Nugroho, B., Azhar, N. F., Pratama , B. M., Syakbani, A. R. A., Wibowo, D. R. and Syam, A. M. A. R. (2025) "Implementation of Sparrow Pest Detection Using YOLOv8 Method on Raspberry Pi and Google Coral USB Accelerator", Sebatik, 29(1).
Submitted: 2024-12-18 Accepted: 2025-02-27 Published: 2025-06-30

Sebatik Vol. 29 No. 1 June 2025
ISSN: 1410-3737(p) 2621-069X(e)

DOI:10.46984/sebatik.v29i1.2558

Implementation of Sparrow Pest Detection Using YOLOv8 Method on Raspberry Pi and Google Coral USB Accelerator

Bowo Nugroho[1)], Nur Fajri Azhar[2)], Boby Mugi Pratama[3)],Ahmad Rusdianto Andarina Syakbani[4)], Darrell Rajendra Wibowo[5)], dan **Andi Muhammad Agung Ramadhani Syam[6)]**

[1,2,3,4,5,6] Informatika, Institut Teknologi Kalimantan
[1,2,3,4,5,6] Jl. Soekarno Hatta, Kota Balikpapan, 76127
E-mail: bowo.nugroho@lecturer.itk.ac.id[1)], fajri@lecturer.itk.ac.id[2)], bmpratama@lecturer.itk.ac.id[3)], 11211005@student.itk.ac.id[4)], 11211028@student.itk.ac.id[5)], 11211011@student.itk.ac.id[6)]

ABSTRACT

Sparrows are one of the most costly pests for farmers, as they can reduce rice yields by 50-60%. Traditional control methods, such as the use of scarecrows, windmills, and pesticides, are often ineffective or cause negative impacts on the environment, such as damage to ecosystems and human health. To overcome this problem, YOLOv8-based object detection technology offers a modern solution to automatically detect bird pests with a high level of accuracy. This research aims to implement the YOLOv8 model on power-efficient embedded devices, such as Raspberry Pi 4 and Google Coral USB TPU Accelerator, to support real-time sparrow detection at an affordable cost. The research was conducted through three main stages, namely collecting bird image datasets to support model training, training the YOLOv8n model to produce reliable bird pest detection, and implementing the model on embedded devices with and without TPU accelerators to evaluate detection performance. The evaluation results show that the YOLOv8 model has high performance with precision 0.91, recall 0.86, mAP50 0.92, and mAP50-95 0.59 after being trained for 300 epochs. Implementation on Raspberry Pi 4 without accelerator only resulted in an inference speed of 0.39 Frame Per Second, while with Google Coral USB TPU, the speed increased significantly to 7 Frame Per Second. This proves that TPU accelerators are highly effective in supporting real-time object detection. This technology is expected to help farmers protect crops efficiently, reduce losses due to pests, support sustainable agricultural productivity, and contribute to the overall improvement of food security.

***Keywords**: Sparrow pest, YOLOv8, Object detection, Raspberry Pi 4, Google Coral TPU*

Implementasi Deteksi Hama Burung Pipit dengan Metode YOLOv8 pada Raspberry Pi dan Google Coral USB Accelerator

ABSTRAK

Burung pipit merupakan salah satu hama yang sangat merugikan petani, karena dapat menurunkan hasil panen padi hingga 50-60%. Metode pengendalian tradisional, seperti penggunaan orang-orangan sawah, kincir angin, dan pestisida, sering kali kurang efektif atau menimbulkan dampak negatif terhadap lingkungan, seperti kerusakan ekosistem dan kesehatan manusia. Untuk mengatasi permasalahan ini, teknologi deteksi objek berbasis YOLOv8 menawarkan solusi modern untuk mendeteksi hama burung secara otomatis dengan tingkat akurasi yang tinggi. Penelitian ini bertujuan untuk mengimplementasikan model YOLOv8 pada perangkat tertanam hemat daya, seperti Raspberry Pi 4 dan Google Coral USB TPU *Accelerator*, guna mendukung deteksi burung pipit secara *real-time* dengan biaya yang terjangkau. Penelitian dilakukan melalui tiga tahap utama, yaitu pengumpulan *dataset* citra burung untuk mendukung pelatihan model, pelatihan model YOLOv8n untuk menghasilkan deteksi hama burung yang andal, serta implementasi model pada perangkat tertanam dengan dan tanpa akselerator TPU untuk mengevaluasi performa deteksi. Hasil evaluasi menunjukkan bahwa model YOLOv8 memiliki performa tinggi dengan *precision* 0,91, *recall* 0,86, mAP50 0,92, dan mAP50-95 0,59 setelah dilatih selama 300 epoch. Implementasi pada Raspberry Pi 4 tanpa akselerator hanya menghasilkan kecepatan inferensi 0,39 *Frame Per Second*, sementara dengan Google Coral USB TPU, kecepatan meningkat signifikan menjadi 7 *Frame Per Second*. Hal ini membuktikan bahwa akselerator TPU sangat efektif dalam mendukung deteksi objek secara *real-time*. Teknologi ini diharapkan dapat membantu petani melindungi hasil panen secara efisien, mengurangi kerugian akibat hama, mendukung produktivitas pertanian yang berkelanjutan, serta berkontribusi pada peningkatan ketahanan pangan secara keseluruhan.

Kata Kunci: Hama Burung Pipit, YOLOv8, Deteksi Objek, Raspberry Pi 4, Google Coral TPU

1. PENDAHULUAN

Burung pipit merupakan hama yang sangat merugikan petani. Burung pipit menyerang tanaman pada saat tumbuhan padi dewasa yang berumur 60-90 hari. Burung pipit dapat mengurangi produksi padi hingga 50-60% (Hamakonda et al., 2023; Kahfiani, 2023). Burung pipit yang menyerang secara berkelompok dapat menyebabkan tanaman padi mengering dan menghasilkan bulir yang kosong. Setiap burung rata-rata mengonsumsi 5 gram padi per hari. Akibatnya, petani mengalami kerugian berupa penurunan hasil panen serta peningkatan biaya produksi untuk mengatasi gangguan hama tersebut (Yenisbar & Asmah, 2021) .

Pengendalian hama burung menjadi tantangan bagi petani, yang mencoba berbagai metode untuk mengatasinya. Beberapa cara yang biasa digunakan antara lain adalah penyemprotan pestisida, pemasangan boneka orang-orangan di sawah, dan penggunaan kincir angin. Namun, metode-metode ini sering kali kurang efektif. Burung memiliki kemampuan beradaptasi, sehingga alat seperti orang-orangan sawah dan kincir angin lambat laun kehilangan efektivitasnya. Sementara itu, penggunaan pestisida kimia menawarkan solusi lain, tetapi membawa resiko besar terhadap keseimbangan ekosistem, lingkungan, serta kesehatan manusia (Nuraeni dkk., 2024).

Kemampuan untuk mendeteksi dan mengklasifikasikan objek dalam gambar atau video sangat penting dalam mengembangkan sistem otomatisasi yang dapat beroperasi secara mandiri, terutama dalam bidang pertanian. Dengan sistem deteksi otomatis yang mampu mengidentifikasi dan mengklasifikasikan burung sebagai ancaman, petani dapat mengelola hama ini secara lebih efisien. Teknologi ini memungkinkan tindakan pencegahan yang cepat dan tepat, seperti mengaktifkan perangkat penangkal atau alarm, sehingga dapat meminimalkan kerusakan pada tanaman. Oleh karena itu, kemampuan deteksi visual yang akurat sangat penting untuk mendukung upaya pengendalian hama burung dan memastikan produktivitas pertanian tetap optimal.

Deteksi objek merupakan salah satu permasalahan fundamental dan menantang dari visi komputer. Teknologi tradisional seperti Viola Jones (VJ) *Detectors* merupakan terobosan pertama dalam deteksi wajah, yang menjadi langkah maju dalam teknologi deteksi objek (Y.-Q. Wang, 2014). Selain itu *Histogram of Oriented Gradients* (HOG) adalah metode ekstraksi fitur dalam visi komputer yang menganalisis distribusi arah gradien intensitas di dalam gambar (Dalal & Triggs, t.t.). Detektor HOG menjadi dasar utama bagi berbagai model pendeteksian objek dan aplikasi visi komputer lainnya (Felzenszwalb dkk., 2010). Namun, kedua pendekatan tersebut memiliki kelemahan dalam menghadapi variasi yang tinggi pada bentuk objek, ukuran, gangguan *noise*, dan kondisi pencahayaan. Sebagai solusi, *Convolutional Neural Networks* (CNN) menawarkan keunggulan dalam mengenali pola kompleks secara otomatis melalui ekstraksi fitur hierarkis, sehingga lebih handal dalam berbagai kondisi dan variasi objek (Bhatt dkk., 2021). Perkembangan terbaru dalam deteksi objek mencakup pengembangan *framework* detektor satu tahap (*one-stage*) dan dua tahap (*two-stage*), di mana detektor dua tahap menganggap deteksi sebagai proses "kasar ke halus", sementara detektor satu tahap menyelesaikannya "dalam satu langkah" (Zou dkk., 2023). Arsitektur *two-stage detectors* tahap membagi proses deteksi menjadi dua tahap, yaitu proposal wilayah dan klasifikasi. Beberapa model terkenal yang mengadopsi arsitektur ini antara lain RCNN, SPPNet, Fast RCNN, Faster RCNN, dan *Feature Pyramid* Networks (FPN), yang menjadi dasar dalam pengembangan model deteksi objek terkini (Girshick dkk., 2014; Lin dkk., 2017; Purkait dkk., 2017; Ren dkk., 2017; X. Wang dkk., 2017; Zou dkk., 2023). Selanjutnya, *one-stage detectors* menyederhanakan proses deteksi dengan melakukan deteksi objek secara langsung dalam satu langkah, tanpa memerlukan tahap proposal wilayah terpisah. Contoh terkenal dari detektor satu tahap yang memberikan kecepatan dan efisiensi lebih tinggi termasuk model-model seperti YOLO (You Only Look Once), yang memiliki berbagai varian seperti YOLOv3, YOLOv4, YOLOv5, YOLOv6, YOLOv7, YOLOv8, serta SSD (*Single Shot Multibox Detector*) (Bochkovskiy dkk., 2020; Diwan dkk., 2023; Karthi dkk., 2021; Li dkk., 2022; Liu dkk., 2016; Redmon & Farhadi, 2018; Sohan dkk., 2024; C.-Y. Wang dkk., 2022).

Untuk membantu petani dalam mendeteksi hama burung di sawah secara *real-time*, dibutuhkan solusi yang efisien dan terjangkau, yang menggabungkan teknologi deteksi objek dengan perangkat tertanam hemat daya. Model deteksi objek berbasis YOLOv8, yang telah terbukti efektif dalam berbagai aplikasi deteksi objek, diimplementasikan pada Raspberry Pi 4, menjadikannya pilihan yang sangat tepat untuk tujuan ini. Raspberry Pi 4 dipilih karena ukurannya yang kompak, konsumsi daya yang rendah, serta kemampuannya untuk menjalankan model deteksi objek secara *real-time* dengan performa yang cukup baik meskipun memiliki keterbatasan daya komputasi (Anandan dkk., 2020; Benkirat, 2023; Jiang dkk., 2020; Jolles, 2021). Implementasi pada Raspberry Pi 4 memungkinkan deteksi hama burung yang efisien tanpa memerlukan perangkat keras mahal, membuat solusi ini lebih terjangkau dan mudah diakses oleh petani di berbagai wilayah, terutama di daerah yang memiliki keterbatasan anggaran.

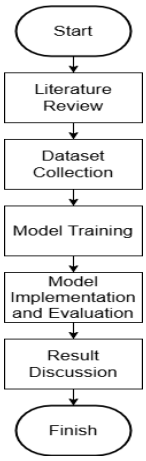
2. RUANG LINGKUP

Ruang lingkup permasalahan dalam penelitian ini mencakup analisis performa model YOLOv8n dalam mendeteksi objek burung, yang merupakan hama pada lingkungan tertentu. Penelitian ini fokus pada implementasi model pada perangkat tertanam yang irit

daya, yaitu Raspberry Pi4 dan Google Coral USB TPU Accelerator, untuk mengevaluasi kemampuannya dalam mendeteksi objek secara *real-time*. Kajian dilakukan untuk mengukur sejauh mana model YOLOv8n dapat dioptimalkan pada perangkat dengan sumber daya terbatas, mencakup aspek akurasi deteksi, dan waktu deteksi setiap gambarnya. Dengan batasan pada penggunaan perangkat hemat daya, penelitian ini bertujuan menghasilkan solusi praktis yang mendukung deteksi hama burung secara efisien dan real-time, sekaligus memastikan bahwa model yang diimplementasikan dapat bekerja secara andal dalam lingkungan sistem benam.

3. BAHAN DAN METODE

Penelitian ini dimuai dari studi literatur. Kemudian penelitian terdiri dari tiga tahap utama, yaitu pengumpulan dataset, pelatihan model, dan implementasi model. Yang selanjutnya melakukan evaluasi dan diskusi hasil model. Proses diagram alir penelitian dapat dilihat pada gambar 1.



Gambar 1. Diagram Alir Penelitian
Figure 1. Research Flowchart

Mengacu pada diagram alir yang terdapat di gambar 1, berikut penjabaran tahapan penelitian:

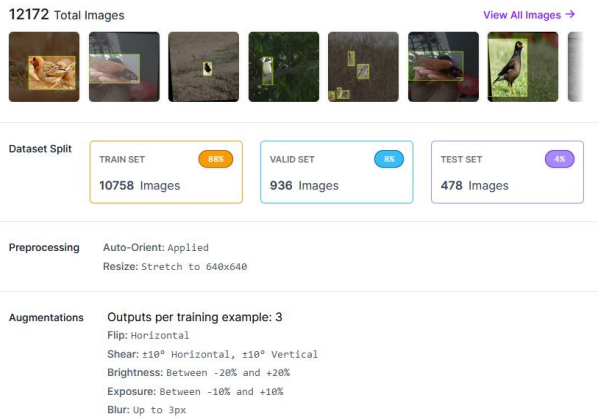
1. Studi Literatur: Dilakukan untuk mendalami dan memahami sistem yang digunakan. Mengeksplorasi pembelajaran mesin, pengumpulan dataset, evaluasi dataset yang baik untuk digunakan, dan memahami dasar model serta arsitetur YOLO.
2. Pengumpulan *Dataset*: Melibatkan pengumpulan dan anotasi data citra burung yang relevan untuk skenario deteksi hama burung.
3. Pelatihan Model: Model YOLOv8n dilatih menggunakan dataset yang telah disiapkan untuk menghasilkan model yang mampu mendeteksi objek burung dengan akurasi tinggi.
4. Evaluasi dan Implementasi Model: Model dan Implementasi Model yang telah dilatih dioptimalkan untuk perangkat tertanam, yaitu Raspberry Pi 4 dengan dan tanpa Google Coral USB TPU Accelerator. Evaluasi performa dilakukan berdasarkan perbandingan waktu inferensi antara kedua konfigurasi tersebut untuk menilai efektivitas akselerator dalam mendukung deteksi real-time.
5. Diskusi Hasil: Dilakukan analisis akurasi deteksi model YOLOv8n menggunakan metrik *precision*, *recall*, dan mAP serta membahas hasil *Frame Per Second* (FPS) yang didapat. Performa inferensi dibandingkan berdasarkan waktu pemrosesan pada Raspberry Pi 4 dengan dan tanpa Google Coral USB TPU Accelerator.

4. PEMBAHASAN

Pembahasan mencakup tahapan implementasi, pengujian, dan evaluasi sistem deteksi hama burung menggunakan model YOLOv8n pada perangkat Raspberry Pi 4 dengan dan tanpa akselerator Google Coral USB TPU. Penjelasan meliputi proses pengumpulan dataset, pelatihan model, evaluasi performa, serta analisis hasil inferensi dalam skenario deteksi real-time.

4.1. Pengumpulan *Dataset*

Dataset dalam penelitian ini terdiri dari 12.172 gambar yang diambil dari berbagai sumber yang relevan dengan konteks deteksi objek burung. Gambar-gambar tersebut kemudian dibagi menjadi tiga *subset* utama yaitu data pelatihan, validasi, dan pengujian, untuk memastikan model yang dilatih mampu mengatasi data baru dengan baik. Data pelatihan digunakan untuk membantu model mempelajari pola-pola dalam gambar, sementara data validasi berperan dalam mengoptimalkan *hyperparameter* dan mencegah *overfitting*. Selain itu, data pengujian dimanfaatkan untuk mengevaluasi kemampuan model dalam menggeneralisasi data yang belum pernah dilihat sebelumnya, sehingga memberikan penilaian objektif terhadap kinerjanya. Pembagian *dataset* yang tepat sangat penting untuk mencapai akurasi tinggi dan kemampuan generalisasi model dalam berbagai skenario.



Gambar 2. Sebaran Dataset
Figure 2. Dataset Distribution

Sebaran dataset ditampilkan pada gambar 2, di mana 88% atau 10.758 gambar dialokasikan untuk data pelatihan, 8% atau 936 gambar untuk data validasi, dan 4% atau 478 gambar digunakan untuk pengujian model. Data pelatihan membantu model mempelajari objek dan pola, sementara data validasi digunakan untuk mengukur kinerja model dan menyesuaikan *hyperparameter* selama proses pelatihan. Terakhir, data pengujian digunakan untuk mengevaluasi kemampuan model dalam menghadapi data baru, memberikan gambaran mengenai kinerja model di dunia nyata.

4.2. Pelatihan Model

Pada tahap sebelum pelatihan, dilakukan *preprocessing* untuk mempersiapkan data gambar. Proses ini meliputi penyesuaian orientasi gambar menggunakan *auto-orient* dan pengubahan ukuran gambar menjadi 640x640 piksel sesuai kebutuhan input model. Untuk meningkatkan variasi data dan mencegah *overfitting*, augmentasi data dilakukan dengan teknik seperti *flipping horizontal*, *shear*, penyesuaian kecerahan, *exposure*, dan penerapan efek *blur*. Setiap gambar juga dilabeli dengan *bounding box* yang mengelilingi objek burung serta diberi label kelas agar model mampu mengenali kategori objek dengan baik.

Tahap pelatihan menggunakan *dataset* yang telah diproses, di mana model YOLOv8n secara iteratif mendeteksi objek pada gambar pelatihan dan mengevaluasi hasil deteksinya dengan membandingkannya dengan label *ground truth*. Proses ini berlangsung selama 300 *epoch* untuk mengoptimalkan performa model dalam mendeteksi objek burung secara akurat pada berbagai kondisi gambar.

Model menggunakan *hyperparameter* yang dioptimalkan, termasuk *learning rate*, momentum, dan pengaturan lainnya. *Hyperparameter* ini telah dioptimalkan dan dicantumkan dalam Tabel 1, di mana setiap pengaturan dipilih secara cermat untuk mencapai kinerja model yang optimal. Dengan pendekatan sistematis ini, model diharapkan memiliki kemampuan generalisasi yang baik dan mampu mendeteksi objek burung dengan akurat dalam kondisi dunia nyata.

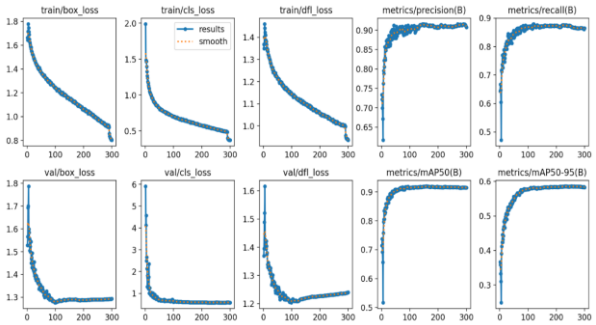
Tabel 1. *Hyperparameter* Pelatihan Model
Table 1. Model Training Hyperparameters

Hyperparameter	**Value**
ImageSize	640x640
Optimizer	Stochastic Gradient Descent
Learning Rate	0.01
Momentum	0.937
IoU	0.7
Epoch	300
Batch Size	64
Weight Decay	0.0005

4.3. Evaluasi dan Implementasi Model

Setelah tahap pelatihan selesai, model dievaluasi menggunakan *dataset* validasi untuk mengukur kemampuannya mendeteksi objek pada gambar-gambar baru yang belum pernah dilihat sebelumnya. Evaluasi ini dilakukan dengan menggunakan metrik seperti *Precision*, *Recall*, dan *Mean Average Precision* (mAP) untuk menilai performa model. Jika hasil evaluasi menunjukkan bahwa model masih memerlukan perbaikan, maka langkah-langkah seperti *preprocessing*, konfigurasi, atau proses pelatihan diulang dengan penyesuaian tertentu hingga kinerja yang diinginkan tercapai.

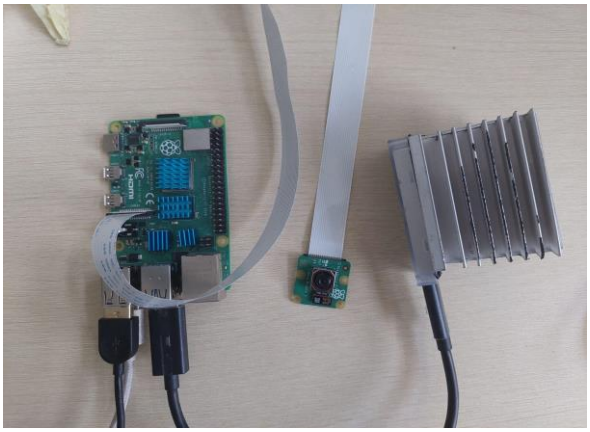
Gambar 3 menunjukkan diagram yang menggambarkan hasil pelatihan model. Setelah melatih model selama 300 *epoch*, hasil yang diperoleh adalah sebagai berikut: training box loss sebesar 0,74, training *class loss* sebesar 0,34, presisi sebesar 0,91, *recall* sebesar 0,86, mAP50 sebesar 0,92, dan mAP50-95 sebesar 0,59. Pada tahap validasi, *validation box loss* tercatat sebesar 1,29, *validation class loss* sebesar 0,58, dan *validation distribution focal loss* sebesar 1,34. Hasil ini menunjukkan bahwa model YOLOv8 memiliki performa yang baik dalam mendeteksi dan mengklasifikasikan objek. Presisi yang tinggi (0,91) dan *recall* sebesar 0,86 mengindikasikan bahwa model mampu mendeteksi sebagian besar objek yang relevan sekaligus meminimalkan kesalahan prediksi (*false positives*). Nilai mAP50 sebesar 0,92 menegaskan tingkat akurasi yang tinggi pada ambang batas tertentu, sementara nilai mAP50-95 sebesar 0,59 mencerminkan kemampuan model untuk mempertahankan performa di berbagai tingkat ambang batas.



Gambar 3. Hasil Proses Pelatihan Model
Figure 3. Model Training Process Results

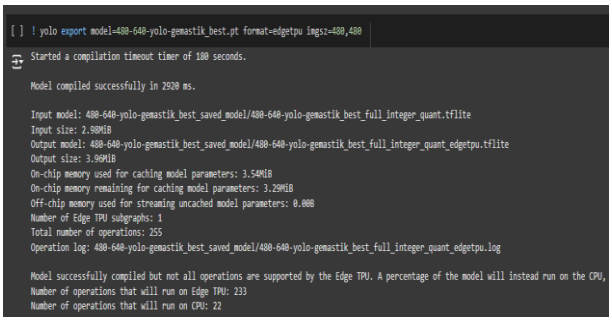
Setelah model selesai dibangun dan menghasilkan performa yang baik selama percobaan, model diimplementasikan ke dalam alat pendeteksi burung menggunakan perangkat keras. gambar 4 menunjukkan alat yang digunakan pada penelitian ini, di mana model yang telah dibangun dan menunjukkan performa baik selama percobaan diimplementasikan ke dalam alat pendeteksi burung menggunakan perangkat keras tertentu. Alat ini terdiri dari Raspberry Pi 4 sebagai pusat pemrosesan utama, dilengkapi dengan *heatsink* untuk

mencegah panas berlebih saat digunakan. Kamera Raspberry Pi terhubung melalui kabel pita untuk menangkap gambar atau video sebagai *input* ke sistem. Selain itu, Google Coral USB TPU Accelerator digunakan untuk mempercepat proses inferensi model deteksi dengan dukungan pemrosesan *tensor* yang efisien, memastikan sistem mampu menjalankan deteksi objek secara *real-time.*



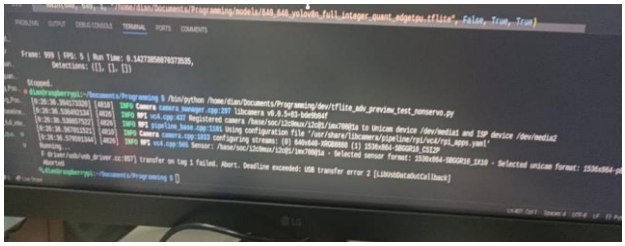
Gambar 4. Perangkat Keras yang Digunakan
Figure 4. Hardware Used

Sebelum diimplementasikan pada Raspberry Pi dengan Google Coral TPU, model YOLOv8n yang telah dilatih perlu melalui proses konversi ke format yang kompatibel. Model awal yang masih dalam format *.pt* (PyTorch) harus diubah menjadi format *.edgetpu.tflite* (EdgeTPU TensorFlow Lite) untuk memastikan kompatibilitas dan memanfaatkan akselerasi perangkat keras dari Google Coral TPU. Proses konversi ini melibatkan penyesuaian ukuran gambar *input* menjadi 480x480 piksel dan penerapan format khusus yang mendukung optimalisasi inferensi pada perangkat tertanam. Setelah model berhasil dikonversi, langkah selanjutnya adalah mengimplementasikan model tersebut pada Raspberry Pi untuk menjalankan proses deteksi objek burung secara *real-time* dengan kinerja yang lebih efisien. Gambar 5 menggambarkan tahapan proses konversi model secara keseluruhan.



Gambar 5. Proses Konversi Model
Figure 5. Model Conversion Process

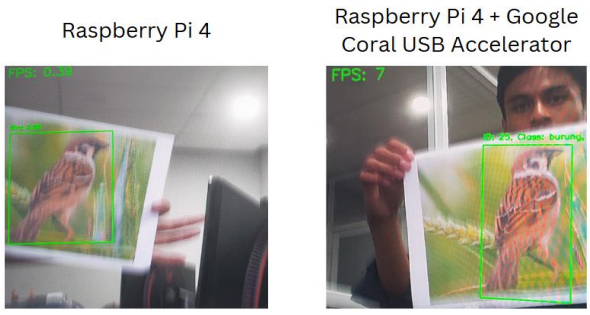
Gambar 6 merupakan pesan *error* jika mengkonversi model lebih dari resolusi gambar 480x480. Program tidak dapat bekerja jika *ImageSize* lebih besar dari ukuran 480x480. Sebagai contoh program dipasang dengan menggunakan *ImageSize* dengan besar 640x640 maka akan menghasilkan *error* pada TPU yang digunakan dikarenakan model yang terlalu besar.



Gambar 6. Pesan Error Model Terlalu besar
Figure 6. Error Message Model is Too Big

Implementasi model dilakukan dengan mengintegrasikan model yang telah dilatih ke dalam aplikasi yang dirancang untuk mendeteksi burung. Kamera akan menangkap gambar, yang kemudian diproses untuk menentukan keberadaan burung di dalamnya menggunakan metode YOLO serta model yang telah dibangun sebelumnya. Hasil deteksi menunjukkan adanya burung, dan program akan menampilkan lokasi burung tersebut dengan menggunakan kotak pembatas (*bounding box*).

Gambar 7 menunjukkan hasil pengujian implementasi kedua konfigurasi, baik tanpa akselerator maupun dengan Google Coral USB TPU Accelerator, berupa tangkapan layar *Frame per Second* (FPS) yang menggambarkan peningkatan performa inferensi dengan akselerator TPU. Pengujian menunjukkan bahwa model yang dijalankan tanpa Google Coral menghasilkan rata-rata 0,39 FPS, yang mengindikasikan keterbatasan Raspberry Pi4 dalam mendukung deteksi objek real-time. Sebaliknya, dengan Google Coral, FPS meningkat drastis menjadi 7, menunjukkan kontribusi signifikan akselerator TPU dalam mempercepat inferensi dan memungkinkan deteksi burung lebih cepat dan efisien.



Gambar 7. Perbandingan Performa Deteksi Objek
Figure 7. Object Detection Performance Comparison

5. KESIMPULAN

Model YOLOv8n menunjukkan performa yang baik dalam mendeteksi dan mengklasifikasikan objek burung. Setelah dilatih selama 300 *epoch*, hasil evaluasi pada dataset pelatihan menunjukkan training box loss sebesar 0,74 dan *training class loss* sebesar 0,34, yang menunjukkan bahwa model mampu meminimalkan kesalahan dalam mendeteksi objek dan mengklasifikasikannya dengan baik. Selain itu, nilai *precision* sebesar 0,91 dan *recall* sebesar 0,86 mengindikasikan bahwa model memiliki kemampuan tinggi dalam mendeteksi sebagian besar objek yang relevan sambil meminimalkan kesalahan prediksi (false positives). Nilai mAP50 sebesar 0,92 menegaskan tingkat akurasi yang tinggi pada ambang batas tertentu, sementara nilai mAP50-95 sebesar 0,59 menunjukkan kemampuan model untuk mempertahankan performa yang baik di berbagai tingkat ambang batas.

Implementasi pada perangkat Raspberry Pi dengan Google Coral TPU menunjukkan peningkatan performa deteksi yang signifikan. Tanpa menggunakan akselerator, model hanya mampu mencapai 0,39 FPS, yang menunjukkan keterbatasan Raspberry Pi 4 dalam mendukung deteksi objek secara *real-time*. Namun, ketika model diimplementasikan dengan menggunakan Google Coral USB TPU Accelerator, FPS meningkat menjadi 7, yang menunjukkan bahwa akselerator TPU sangat membantu dalam mempercepat proses inferensi. Dengan demikian, model yang dilatih dengan baik mampu memberikan performa deteksi yang efisien dan cepat, terutama ketika dipadukan dengan akselerator TPU untuk aplikasi deteksi objek *real-time* pada perangkat tertanam.

6. SARAN

Penelitian selanjutnya dapat mengoptimalkan performa deteksi objek dengan menggunakan model yang lebih ringan atau menerapkan teknik kompresi model untuk meningkatkan kecepatan inferensi tanpa mengurangi akurasi. Penggunaan akselerator perangkat keras seperti NVIDIA Jetson Nano juga dapat dieksplorasi guna mempercepat proses inferensi pada perangkat tertanam dengan keterbatasan daya. Variasi augmentasi data perlu ditingkatkan untuk mendukung kemampuan generalisasi model dalam mendeteksi burung di berbagai kondisi lingkungan. Selain itu, penggunaan dataset yang lebih besar dan beragam serta penerapan teknik fine-tuning pada model diharapkan mampu meningkatkan akurasi deteksi lebih lanjut. Penelitian mendatang juga dapat menguji implementasi model pada lingkungan yang lebih kompleks untuk menilai kinerja model dalam menghadapi skenario dunia nyata yang lebih menantang.

7. DAFTAR PUSTAKA

Anandan, M., Manikandan, M., & Karthick, T. (2020). Advanced Indoor and Outdoor Navigation System for Blind People Using Raspberry-Pi. *Journal of Internet Technology*, *21*(1), 183–195.

Benkirat, I. (2023). *Design and implementation of a Real-Time Object Detection and Understanding System using Deep Neural Network to assist the visually impaired persons, running on Raspberry Pi*. university of guelma.

Bhatt, D., Patel, C., Talsania, H., Patel, J., Vaghela, R., Pandya, S., Modi, K., & Ghayvat, H. (2021). CNN Variants for Computer Vision: History, Architecture, Application, Challenges and Future Scope. *Electronics*, *10*(20), 2470. https://doi.org/10.3390/electronics10202470

Bochkovskiy, A., Wang, C.-Y., & Liao, H.-Y. M. (2020). *YOLOv4: Optimal Speed and Accuracy of Object Detection*. https://arxiv.org/abs/2004.10934

Dalal, N., & Triggs, B. (n.d.). Histograms of Oriented Gradients for Human Detection. *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, 886–893. https://doi.org/10.1109/CVPR.2005.177

Diwan, T., Anirudh, G., & Tembhurne, J. V. (2023). Object detection using YOLO: challenges, architectural successors, datasets and applications. *Multimedia Tools and Applications*, *82*(6), 9243–9275. https://doi.org/10.1007/s11042-022-13644-y

Felzenszwalb, P. F., Girshick, R. B., & McAllester, D. (2010). Cascade object detection with deformable part models. *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2241–2248. https://doi.org/10.1109/CVPR.2010.5539906

Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2014). Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. *2014 IEEE Conference on Computer Vision and Pattern Recognition*, 580–587. https://doi.org/10.1109/CVPR.2014.81

Hamakonda, U. A., Taus, I., Puspita, V. A., Lea, V. C., Bure, V., Soba, K., & Mamo, N. (2023). IDENTIFIKASI HAMA PADA TANAMAN PADI INPARI 30 (Oriza sativa L) DI DESA PAPE KECAMATAN BAJAWA KABUPATEN NGADA. *Jurnal Pertanian Agros*, *25*(4), 3635–3639.

Jiang, Z., Zhao, L., Li, S., & Jia, Y. (2020). *Real-time object detection method based on improved YOLOv4-tiny*. https://arxiv.org/abs/2011.04244

Jolles, J. W. (2021). Broad-scale applications of the Raspberry Pi: A review and guide for biologists. *Methods in Ecology and Evolution*, *12*(9), 1562–1579. https://doi.org/10.1111/2041-210X.13652

Kahfiani, A. (2023). *Jaring, Cara Jitu Kendalikan Hama Burung Pipit*. *2*(Vol. 2 No. 1 (2023): April 2023), 16–20.

https://epublikasi.pertanian.go.id/berkala/btip/article/view/3487/3493

Karthi, M., Muthulakshmi, V., Priscilla, R., Praveen, P., & Vanisri, K. (2021). Evolution of YOLO-V5 Algorithm for Object Detection: Automated Detection of Library Books and Performace validation of Dataset. *2021 International Conference on Innovative Computing, Intelligent Communication and Smart Electrical Systems (ICSES)*, 1–6. https://doi.org/10.1109/ICSES52305.2021.9633834

Li, C., Li, L., Jiang, H., Weng, K., Geng, Y., Li, L., Ke, Z., Li, Q., Cheng, M., Nie, W., Li, Y., Zhang, B., Liang, Y., Zhou, L., Xu, X., Chu, X., Wei, X., & Wei, X. (2022). *YOLOv6: A Single-Stage Object Detection Framework for Industrial Applications*. https://arxiv.org/abs/2209.02976

Lin, T.-Y., Dollar, P., Girshick, R., He, K., Hariharan, B., & Belongie, S. (2017). Feature Pyramid Networks for Object Detection. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 936–944. https://doi.org/10.1109/CVPR.2017.106

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., & Berg, A. C. (2016). SSD: Single Shot MultiBox Detector. In *Computer Vision – ECCV 2016* (pp. 21–37). Springer International Publishing. https://doi.org/10.1007/978-3-319-46448-0_2

Nuraeni, I., Alfauzi, R. D., Sopyan, S., & Wahyudi, D. (2024). 33 Penggunaan Jaring Sebagai Pengendali Hama Burung Pipit (Estrildid Finches) pada Tanaman padi (Oryza Sativa L.) di Kampung Tinggarjaya Hilir Desa Cimaung. *PROCEEDINGS UIN SUNAN GUNUNG DJATI BANDUNG*, *4*(9), 278–288.

Purkait, P., Zhao, C., & Zach, C. (2017). *SPP-Net: Deep Absolute Pose Regression with Synthetic Views*. https://arxiv.org/abs/1712.03452

Redmon, J., & Farhadi, A. (2018). *YOLOv3: An Incremental Improvement*. https://arxiv.org/abs/1804.02767

Ren, S., He, K., Girshick, R., & Sun, J. (2017). Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *39*(6), 1137–1149. https://doi.org/10.1109/TPAMI.2016.2577031

Sohan, M., Sai Ram, T., & Rami Reddy, Ch. V. (2024). *A Review on YOLOv8 and Its Advancements* (pp. 529–545). https://doi.org/10.1007/978-981-99-7962-2_39

Wang, C.-Y., Bochkovskiy, A., & Liao, H.-Y. M. (2022). *YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors*. https://arxiv.org/abs/2207.02696

Wang, X., Shrivastava, A., & Gupta, A. (2017). *A-Fast-RCNN: Hard Positive Generation via Adversary for Object Detection*. https://arxiv.org/abs/1704.03414

Wang, Y.-Q. (2014). An Analysis of the Viola-Jones Face Detection Algorithm. *Image Processing On Line*, *4*, 128–148. https://doi.org/10.5201/ipol.2014.104

Yenisbar, & Asmah, Y. (2021). *Laporan Penelitian Stimulus Universitas Nasional : Beberapa Hama Tanaman Padi (Oryza sativa L.) di Desa Undrusbinangun, Kecamatan Kadudampit, Kabupaten Sukabumi, Jawa Barat*. http://repository.unas.ac.id/4739/

Zou, Z., Chen, K., Shi, Z., Guo, Y., & Ye, J. (2023). Object Detection in 20 Years: A Survey. *Proceedings of the IEEE*, *111*(3), 257–276. https://doi.org/10.1109/JPROC.2023.3238524