

Implementasi Algoritma Yolo untuk Deteksi Buah Durian dan Manggis

I Putu Aditya Pradana^{a1}, Ngurah Agus Sanjaya ER^{a2}

^aProgram Studi Informatika, Fakultas Matematika dan Ilmu Pengetahuan Alam,
Universitas Udayana
Jalan Raya Kampus UNUD, Bukit Jimbaran, Kuta Selatan, Badung, Bali, Indonesia
¹adityapradana040604@gmail.com
²agus_sanjaya@unud.ac.id

Abstract

This study aims to implement the YOLOv8 algorithm in detecting images of durian and mangosteen fruits. The research methodology includes literature review, image collection, data processing, YOLOv8 algorithm implementation, model evaluation on validation data, and drawing conclusions. Image collection is done through online sources, and data is processed through annotation, pre-processing, and augmentation using the Roboflow platform before exporting to YOLOv8 format. The algorithm implementation is carried out in Google Colab with model training, object detection, and evaluation stages on validation data. Evaluation results include accuracy, recall, precision, and F1 score values, with model performance evaluated using mean average precision (mAP) metric. The results indicate that the model can recognize objects well, with a mAP above 0.27%. This study successfully implements YOLOv8 for durian and mangosteen fruit detection with satisfactory evaluation results.

Keywords: You Only Look Once (YOLO), Google Colab, Roboflow

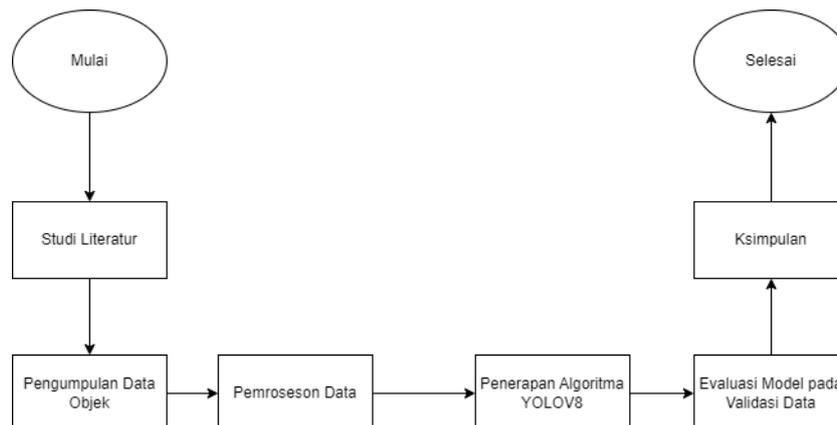
1. Pendahuluan

Perkembangan dunia semakin pesat dan dunia perteknologian semakin canggih, maju dan modern. Salah satunya menguraikan kerangka kerja eksperimental yang digunakan untuk mengidentifikasi dan memproses citra buah durian dan manggis menggunakan algoritma YOLOv8. Metodologi penelitian tersebut mencakup serangkaian tahapan mulai dari studi literatur, pengumpulan gambar, pemrosesan data, penerapan algoritma YOLOv8, hingga evaluasi model pada data validasi. Output dari penelitian ini adalah pemodelan dataset dengan nilai metrik seperti akurasi, recall, precision, dan f1 score. Dalam konteks ini, studi literatur menjadi langkah pertama yang diperlukan untuk memahami konsep dan teknik yang relevan dengan sistem deteksi menggunakan algoritma YOLOv8 pada citra buah durian dan manggis. Kemudian, pengumpulan gambar dilakukan untuk mempersiapkan data primer yang diperlukan untuk pelatihan dan pengujian model. Tahap selanjutnya adalah pemrosesan data, di mana citra yang telah dikumpulkan diproses melalui anotasi, pra-pemrosesan, augmentasi, dan ekspor ke format YOLOv8 menggunakan platform roboflow. Setelah dataset ter-ekspor, proses berlanjut dengan penerapan algoritma YOLOv8. Ini melibatkan pelatihan model menggunakan platform Google Collab dengan mengunggah kode dan file model yang diperlukan. Langkah ini diikuti oleh proses pendeteksian dan evaluasi dataset untuk mengukur kinerja model. Hasil dari evaluasi ini akan memberikan wawasan tentang seberapa baik model dapat mendeteksi objek pada data yang belum pernah dilihat sebelumnya. Dalam hasil dan diskusi, implementasi YOLOv8 dijelaskan secara rinci, termasuk langkah-langkah anotasi, pra-pemrosesan, dan pelatihan model. Evaluasi kinerja model dilakukan menggunakan berbagai metrik, seperti mean average precision (mAP), untuk mengevaluasi akurasi dan recall dari model. Pendahuluan ini memberikan gambaran menyeluruh tentang tujuan, metodologi, dan hasil yang diharapkan dari penelitian ini, serta memperkenalkan pembaca pada topik yang akan dibahas lebih lanjut dalam laporan penelitian.

2. Metode Penelitian

Penelitian ini bersifat eksperimental dengan objek penelitian yaitu durian dan manggis diperoleh dengan cara mengambil foto secara langsung dari Google. Citra yang telah didapatkan

selanjutnya akan diproses melalui beberapa tahapan seperti anotasi, pra- pemrosesan, augmentasi (pelebaran dataset), serta mengekspor dataset ke dalam format *YOLOv8*. Proses tersebut dilakukan secara otomatis dengan memanfaatkan *platform roboflow*. Setelah dataset ter-ekspor ke dalam format *YOLOv8* langkah selanjutnya masuk ke *platform* Google Collab memanggil model *YOLOv8* dengan cara menjalankan code cloud, kemudian mengupload show donload code. Kemudian, dilanjutkan ke beberapa proses yaitu pelatihan, pendeteksian dan evaluasi dataset. *Output* dari seluruh proses ini adalah pemodelan dataset dengan nilai *confusion metric* seperti akurasi, recall, precision, serta f1 score. Berikut merupakan tahapan dalam metodologi penelitian yang dijelaskan melalui diagram alur penelitian dan dapat dilihat pada Gambar 1.



Gambar 1. Diagram Alur Penelitian

a. Studi Literatur

Pada tahap ini mempelajari berbagai referensi atau rujukan yang mendukung penelitian bersumber dari internet, jurnal, tugas akhir, penelitian dan modul -modul yang berkaitan dengan penerapan algoritma *YOLOv8* untuk sistem deteksi pada dataset baru citra durian dan manggis serta penelitian - penelitian sejenis.

b. Pengumpulan Gambar

Tahap ini merupakan persiapan yang harus dilakukan sebelum melakukan penelitian. Tahap yang dimaksud adalah pengumpulan gambar buah durian dan manggis. Data yang tersaji merupakan data primer yang diambil dari mesin Google. Gambar yang diambil terdiri dari 2 jenis buah yaitu durian dan manggis

c. Pemrosesan Data

Setelah data mentah dikumpulkan, langkah selanjutnya adalah tahap pemrosesan data. Data yang sudah ada dimasukkan ke dalam *roboflow* untuk menjalani beberapa proses seperti:

- Anotasi gambar atau pemberian nama gambar berdasarkan 2 jenis buah yaitu buah durian dan manggis
- Pra pemrosesan data dilakukan dengan mengekstrak bagian - bagian yang tidak penting seperti latar belakang yang tidak relevan, objek orientasi otomatis sekaligus mengubah ukuran menjadi 640 x 640 piksel.
- Ekspor data ke dalam format *YOLOv8*

d. Penerapan Algoritma *YOLOv8*

Setelah data melewati proses augmentasi, tahap selanjutnya adalah melakukan proses penerapan algoritma *YOLOv8* pada data tindakan ini merupakan proses training data yang ada untuk mengupload file Lantana zip dan data yaml ke dalam codingan Google Collab lalu melakukan proses runing di dalam coding setelah itu menginput nilai epoch berapa kali data akan dilatih. Setelah hasil nilai keluar lanjut ke proses pendeteksian dengan cara

memasukan gambar yang ingin dideteksi ke codingan dalam Google Collab setelah itu model akan segera membaca dan menampilkan hasil deteksi. Setelah objek benar maka tahap selanjutnya melihat dan mencocokkan hasilnya. Jika objek yang ada sudah terdeteksi sama dengan jenis warna pada lantana camara untuk tiap klasifikasi atau kelasnya, maka dapat dikatakan proses training nya berhasil

e. Evaluasi Model pada Validasi Data

Setelah data melewati proses augmentasi, tahap selanjutnya adalah melakukan proses penerapan algoritma YOLOv8 pada data tindakan ini merupakan proses training data yang ada untuk mengupload show donload code ke dalam codingan Google Collab lalu melakukan proses runing di dalam coding, setelah itu menginput nilai epoch berapa kali data akan dilatih. Setelah hasil nilai keluar lanjut ke proses pendeteksian dengan cara memasukan gambar yang ingin dideteksi ke codingan dalam Google Collab setelah itu model akan segera membaca dan menampilkan hasil deteksi. Setelah objek benar maka tahap selanjutnya melihat dan mencocokkan hasilnya. Jika objek yang ada sudah terdeteksi sama dengan jenis warna pada lantana camara untuk tiap klasifikasi atau kelasnya, maka dapat dikatakan proses trainingnya berhasil.

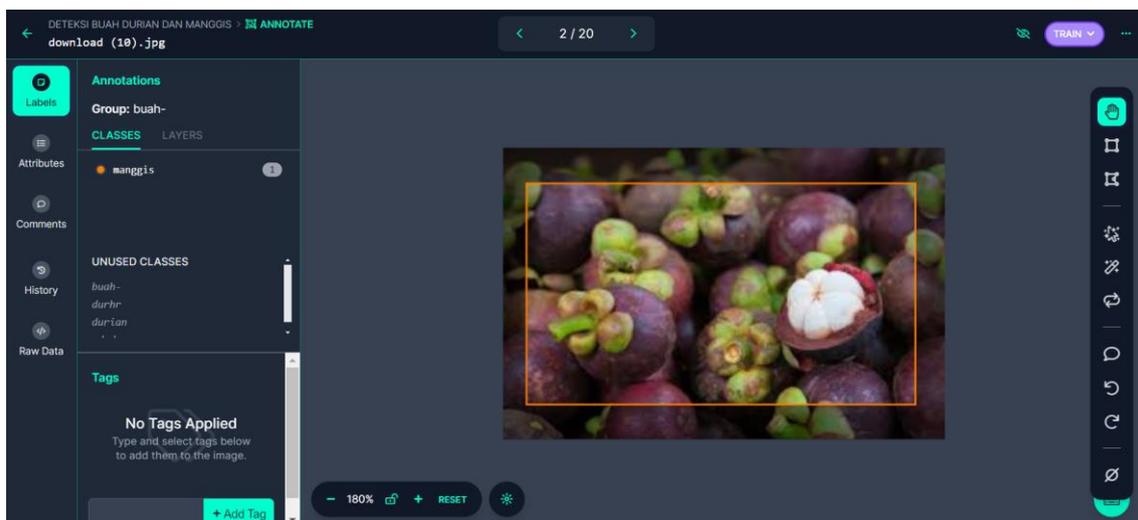
f. Kesimpulan

Tahap ini model melakukan prediksi pada validasi data, dan hasil prediksi bisa dibandingkan dengan label sebenarnya. Evaluasi ini memberikan wawasan tentang seberapa baik model dapat melakukan prediksi pada data yang belum pernah dilihat sebelumnya hasil ini dapat dilihat dari hasil best epoch dengan nilai rata-rata accuracy, precision, recall dan MAP yang dihasilkan oleh Google Collab.

3. Hasil dan Diskusi

3.1 Implementasi YOLOv8

Tahap Implementasi YOLO dimulai dengan mempersiapkan dataset 2 jenis buah yaitu buah durian dan manggis Dengan total gambar sebanyak 20 gambar. Gambar yang dikumpulkan kemudian dimasukkan ke dalam roboflow dilakukan beberapa perlakuan seperti anotasi gambar atau pemberian nama kelas pada objek, hal ini dapat dilihat pada Gambar 2



Gambar 2. Contoh Anotasi Objek

Setelah melalui tahap anotasi lanjut ke tahap preprocessing di mana tahapan ini dilakukan dengan cara mengekstrak bagian - bagian yang tidak penting seperti latar belakang yang tidak relevan, objek orientasi diatur otomatis sekaligus mengubah ukuran menjadi 640 x 640 piksel. Seperti yang diperlihatkan pada Gambar 3.

```
✓ 6s ▶ !mkdir {HOME}/datasets
      %cd {HOME}/datasets

      !pip install roboflow

      from roboflow import Roboflow
      rf = Roboflow(api_key="1Zg7spxmwliEFjV41U2I")
      project = rf.workspace("ditya-pradana").project("deteksi-buah-durian-dan-manggis")
      version = project.version(1)
      dataset = version.download("yolov8")
```

Gambar 3. Tampilan source code untuk mengunduh *YOLOv8*

Setelah mengunduh *YOLOv8* ke dalam Google Collab, selanjutnya mengunduh Show Donlowad Code yang didapatkan di *Platform roboflow*. Hal ini perlu dilakukan agar memungkinkan Google Collab untuk mengakses ke data atau berkas yang diperlukan untuk melatih ataupun menguji model. Hal ini dapat dilihat pada Gambar 4 dimana Show Donlowad Code.

```
✓ 1m [13] %cd {HOME}

!yolo task=detect mode=train model=yolov8s.pt data={dataset.location}/data.yaml epochs=25 imgsz=800 plots=True
```

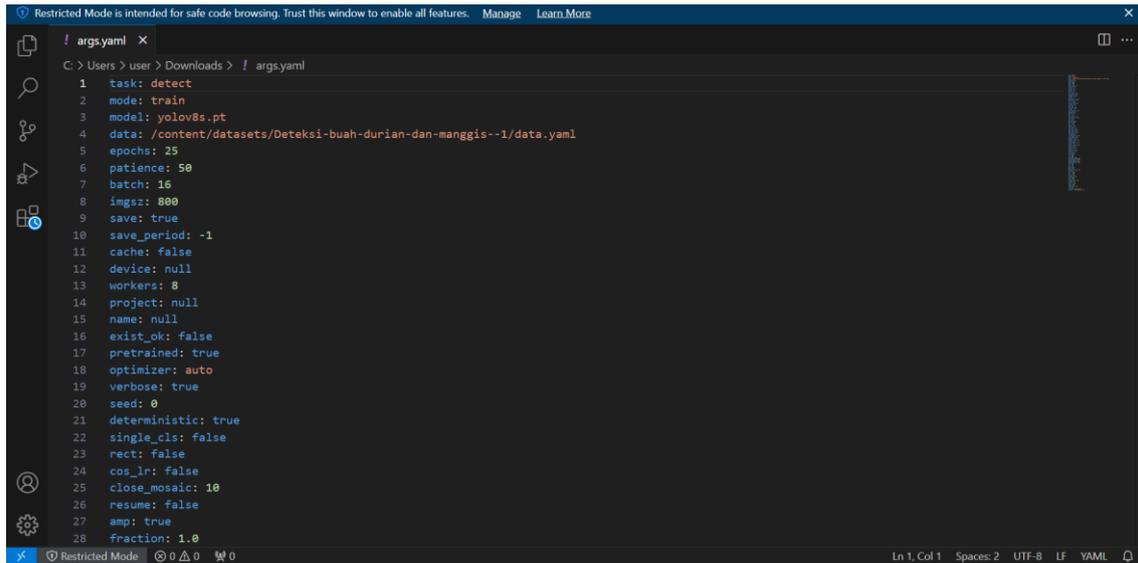
Gambar 4. Tampilan *Show Donlowad Code*

Langkah selanjutnya masuk ke tahap pelatihan data yang dapat dilihat pada Gambar 5, dimana tahap! `yolo task=detect mode=train model=YOLOv8s.pt data={dataset.location}/data.yaml epochs=25 imgsz=800 plots=True`, bagian ini menggunakan program *YOLOv8* untuk melaksanakan training data. `model=YOLOv8s.pt`, menentukan model awal yang akan digunakan untuk pelatihan. Dalam hal ini, *YOLOv8s.pt* adalah model *YOLOv8* versi ringan yang sudah disediakan. `data={dataset.location}/data.yaml`, menentukan lokasi file konfigurasi data pelatihan Anda. Ganti `{dataset.location}` dengan nama folder aktual dataset Anda. File `.yaml` ini berisi informasi penting tentang dataset, seperti lokasi gambar, label objek, dan lainnya. `epochs=25`, menentukan jumlah iterasi pelatihan. Model akan dilatih selama 25 iterasi. `imgsz=800`, menentukan ukuran gambar yang akan digunakan selama pelatihan. Dalam hal ini, gambar akan diubah ukurannya menjadi 800x800 piksel. `plots=True`, menampilkan grafik pelatihan selama proses berlangsung.

```
✓ 1m [13] %cd {HOME}

!yolo task=detect mode=train model=yolov8s.pt data={dataset.location}/data.yaml epochs=25 imgsz=800 plots=True
```

Gambar 5. Tampilan *Source Code Untuk Training Data*



Gambar 6. Tampil Data Yaml

Setelah melakukan training data, langkah berikutnya melakukan pendeteksian objek seperti terlihat pada Gambar 7. Proses ini diperlukan untuk menguji apakah data train berhasil dikenali oleh model atau tidak. Proses ini dilakukan dengan menjalankan perintah python task=detect. Untuk %cd {HOME}, Ini akan mengubah direktori kerja ke direktori home Anda di Google Colab. !yolo task=detect mode=predict model={HOME}/runs/detect/train/weights/best.pt conf=0.25 source={dataset.location}/test/images save=True, bagian ini menggunakan program YOLOv8 untuk melakukan deteksi objek. Deteksi objek, mencari dan mengenali objek tertentu dalam gambar. Mode prediksi, menggunakan model yang sudah dilatih sebelumnya untuk memprediksi objek dalam gambar baru. model={HOME}/runs/detect/train/weights/best.pt, menentukan lokasi file bobot model terlatih Anda. Bobot model, parameter yang dipelajari model selama proses pelatihan dan berisi informasi penting untuk melakukan deteksi objek. conf=0.25, mengatur ambang kepercayaan minimum. Model hanya akan menampilkan deteksi dengan skor kepercayaan minimal 25% (0.25). Skor kepercayaan, Nilai antara 0 dan 1 yang menunjukkan seberapa yakin model terhadap deteksi objek. source={dataset.location}/test/images, Menentukan lokasi gambar yang ingin Anda deteksi objeknya. Ganti {dataset.location} dengan nama folder aktual dataset Anda. save=True, menyimpan hasil deteksi ke dalam gambar baru. Gambar ini akan memiliki kotak pembatas yang digambar di sekitar objek yang terdeteksi beserta labelnya.



Gambar 7. Tampilan Source Code Untuk Menjalankan Deteksi Objek

3.2. Performance Metric

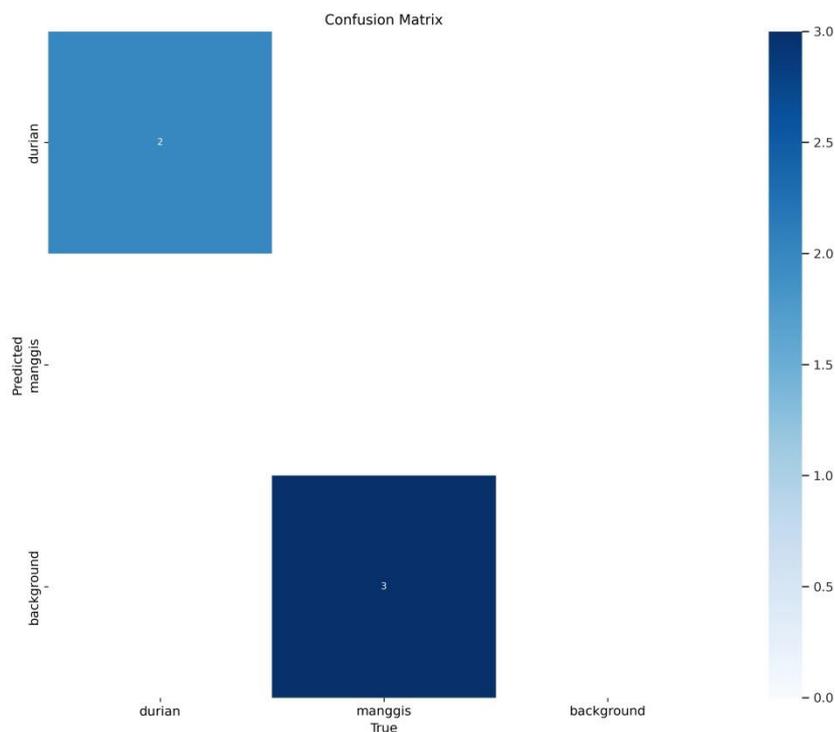
Evaluasi pelatihan dengan epoch atau iterasi 50 dapat dilihat pada Gambar 8, dimana pada 2 jenis buah durian dan manggis, dengan nilai image 360 merujuk pada gambar panorama 360 derajat. Gambar panorama 360 derajat adalah gambar yang mencangkup seluruh lingkaran visual sehingga dapat dilihat ke segala arah secara vertikal maupun horizontal. Untuk jumlah instances pada setiap kelas berbeda-beda karena model deteksi objek dilatih untuk mengenali dan menempatkan objek-objek yang muncul dalam gambar panorama 360 derajat sehingga dari masing-masing kelas secara berturut-turut model mampu mengenali sebanyak 5, 2, 3, dimana hasil ini bergantung pada karakteristik objek, kerumitan dataset, ukuran dataset pelatihan, kualitas anotasi, parameter pelatihan seperti jumlah Epoch dapat mempengaruhi jumlah

instances. Untuk mAP50 (mean average precision) atau metrik yang mencakup precision dan recall pada bagian ambang batas, mAP dihitung dengan cara menghitung nilai rata-rata dari precision sehingga mAP pada rentang nilai ambang deteksi 0,5 hingga 0,95 (mAP50-95) dapat memberikan gambaran lebih tentang evaluasi kinerja model secara menyeluruh di berbagai tingkat ambang deteksi, sehingga nilai mAP yang didapat pada masing-masing kelas di atas 0.27%.

```

11s [16]
!yolo task=detect mode=val model={HOME}/runs/detect/train/weights/best.pt data={dataset.location}/da
/content
Ultralytics YOLOv8.0.196 Python-3.10.12 torch-2.2.1+cu121 CUDA:0 (Tesla T4, 15102MiB)
Model summary (fused): 168 layers, 11126358 parameters, 0 gradients, 28.4 GFLOPs
val: Scanning /content/datasets/Deteksi-buah-durian-dan-manggis--1/valid/labels.cache... 4 images, 0
/usr/lib/python3.10/multiprocessing/popen_fork.py:66: RuntimeWarning: os.fork() was called. os.fork()
self.pid = os.fork()
      Class      Images  Instances   Box(P          R      mAP50  mAP50-95): 100% 1/1 [0
        all         4           5   0.942         0.5   0.749   0.352
        durian       4           2   0.883         1   0.995   0.554
        manggis      4           3         1         0   0.504   0.149
Speed: 0.4ms preprocess, 37.3ms inference, 0.0ms loss, 162.4ms postprocess per image
Results saved to runs/detect/val
💡 Learn more at https://docs.ultralytics.com/modes/val
    
```

Gambar 8. Tampilan Parameter Hasil Training Dengan Google Collab



Gambar 9. Funfusion Matrix

4. Kesimpulan

Hasil penelitian ini mengimplementasikan algoritma YOLOv8 untuk mendeteksi buah durian dan manggis menggunakan citra yang diperoleh dari Google. Tahapan metodologi penelitian meliputi studi literatur, pengumpulan gambar, pemrosesan data, penerapan algoritma YOLOv8, evaluasi model pada data validasi, serta pembuatan kesimpulan. Pada tahap studi literatur, peneliti mempelajari berbagai referensi yang mendukung penggunaan algoritma YOLOv8 untuk deteksi pada dataset citra durian dan manggis. Selanjutnya, dilakukan pengumpulan gambar durian dan

manggis dari Google sebagai data primer. Data tersebut kemudian diproses melalui tahap anotasi, pra-pemrosesan, dan augmentasi menggunakan platform Roboflow. Setelah itu, dataset diekspor ke format YOLOv8. Proses penerapan algoritma YOLOv8 dilakukan di Google Colab, di mana dilakukan pelatihan model, pendeteksian objek, dan evaluasi model pada data validasi. Hasil dari evaluasi termasuk nilai akurasi, recall, precision, dan F1 score. Selanjutnya, hasil dan diskusi dari penelitian ini mencakup implementasi YOLOv8 dengan menunjukkan tahapan anotasi, pra-pemrosesan, dan pelatihan model secara rinci. Dalam tahap evaluasi, performa model dievaluasi menggunakan berbagai metrik seperti mean average precision (mAP). Hasil evaluasi menunjukkan bahwa model dapat mengenali objek dengan baik, dengan mAP yang cukup tinggi di atas 0.27%. Dari keseluruhan penelitian, dapat disimpulkan bahwa penelitian ini berhasil mengimplementasikan algoritma YOLOv8 untuk mendeteksi buah durian dan manggis dengan cukup baik. Model yang dihasilkan mampu mengenali objek dengan akurasi yang memadai, seperti terlihat dari nilai-nilai performa yang dicapai dalam evaluasi.

Daftar Pustaka

- [1] A. Muliantara and N. A. S. ER, "Pengembangan Pengklasifikasi jenis Tanaman Menggunakan pendekatan backpropagation Dan Nguyen-Widrow," *Jurnal Ilmu Komputer*, <https://ojs.unud.ac.id/index.php/jik/article/view/18779> (accessed May 6, 2024).
- [2] Bagus Janapriya, A. A. G. (2023). Pengenalan Jenis Rambu Lalu Lintas menggunakan Metode YOLO V5. *JELIKU (Jurnal Elektronik Ilmu Komputer Udayana)*, 11(4). <https://doi.org/10.24843/jlk.2023.v11.i04.p32>.
- [3] Anggotra, P., & Muliantara, A. (2023). Deteksi Relief Candi Borobudur Menggunakan Metode Template Matching. *JELIKU (Jurnal Elektronik Ilmu Komputer Udayana)*, 12(1). <https://doi.org/10.24843/jlk.2023.v12.i01.p03>.
- [4] Bochkovskiy, Alexey, Chien-Yao Wang, and Hong-Yuan Mark Liao. "YOLOv4: Optimal Speed and Accuracy of Object Detection." *arXiv preprint arXiv:2004.10934* (2020).
- [5] Saputra, D. H., & Imran, B. (2023). Object Detection Untuk Mendeteksi Citra Buah-Buahan Menggunakan Metode Yolo. *Jurnal Kecerdasan Buatan dan Teknologi Informasi*, 2(2), 70-80.
- [6] Redmon, Joseph, and Ali Farhadi. "YOLOv3: An Incremental Improvement." *arXiv preprint arXiv:1804.02767* (2018).
- [7] Bochkovskiy, Alexey, et al. "YOLOv5: An Incremental Improvement." *arXiv preprint arXiv:2006.02541* (2020).
- [8] HWB, N., Mailoa, E., & Purnomo, H. D. (2020). Deteksi Buah untuk Klasifikasi Berdasarkan Jenis dengan Algoritma CNN Berbasis YOLOv3. *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, 4(3), 476-481.

Halaman ini sengaja dibiarkan kosong