitian ini bertujuan untuk mengatasi tantangan teknis dalam mengembangkan sistem real-time object detection yang optimal pada perangkat Raspberry Pi 3. Disamping itu, penelitian ini juga bertujuan untuk melakukan eksplorasi terkait proses tuning daripada model real time object detection sehingga menghasilkan sistem real time object detection yang memiliki kinerja optimal pada perangkat single board computer. Hasil penelitian ini akan memberikan wawasan tentang kemungkinan dan batasan dalam mengintegrasikan teknologi deteksi objek dengan perangkat SBC yang memiliki sumber daya terbatas, serta memberikan kontribusi pada perkembangan solusi aplikatif yang dapat diimplementasikan dalam berbagai konteks praktis.

2. METODOLOGI PENELITIAN

2.1 Tahapan Penelitian

Penelitian ini dilakukan dengan tujuan untuk membangun model object detection pada perangkat Single Board Computer. Adapun alur penelitian yang dilakukan dapat dilihat pada gambar berikut ini:



Gambar 1. Tahapan Penelitian

- a. Persiapan alat dan bahan, dimana pada tahap ini dilakukan persiapan alat dan bahan sebagai berikut:
 1. Perangkat keras, yang terdiri dari Raspberry Pi 3 Model B+, Raspberry Pi Case, USB Webcam Camera, USB Wireless Mouse, USB Wireless Keyboard, LCD Monitor, Power Adaptor untuk Raspberry Pi, dan Laptop untuk mendukung kebutuhan penelitian, mulai pengumpulan informasi hingga penulisan laporan dan publikasi.
 2. Perangkat Lunak, adapun perangkat lunak yang digunakan dalam penelitian ini yaitu Windows 11, Notepad, TensorFlow Lite, Raspbian Stretch, Chromium sebagai browser pada perangkat Raspberry Pi, Shell Terminal sebagai command prompt application pada OS Raspbian, Github dan Python sebagai bahasa pemrograman yang digunakan.
- b. Instalasi perangkat Single Board Computer, dimana pada tahap ini dilakukan instalasi Raspbian Stretch pada perangkat Raspberry Pi serta konfigurasi awal perangkat SBC Raspberry Pi yang meliputi perubahan kata sandi default, konfigurasi jaringan, update sistem, konfigurasi bahasa dan wilayah, ekspansi partisi, aktivasi SSH, Konfigurasi tanggal dan waktu, serta pengaturan layar. Pada tahap ini juga dilakukan konfigurasi kamera melalui menu “Raspberry Pi Configuration – Interface – Camera – Enable”.
- c. Download Model Resources, pada tahap ini dilakukan pengunduhan model object detection menggunakan perintah “git clone linkresources”
- d. Konfigurasi Resources, dimana pada tahap ini dilakukan konfigurasi file resources yang telah didownload pada perangkat SBC Raspberry Pi dengan melakukan ekstraksi file resources tersebut.
- e. Instalasi dan Konfigurasi Virtual Environment, dimana pada tahap ini dilakukan instalasi Virtual Environment khusus untuk object detection menggunakan perintah “sudo pip3 install virtual environment”. Pada tahap ini juga dilakukan konfigurasi terhadap virtual environment yang digunakan yaitu:
 1. Create New Environment, dimana langkah ini dilakukan untuk membuat environment baru menggunakan perintah “python3 -m venv rtod-env”.

2. Masuk kedalam environment yang telah dibuat menggunakan perintah “source rtod-env/bin/activate”.
- f. Download Python Packages and Dependencies, dimana pada tahap ini beberapa package pada python diunduh agar model yang dibangun dapat berjalan dengan baik. Adapun pada tahap ini diunduh file resource sebesar 400MB dan Library TensorFlow Lite.
- g. Download Model Object Detection, dimana pada tahap ini dilakukan download model object detection yang bersumber dari Google’s Sample Model menggunakan perintah “wget linkresources”. Pada tahap ini juga dilakukan ekstraksi data menggunakan perintah “unzip namafile”. Adapun pada model yang di download, terdapat 80 object yang dapat di deteksi sebagai berikut:

```
person, bicycle, car, motorcycle, airplane, bus, train, truck,
boat, traffic light, fire hydrant, stop sign, parking meter,
bench, bird, cat, dog, horse, sheep, cow, elephant, bear,
zebra, giraffe, backpack, umbrella, handbag, tie, suitcase,
frisbee, skis, snowboard, sports ball, kite, baseball bat,
baseball glove, skateboard, surfboard, tennis racket, bottle,
wine glass, cup, fork, knife, spoon, bowl, banana, apple,
sandwich, orange, broccoli, carrot, hot dog, pizza, donut,
cake, chair, couch, potted plant, bed, dining table, toilet,
tv, laptop, mouse, remote, keyboard, cell phone, microwave,
oven, toaster, sink, refrigerator, book, clock, vase, scissors,
teddy bear, hair drier, toothbrush|
```

Gambar 2. Objek yang dideteksi pada Model

- h. Deteksi Objek, dimana pada tahap ini digunakan code python untuk mengaktifkan camera dan mendeteksi objek berdasarkan model yang dibangun, yaitu menggunakan model Google Sample MobileNet.
- i. Analisis Hasil Penelitian, dimana pada tahap ini dilakukan analisis hasil penelitian yang telah dilakukan
- j. Pembahasan dan Kesimpulan, dimana pada tahap ini dilakukan pembahasan terhadap hasil yang didapatkan serta ditarik beberapa kesimpulan yang menyeluruh terhadap proses penelitian dari awal hingga akhir.
- k. Penulisan Laporan, pada tahap ini dilakukan penulisan laporan penelitian serta publikasi hasil penelitian pada artikel jurnal ilmiah.

2.2 Alat dan Bahan

Adapun alat dan bahan yang digunakan dalam penelitian ini yaitu meliputi perangkat keras dan perangkat lunak sebagai berikut:

- a. Perangkat Keras:
1. Raspberry Pi 3 Model B+, sebagai perangkat Single Board Computer yang digunakan dalam penelitian ini, dengan spesifikasi BroadCom 64-bit SoC @1,4GHz, 1 GB LPDDR2 SDRAM, 2,4 GHz dan 5 GHz IEEE 802,11 b/g/n wireless LAN, Bluetooth 4,2 BLE, 40-pin Extended GPIO Header, Full Size HDMI dan 4 USB 2 Ports.
 2. Raspberry Pi Case, sebagai pelindung perangkat Single Board Computer Raspberry Pi.
 3. USB Webcam Camera, sebagai perangkat image / video capture, dimana perangkat Webcam yang digunakan memiliki resolusi maksimal 720p/30fps, 0,9 mega pixel, fixed focus, dan dFoV sebesar 55°.
 4. USB Wireless Mouse
 5. USB Wireless Keyboard
 6. Micro SD Card, sebagai media penyimpanan data untuk perangkat Raspberry Pi.
 7. LCD Monitor sebagai output display perangkat SBC Raspberry Pi.
 8. Power Adaptor 5 Volt untuk Raspberry Pi
 9. Laptop untuk mendukung kebutuhan penelitian, mulai pengumpulan informasi hingga penulisan laporan dan publikasi.
- b. Perangkat Lunak:
1. Windows 11
 2. Notepad
 3. TensorFlow Lite
 4. Raspbian Stretch, sebagai OS untuk perangkat Single Board Computer Rapsberry Pi.
 5. Chromium sebagai browser pada perangkat Raspberry Pi
 6. Shell Terminal sebagai command prompt application pada OS Raspbian
 7. Github, sebagai website untuk mengunduh beberapa resources
 8. Python sebagai bahasa pemrograman yang digunakan.

2.3 TensorFlow Lite

TensorFlow Lite adalah versi ringan (lightweight) dari TensorFlow, sebuah framework deep learning yang digunakan untuk membuat dan menjalankan model machine learning pada perangkat dengan sumber daya terbatas, seperti perangkat seluler (smartphone), mikrokontroler, Raspberry Pi, dan perangkat berdaya rendah lainnya [30]. TensorFlow Lite dikembangkan oleh Google dan dirancang khusus untuk memungkinkan inferensi model machine learning (penggunaan

model yang sudah dilatih) dengan cepat dan efisien di perangkat yang memiliki keterbatasan perangkat keras dan daya [31]. Berikut adalah beberapa konsep dasar tentang TensorFlow Lite [32]–[34]:

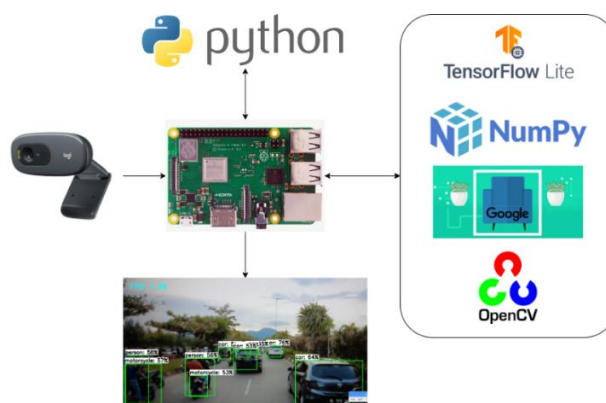
- a. Model Machine Learning, TensorFlow Lite memungkinkan untuk mengambil model machine learning yang sudah dilatih dengan TensorFlow (atau framework lainnya) dan mengonversinya menjadi format yang dapat dijalankan di TensorFlow Lite. Model ini dapat berupa model klasifikasi, deteksi objek, pengenalan suara, atau jenis model machine learning lainnya. Model machine learning pada TensorFlow Lite adalah model yang telah dilatih untuk tugas tertentu dan diubah ke format ringan (.tflite) untuk eksekusi di perangkat dengan sumber daya terbatas. Model ini dapat digunakan untuk berbagai tugas seperti klasifikasi gambar, deteksi objek, dan pengenalan suara. TensorFlow Lite menyediakan lingkungan yang dioptimalkan untuk menjalankan model-machine learning ini secara efisien, dengan kecepatan tinggi dan penggunaan daya yang rendah. Hal ini membuatnya sangat cocok untuk aplikasi perangkat seluler, IoT, dan mikrokontroler, di mana sumber daya terbatas memainkan peran penting. Model TensorFlow Lite dapat diintegrasikan ke dalam berbagai aplikasi untuk memberikan prediksi real-time dan pemrosesan data cerdas dengan cepat, sehingga menjadi alat yang kuat dalam penerapan machine learning di berbagai perangkat terbatas.
- b. Konversi Model, proses konversi model pada TensorFlow Lite adalah langkah penting untuk mengubah model machine learning yang telah dilatih (biasanya dalam format TensorFlow) ke format ringan (.tflite) sehingga dapat dijalankan di perangkat dengan sumber daya terbatas. Berikut adalah langkah-langkah umum dalam proses konversi model TensorFlow ke TensorFlow Lite:
 1. Pelatihan Model (Opsional): Proses dimulai dengan melatih model machine learning menggunakan data yang sesuai. Ini adalah langkah di mana model Anda belajar dari data pelatihan untuk menyelesaikan tugas tertentu.
 2. Ekstraksi Model TensorFlow: Model yang telah dilatih dalam format TensorFlow (umumnya dengan ekstensi .pb atau .h5) harus diekstraksi. Anda memerlukan model yang sudah ada untuk dikonversi.
 3. Konversi Model dengan TensorFlow Converter: TensorFlow menyediakan TensorFlow Converter, yang memungkinkan Anda mengonversi model ke format TensorFlow Lite. Anda harus menentukan tipe data target (misalnya, float16 atau int8) dan model Anda akan dioptimalkan untuk kinerja pada perangkat target.
 4. Optimisasi (Opsional): Setelah konversi, Anda dapat memilih untuk mengoptimalkan model Anda lebih lanjut untuk meningkatkan kinerja pada perangkat terbatas. Optimisasi ini dapat mencakup kompresi model, kuantisasi berat, dan lainnya.
 5. Validasi dan Pengujian: Model TensorFlow Lite yang dihasilkan harus divalidasi dan diuji untuk memastikan bahwa model tersebut masih memberikan hasil yang akurat dan sesuai dengan model asli.
 6. Integrasi ke Aplikasi: Model TensorFlow Lite yang sudah dikonversi kemudian diintegrasikan ke dalam aplikasi atau perangkat yang akan menggunakannya.
- c. Inferensi pada Perangkat, TensorFlow Lite dirancang untuk menjalankan inferensi (prediksi) model pada perangkat target. Ini memungkinkan aplikasi di perangkat seluler atau perangkat berdaya rendah untuk menggunakan model machine learning tanpa koneksi internet atau ketergantungan pada server jarak jauh. Proses inferensi pada TensorFlow Lite adalah tahap ketika model machine learning yang telah dikonversi ke format ringan (.tflite) dijalankan di perangkat terbatas untuk melakukan tugas spesifik, seperti klasifikasi gambar, deteksi objek, atau pengenalan suara. Saat inferensi, model menerima data input (misalnya gambar atau suara) dan menghasilkan prediksi atau output yang sesuai berdasarkan apa yang telah dipelajari selama pelatihan. Proses ini berlangsung dengan sangat cepat dan efisien, memungkinkan perangkat dengan sumber daya terbatas untuk memberikan respons real-time yang akurat. Ini sangat bermanfaat dalam berbagai aplikasi, termasuk deteksi objek dalam kamera ponsel, penerjemahan bahasa di perangkat seluler, dan banyak lagi, di mana kinerja dan respons cepat sangat penting.
- d. Optimasi Kinerja, TensorFlow Lite melakukan berbagai teknik optimasi untuk meningkatkan kinerja inferensi model di perangkat yang memiliki sumber daya terbatas. Ini mencakup optimasi untuk CPU, GPU, dan accelerators khusus (seperti TPU). Optimasi kinerja pada TensorFlow Lite adalah langkah penting untuk memastikan model machine learning berjalan secara efisien di perangkat dengan sumber daya terbatas. Berikut adalah beberapa strategi umum yang digunakan untuk mengoptimalkan kinerja model TensorFlow Lite:
 1. Kuantisasi, dimana kuantisasi adalah proses mengubah parameter model dari tipe data float32 menjadi tipe data int8 atau int16 yang memungkinkan perhitungan yang lebih cepat. Ini mengurangi konsumsi memori dan meningkatkan kecepatan inferensi.
 2. Kompresi Model, pada tahap ini digunakan teknik kompresi seperti pruning (menghapus parameter yang tidak penting), quantization-aware training, atau teknik kompresi model lainnya untuk mengurangi ukuran model dan meningkatkan efisiensi.
 3. Accelerated Hardware, menggunakan akselerator perangkat keras seperti unit pemrosesan tensor (TPU) atau unit pemrosesan grafik (GPU) jika tersedia pada perangkat Anda.
 4. Optimasi Kernel, memastikan bahwa kernel (inti) operasi matematis yang digunakan oleh model telah dioptimalkan untuk perangkat target. TensorFlow Lite menyediakan kernel-kernel yang dioptimalkan untuk berbagai arsitektur perangkat keras.
 5. Pemangkasan Model, menghapus bagian model yang mungkin tidak dibutuhkan untuk tugas tertentu, mengurangi jumlah operasi dan meningkatkan kecepatan inferensi.
 6. Penggunaan Format Kompresi dengan menggunakan format kompresi gambar atau audio yang lebih efisien jika model berurusan dengan data gambar atau audio.

7. Caching Hasil Inferensi, menyimpan hasil inferensi sebelumnya untuk data input yang sama dan mengembalikan hasil ini jika input yang sama diberikan lagi, mengurangi waktu komputasi.
 8. Mengurangi Resolusi Gambar, jika model bekerja dengan gambar, mengurangi resolusi gambar saat preprocessing dapat mengurangi beban komputasi.
 9. Paralelisasi Operasi, dengan memanfaatkan kemampuan perangkat keras untuk menjalankan beberapa operasi inferensi secara paralel untuk meningkatkan kecepatan.
 10. Memahami Perangkat Terbatas, menganalisis sumber daya perangkat target (memori, daya komputasi, dll.) dan mengoptimalkan model dan proses inferensi sesuai dengan batasan ini.
- e. Dukungan untuk Berbagai Perangkat, TensorFlow merupakan sebuah platform machine learning open-source yang memiliki dukungan yang luas untuk berbagai perangkat dan platform. TensorFlow mendukung berbagai perangkat keras termasuk CPU, GPU, dan perangkat keras AI spesifik seperti Tensor Processing Units (TPUs). Selain itu, TensorFlow juga mendukung berbagai platform, termasuk Android dan iOS untuk pengembangan aplikasi seluler, serta mikrokontroler seperti Raspberry Pi untuk perangkat IoT. Dukungan ini memungkinkan para pengembang untuk mengimplementasikan model machine learning di berbagai perangkat dengan sumber daya yang berbeda, termasuk perangkat dengan keterbatasan daya komputasi dan memori. Dengan berbagai opsi untuk deployment, TensorFlow memberikan fleksibilitas yang diperlukan untuk membangun solusi machine learning yang sesuai dengan berbagai kebutuhan aplikasi.
 - f. Berbagai Bahasa Pemrograman, TensorFlow Lite mendukung beberapa bahasa pemrograman yang berbeda untuk integrasi dan pengembangan model-machine learning. Bahasa utama yang digunakan dalam TensorFlow Lite adalah Python, yang menyediakan antarmuka yang kuat untuk mengembangkan, mengonversi, dan menjalankan model. Selain Python, TensorFlow Lite juga memiliki dukungan untuk bahasa pemrograman seperti C++, Java, dan Go, yang memungkinkan integrasi dengan aplikasi dan sistem yang ditulis dalam bahasa-bahasa ini. Hal ini memungkinkan pengembang untuk mengintegrasikan model-machine learning TensorFlow Lite ke dalam berbagai jenis aplikasi dan perangkat dengan menggunakan bahasa pemrograman yang sesuai dengan kebutuhan mereka. Dukungan untuk berbagai bahasa pemrograman ini membuat TensorFlow Lite menjadi alat yang fleksibel untuk pengembangan dan integrasi model-machine learning di berbagai platform dan lingkungan.
 - g. Dukungan Kustomisasi, TensorFlow Lite memberikan dukungan yang signifikan untuk disesuaikan (kustomisasi) agar sesuai dengan kebutuhan pengembangan dan aplikasi yang beragam. Dalam proses konversi model ke format TensorFlow Lite, pengguna memiliki kontrol atas beberapa aspek kustomisasi utama, termasuk tipe data target, tingkat kuantisasi (misalnya, float16 atau int8), dan konfigurasi optimalisasi. Hal ini memungkinkan pengguna untuk menentukan kompromi antara akurasi dan kinerja, yang sangat berguna ketika berurusan dengan perangkat keras terbatas. Selain itu, TensorFlow Lite menyediakan antarmuka Native C++ yang memungkinkan pengguna untuk memasukkan kode tambahan atau memodifikasi operasi-operasi khusus dalam model. Ini membuka pintu bagi kustomisasi yang lebih lanjut, termasuk penambahan operasi-operasi khusus atau integrasi dengan perangkat keras tertentu. Dengan kontrol tingkat rendah ini, pengembang dapat memastikan bahwa TensorFlow Lite dapat dioptimalkan secara maksimal untuk kebutuhan aplikasi mereka, terlepas dari berbagai kendala dan persyaratan yang mungkin ada. TensorFlow Lite secara efektif menyediakan kustomisasi yang diperlukan untuk menjalankan model machine learning dengan efisien pada berbagai perangkat dan platform.

TensorFlow Lite memiliki peran penting dalam memungkinkan model machine learning digunakan secara luas di berbagai perangkat yang mungkin memiliki keterbatasan daya dan perangkat keras. Hal ini membuat model machine learning menjadi lebih dapat diakses dan digunakan dalam berbagai aplikasi, mulai dari pengenalan gambar hingga pemrosesan bahasa alami dan banyak aplikasi lainnya.

2.4 Rancangan Sistem

Penelitian ini bertujuan untuk mengembangkan model object detection pada perangkat single board computer, dimana pada penelitian ini digunakan perangkat Raspberry Pi 3 Model B+ sebagai perangkat Single Board Computer. Adapun rancangan sistem secara keseluruhan dapat dilihat pada gambar berikut:



Gambar 3. Rancangan sistem

Pada gambar 3 dapat dilihat bahwa model real time object detection dibangun menggunakan gabungan beberapa perangkat keras dan perangkat lunak. Raspberry Pi digunakan sebagai perangkat Single Board Computer yang bertugas menjalankan semua proses komputasi. USB Webcam yang tersambung pada perangkat USB input SBC digunakan untuk menangkap citra dari lingkungan sekitar untuk kemudian diproses menggunakan sistem real time object detection. Adapun model image processing dibangun menggunakan TensorFlow Lite, Numpy, OpenCV dan Sample Model dari Google berbasis MobileNet. Adapun outputnya adalah sistem berhasil membuat label untuk setiap objek yang muncul pada halaman Image/Video Capture.

3. HASIL DAN PEMBAHASAN

3.1 Rangkaian Sistem

Penelitian ini dilakukan dengan mengembangkan sistem real time object detection menggunakan TensorFlow Lite, melalui proses tuning yang dilakukan untuk menemukan pengaturan yang tepat sehingga dapat berjalan pada perangkat Raspberry Pi sebagai perangkat Single Board Computer. Proses pengembangan sistem real time object detection dalam penelitian ini dilakukan melalui tahapan sebagai berikut:

- a. Persiapan SBC, dimana pada tahap ini dilakukan persiapan Raspberry Pi sebagai perangkat SBC yang digunakan, mulai dari:
 1. Instalasi Raspbian Stretch OS
 2. Download Repository yang dibutuhkan
 3. Membuat Virtual Environment
 4. Install TensorFlow dan OpenCV
- b. Setup TensorFlow Lite untuk perangkat Raspberry Pi, dimana pada tahap ini digunakan Pre-Trained Google untuk object detection yang memuat 80 objek citra yang telah dilatih seperti yang ditunjukkan pada Gambar 1.
- c. Melakukan setup webcam untuk real time object detection pada Raspberry Pi menggunakan bahasa pemrograman Python.

Berdasarkan proses pengembangan yang telah dilakukan tersebut, maka pada bagian ini dapat dilihat hasil yang telah dicapai berupa rangkaian sistem secara keseluruhan yang dapat dilihat pada gambar berikut:



Gambar 4. Rangkaian Sistem

Pada gambar 4 dapat dilihat bahwa web cam dihubungkan pada Raspberry Pi melalui port USB. Konektivitas jaringan internet dilakukan dengan menggunakan Ethernet port pada Raspberry Pi. Pada rangkaian sistem ini, juga digunakan HDMI port untuk menghubungkan monitor sebagai output display dari Raspberry Pi.

3.2 Hasil Yang Didapatkan

Berdasarkan penelitian yang dilakukan melalui tahapan-tahapan yang ada yang telah dijelaskan pada bagian sebelumnya, maka didapatkan hasil bahwa pengembangan real time object detection telah berhasil dilakukan. Sistem yang dibangun menggunakan perangkat Raspberry Pi sebagai perangkat Single Board Computer mendapatkan hasil yang baik. Adapun hasil capture video/image menggunakan webcam dapat dilihat pada gambar berikut:



Gambar 5. Hasil Real Time Object Detection

Pada gambar 5 terlihat bahwa sistem mampu mengenali objek yang ditangkap oleh web cam, sesuai dengan labelmap yang ada. Hasil ini didapatkan berdasarkan hasil pengujian yang dilakukan menggunakan USB web cam yang diarahkan pada layar laptop yang sedang memutar video street dashcam yang ada pada salah satu kanal Youtube. Sistem berhasil mengidentifikasi beberapa objek seperti Car, Person dan Motorcycle secara real time.

3.3 Keterbatasan Penelitian

Berdasarkan proses penelitian serta pembahasan yang dilakukan pada bagian sebelumnya, terdapat beberapa keterbatasan yang dihadapi, diantaranya adalah sebagai berikut:

- Raspberry Pi yang digunakan sebagai perangkat Single Board Computer memiliki kekurangan pada kemampuan komputasi, namun hal ini dianggap wajar mengingat spesifikasi dari Raspberry Pi yang pada dasarnya dirancang untuk memproses komputasi ringan dengan dukungan GPIO pin. Namun penggunaan TensorFlow Lite untuk object detection sudah sangat meringankan proses klasifikasi citra pada real time object detection.
- Kamera yang digunakan merupakan perangkat web cam yang memiliki resolusi 720p dengan 0,9 megapixel, sehingga kualitas perekaman yang dilakukan menjadi kurang maksimal. Hal ini menyebabkan ada beberapa objek yang gagal dideteksi oleh sistem yang dibangun.
- Kemampuan Raspberry Pi dengan spesifikasi terbatas hanya mampu memproses video dengan rata-rata 1,3 fps. Hal ini menyebabkan pemrosesan video menjadi tidak baik, sehingga berdampak juga pada kemampuan model yang dibangun dalam mengenali objek yang ada. Kekurangan ini sebetulnya dapat diatasi dengan menggunakan perangkat machine learning inferencing berbasis edge TPU yang digunakan melalui port USB pada perangkat Raspberry Pi. Perangkat tersebut diharapkan dapat meningkatkan kemampuan proses komputasi pada real time object detection menjadi lebih baik.

4. KESIMPULAN

Berdasarkan penelitian yang dilakukan melalui proses yang telah disebutkan pada bagian sebelumnya, maka penulis menarik beberapa kesimpulan diantaranya yaitu penelitian ini berhasil mengimplementasikan real-time object detection menggunakan TensorFlow Lite pada perangkat Raspberry Pi. Hal ini menunjukkan kemampuan Raspberry Pi sebagai platform embedded yang mampu menjalankan aplikasi deep learning dengan baik, meskipun dengan sumber daya perangkat keras yang terbatas. TensorFlow Lite telah terbukti efisien dalam hal penggunaan sumber daya, dimana dalam penelitian ini, penggunaan CPU Raspberry Pi tetap dalam batas yang dapat diterima selama proses inferensi model. Ini menunjukkan keunggulan TensorFlow Lite dalam menjalankan model machine learning pada perangkat berdaya rendah. Sistem real time object detection yang diimplementasikan mampu mendeteksi objek secara real-time dari feed webcam. Dalam hal ini sistem yang dikembangkan memiliki potensi aplikasi yang luas, termasuk pemantauan keamanan, analisis gambar, dan banyak lagi. Meskipun Raspberry Pi dapat menjalankan object detection secara real-time, ada beberapa kendala dalam hal kecepatan deteksi tergantung pada kompleksitas model dan jumlah objek yang harus dideteksi. Model dengan tingkat komputasi yang lebih tinggi atau banyak objek pada frame dapat mengurangi kecepatan deteksi. Penelitian ini memunculkan peran pentingnya optimasi model untuk Raspberry Pi. Untuk performa terbaik, perlu dilakukan optimasi pada model machine learning agar sesuai dengan sumber daya perangkat keras yang tersedia. Penelitian ini juga membuka peluang untuk penelitian lanjutan dalam optimasi performa real-time object detection pada Raspberry Pi menggunakan

TensorFlow Lite. Disamping itu, perlu penelitian lanjutan terkait dengan objek custom yang lebih kompleks dan beragam untuk di proses. Hal lain yang dapat dilakukan juga adalah eksplorasi penggunaan perangkat embedded dalam aplikasi yang lebih kompleks.

REFERENCES

- [1] I. Muslem, "Prototype Kunci RFID (Radio Frequency Identification) dalam Meningkatkan Keamanan Kendaraan Bermotor," JURNAL TIKA, vol. 5, no. 3, 2021, doi: 10.51179/tika.v5i3.104.
- [2] Ferdinand Louis, M. Ficky Duskarnaen, and Hamidillah Ajie, "UJI KECEPATAN RASPBERRY PI SEBAGAI PRIVATE CLOUD STORAGE UNTUK SMALL OFFICE HOME OFFICE: DENGAN STUDI KASUS DI UPT TIK," PINTER : Jurnal Pendidikan Teknik Informatika dan Komputer, vol. 5, no. 2, 2021, doi: 10.21009/pinter.5.2.7.
- [3] W. Najib, S. Sulisty, and Widyawan, "Tinjauan Ancaman dan Solusi Keamanan pada Teknologi Internet of Things," Jurnal Nasional Teknik Elektro dan Teknologi Informasi, vol. 9, no. 4, 2020, doi: 10.22146/jnteti.v9i4.539.
- [4] P. Hatta and C. W. Budiyo, "Alleviate the contending issues in network operating system courses: Psychomotor and troubleshooting skill development with Raspberry Pi," Open Engineering, vol. 11, no. 1, 2021, doi: 10.1515/eng-2021-0076.
- [5] G. Hollingworth, "Introducing Raspberry Pi Imager, our new imaging utility - Raspberry Pi," 5 Mar, 2020.
- [6] M. Manfaluthy, S. Wilyanti, and Y. Lasito, "Face Recognition Berbasis Raspberry Pi Pada Keamanan Pintu Otomatis," Prosiding Seminar Nasional Teknoka, vol. 4, 2020, doi: 10.22236/teknoka.v4i0.4274.
- [7] I. R. Muslem and T. M. Johan, "KLIK: Kajian Ilmiah Informatika dan Komputer Klasifikasi Citra Ikan Menggunakan Algoritma Convolutional Neural Network dengan Arsitektur VGG-16," Media Online, vol. 4, no. 2, pp. 978–985, 2023, doi: 10.30865/klik.v4i2.1209.
- [8] Z. Jiang, L. Zhao, L. I. Shuaiyang, and J. I. A. Yanfei, "Real-Time Object Detection Method For Embedded Devices," ArXiv, vol. 3, 2020.
- [9] J. Lee and K. il Hwang, "YOLO with adaptive frame control for real-time object detection applications," Multimed Tools Appl, vol. 81, no. 25, 2022, doi: 10.1007/s11042-021-11480-0.
- [10] S. Zhou, Y. Chen, X. Li, and A. Sanyal, "Deep SCNN-Based real-time object detection for self-driving vehicles using LiDAR temporal data," IEEE Access, vol. 8, 2020, doi: 10.1109/ACCESS.2020.2990416.
- [11] S. Moon, J. Jeon, D. Kim, and Y. Kim, "Swarm Reconnaissance Drone System for Real-Time Object Detection Over a Large Area," IEEE Access, vol. 11, 2023, doi: 10.1109/ACCESS.2022.3233841.
- [12] L. Dong, Z. Yang, X. Cai, Y. Zhao, Q. Ma, and X. Miao, "WAVE: Edge-Device Cooperated Real-time Object Detection for Open-air Applications," IEEE Trans Mob Comput, 2022, doi: 10.1109/TMC.2022.3150401.
- [13] W. Gai, Y. Liu, J. Zhang, and G. Jing, "An improved Tiny YOLOv3 for real-time object detection," Systems Science and Control Engineering, vol. 9, no. 1, 2021, doi: 10.1080/21642583.2021.1901156.
- [14] W. He, Z. Huang, Z. Wei, C. Li, and B. Guo, "TF-YOLO: An improved incremental network for real-time object detection," Applied Sciences (Switzerland), vol. 9, no. 16, 2019, doi: 10.3390/app9163225.
- [15] K. M. Hosny, A. Magdi, A. Salah, O. El-Komy, and N. A. Lashin, "Internet of things applications using Raspberry-Pi: a survey," International Journal of Electrical and Computer Engineering, vol. 13, no. 1, 2023, doi: 10.11591/ijece.v13i1.pp902-910.
- [16] I. W. Muttaqin and A. Rahman, "Sistem Presensi Berbasis RFID Menggunakan Raspberry Pi 3," Buletin Ilmiah Sarjana Teknik Elektro, vol. 1, no. 1, 2019, doi: 10.12928/biste.v1i1.850.
- [17] G. Widya Dharma, I. N. Piarsa, and I. M. Agus Dwi Suarjaya, "Kontrol Kunci Pintu Rumah Menggunakan Raspberry Pi Berbasis Android," Jurnal Ilmiah Merpati (Menara Penelitian Akademika Teknologi Informasi), 2018, doi: 10.24843/jim.2018.v06.i03.p02.
- [18] M. Irfan Pure, A. Ma'arif, and A. Yudhana, "Alat Deteksi Detak Jantung Pada Atlet Maraton Menggunakan Raspberry Pi 3B," JTEV (Jurnal Teknik Elektro dan Vokasional), vol. 7, no. 2, 2021, doi: 10.24036/jtev.v7i2.113526.
- [19] B. Yuwono, S. P. Nugroho, and H. Heriyanto, "PENGEMBANGAN MODEL PUBLIC MONITORING SYSTEM MENGGUNAKAN RASPBERRY PI," Telematika, vol. 12, no. 2, 2015, doi: 10.31315/telematika.v12i2.1409.
- [20] I. G. M. N. Desnanjaya, I. N. B. Hartawan, I. K. D. G. Supartha, and K. C. Kombonglangi, "IMPLEMENTASI COMPUTER VISION PADA MESIN FILLING CUPCAKE MENGGUNAKAN RASPBERRY PI," JST (Jurnal Sains dan Teknologi), vol. 11, no. 1, 2022, doi: 10.23887/jstundiksha.v11i1.39048.
- [21] I. W. Pande Agustiana Putra, I. N. Piarsa, and K. Suar Wibawa, "Sistem Pendeteksi Kebakaran Menggunakan Raspberry Pi Berbasis Android," Jurnal Ilmiah Merpati (Menara Penelitian Akademika Teknologi Informasi), 2018, doi: 10.24843/jim.2018.v06.i03.p03.
- [22] T. V. Sai, B. Aditya, A. M. Reddy, and Dr. Y. Srinivasulu, "Real Time Object Detection Using Raspberry Pi," Int J Res Appl Sci Eng Technol, vol. 11, no. 1, 2023, doi: 10.22214/ijraset.2023.48549.
- [23] S. Patil, "Real time object detection on a Raspberry Pi," Electronicwings, vol. V, no. Vi, 2017.
- [24] A. Gunnarsson, "Real time object detection on a Raspberry Pi," Electronicwings, vol. V, no. Vi, 2019.
- [25] S. Bantun, A. Ashari, and R. Karim, "Analisis Kinerja Raspberry Pi Sebagai SIP Server Untuk Aplikasi Video Phone," Techno.Com, vol. 19, no. 2, 2020, doi: 10.33633/tc.v19i2.3220.
- [26] M. Data, W. Yahya, and A. Kurniawan, "Implementasi Teknologi Virtualisasi Berbasis Kontainer untuk Perangkat Internet of Things pada Pertanian Presisi," CYBERNETICS, vol. 3, no. 01, 2020, doi: 10.29406/cbn.v3i01.1448.
- [27] H. Yenni and M. A. Ardianto, "ALAT DIGITAL PENCETAK KUE BAWANG MENGGUNAKAN RASPBERRY PI 3 MODEL B BERBASIS ANDROID," JTT (Jurnal Teknologi Terapan), vol. 6, no. 1, 2020, doi: 10.31884/jtt.v6i1.246.
- [28] T. Maya Kadarina, "Otomatisasi Perekaman Foto Intraoral Gigi untuk Rekam Medis Elektronik Menggunakan Internet of Things," Jurnal Teknologi Elektro, vol. 11, no. 1, 2020, doi: 10.22441/jte.2020.v11i1.008.
- [29] M. Data, W. Yahya, and A. Kurniawan, "Implementasi Teknologi Virtualisasi Berbasis Kontainer untuk Perangkat Internet of Things pada Pertanian Presisi," CYBERNETICS, vol. 3, no. 01, 2020, doi: 10.29406/cbn.v3i01.1448.
- [30] S. Bramasto, "Tensorflow Lite Pada Perangkat Bergerak Guna Prediksi Waktu Panen pada Operasi Pertanian Vertikal," Technopex-2020, 2020.

- [31] S. Li, "TensorFlow Lite: On-Device Machine Learning Framework," *Jisuanji Yanjiu yu Fazhan/Computer Research and Development*, vol. 57, no. 9, 2020, doi: 10.7544/issn1000-1239.2020.20200291.
- [32] Google, "TensorFlow Lite guide," TensorFlow, 2020.
- [33] A. Taufiq, M. Pratama, and A. R. Pratama, "Rancang Bangun Aplikasi Android 'Kuliah Apa?' Berbasis Flutter dan TensorFlow Lite," *Automata*, vol. 2, no. 1, 2021.
- [34] E. Manor and S. Greenberg, "Custom Hardware Inference Accelerator for TensorFlow Lite for Microcontrollers," *IEEE Access*, vol. 10, 2022, doi: 10.1109/ACCESS.2022.3189776.