

Implementasi Deteksi Objek Boneka Korban pada Kontes Robot SAR Indonesia Menggunakan ESP32-cam

Arahmad Taupiq^{1,*}, Yovi Pratama¹, M Irwan Bustami²

¹ Fakultas Ilmu Komputer, Program Studi Teknik Informatika, Universitas Dinamika Bangsa, Jambi, Indonesia

² Fakultas Ilmu Komputer, Program Studi Sistem Komputer, Universitas Dinamika Bangsa, Jambi, Indonesia

Email: ^{1,*}arahmadtaupiq612@gmail.com, ²yovi.pratama@gmail.com, ³irwan.sk05@gmail.com

Email Penulis Korespondensi: arahmadtaupiq612@gmail.com

Submitted: 27/09/2024; Accepted: 29/12/2024; Published: 30/12/2024

Abstrak—Kontes Robot SAR Indonesia tahun 21024 menuntut kemampuan robot untuk membedakan antara boneka dummy dan boneka korban dalam situasi darurat. Robot SAR ini mempunyai tujuan utama untuk menyelamatkan dan membawa boneka korban ke zona aman, sehingga penulis mengeksplorasi implementasi object detection pada robot SAR menggunakan ESP32-cam untuk mendeteksi boneka korban. Penulis menggunakan platform Edge Impulse, sebuah platform TinyML, untuk melatih model deteksi objek menggunakan arsitektur Faster Objects, More Objects (FOMO). FOMO adalah sebuah algoritma deteksi objek yang dirancang untuk berjalan dengan efisien pada perangkat dengan sumber daya terbatas seperti pada mikrokontroler ESP32-CAM. Data pelatihan diperoleh melalui pengambilan gambar boneka dummy dan boneka korban dalam berbagai sudut, kondisi pencahayaan, dan latar belakang menggunakan kamera dari ESP32-cam. Hasil confusion matrix, yang menunjukkan tingkat keberhasilan model dalam mendeteksi objek secara akurat, yang selanjutnya disebut sebagai F1 Score dari proses training model mencapai nilai 100%. Saat dilakukan pengujian, dari 10 kali percobaan, model berhasil mendeteksi boneka dummy dengan benar sebanyak 8 kali dan boneka korban sebanyak 9 kali. Peneliti menghadapi tantangan seperti variasi posisi dan kondisi lingkungan sehingga peneliti menggunakan algoritma tambahan dengan melakukan counter pada objek korban yang dideteksi untuk meningkatkan akurasi deteksi. Penggunaan FOMO memungkinkan deteksi objek yang lebih cepat dan mampu mendeteksi lebih banyak objek dalam satu frame. Implementasi ini menunjukkan potensi besar dalam pengembangan robot SAR yang lebih efisien dan mandiri untuk misi penyelamatan. Temuan ini berkontribusi pada peningkatan teknologi robotik, salah satunya dalam operasi SAR dan memberikan dasar bagi penelitian lebih lanjut dalam penerapan deteksi objek.

Kata Kunci: Deteksi Objek; ESP32-Cam; TinyML; edge Impulse; FOMO(Faster Objects, More Objects)

Abstract—The 2024 Indonesian SAR Robot Contest requires the robot's ability to distinguish between a dummy doll and a victim doll in an emergency situation. This SAR robot has the main goal of saving and bringing the victim doll to a safe zone, so the author explores the implementation of object detection on the SAR robot using ESP32-cam to detect the victim doll. The author uses the Edge Impulse platform, a TinyML platform, to train an object detection model using the Faster Objects, More Objects (FOMO) architecture. FOMO is an object detection algorithm designed to run efficiently on devices with limited resources such as the ESP32-CAM microcontroller. Training data was obtained by taking pictures of the dummy doll and the victim doll at various angles, lighting conditions, and backgrounds using a camera from the ESP32-cam. The results of the confusion matrix, which shows the level of success of the model in detecting objects accurately, hereinafter referred to as the F1 Score from the model training process reached a value of 100%. When tested, out of 10 trials, the model successfully detected the dummy doll correctly 8 times and the victim doll 9 times. Researchers face challenges such as variations in position and environmental conditions so that researchers use additional algorithms by performing counters on detected victim objects to improve detection accuracy. The use of FOMO allows faster object detection and is able to detect more objects in one frame. This implementation shows great potential in the development of more efficient and independent SAR robots for rescue missions. These findings contribute to the improvement of robotic technology, one of which is in SAR operations and provide a basis for further research in the application of object detection.

Keywords: Object Detection; ESP32-Cam; TinyML; edge Impulse; FOMO(Faster Objects, More Objects)

1. PENDAHULUAN

Dunia Teknologi robotika terus mengalami kemajuan setiap harinya. Tujuan diciptakannya robot adalah untuk membantu mempermudah kehidupan manusia modern saat ini dalam berbagai aspek [1]. Robot merupakan alat mekanik yang dapat melakukan aktivitas fisik yang dapat dikendalikan langsung oleh manusia atau melalui program yang sudah ditentukan sebelumnya (kecerdasan buatan) [2]. Saat ini, robot hadir dalam berbagai bentuk yang disesuaikan dengan tujuan dan fungsinya, seperti robot quadcopter, humanoid, manipulator, hexapod, dan masih banyak lagi. Jenis robot berbentuk *hexapod* memiliki potensi yang besar untuk digunakan dalam misi pencarian dan penyelamatan atau *Search and Rescue*(SAR), karena kemampuannya untuk beradaptasi dengan medan yang sulit [3]. KRSRI(Kontes Robot SAR Indonesia) merupakan salah satu cabang lomba robotika yang bernama KRI(Kontes Robot Indonesia), yang diselenggarakan oleh Balai Pengembangan Talenta Indonesia (BPTI), Pusat Prestasi Nasional (Puspresnas), Kementerian Pendidikan, Kebudayaan, Riset dan Teknologi Republik Indonesia [4]. Ajang lomba KRSRI memiliki misi utama yaitu menyelamatkan dan membawa boneka korban ke zona aman [5].

Pada ajang KRSRI tahun 2024 terdapat 2 objek boneka, yaitu boneka korban berbentuk setengah badan anak kecil berwarna oranye yang sedang menangis, dan boneka dummy berbentuk setengah badan berwarna oranye tanpa tangan dan kepala, yang harus dibedakan oleh robot sehingga membutuhkan teknologi *object detection*. Dalam beberapa tahun terakhir, teknologi *object detection* atau deteksi objek berkembang dengan sangat pesat dan banyak diterapkan dalam berbagai bidang, termasuk dalam kontes robot *Search and Rescue* (SAR). Deteksi objek bertujuan

untuk mengidentifikasi lokasi objek dalam sebuah citra atau video[6]. Deteksi Objek bisa dilakukan dengan berbagai metode, salah satunya dengan menggunakan metode *Haar Cascade Classifier* seperti yang pernah dilakukan oleh Joshua Sitompul et. Al [7], objek dapat dideteksi dari rentang jarak 100 cm sampai 150 cm. Pada robot SAR ini, robot harus dapat membedakan antara objek boneka dummy, boneka korban, dan lingkungan sekitarnya. Akurasi pendeteksian boneka korban sangat penting dalam ajang lomba ini agar robot dapat mengangkat objek yang benar lalu korban diletakkan di safety zone agar mendapatkan point tinggi. Teknologi objek deteksi ini dapat meningkatkan efisiensi dan akurasi dalam operasi penyelamatan, sehingga dapat menyelamatkan lebih banyak nyawa jika diterapkan dalam dunia nyata kedepannya.

TinyML merupakan jawaban atas ketidakjangkauan *AI(Artificial Intelligence)* yang memerlukan komputasi besar dan berat, serta memerlukan perangkat keras khusus dengan biaya yang sangat mahal dan menghabiskan sumber daya yang sangat besar [8], [9], [10]. Deteksi objek adalah salah satu aplikasi utama dari *tinyML*. *TinyML* dapat berjalan tanpa koneksi internet dan dapat diterapkan pada perangkat dengan sumberdaya terbatas seperti mikrokontroler ESP32-cam salah satunya dengan menggunakan platform *edge impulse*, yang dimana output dari model *tinyML* dapat di-deployment menjadi *arduino library* lalu diupload ke ESP32-cam..

ESP32-cam adalah sebuah mikrokontroler dilengkapi dengan modul kamera yang dapat menangkap gambar secara langsung dan mengolah citra. Sebagaimana mikrokontroler pada umumnya, ESP32-cam dapat digunakan sebagai otak dari berbagai macam *IoT* contohnya sebagai otak dari mobil remote control selain mengontrol mobil, ESP32-cam dapat digunakan untuk memantau lingkungan sekitar layaknya kamera pada umumnya [11].

Edge impulse adalah sebuah platform *machine learning operations(MLOps)* berbasis cloud yang digunakan untuk mengembangkan *embedded system* dan *tinyML* yang dapat diterapkan ke berbagai mikrokontroler dengan sumber daya terbatas karna menggunakan ukuran memori yang terbatas dan konsumsi daya rendah [12], [13], [14]. Model *tinyML* dari *edge impulse* lebih cepat dan mudah dalam hal pengembangan dan *deployment* [15], [16].

Salah satu algoritma deteksi objek yang dikembangkan oleh *edge impulse* adalah *FOMO(Faster Objects, More Objects)* yang memungkinkan untuk menghitung banyak objek dan mengetahui posisinya dalam gambar secara *real-time* dengan menggunakan sumber daya dan memori hingga 30x lebih sedikit daripada *MobileNet SSD* atau *YOLOv5* [17]. Peneliti menggunakan model *FOMO(Faster Objects, More Objects) MobileNetV2 0.1* untuk training model deteksi objek, yaitu model deteksi objek berbasis *MobileNetV2* yang merupakan salah satu arsitektur yang menggunakan metode *Convolutional Neural Network (CNN)* [18], [19], [20]. Model tersebut dirancang dengan ukuran lebih kecil dari 100KB dan mendukung input gambar hitam-putih (*grayscale*) atau berwarna (*RGB*) pada resolusi apapun. Penelitian sebelumnya menunjukkan bahwa ESP32-CAM memiliki potensi besar untuk deteksi objek dalam berbagai aplikasi. Misalnya, Ziliwu et al. mencapai akurasi deteksi hingga 91% dalam klasifikasi buah dan sayuran[21], sementara Gutti et al. menggunakan model *MobileNetV1* untuk klasifikasi produk pertanian dengan akurasi prediksi kelas 77% dan penggunaan memori sebesar 66,1 KB [22]. Selain itu, Nurdiansyah et al. menggunakan algoritma *YOLOv5* pada ESP32-CAM untuk mendeteksi 27 jenis objek di sekitar robot, mencapai mAP 95%, precision 97%, dan recall 100% [23]. Penelitian lainnya, seperti Aprilianto et al., menunjukkan kemampuan ESP32-CAM dalam membaca lampu navigasi kapal pada jarak 100 cm [24], dan Nugroho et al. mendemonstrasikan waktu respon rata-rata 0,762 detik untuk pemrosesan gambar [25]. Penelitian-penelitian tersebut mengindikasikan bahwa ESP32-CAM merupakan perangkat yang potensial untuk deteksi objek, termasuk dalam aplikasi SAR seperti yang dikaji dalam penelitian ini.

Penelitian ini bertujuan untuk deteksi objek dan mengklasifikasi boneka dummy dan boneka korban dalam Kontes Robot SAR Indonesia tahun 2024 yang diterapkan pada ESP32-cam dengan menggunakan model *FOMO(Faster Objects, More Objects) MobileNetV2 0.1* dari platform *edge impulse*. Hasil dari penelitian ini diharapkan dapat mampu mendeteksi dengan akurat boneka korban pada ajang Kontes Robot SAR Indonesia tahun 2024, selain itu memberikan dasar untuk aplikasi robot SAR dalam misi penyelamatan di dunia nyata, menambah wawasan mengenai objek deteksi, dan dapat digunakan sebagai bahan studi literatur bagi peneliti lain yang berminat dengan deteksi objek menggunakan ESP32-cam.

2. METODOLOGI PENELITIAN

Metodologi ini dirancang untuk memenuhi kebutuhan Kontes Robot SAR Indonesia dengan memanfaatkan perangkat keras sederhana namun efisien. Terdapat beberapa tahapan yang dilalui untuk melakukan penelitian ini, dimulai dari: Studi Literatur dan Analisis Kebutuhan, Desain Hardware dan Software, pengumpulan dan pelabelan dataset, pelatihan model di platform *edge impulse*, lalu diakhiri pengujian akurasi dan validasi dari hasil model training, seperti pada Gambar 1 berikut yang menggambarkan tahapan penelitian dari awal hingga akhir:



Gambar 1. Tahapan Penelitian

2.1 Studi Literatur dan Analisis Kebutuhan

Ini merupakan langkah awal dalam tahapan penelitian, yaitu melakukan kajian literatur atau riset terhadap penelitian terdahulu yang telah dilakukan untuk mendapatkan informasi dan teori pendukung. Pada tahap ini peneliti juga melakukan analisa kebutuhan, berikut adalah hal-hal yang dibutuhkan dalam penelitian ini :

- a. Boneka korban dan boneka dummy Kontes Robot Sar Indonesia tahun 2024 sebagai objek untuk dideteksi, pada Gambar 2 berikut dapat dilihat bentuk dari boneka dummy dan boneka korban :



Gambar 2. Boneka korban dan boneka dummy

- b. Platform *edge impulse*, yang digunakan untuk training model deteksi objek dan *mendeploy* menjadi *arduino library*, pada situs studio.edgeimpulse.com.
- c. Mikrokontroler ESP32-cam sebagai perangkat keras yang digunakan untuk pengumpulan dataset dan digunakan untuk implementasi deteksi objek.
- d. Dataset gambar boneka dummy dan boneka korban.
- e. Software Arduino IDE yang digunakan untuk memprogram ESP32-cam.
- f. Kabel micro USB.
- g. Laptop dengan spesifikasi RAM 16 GB dengan Windows 11 sebagai sistem operasi.

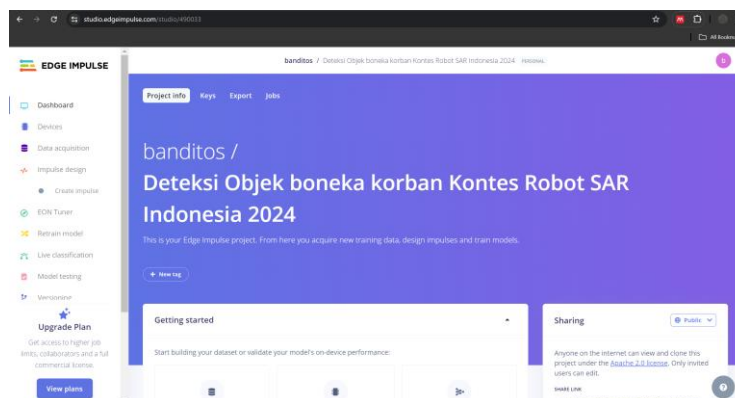
2.2 Desain Hardware dan Software

Hardware yang akan peneliti butuhkan sesuai dengan analisis kebutuhan adalah sebuah ESP32-cam, kabel micro USB, dan sebuah laptop, desain yang peneliti buat sangat sederhana yaitu sebuah ESP32-cam yang dihubungkan dengan sebuah kabel micro USB ke sebuah laptop seperti yang terlihat pada Gambar 3.



Gambar 3. Desain Hardware yang digunakan dalam penelitian

Terdapat dua software yang peneliti gunakan dalam penelitian ini yaitu *edge impulse* sebagai platform untuk training model deteksi objek, dan Arduino IDE yang digunakan untuk mengunggah program ke mikrokontroler ESP32-cam[26]. Tampilan dari dashboard platform *edge impulse* dapat diamati pada Gambar 4.



Gambar 4. Tampilan dashboard platform *edge impulse*

2.3 Pengumpulan Dataset dan Labelling Dataset

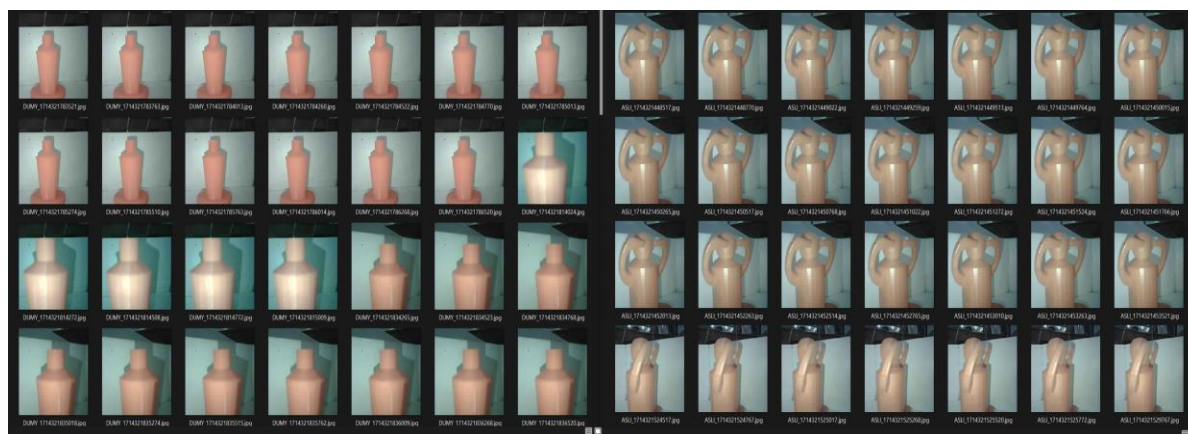
a. Pengumpulan Dataset

Pada tahap ini, peneliti mengumpulkan gambar boneka dummy dan boneka korban untuk dijadikan dataset model training deteksi objek, boneka korban dan boneka dummy peneliti letakkan di beberapa titik arena lomba dengan arah sesuai petunjuk teknis Kontes Robot SAR Indonesia tahun 2024 yaitu dengan arah boneka dummy menghadap lurus ke depan, sedangkan boneka korban menghadap serong 45° ke kiri ataupun kanan, dataset yang diambil menggunakan kamera dari ESP32-cam dengan lampu flash menyala untuk menyesuaikan kualitas gambar training dengan ESP32-cam tersebut. Gambar yang digunakan untuk dataset akan dibagi menjadi dua bagian yaitu train digunakan untuk training model objek deteksi dan test digunakan untuk mengetes model dari deteksi objek ,berikut adalah jumlah gambar yang digunakan untuk dataset training model deteksi objek dapat dilihat pada Tabel 1 berikut :

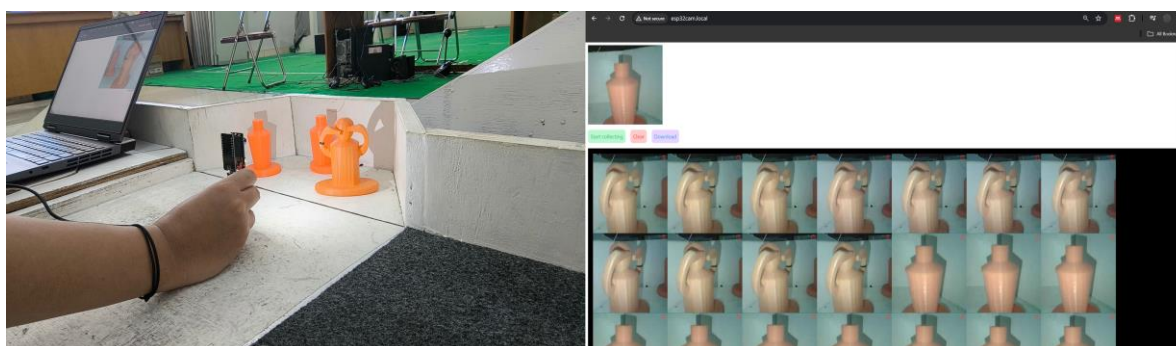
Tabel 1. Jumlah dataset yang digunakan untuk *training* dan *testing* model deteksi objek

Objek	Train	Test
Boneka Dummy	71	17
Boneka Korban	83	20
Total Dataset :		191 Gambar

Dataset gambar yang peneliti ambil sudah relevan dan representatif terhadap kondisi nyata yang nantinya akan dihadapi oleh model deteksi objek, sehingga jumlah dataset yang peneliti perlukan relatif kecil. Berikut contoh dari dataset gambar boneka dummy dan boneka korban yang dapat dilihat pada Gambar 5, serta proses pengambilan gambar dataset dapat dilihat pada Gambar 6.



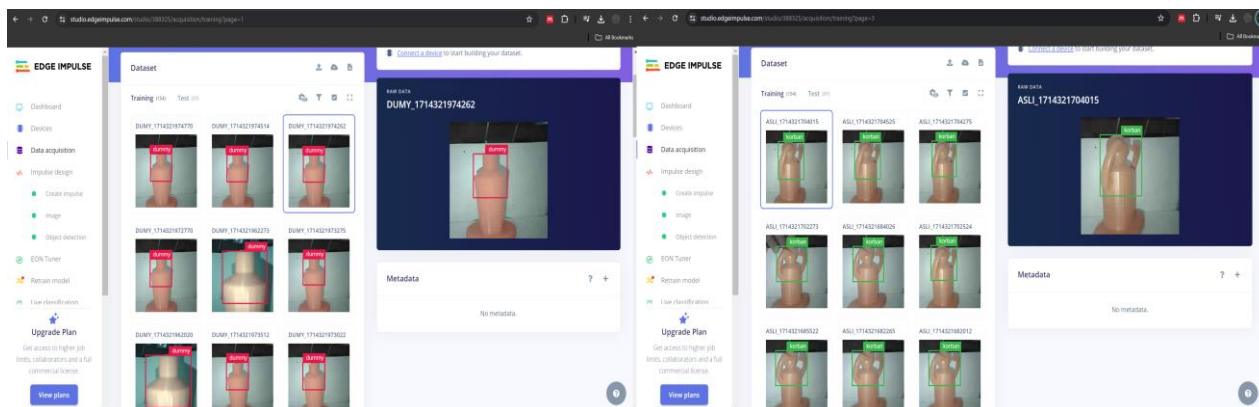
Gambar 5. Contoh dataset gambar boneka dummy dan boneka korban



Gambar 6. Proses pengambilan gambar untuk dataset

b. Labelling Dataset

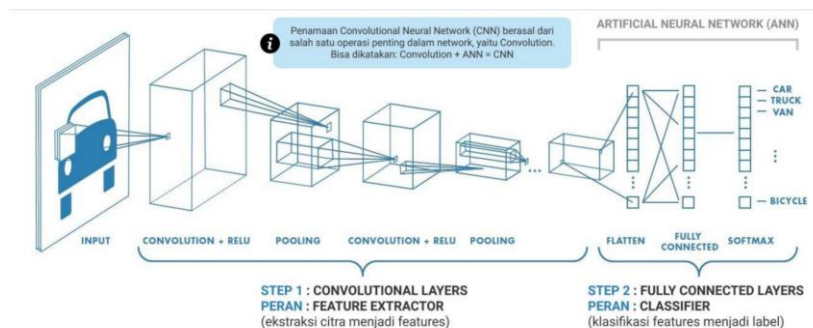
Pada tahap ini, dataset dari gambar boneka dummy dan boneka korban akan diunggah pada bagian “data acquisition” di platform *edge impulse*, semua gambar yang telah diunggah akan dilabeli satu per-satu berdasarkan klasifikasi gambar tersebut, pada tahap ini peneliti membuat dua label kelas klasifikasi yaitu label “dummy” untuk boneka dummy, dan label “korban” untuk boneka korban. Labelling dataset akan ditandai dengan sebuah garis berbentuk persegi yang disebut *bounding box* agar memudahkan model mengetahui posisi objek yang dilabeli pada gambar, yang dimana *bounding box* tersebut posisinya akan menjadi bingkai dari objek yang akan diklasifikasi. Pada penelitian ini, peneliti meletakkan bounding box pada bagian setengah badan boneka ke atas, karna pada bagian tersebut yang peneliti anggap akan mudah untuk dipelajari oleh model objek deteksi, berikut pada Gambar 7 akan diperlihatkan beberapa contoh dataset yang sudah dilabeli.



Gambar 7. Dataset yang sudah dilabeli

2.4 Training Model di Edge Impulse

Selanjutnya, peneliti melakukan training model deteksi objek menggunakan algoritma *FOMO* (*Faster Objects, More Objects*) dengan *MobileNetV2 0.1* dari *Edge Impulse*. Algoritma ini menggunakan arsitektur *MobileNetV2*, yang merupakan bagian dari metode *Convolutional Neural Network* (CNN). CNN dikembangkan dari *Multi-Layer Perceptron* (MLP) yang diciptakan untuk mengerjakan data dua dimensi, CNN adalah salah satu metode *Machine Learning*[27]. CNN difungsikan untuk mengelola data citra, yang merupakan jenis metode yang dikembangkan dari jaringan syaraf tiruan[28]. Secara umum, CNN terdiri dari dua tahapan yaitu pertama *feature extractor* berperan meng-ekstrak sebuah citra(image) menjadi data angka-angka yang merepresentasikan citra tersebut, lalu yang kedua tahap *classification* berperan untuk memproses data angka-angka yang sudah di-ekstrak sebelumnya untuk dilakukan klasifikasi, tahapan klasifikasi citra pada metode CNN dapat dilihat pada Gambar 8 berikut.



Gambar 8. Tahapan klasifikasi citra dengan metode CNN

Algoritma *FOMO* bekerja dengan tujuan untuk memberikan informasi jumlah objek dan lokasi koordinat x, y objek pada sebuah citra dengan komputasi yang efisien. *FOMO* memanfaatkan *heat map* yaitu representasi visual dengan memetakan pixel yang nantinya menjadi titik tengah (*centroid*) dari lokasi objek yang terdeteksi pada citra, resolusi *heat map* pada algoritma *FOMO* dapat dikonfigurasi, contohnya dengan menggunakan *heat map* 8 kali lebih kecil dari gambar yang diinput sehingga input gambar dengan resolusi 160×160 akan mempunyai *heat map* 20×20 . Perbedaan *FOMO* dengan algoritma deteksi objek lainnya yaitu *FOMO* tidak memberikan output *bounding box* tetapi hanya memberikan *centroid* dari objek yang terdeteksi pada citra[17].

Setelah dataset terkumpul dan dilabeli, langkah berikutnya adalah *Impulse Design* yang terdiri dari tiga fase utama yaitu pertama *create impulse*, *image*, dan *object detection*, masing masing fase tersebut akan berisi beberapa parameter yang memiliki peran penting dalam proses pengolahan data dan pelatihan model.

a. Create Impulse

Pada fase ini, sebuah impulse mengambil data mentah, menggunakan pemrosesan sinyal untuk mengekstrak fitur, dan kemudian menggunakan blok pembelajaran untuk mengklasifikasikan data baru. Image data 48×48 dipilih agar model tetap dapat melakukan deteksi objek secara efisien tanpa mengurangi kinerja perangkat karena ESP32-CAM memiliki sumber daya terbatas. Berikut beberapa parameter yang peneliti isi.

1. Image data (resolusi dari gambar) : 48×48
2. Processing block : image
3. Learning block : object detection (images)

b. Image

Pada fase ini akan ditentukan bagaimana *color depth* dari image tersebut akan diolah, apakah menggunakan berwarna (*RGB*) ataupun hitam-putih (*grayscale*) lalu akan dilakukan *generate features*. Color depth : RGB

c. Object Detection

Fase ini akan dilakukan training dari model deteksi objek, nilai learning rate 0.003 dipilih agar memberikan keseimbangan yang baik antara kecepatan pembelajaran dan stabilitas selama proses pelatihan, berikut beberapa parameter yang peneliti isi yang digunakan dalam model training.

1. Number of training cycles (epochs dari training model) : 90
2. Learning rate (seberapa cepat sebuah model belajar) : 0.003
3. Training Processor : CPU
4. Data Augmentation (mengubah data untuk meningkatkan jumlah latihan, serta meningkatkan akurasi) : YES
5. Neural Network Architecture : FOMO(Faster Objects, More Objects) MobileNetV2 0.1

Setelah melakukan konfigurasi parameter dari ketiga fase tersebut, model siap untuk ditraining dan hasilnya akan ditampilkan selanjutnya.

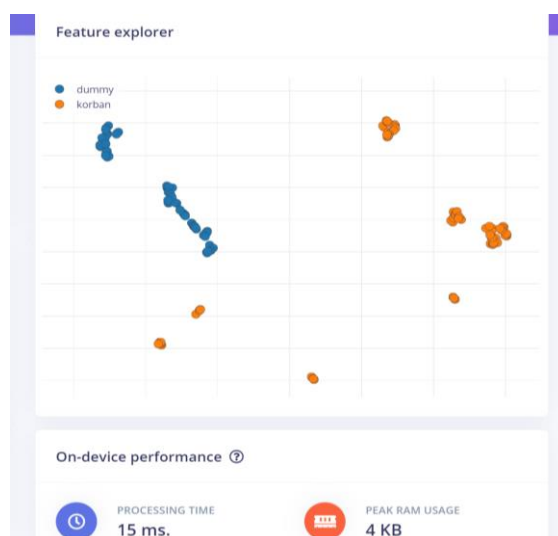
2.5 Pengujian Akurasi dan Validasi Model

Pada platform *Edge Impulse* akan dilakukan perhitungan akurasi model dengan menghitung setiap hasil training model menggunakan *confusion matrix* pada saat proses training model. Nilai akurasi setiap objek akan dirata-ratakan lalu menjadi nilai akurasi akhir yang disebut dengan *F1 Score*. Semakin tinggi nilai *F1 Score* maka akan semakin akurat model deteksi objek. Proses pengujian klasifikasi dari model objek deteksi sebelum diimplementasikan ke ESP32-cam adalah dengan cara peneliti memanfaatkan fitur *model testing* dan *live classification* yang disediakan oleh *edge impulse*. Hasil dari *confusion matrix*, pengujian dengan fitur *model testing* dan *live classification*, serta pengujian setelah diimplementasikan pada ESP32-cam akan ditampilkan selanjutnya.

3. HASIL DAN PEMBAHASAN

Sebelum menjelaskan hasil dari training model deteksi objek. Pada fase *image*, langkah yang dilakukan adalah mengatur *color depth* dari image yang akan diolah, setelah itu akan dilakukan proses *generate features* menggunakan *edge impulse*. Pada tahap ini, *edge impulse* akan menghasilkan sebuah diagram *feature explorer* dari image dataset yang telah melalui proses generate features. Diagram ini memberikan informasi penting tentang image dataset yang dimiliki.

Pada diagram *feature explorer*, setiap objek dalam dataset akan diwakili oleh titik-titik dengan warna yang berbeda sesuai dengan labelnya. Pada penelitian ini titik oranye mewakili label objek korban, sedangkan titik biru mewakili label objek dummy. Jika titik-titik ini terpisah dengan jelas, tidak tumpang tindih antara label ‘korban’ dengan ‘dummy’ dan berkelompok sesuai dengan labelnya masing-masing, ini menunjukkan bahwa dataset yang dimiliki pada penelitian ini, memiliki kualitas yang baik untuk mendukung proses training model. Diagram *feature explorer* ini dapat membantu peneliti dalam mengidentifikasi kualitas dataset yang digunakan, dan peneliti dapat memastikan objek-objek dalam dataset tersebut dapat dibedakan dengan baik dengan algoritma deteksi, sehingga proses training dapat menghasilkan model yang akurat dan efisien. Pada Gambar 9 dapat dilihat diagram *feature explorer* dari penelitian ini.

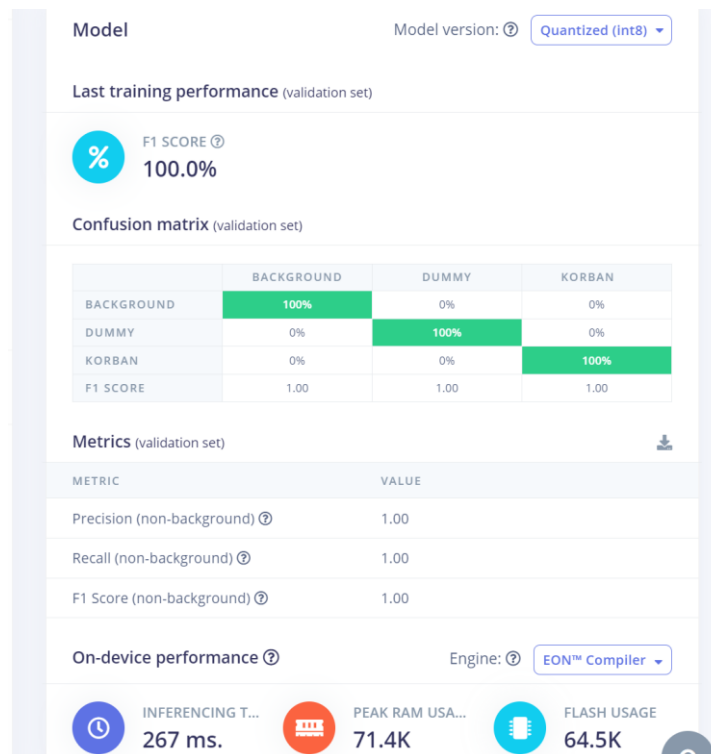


Gambar 9. Diagram *feature explorer* hasil dari *generate features*

3.1 Hasil Model Training

Setelah proses training model deteksi objek selesai, platform *edge impulse* akan memberikan hasil nilai akurasi setiap objek yang telah ditraining. Hasil training disajikan dalam bentuk *confusion matrix*. *Confusion matrix* adalah sebuah tabel yang digunakan dalam *machine learning* untuk mengevaluasi kinerja klasifikasi model deteksi objek, yang

digunakan untuk membandingkan antar nilai prediksi model dan nilai sebenarnya dari data uji. Pada Gambar 10 berikut dapat dilihat hasil *confusion matrix* yang peneliti dapatkan setelah training model selesai.



Gambar 10. Hasil model training dalam bentuk tabel *confusion matrix*

Dari *confusion matrix* tersebut, hasil yang dapat diperoleh adalah sebagai berikut:

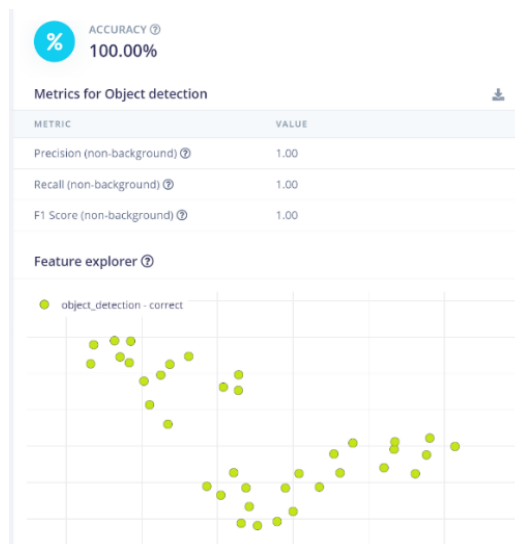
- Dummy dengan nilai akurasi 100%
- Korban dengan nilai akurasi 100%
- F1 Score dengan nilai akurasi 100%

Dari hasil training model deteksi objek diatas, didapatkan bahwa model objek deteksi dapat mendeteksi objek boneka korban dan boneka dummy dengan sangat baik, yang direpresentasikan nilai dari tabel *confusion matrix*. Nilai skor yang ditampilkan adalah nilai yang didapatkan dari hasil evaluasi model deteksi objek terhadap dataset yang digunakan selama proses training model deteksi objek.

Namun demikian, perlu dicatat bahwa dalam implementasi di dunia nyata, ketika peneliti mengaplikasikan model deteksi objek ini ke dalam perangkat ESP32-CAM, hasil deteksinya mungkin tidak akan mencapai tingkat akurasi 100% seperti yang terlihat dalam confusion matrix. Perbedaan hasil ini bisa disebabkan oleh berbagai faktor eksternal yang dapat mempengaruhi performa model deteksi objek. Faktor-faktor tersebut meliputi pencahayaan yang mungkin berubah-ubah, sudut pandang dari kamera yang berbeda saat pengambilan gambar dengan Ketika dilakukan pengujian, kondisi lingkungan yang tidak terkendali, dan banyak variabel lain yang dapat mempengaruhi kualitas deteksi. Oleh karena itu, meskipun model deteksi objek dari hasil training menunjukkan performa yang sangat baik, peneliti harus siap menghadapi tantangan dan variasi yang muncul saat mengimplementasikan model ini dalam situasi dunia nyata.

3.2 Model Testing

Pada tahapan ini, peneliti melakukan pengujian pertama model, yaitu dengan menggunakan fitur *model testing* yang merupakan salah satu opsi dari fitur yang disediakan oleh platform *edge impulse*. Pengujian model digunakan untuk mengevaluasi kinerja model deteksi objek yang telah dihasilkan dari proses *training* sebelumnya. Fitur ini memungkinkan peneliti untuk menguji model deteksi objek menggunakan dataset yang telah dibagi secara khusus sebelumnya yaitu *training* dan *test*, *dataset test*-lah yang digunakan untuk tujuan pengujian. Dalam penelitian ini, peneliti telah menyiapkan dataset pengujian yang terdiri dari 37 gambar. *Dataset test* ini dibagi menjadi dua kategori: 17 gambar untuk kategori boneka dummy dan 20 gambar untuk kategori boneka korban, pembagian dataset *training* dan *test* dapat diatur oleh pengguna pada bagian *data acquisition* pada platform *edge impulse*. Pembagian ini dilakukan untuk memastikan bahwa model diuji secara menyeluruh dan mampu mendeteksi kedua jenis objek dengan akurasi yang tinggi. Hal tersebut sangat penting untuk memastikan bahwa model tidak hanya bekerja dengan baik pada *dataset training*, tetapi juga pada dataset yang belum pernah dilihat sebelumnya yaitu *dataset test*, sehingga memberikan gambaran yang lebih realistis tentang performa model dalam aplikasi dunia nyata. Pada Gambar 11 di bawah ini dapat dilihat hasil dari *metrics model testing*.



Gambar 11. Hasil metrics model testing

Dari gambar di atas, diperoleh informasi bahwa model deteksi objek dapat mengenali dan mengklasifikasikan gambar-gambar dari dataset, pengujian dengan sangat baik. Dataset pengujian ini terdiri dari total 37 gambar. Hasil pengujian menunjukkan akurasi mencapai 100%, yang berarti semua gambar dalam *dataset test* berhasil dideteksi dan diklasifikasikan dengan sangat baik oleh model deteksi objek. Pada bagian *feature explorer*, terdapat titik-titik berwarna hijau yang menunjukkan hasil deteksi objek oleh model deteksi objek. Titik-titik tersebut menunjukkan bahwa objek tersebut telah dideteksi dan diklasifikasikan dengan benar sesuai dengan kategorinya. Sebaliknya, jika terdapat titik-titik berwarna merah menandakan bahwa objek tersebut tidak diklasifikasikan dengan benar, atau dengan kata lain, model melakukan kesalahan dalam mendeteksi dan mengklasifikasikan objek tersebut. Dalam penelitian ini semua titik-titik pada data *feature explorer* berwarna hijau ini menunjukkan performa yang sangat baik dari model deteksi objek yang telah dilatih.

3.3 Live Classification Model Deteksi Objek

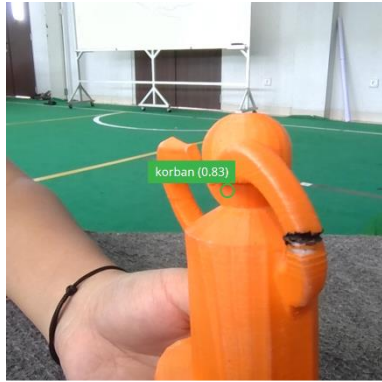
Pada tahap ini, peneliti melakukan pengujian kedua pada model deteksi objek. Pengujian ini memanfaatkan fitur yang disediakan oleh platform *edge impulse* yaitu fitur *live classification*. Fitur *live classification* ini memungkinkan pengguna untuk memvalidasi model objek deteksi dengan data dari dunia nyata yang diambil langsung dari perangkat atau *development board* yang didukung oleh platform *edge impulse*. Fitur ini dapat memberikan gambaran nyata tentang bagaimana performa model deteksi objek dalam kondisi dunia nyata. Dengan fitur ini, peneliti dapat melihat seberapa baik model deteksi objek secara *real-time*, yang mencakup berbagai parameter acak seperti pencahayaan, posisi objek, posisi kamera dan lain lain yang mungkin tidak tercakup dalam dataset yang digunakan. Dengan ini peneliti dapat mengidentifikasi kekuatan dan kelemahan model deteksi objek, sehingga dapat melakukan penyesuaian yang diperlukan untuk meningkatkan performa model deteksi objek tersebut. Pada penelitian ini, peneliti menggunakan kamera webcam dari laptop yang digunakan untuk melakukan *live classification* sehingga mungkin nantinya hasil ini akan berbeda dengan performa sebenarnya pada ESP32-cam, tetapi tetap dapat menjadi patokan performa yang penting bagi peneliti. Berikut adalah beberapa hasil deteksi dan klasifikasi dari model objek deteksi selama proses *live classification*.



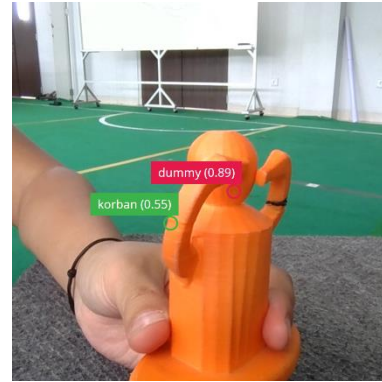
Gambar 12. Model mendeteksi kedua objek dengan baik



Gambar 13. Model tidak dapat mendeteksi boneka korban



Gambar 14. Model mendeteksi boneka korban dengan baik



Gambar 15. Model mengalami kesalahan dalam mendeteksi boneka korban

Dari hasil *live classification* dapat disimpulkan bahwa model deteksi objek yang sudah *ditraining* dapat mendeteksi objek boneka dummy dengan sangat baik, seperti yang ditampilkan pada Gambar 13 dan Gambar 14. Namun, untuk boneka korban, model mengalami kesulitan untuk mendeteksi dan mengklasifikasikan objek tersebut dengan konsistensi yang sama. Hal ini terlihat pada Gambar 16 dan Gambar 14 yang mana model gagal mendeteksi korban asli dengan tepat. Sebaliknya pada Gambar 13 dan Gambar 15, model deteksi objek berhasil mendeteksi boneka korban, tetapi hanya jika objek berada pada sudut pandang tertentu, hal ini dapat disebabkan oleh perbedaan sudut pandang, pencahayaan, serta perbedaan sumber input, yang mana pada saat *live classification*, peneliti menggunakan kamera dari webcam laptop, bukan dari kamera ESP32-CAM, yang memiliki kualitas gambar dan kinerja pemrosesan berbeda.

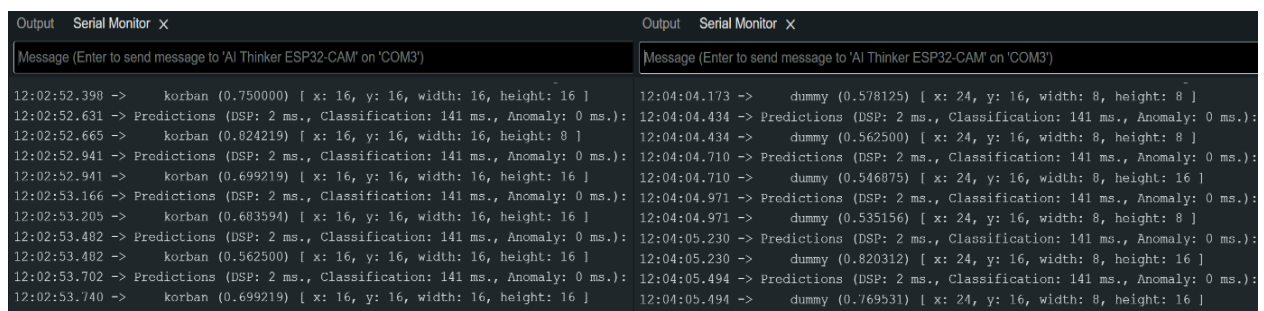
3.4 Deployment Dan Uji Coba Model Deteksi Objek Pada ESP32-cam

Setelah menyelesaikan semua tahapan di atas, langkah selanjutnya peneliti akan melakukan *deployment* model deteksi objek yang sudah *ditraining*. Pada tahap ini, peneliti akan melakukan *deployment* model menjadi *library arduino* yang nantinya akan diunggah ke perangkat ESP32-cam melalui software *Arduino IDE*. Pada platform *edge impulse*, untuk *men-deploy* model, peneliti masuk ke menu *deployment* lalu memilih *Arduino Library* sebagai opsi *deployment*, lalu memilih *Tensorflow Lite* dengan *Quantized (int8)* sebagai optimasi model untuk memastikan bahwa model objek deteksi tersebut dapat berjalan efisien pada perangkat ESP32-cam. Selanjutnya, *edge impulse* akan *men-generate library arduino* dan saat sudah selesai digenerate maka *library arduino* tersebut dapat diunduh. Setelah itu peneliti menginstall *library* yang sudah diunduh tadi pada software *Arduino IDE*, lalu kemudian peneliti mengunggah program tersebut ke perangkat ESP32-cam. Selanjutnya, peneliti menguji model deteksi objek yang telah diunggah ke ESP32-cam langsung pada objek boneka dummy dan boneka korban.

Pengujian peneliti lakukan dengan lampu flash menyala pada ESP32-cam, seperti yang dilakukan pada saat pengambilan dataset. Pengujian peneliti lakukan dengan metode kamera pada ESP32-cam menghadap ke objek dengan jarak 6 cm sampai 9 cm selama 5 detik, kemudian kamera dialihkan menghadap ke arah yang lain, dan kembali diulang sebanyak 10 kali untuk masing-masing objek boneka korban dengan label korban dan boneka dummy dengan label dummy, peneliti melakukan pemantauan hasil deteksi dan klasifikasi model deteksi objek yaitu melalui *serial monitor* pada software *Arduino IDE*. Pada Tabel 2 berikut peneliti menyajikan hasil dari pengujian yang peneliti lakukan.

Tabel 2. Hasil Pengujian deteksi objek langsung pada ESP32-cam

Objek	Berhasil	Tidak Berhasil
Boneka dummy	8	2
Boneka korban	9	1



Gambar 16. Tampilan Serial Monitor pada Arduino IDE



Gambar 17. Proses Pengujian Model Deteksi Objek

Hasil pengujian yang terlihat pada tabel 2, model deteksi objek menunjukkan kinerja yang sangat baik pada perangkat ESP32-cam. Hal ini dibuktikan dengan hasil dari 10 kali percobaan, model deteksi objek berhasil untuk mendeteksi objek boneka dummy sebanyak 8 kali dan berhasil mendeteksi objek boneka korban sebanyak 9 kali, dengan masing masing waktu untuk klasifikasi sebesar 141 ms.

Dapat disimpulkan bahwa model deteksi objek pada ESP32-cam berjalan lebih baik untuk mendeteksi boneka korban daripada boneka dummy. Namun hal ini bertolak belakang dengan hasil uji pada tahap *live classification* yang dimana pada tahap tersebut justru model deteksi objek mempunyai performa yang lebih baik dalam mendeteksi objek boneka dummy dibandingkan objek boneka korban. Salah satu faktor yang dapat mempengaruhi terjadinya perbedaan performa ini yaitu penggunaan kamera yang berbeda pada saat pengumpulan dataset untuk *training* model. Kamera yang digunakan untuk mengumpulkan dataset training model adalah kamera dari perangkat ESP32-cam itu sendiri. Oleh karena itu, performa model deteksi objek pada ESP32-cam menjadi lebih optimal karena model tersebut dilatih dengan data yang diambil dari perangkat yang sama. Selain faktor penggunaan kamera yang berbeda, faktor yang dapat mempengaruhi perbedaan performa model deteksi objek adalah pada pengujian ini, peneliti menyalakan lampu flash pada perangkat ESP32-cam untuk menjaga konsistensi pencahayaan objek, seperti yang dilakukan pada saat peneliti mengambil dataset *training* untuk model deteksi objek.

Secara menyeluruh, meskipun ESP32-cam mempunyai sumber daya terbatas sehingga resolusi dari image data hanya sebesar 48×48 agar model dapat berjalan secara efisien, dari hasil pengujian ini model deteksi objek yang di-*deploy* pada ESP32-cam menunjukkan performa yang baik, terutama dalam mendeteksi objek boneka korban dengan waktu pemrosesan klasifikasi sebesar 141 ms. Hasil ini memberikan indikasi positif untuk penggunaan model deteksi objek ini dalam Kontes Robot SAR Indonesia, meskipun masih perlu ditingkatkan dan diperbaiki pada beberapa aspek untuk mencapai akurasi yang lebih tinggi.

3.5 Optimasi Model Deteksi Objek Untuk Kontes Robot SAR Indonesia

Dikarnakan model deteksi objek ini akan peneliti gunakan pada ajang Kontes Robot SAR Indonesia tahun 2024, maka tingkat akurasi yang tinggi untuk mendeteksi objek boneka korban sangatlah penting. Dalam pengujian yang dilakukan, peneliti menemukan bahwa model deteksi objek terkadang mengalami kesalahan, di mana boneka dummy terdeteksi sebagai boneka korban, seperti yang diperlihatkan pada Gambar 18 berikut.

```

Output  Serial Monitor  x
Message (Enter to send message to 'AI Thinker ESP32-CAM' on 'COM3')

10:55:10.197 -> Predictions (DSP: 2 ms., Classification: 141 ms., Anomaly: 0 ms.):
10:55:10.233 ->   dummy (0.804688) [ x: 24, y: 8, width: 8, height: 8 ]
10:55:10.233 ->   korban (0.718750) [ x: 24, y: 16, width: 8, height: 8 ]
10:55:10.471 -> Predictions (DSP: 2 ms., Classification: 141 ms., Anomaly: 0 ms.):
10:55:10.505 ->   dummy (0.644531) [ x: 24, y: 8, width: 8, height: 8 ]
10:55:10.763 -> Predictions (DSP: 2 ms., Classification: 141 ms., Anomaly: 0 ms.):
10:55:10.763 ->   dummy (0.789062) [ x: 24, y: 8, width: 8, height: 8 ]
10:55:10.763 ->   korban (0.792969) [ x: 24, y: 16, width: 8, height: 8 ]
10:55:11.021 -> Predictions (DSP: 2 ms., Classification: 141 ms., Anomaly: 0 ms.):
10:55:11.021 ->   dummy (0.828125) [ x: 24, y: 8, width: 8, height: 8 ]
10:55:11.054 ->   korban (0.550781) [ x: 24, y: 16, width: 8, height: 8 ]

```

Gambar 18. Model Deteksi Objek Mengalami Kesalahan Deteksi

Untuk menghindari kesalahan ini, peneliti menambahkan algoritma tambahan pada model deteksi objek. Algoritma ini melibatkan pembuatan sebuah variabel *integer* yang digunakan sebagai counter. Counter ini akan menghitung berapa kali label "korban" muncul. Jika label "korban" muncul secara berturut-turut, nilai counter akan bertambah, dan akan direset jika label "dummy" muncul atau ketika tidak ada objek yang dideteksi. Ketika nilai counter mencapai 3, label "korban" akan ditampilkan pada *serial monitor*. Pada Gambar 19 dapat dilihat kode dari algoritma ini.

```

esp32_camera.ino
191 // print the predictions
192 // ei_printf("Predictions (DSP: %d ms., Classification: %d ms., Anomaly: %d ms.): \n",
193 //         result.timing.dsp, result.timing.classification, result.timing.anomaly);
194
195 #if EI_CLASSIFIER_OBJECT_DETECTION == 1
196 bool bb_found = result.bounding_boxes[0].value > 0;
197 for (size_t ix = 0; ix < result.bounding_boxes_count; ix++) {
198     auto bb = result.bounding_boxes[ix];
199     if (bb.value == 0) {
200         continue;
201     }
202     if (bb.label == "dummy") {
203         counter = 0;
204         ei_printf(" %s (%f) [ x: %u, y: %u, width: %u, height: %u ]\n", bb.label, bb.value, bb.x, bb.y, bb.width, bb.height);
205     } else if (bb.label == "korban") {
206         counter++;
207         if (counter >= 3) {
208             counter = 0;
209             ei_printf(" %s (%f) [ x: %u, y: %u, width: %u, height: %u ]\n", bb.label, bb.value, bb.x, bb.y, bb.width, bb.height);
210         }
211     }
212 }
213 if (!bb_found) {
214     counter = 0;
215     ei_printf(" No objects found\n");
216 }

```

Gambar 19. Kode Dari Algoritma counter

Meskipun algoritma ini meningkatkan akurasi, proses penghitungan menyebabkan penurunan kecepatan deteksi boneka korban waktu pemrosesan sebelumnya 141 ms kini menjadi 423 ms karena perlu menunggu label 'korban' muncul tiga kali berturut-turut, algoritma ini diharapkan dapat mengurangi kesalahan deteksi di mana boneka dummy terdeteksi sebagai boneka korban. Dengan demikian, akurasi pendeteksian dapat meningkat, yang sangat penting untuk keberhasilan dalam mengikuti Kontes Robot SAR Indonesia.

4. KESIMPULAN

Penelitian ini berhasil memenuhi tujuan utama yaitu pengembangan dan optimasi model deteksi objek untuk pengaplikasian dalam Kontes Robot SAR Indonesia. Model deteksi objek ini menggunakan platform Edge Impulse dengan algoritma FOMO (Faster Objects, More Objects) untuk proses training model, dengan tujuan utama mendeteksi boneka korban dan boneka dummy. Penelitian ini menemukan bahwa model deteksi objek yang telah ditraining memiliki akurasi tinggi pada confusion matrix, mencapai 100% untuk semua kategori. Peneliti melakukan beberapa pengujian sebelum diimplementasikan pada ESP32-cam diantaranya dengan fitur model testing, model mampu mendeteksi objek dalam dataset test dengan baik. Selanjutnya dengan fitur live classification menggunakan webcam laptop, model mengalami variasi performa yang disebabkan oleh perubahan pencahayaan, sudut pandang kamera, dan kualitas kamera. Peneliti juga mengidentifikasi bahwa performa model deteksi objek dapat dipengaruhi oleh perbedaan kamera yang digunakan selama pengumpulan dataset dan pengujian. Saat diimplementasikan pada perangkat ESP32-cam dan diuji dalam situasi nyata, model ini menunjukkan variasi dalam performanya dibuktikan dengan hasil dari 10 kali percobaan, model deteksi objek berhasil untuk mendeteksi objek boneka dummy sebanyak 8 kali dan berhasil mendeteksi objek boneka korban sebanyak 9 kali dengan waktu proses klasifikasi sebesar 141 ms. Untuk meningkatkan akurasi deteksi boneka korban, peneliti mengimplementasikan algoritma counter yang menghitung kemunculan label "korban" sebanyak 3 kali secara berturut-turut. Meskipun algoritma ini meningkatkan akurasi deteksi, tetapi terdapat penurunan kecepatan deteksi untuk boneka korban menjadi sebesar 423 ms karena proses penghitungan yang diperlukan. Secara menyeluruh, model deteksi objek yang telah dikembangkan dari penelitian ini dapat diandalkan untuk digunakan dalam Kontes Robot SAR Indonesia, meskipun masih ada beberapa tantangan yang harus diatasi. Keterbatasan penelitian ini mencakup performa yang beragam dalam kondisi dunia nyata dan penurunan kecepatan deteksi akibat penambahan algoritma counter. Oleh karena itu, Penelitian selanjutnya dapat fokus pada optimasi model dan penanganan faktor eksternal yang lebih kompleks untuk mencapai akurasi dan efisiensi yang lebih tinggi.

REFERENCES

- [1] S. Suhada and H. Helmi, "Aplikasi Mikrokontroler Atmega8535 Pada Robot Cerdas Pengangkut Tempat Sampah (Box) Menggunakan Sensor Warna Tcs3200," *JURNAL MEDIA INFORMATIKA BUDIDARMA*, vol. 3, no. 4, pp. 293–298, Oct. 2019, doi: 10.30865/mib.v3i4.1251.
- [2] I. A. Wicaksono, "Implementasi Kontrol PID Pada Gerakan Robot Line Follower Berkaki Menggunakan Sensor Kamera," *Jurnal Elektronika dan Otomasi Industri*, vol. 7, no. 3, 2020, Accessed: Jun. 25, 2024. [Online]. Available: <https://jurnal.polinema.ac.id/index.php/elkolind/article/view/4346>
- [3] Y. Pratama *et al.*, "Implementasi Kontrol PID untuk Percepatan Rotasi pada Robot Hexapod," *JURNAL MEDIA INFORMATIKA BUDIDARMA*, vol. 7, no. 4, pp. 1869–1876, Oct. 2023, doi: 10.30865/MIB.V7I4.6583.
- [4] B. Pengembangan Talenta, I. Pusat, P. Nasional, K. Pendidikan, and D. Teknologi, "Pedoman Kontes Robot Indonesia (Kri) Pendidikan Tinggi Tahun 2024." 2024
- [5] R. Y. Adhitya *et al.*, "Sistem Pengendalian Robot KRSRI Menggunakan Logika Fuzzy Sugeno Orde Nol," *Jurnal Elektronika dan Otomasi Industri*, vol. 11, no. 1, pp. 159–168, May 2024, doi: 10.33795/ELKOLIND.V11I1.4057.



- [6] K. Khairunnas, E. M. Yuniarno, and A. Zaini, “Pembuatan Modul Deteksi Objek Manusia Menggunakan Metode YOLO untuk Mobile Robot,” *Jurnal Teknik ITS*, vol. 10, no. 1, pp. A50–A55, Aug. 2021, doi: 10.12962/j23373539.v10i1.61622.
- [7] J. Sitompul, M. I. Bustami, and D. Kisbianty, “Implementasi Algoritma Haar Cascade Classifier Dalam Mendeteksi Robot Sepak Bola Beroda,” *JURNAL MEDIA INFORMATIKA BUDIDARMA*, vol. 6, no. 4, pp. 2032–2039, Oct. 2022, doi: 10.30865/mib.v6i4.3929.
- [8] Y. Abadade, A. Temouden, H. Bamoumen, N. Benamar, Y. Chtouki, and A. S. Hafid, “A Comprehensive Survey on TinyML,” *IEEE Access*, vol. 11, pp. 96892–96922, 2023, doi: 10.1109/ACCESS.2023.3294111.
- [9] Steven Reynandi Owen, “Pengembangan Aplikasi Untuk Mendeteksi Penggunaan Masker Berbasis Tinyml Di Raspberry Pi,” 2022, Accessed: Jul. 16, 2024. [Online]. Available: <http://repository.ithb.ac.id>
- [10] Pete Warden and Daniel Situnayake, *TinyML: Machine Learning With TensorFlow Lite on Arduino and Ultra-Low-Power Microcontroller*. O’Reilly Media, 2019. Accessed: Sep. 24, 2024. [Online]. Available: <https://www.google.co.id/books/edition/TinyML/tN3EDwAAQBAJ?hl=id&gbpv=0>
- [11] B. Fandidarma, R. Dwi Laksono, K. Warih, and B. Pamungkas, “Rancang Bangun Mobil Remote Control Pemantau Area berbasis IoT menggunakan ESP 32 Cam,” vol. 2, no. 1, pp. 2745–598, 2021.
- [12] S. Hymel *et al.*, “Edge Impulse: An MLOps Platform for Tiny Machine Learning,” Nov. 2022, Accessed: Jul. 16, 2024. [Online]. Available: <https://arxiv.org/abs/2212.03332v3>
- [13] C. K. Kwon, “Development of Embedded Machine Learning Finger Number Recognition Application using Edge Impulse Platform,” *Proceedings - 2023 Congress in Computer Science, Computer Engineering, and Applied Computing, CSCE 2023*, pp. 2697–2699, 2023, doi: 10.1109/CSCE60160.2023.00433.
- [14] A. Dharani, S. A. Kumar, and P. N. Patil, “Object Detection at Edge Using TinyML Models,” *SN Comput Sci*, vol. 5, no. 1, pp. 1–6, Jan. 2024, doi: 10.1007/S42979-023-02304-Z/METRICS.
- [15] I. N. Mihigo, M. Zennaro, A. Uwitonze, J. Rwigema, and M. Rovai, “On-Device IoT-Based Predictive Maintenance Analytics Model: Comparing TinyLSTM and TinyModel from Edge Impulse,” *Sensors 2022, Vol. 22, Page 5174*, vol. 22, no. 14, p. 5174, Jul. 2022, doi: 10.3390/S22145174.
- [16] A. Mellit, N. Blasutigh, and A. M. Pavan, “TinyML for fault diagnosis of Photovoltaic Modules using Edge Impulse Platform,” *11th International Conference on Smart Grid, icSmartGrid 2023*, 2023, doi: 10.1109/ICSMARTGRID58556.2023.10171088.
- [17] “FOMO: Object detection for constrained devices | Edge Impulse Documentation.” Accessed: Jul. 16, 2024. [Online]. Available: <https://docs.edgeimpulse.com/docs/edge-impulse-studio/learning-blocks/object-detection/fomo-object-detection-for-constrained-devices>
- [18] F. Marpaung, N. Khairina, R. Muliono, M. Muhathir, and S. Susilawati, “Klasifikasi Daun Teh Siap Panen Menggunakan Convolutional Neural Network Arsitektur Mobilenetv2,” *Jurnal Teknoinfo*, vol. 18, no. 1, pp. 215–225, Jan. 2024, doi: 10.33365/JTI.V18I1.3435.
- [19] Y. Hamid, S. Wani, A. B. Soomro, A. A. Alwan, and Y. Gulzar, “Smart Seed Classification System based on MobileNetV2 Architecture,” *Proceedings of 2022 2nd International Conference on Computing and Information Technology, ICCIT 2022*, pp. 217–222, 2022, doi: 10.1109/ICCIT52419.2022.9711662.
- [20] A. Nada Nafisa, E. Nia Devina Br Purba, F. Aulia Alfarisi Harahap, N. Adawiyah Putri, I. Komputer, and F. Matematika dan Ilmu Pengetahuan Alam, “Implementasi Algoritma Convolutional Neural Network Arsitektur Model Mobilenetv2 Dalam Klasifikasi Penyakit Tumor Otak Glioma, Pituitary Dan Meningioma,” *Jurnal Teknologi Informasi, Komputer, dan Aplikasinya (JTika)*, vol. 5, no. 1, pp. 53–61, Mar. 2023, doi: 10.29303/JTIKA.V5I1.234.
- [21] J. R. Ziliwu, G. C. Setyawan, and H. Budiati, “Penerapan ESP32-CAM dan TinyML dalam Klasifikasi Gambar Buah dan Sayuran,” *Jurnal Ilmiah Teknik Informatika dan Sistem Informasi*, vol. 13, no. 1, pp. 584–595, Apr. 2024, doi: 10.35889/JUTISI.V13I1.1869.
- [22] V. Gutti and R. Karthi, “Real Time Classification of Fruits and Vegetables Deployed on Low Power Embedded Devices Using Tiny ML,” *Lecture Notes in Networks and Systems*, vol. 514 LNNS, pp. 347–359, 2022, doi: 10.1007/978-3-031-12413-6_27.
- [23] D. Nurdiansyah, S. Satrianansyah, and A. Sobri, “SMART ROBOT OBJECT DETECTION MENGGUNAKAN ESP-32 CAM,” *Jurnal Tekinkom (Teknik Informasi dan Komputer)*, vol. 7, no. 1, pp. 272–280, Jun. 2024, doi: 10.37600/TEKINKOM.V7I1.1296.
- [24] A. Rizqi Aprilianto, S. H. Mulyanto, S. Fatimah, H. Nurdiansari, A. Kasan Gupron, and P. Pelayaran Surabaya, “Rancang Bangun Kapal Tanpa Awak Guna Mendeteksi Navigation Lamp Untuk Menghindari Tubrukan,” *Ocean Engineering : Jurnal Ilmu Teknik dan Teknologi Maritim*, vol. 3, no. 2, pp. 63–79, Jun. 2024, doi: 10.58192/OCEAN.V3I2.2215.
- [25] S. A. Nugroho *et al.*, “Rancang Bangun Sistem Deteksi Label Kardus Berbasis Model Kecerdasan Buatan YOLO dan EasyOCR serta ESP32-CAM,” *JURNAL TEKNIK ELEKTRO*, vol. 11, no. 2, pp. 190–200, Jun. 2022, doi: 10.26740/JTE.V11N2.P190-200.
- [26] S. Samsugi, Z. Mardiyansyah, and A. Nurkholis, “Sistem Pengontrol Irigasi Otomatis Menggunakan Mikrokontroler Arduino Uno,” *Jurnal Teknologi dan Sistem Tertanam*, vol. 1, no. 1, pp. 17–22, Aug. 2020, doi: 10.33365/JTST.V1I1.719.
- [27] Nisa Hanum Harani and Miftahul Hasanah, *Deteksi Objek Dan Pengenalan Karakter Plat Nomor Kendaraan Indonesia Berbasis Python*. Kreatif Industri Nusantara, 2020. Accessed: Sep. 24, 2024. [Online]. Available: https://books.google.co.id/books?hl=id&lr=&id=saD6DwAAQBAJ&oi=fnd&pg=PR6&dq=buku+deteksi+objek&ots=GJY75XPPxR&sig=0gMJSIcsmiN9QDZ_SiXE-oCImEw&redir_esc=y#v=onepage&q=buku%20deteksi%20objek&f=false
- [28] P. N. Dacipta and R. E. Putra, “Sistem Klasifikasi Limbah Menggunakan Metode Convolutional Neural Network (CNN) Pada Webservice Berbasis Framework Flask,” *Journal of Informatics and Computer Science (JINACS)*, vol. 3, no. 04, pp. 394–402, May 2022, doi: 10.26740/JINACS.V3N04.P394-402.