



IMPLEMENTASI METODE YOU ONLY LOOK ONCE (YOLO) UNTUK DETEKSI KESEGERAN TELUR AYAM BERDASARKAN CITRA CANGKANG

Tri Handayani

Teknik Informatika, Universitas Muhammadiyah Ponorogo
tri128184@gmail.com

Abstrak

Penelitian ini mengimplementasikan metode You Only Look Once (YOLO) untuk mendiagnosis kesegaran telur ayam berdasarkan citra cangkang. Telur ayam merupakan makanan yang populer dan permintaannya tinggi, namun penerapan teknologi dalam pemilahan telur masih terbatas di Indonesia. Melalui pengumpulan data citra cangkang telur segar dan tidak segar serta proses pelatihan dan pengujian model YOLO, hasil penelitian menunjukkan bahwa model mampu mendeteksi dan mengklasifikasikan kualitas telur dengan baik. Evaluasi model menunjukkan presisi dan recall yang baik untuk kedua kelas telur segar dan tidak segar. Rekomendasi pengembangan termasuk peningkatan dataset untuk pelatihan model dan penelitian lebih lanjut untuk meningkatkan kemampuan model dalam membedakan antara telur segar dan tidak segar.

Kata kunci: Telur Ayam; You Only Look Once (YOLO); Citra Cangkang;

Abstract

This research implements the You Only Look Once (YOLO) method to diagnose the freshness of chicken eggs based on shell images. Chicken eggs are a popular food item with high demand, yet the technological implementation in egg sorting remains limited in Indonesia. Through the collection of fresh and unfresh eggshell image data and the training and testing process of the YOLO model, the study results show that the model can effectively detect and classify egg quality. Model evaluation demonstrates good precision and recall for both fresh and unfresh egg classes. Recommendations for further development include expanding the dataset for model training and conducting further research to enhance the model's ability to distinguish between fresh and unfresh eggs.

Keywords: chicken eggs; You Only Look Once (YOLO); shell image;

1. Pendahuluan

Telur adalah salah satu lauk yang sangat populer dan sering dikonsumsi. Berdasarkan data dari Badan Pusat Statistik (BPS) pada tahun 2020, permintaan telur ayam di Indonesia mencapai 4.895 ribu ton. Selain permintaan dalam negeri, telur ayam juga diminati oleh pasar internasional. Dari Januari hingga Mei 2021, sebanyak 5.522 kg telur ayam diekspor ke Burma/Myanmar. Data ini mengindikasikan bahwa konsumsi telur ayam sangat tinggi. Tingginya permintaan telur ayam tentunya disertai dengan proses pemilahan yang ketat sebelum telur didistribusikan dan dijual [1].

Pemilahan bertujuan untuk menilai kualitas telur sebelum didistribusikan dan dijual. Penggunaan teknologi dalam proses pemilahan mulai diterapkan untuk mempercepat penilaian kualitas telur. Sayangnya, pemanfaatan teknologi dalam pemilahan telur di Indonesia belum optimal. Penggunaan teknologi dalam proses ini masih jarang diterapkan. Banyak peternakan di Indonesia masih mengandalkan metode pemilahan manual. Padahal, berbagai metode deteksi kualitas telur sudah banyak dikembangkan, seperti metode deteksi retakan pada telur dengan menggunakan pengolahan citra [2], [3].

Dua aspek utama yang menentukan kualitas telur ayam adalah bagian dalam (putih, kuning, dan kantong kulit) dan bagian luar (warna, bentuk, dan kebersihan) [4]. Menurut Muchtadi et al. dalam penelitian Djaelani, telur mudah mengalami penurunan kualitas akibat kerusakan fisik serta kehilangan air dan zat kimia penting seperti karbondioksida, ammonia, nitrogen, dan hidrogen sulfida. Ini menunjukkan pentingnya



pengawasan terhadap kedua aspek tersebut untuk memastikan kualitas telur yang dijual kepada konsumen tetap terjaga [5].

Tidak semua telur ayam memiliki kualitas yang layak untuk dikonsumsi, sehingga perlu kehati-hatian dalam memilihnya sebelum diolah menjadi makanan. Biasanya, penilaian kualitas telur ayam yang baik dapat dilakukan dengan memeriksa kondisi cangkang. Selain itu, pengamatan bagian dalam telur bisa dilakukan dengan memecahnya, tetapi cara ini memakan waktu dan mungkin tidak efisien. Oleh karena itu, memahami kualitas telur sangat penting. Kesegaran telur umumnya ditentukan oleh kondisi cangkangnya, termasuk warna dan kebersihannya [6].

Dalam penelitian oleh Bacheramsyah, T.F. [7], dilakukan upaya untuk mengidentifikasi kualitas telur ayam berdasarkan ketebalan kuning dan putih telur menggunakan *Discrete Cosine Transform* (DCT) dan algoritma *K-Nearest Neighbor* (KNN), yang menghasilkan akurasi sebesar 83,63%. Namun, dalam penelitian tersebut, pengamatan citra telur masih dilakukan secara manual dengan menghancurkan telur yang dianalisis.

Selain itu, peneliti lain juga mencoba mengidentifikasi kemurnian cangkang telur ayam melalui pengolahan gambar digital dengan metode *thresholding*. Metode ini menggunakan gambar telur RGB sebagai input yang kemudian diubah menjadi gambar *grayscale* sebagai output [6]. Menurut penelitian yang dilakukan oleh Maimunah dan Rokhman, ada kemungkinan bahwa hasil klasifikasi telur berdasarkan kebersihan cangkang sangat dipengaruhi oleh metode pengolahan citra digital yang digunakan.

Dalam penelitian ini, peneliti melakukan deteksi kualitas telur melalui analisis citra cangkang menggunakan pendekatan *deep learning*, yaitu dengan metode *You Only Look Once* (YOLO) yang telah dimodifikasi.

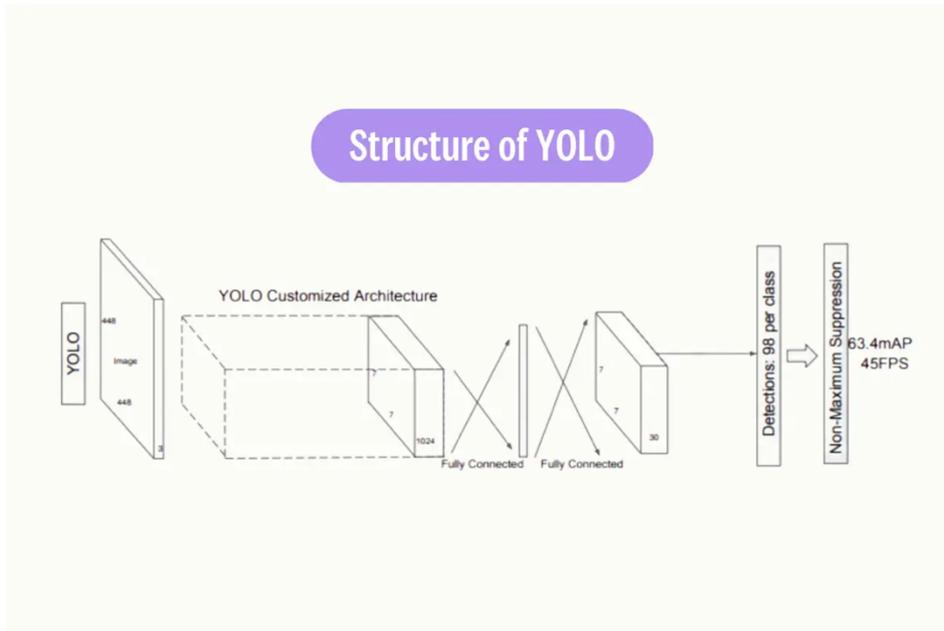
2. Tinjauan Pustaka

2.1. Telur Ayam

Telur ayam adalah salah satu makanan hewani unggas yang paling bergizi. Secara umum, telur terdiri dari putih telur, kuning telur, dan kulit telur. Putih telur merupakan bagian terbesar dari telur, mengandung komponen bioaktif berupa protein yang dapat diidentifikasi berdasarkan urutan isoelektrolit. Protein bernilai tinggi dalam putih telur ayam meliputi *ovalbumin*, *ovotransferrin*, *ovomucin*, *lizozim*, *ovoinhibitor*, *ovoflavoprotein*, *avidin*, dan *cystatin* [8].

2.2. Metode *You Only Look Once* (YOLO)

Metode *You Only Look Once* (YOLO) adalah salah satu teknik yang digunakan untuk memprediksi jenis dan lokasi objek dalam sebuah gambar secara *real-time* menggunakan jaringan konvolusi (*convolutional network*). Metode ini dapat memprediksi kotak-kotak pembatas (*bounding boxes*) dan kelas probabilitas untuk setiap kotak, sehingga menghasilkan prediksi objek yang akurat [9]. Metode ini merupakan hasil dari perkembangan teknologi dalam bidang *computer vision*, yang memungkinkan perangkat untuk mendeteksi, mengklasifikasikan, dan menghitung objek dalam gambar secara otomatis. Data yang dihasilkan dapat dimanfaatkan dalam berbagai bidang kehidupan dan dapat dikombinasikan dengan sistem tertanam untuk aplikasi langsung [8].



Gambar 1. Struktur Model *You Only Look Once* (YOLO)

3. Metodologi

3.1. Pengumpulan Data

Pada tahap ini, digunakan kumpulan data citra yang terdiri dari dua jenis gambar dengan kategori cangkang telur segar dan cangkang telur tidak segar, dengan total 1.898 gambar cangkang telur. Data ini diperoleh dari Kaggle dan akan digunakan untuk pelatihan serta pengujian model. Langkah pertama dalam penelitian ini adalah mengumpulkan data yang diperlukan, yang melibatkan proses pelatihan. Setelah itu, data pelatihan dimuat dan diproses melalui pra-pemrosesan, termasuk pemberian label pada setiap data serta penyesuaian ukuran citra. Selanjutnya, arsitektur YOLO dirancang dan parameter modelnya dioptimalkan.

Langkah berikutnya adalah pengujian, di mana data uji dimuat, diproses melalui pra-pemrosesan, dan diuji menggunakan YOLO. Hasil klasifikasi dari YOLO kemudian dievaluasi menggunakan parameter optimal yang diperoleh selama proses pelatihan.

Tabel 1. Data Cangkang Telur Ayam

No	Kategori Data	Jumlah data
1	<i>Data Training</i>	1599
2	<i>Data Validation</i>	299
Total		1898

3.2. Pre-processing

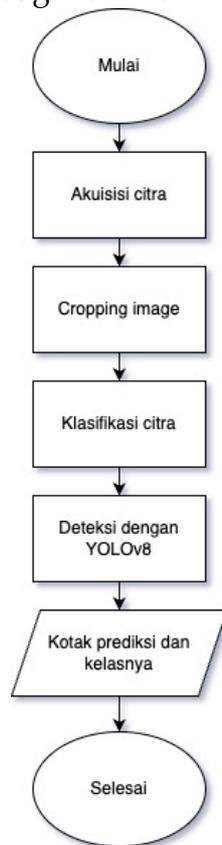
Sebelum memulai proses pelatihan, langkah pertama yang dilakukan adalah pra-pemrosesan (*pre-processing*). Tahap ini melibatkan penyesuaian ukuran citra agar seragam dan pelabelan data yang sesuai. Ukuran citra yang lebih besar biasanya meningkatkan kompleksitas proses, tetapi juga dapat meningkatkan akurasi hasil klasifikasi secara signifikan. Dalam penelitian ini, gambar-gambar dengan ukuran 640 x 640 piksel digunakan untuk mengevaluasi kemampuan model dalam mengidentifikasi hasil dari gambar tersebut. Ukuran ini dipilih karena memberikan keseimbangan antara

kompleksitas proses dan akurasi klasifikasi, memungkinkan model untuk mempelajari fitur-fitur penting dengan lebih efektif.

3.3. Desain Model *You Only Look Once*

Metode YOLO, atau *You Only Look Once*, digunakan untuk mendeteksi kesegaran telur ayam dengan menganalisis citra cangkang telur. Citra telur dipotong menjadi gambar kecil dan diklasifikasikan berdasarkan jenis telur. YOLOV8, versi terbaru YOLO, kemudian digunakan untuk mendeteksi telur ayam dalam gambar dan mengidentifikasi kesegarannya. Hasil deteksi ditampilkan pada layar atau disimpan dalam file.

Metode ini menawarkan beberapa keuntungan, seperti kecepatan, akurasi, dan kemudahan penggunaan. Dengan melatih model YOLO dengan dataset citra telur yang lebih besar dan beragam, akurasi deteksi dapat ditingkatkan. Metode ini dapat dikombinasikan dengan metode lain untuk meningkatkan kontrol kualitas telur ayam di industri peternakan. Diagram alir sebagai berikut.

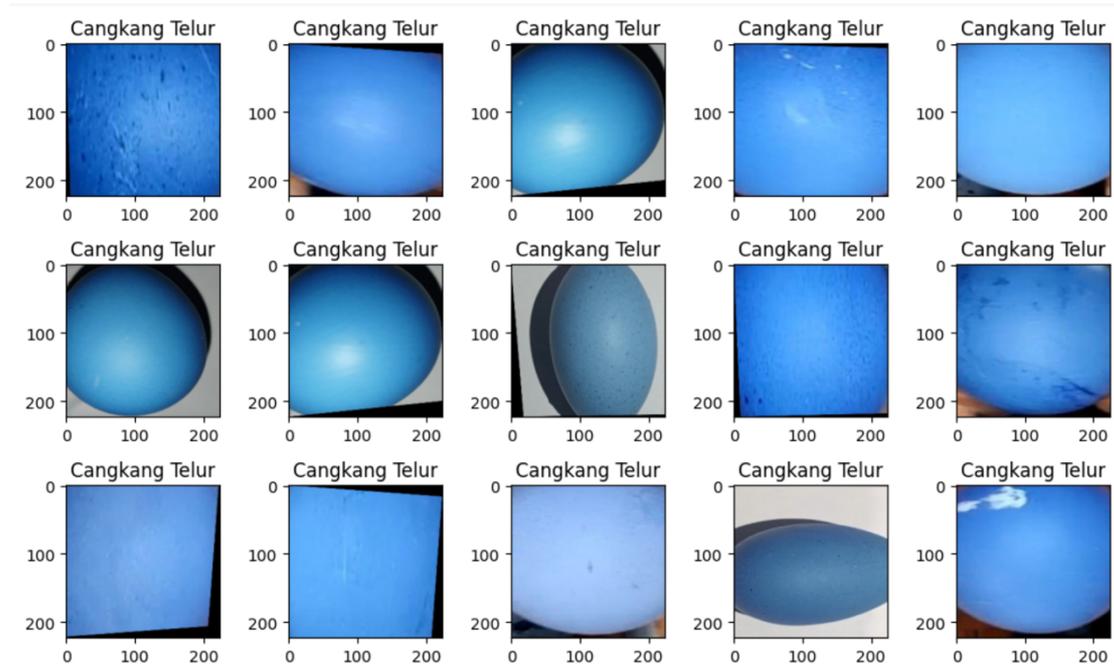


Gambar 2. Desain Model YOLO

4. Hasil dan Pembahasan

Bagian ini berfokus pada analisis penelitian dan temuan terbaru. Hasil eksperimen dievaluasi untuk menilai sejauh mana kesesuaiannya dengan hipotesis yang telah ditetapkan, jika ada. Pembahasan hasil tersebut juga dikaitkan dengan referensi-referensi yang telah digunakan.

Dataset berhasil dimuat ke dalam *Google Colab* dan kemudian diubah ukurannya menjadi 640 x 640 piksel seperti yang ditunjukkan pada gambar di bawah ini.



Gambar 3. Dataset Cangkang Telur

Model disesuaikan untuk mendeteksi dua kelas objek, berbeda dengan konfigurasi aslinya yang mendeteksi 80 kelas. Model ini menggunakan arsitektur jaringan konvolusi yang terdiri dari berbagai lapisan, termasuk lapisan konvolusi, blok C2f, blok SPPF, dan lapisan *upsampling*. Lapisan-lapisan ini bekerja sama untuk memproses gambar input dan menghasilkan prediksi *bounding box* serta kelas objek dengan akurasi tinggi. Setiap lapisan memiliki peran spesifik, seperti mengubah ukuran gambar, menggabungkan fitur dari berbagai lapisan, dan mendeteksi objek pada berbagai skala.

Terdiri dari 225 lapisan dengan total 3.011.238 parameter yang dapat dilatih, model ini memiliki kompleksitas komputasi sebesar 8.2 GFLOPs. Data *input* awalnya diproses melalui *pre-processing*, termasuk penyesuaian ukuran gambar dan pelabelan. Setelah itu, model dilatih dan diuji menggunakan data tersebut. Arsitektur YOLO ini dirancang untuk memberikan hasil deteksi objek yang akurat dan efisien, dengan penggunaan parameter yang optimal untuk meningkatkan kinerja dalam mengidentifikasi objek dari gambar.



	from	n	params	module	arguments
0	-1	1	464	ultralytics.nn.modules.conv.Conv	[3, 16, 3, 2]
1	-1	1	4672	ultralytics.nn.modules.conv.Conv	[16, 32, 3, 2]
2	-1	1	7360	ultralytics.nn.modules.block.C2f	[32, 32, 1, True]
3	-1	1	18560	ultralytics.nn.modules.conv.Conv	[32, 64, 3, 2]
4	-1	2	49664	ultralytics.nn.modules.block.C2f	[64, 64, 2, True]
5	-1	1	73984	ultralytics.nn.modules.conv.Conv	[64, 128, 3, 2]
6	-1	2	197632	ultralytics.nn.modules.block.C2f	[128, 128, 2, True]
7	-1	1	295424	ultralytics.nn.modules.conv.Conv	[128, 256, 3, 2]
8	-1	1	460288	ultralytics.nn.modules.block.C2f	[256, 256, 1, True]
9	-1	1	164608	ultralytics.nn.modules.block.SPPF	[256, 256, 5]
10	-1	1	0	torch.nn.modules.upsampling.Upsample	[None, 2, 'nearest']
11	[-1, 6]	1	0	ultralytics.nn.modules.conv.Concat	[1]
12	-1	1	148224	ultralytics.nn.modules.block.C2f	[384, 128, 1]
20	[-1, 9]	1	0	ultralytics.nn.modules.conv.Concat	[1]
21	-1	1	493056	ultralytics.nn.modules.block.C2f	[384, 256, 1]
22	[15, 18, 21]	1	751702	ultralytics.nn.modules.head.Detect	[2, [64, 128, 256]]

Model summary: 225 layers, 3011238 parameters, 3011222 gradients, 8.2 GFLOPs

Gambar 4. Hasil Arsitektur Model YOLO

Model YOLO telah dilatih selama 50 *epoch* dalam waktu 9 menit, dengan ukuran file akhir dari model yang disimpan sebesar 6,3MB untuk berat terakhir dan 6,2MB untuk berat terbaik. Saat memvalidasi model terbaik, sistem menggunakan *Ultralytics YOLOv8.0.196* dengan *Python 3.10.12* dan *Torch 2.3.0+cu121* di GPU *NVIDIA L4* dengan memori 22700MiB. Model ini memiliki 168 lapisan, 3.006.038 parameter, dan 8.1 *GFLOPs* tanpa gradien yang dapat dilatih. Namun, terjadi kesalahan saat mencoba menjalankan rencana dengan *cudaException* terkait *CUDNN_STATUS_NOT_SUPPORTED*, yang menunjukkan masalah kompatibilitas dengan deskriptor eksekusi *CUDNN*.



```
50 epochs completed in 0.161 hours.

Optimizer stripped from runs/detect/train/weights/last.pt, 6.3MB

Optimizer stripped from runs/detect/train/weights/best.pt, 6.2MB

Validating runs/detect/train/weights/best.pt...

Ultralytics YOLOv8.0.196 🚀 Python-3.10.12 torch-2.3.0+cu121 CUDA:0 (NVIDIA L4, 22700MiB)

Model summary (fused): 168 layers, 3006038 parameters, 0 gradients, 8.1 GFLOPs

Plan failed with a cudnnException: CUDNN_BACKEND_EXECUTION_PLAN_DESCRIPTOR: cudnnFinalize Descriptor Failed cudnn_status:
CUDNN_STATUS_NOT_SUPPORTED (Triggered internally at ../aten/src/ATen/native/cudnn/Conv_v8.cpp:919.)
```

Gambar 5. Total Epoch

Kemudian evaluasi model YOLO yang dilakukan pada 229 gambar dengan 229 *instance*, menghasilkan rata-rata presisi (P) 0,847, *recall* (R) 0,904, mAP50 0,951, dan mAP50-95 0,932. Untuk telur segar, presisi 0,87, *recall* 0,864, mAP50 0,948, dan mAP50-95 0,925, sementara untuk telur tidak segar, presisi 0,824, *recall* 0,945, mAP50 0,953, dan mAP50-95 0,938. Kecepatan per gambar adalah 0,2ms untuk pra-pemrosesan, 1,0ms untuk inferensi, dan 5,6ms untuk pasca-pemrosesan. Hasil ini disimpan di direktori "*runs/detect/train*". Lebih detailnya bisa dilihat pada gambar dibawah ini.

```
0, 2.77it/s]
Class      Images  Instances  Box(P   R   mAP50  mAP50-95): 100%|██████████| 8/8 [00:02<00:0
all        229     229      0.847  0.904  0.951  0.932
fresh     229     110      0.87   0.864  0.948  0.925
unfresh   229     119      0.824  0.945  0.953  0.938

Speed: 0.2ms preprocess, 1.0ms inference, 0.0ms loss, 5.6ms postprocess per image

Results saved to runs/detect/train
```

Gambar 6. Hasil Evaluasi Model

Objek `DetMetrics` dari *ultralytics* mencakup atribut-atribut yang meliputi indeks kelas, metrik *box*, *confusion matrix*, dan nilai *fitness* sebesar 0,9337573002799661. Objek ini mengevaluasi presisi, *recall*, mAP50, dan mAP50-95 untuk dua kelas (*'fresh'* dan *'unfresh'*), dengan mAP50 masing-masing 0,92542 dan 0,93837. Hasil evaluasi ini disimpan dalam direktori *'runs/detect/train'*, dengan kecepatan rata-rata per gambar untuk *pre-processing* 0,1717ms, inferensi 0,9731ms, dan *after-processing* 5,6215ms. Objek ini juga menyimpan berbagai metrik dalam `results_dict` untuk analisis lebih lanjut. Informasi bisa dilihat pada gambar dibawah ini.



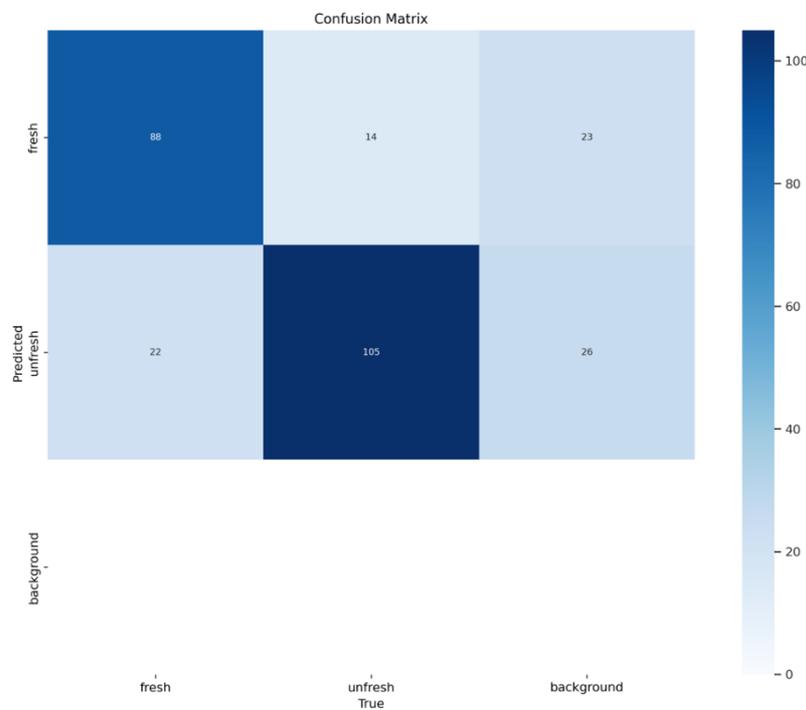
```
ultralalytics.utils.metrics.DetMetrics object with attributes:  
  
ap_class_index: array([0, 1])  
box: ultralytics.utils.metrics.Metric object  
confusion_matrix: <ultralalytics.utils.metrics.ConfusionMatrix object at 0x7f15a799df00>  
fitness: 0.9337573002799661  
keys: ['metrics/precision(B)', 'metrics/recall(B)', 'metrics/mAP50(B)', 'metrics/mAP50-95(B)']  
maps: array([[ 0.92542,  0.93837]])  
names: {0: 'fresh', 1: 'unfresh'}  
plot: True  
results_dict: {'metrics/precision(B)': 0.8469050954883425, 'metrics/recall(B)': 0.9041073477347987, 'metrics/mAP50(B)': 0.9505060167294075, 'metrics/mAP50-95(B)': 0.9318963317855836, 'fitness': 0.9337573002799661}  
save_dir: PosixPath('runs/detect/train')  
speed: {'preprocess': 0.17173321486560536, 'inference': 0.9731496785925986, 'loss': 0.0005882379790060385, 'postprocess': 5.6215050959691215}
```

Gambar 7. Hasil *DetMetrics*

Confusion Metrics ini menunjukkan performa model YOLO dalam mendeteksi kesegaran telur ayam berdasarkan citra cangkang. Matriks terdiri dari tiga baris dan tiga kolom, mewakili tiga kelas: segar, tidak segar, dan *background*.

Nilai diagonal menunjukkan jumlah telur yang diklasifikasikan dengan benar, dengan 88 telur segar, 105 telur tidak segar, dan 80 *background* teridentifikasi dengan tepat. Nilai di luar diagonal menunjukkan kesalahan klasifikasi, seperti 14 telur segar yang salah diklasifikasikan sebagai tidak segar dan 26 latar belakang yang salah diklasifikasikan sebagai telur segar.

Secara keseluruhan, model YOLO mencapai akurasi 83,7%, menunjukkan kemampuannya dalam mendeteksi kesegaran telur ayam. Namun, terdapat ruang untuk peningkatan, terutama dalam membedakan antara telur segar dan tidak segar. Analisis lebih lanjut terhadap matriks konfusi dan data pelatihan dapat membantu meningkatkan performa model.



Gambar 8. Tabel *Confusion Matrix*

Evaluasi validasi model YOLO menunjukkan bahwa model ini terdiri dari 168 lapisan dengan 3.006.038 parameter dan memiliki kompleksitas komputasi sebesar 8.1 GFLOPs. Dari 229 gambar yang divalidasi, model mencapai presisi (P) sebesar 82,2%,



recall (R) 92,45%, mAP@50 sebesar 95,05%, dan mAP@50-95 sebesar 93,16%. Performa ini mencakup dua kelas: telur segar dan telur tidak segar. Kelas telur segar menunjukkan presisi 85,2% dan *recall* 88,9%, sementara kelas telur tidak segar menunjukkan presisi 79,2% dan *recall* 96%. Hasil ini menunjukkan bahwa model memiliki kemampuan yang baik dalam mengidentifikasi dan mengklasifikasikan kualitas telur.

Proses evaluasi dilakukan dengan kecepatan rata-rata 0,2 ms untuk *pre-processing*, 1,7 ms untuk inferensi, dan 4,9 ms untuk pasca-pemrosesan per gambar. F1 Score keseluruhan dari evaluasi ini adalah 87,02%, mengindikasikan keseimbangan yang baik antara presisi dan *recall* dalam performa model. Namun, terdapat catatan bahwa penggunaan *os.fork()* dapat menyebabkan deadlock karena ketidakcocokannya dengan kode *multithreaded* yang digunakan oleh JAX.

```
Ultralytics YOLOv8.0.196 Python-3.10.12 torch-2.3.0+cu121 CUDA:0 (NVIDIA L4, 22700MiB)

Model summary (fused): 168 layers, 3006038 parameters, 0 gradients, 8.1 GFLOPs

val: Scanning /content/telur-4/valid/labels.cache... 229 images, 0 backgrounds, 0 corrupt: 100% |██████████| 229/229 [00:00
<?, ?it/s]

os.fork() was called. os.fork() is incompatible with multithreaded code, and JAX is multithreaded, so this will likely lea
d to a deadlock.

Class      Images  Instances  Box(P      R      mAP50  mAP50-95): 100% |██████████| 15/15 [00:03<0
0:00, 4.33it/s]

      all      229      229      0.822      0.925      0.95      0.932

      fresh    229      110      0.852      0.889      0.948      0.925

      unfresh  229      119      0.792      0.96      0.953      0.939

Speed: 0.2ms preprocess, 1.7ms inference, 0.0ms loss, 4.9ms postprocess per image

Results saved to runs/detect/val2

Precision: 82.20%

Recall: 92.45%

mAP@50: 95.05%

mAP@50-95: 93.16%

F1 Score: 87.02%
```

Gambar 9. Hasil Evaluasi *Data Validation*

Hasil pengujian model YOLO dengan satu gambar cangkang telur baru menunjukkan bahwa model mendeteksi telur tersebut sebagai "*unfresh*". Proses ini dilakukan dengan kecepatan 2,8 ms untuk pra-pemrosesan, 7,2 ms untuk inferensi, dan 2,3 ms untuk pasca-pemrosesan per gambar dengan ukuran 640x640 piksel. Hal ini menunjukkan bahwa model mampu mendeteksi dan mengklasifikasikan telur secara cepat, meskipun terjadi kesalahan deteksi pada telur baru ini.

image 1/1 /content/IMG_20240605_202734.jpg: 640x640 1 unfresh, 7.2ms
Speed: 2.8ms preprocess, 7.2ms inference, 2.3ms postprocess per image at shape (1, 3, 640, 640)
Saving IMG_20240605_202734.jpg to IMG_20240605_202734.jpg



Gambar 10. Hasil *Data Testing*

5. Simpulan

Dalam penelitian ini, dilakukan implementasi metode *You Only Look Once* (YOLO) untuk melakukan diagnosa kesegaran telur ayam berdasarkan citra cangkang. Telur ayam merupakan makanan yang populer dan permintaannya tinggi, baik di pasar domestik maupun internasional. Meskipun pentingnya pemilahan telur untuk menjaga kualitasnya sebelum dijual, penerapan teknologi dalam proses ini masih terbatas di Indonesia. Berbagai metode deteksi kualitas telur telah dikembangkan, termasuk penggunaan citra cangkang telur. Namun, kebanyakan metode masih memerlukan pengamatan manual atau proses yang tidak efisien. Oleh karena itu, penelitian ini mengusulkan pendekatan baru dengan menggunakan metode deep learning, khususnya YOLO, untuk deteksi kualitas telur berdasarkan citra cangkang.

Melalui pengumpulan data citra cangkang telur segar dan tidak segar serta proses pelatihan dan pengujian model YOLO, hasil penelitian menunjukkan bahwa model mampu dengan baik dalam mendeteksi dan mengklasifikasikan kualitas telur. Evaluasi model menunjukkan presisi dan recall yang baik, serta mAP yang tinggi untuk kedua kelas telur segar dan tidak segar. Meskipun demikian, masih ada ruang untuk peningkatan, terutama dalam membedakan antara telur segar dan tidak segar. Hasil ini menunjukkan bahwa metode YOLO dapat menjadi solusi yang efektif dan efisien dalam memonitor dan menjaga kualitas telur ayam.

Rekomendasi untuk pengembangan ke depan termasuk peningkatan dataset untuk pelatihan model agar dapat mengakomodasi variasi lebih lanjut dalam citra cangkang telur. Selain itu, penelitian lebih lanjut dapat dilakukan untuk meningkatkan kemampuan model dalam membedakan antara telur segar dan tidak segar dengan lebih baik, serta eksplorasi integrasi metode YOLO dengan teknologi lain untuk meningkatkan kontrol kualitas telur secara menyeluruh di industri peternakan.



Daftar Referensi

- [1] M. F. A. Pratama, A. L. Prasasti dan M. W. Paryasto, "Klasifikasi Ukuran dan Kualitas Telur Ayam Menggunakan Algoritma Convolutional Neural Network," *Klasifikasi Ukuran dan Kualitas Telur Ayam Menggunakan Algoritma Convolutional Neural Network*, vol. 10, no. 1, pp. 473-480, 2023.
- [2] C. Haoran, H. Chuchu, J. Minlan dan L. Xiaoxiao, "Egg Crack Detection Based on Support Vector Machine," *International Conference on Intelligent Computing and Human-Computer Interaction (ICHCI)*, pp. 80-83, 2020.
- [3] B. Narin, S. Buntan, N. Chumuang dan M. Ketcham, "Crack on Eggshell Detection System Based on Image Processing Technique," *International Symposium on Communications and Information Technologies (ISCIT)*, pp. 1-6, 2018.
- [4] J. Thipakorn, P. Riyamongkol dan R. Waranusast, "Egg weight prediction and egg size classification using image processing and machine learning," *International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON)*, pp. 477-480, 2017.
- [5] M. A. Djaelani, "Kualitas Telur Ayam Ras (Gallus L.) Setelah Penyimpanan yang dilakukan Pencelupan pada Air Mendidih dan Air Kapur Sebelum Penyimpanan," *BULETIN ANATOMI DAN FISILOGI dh SELLULA*, vol. 24, no. 1, pp. 122-127, 2016.
- [6] M. dan T. Rokhman, "Klasifikasi Penurunan Kualitas Telur Ayam Ras Berdasarkan Warna Kerabang Menggunakan Support Vector Machine," *INFORMATICS FOR EDUCATORS AND PROFESSIONALS*, vol. 3, no. 1, pp. 43-52, 2018.
- [7] N. Ibrahim, T. F. Bacheramsyah, B. Hidayat dan S. Darana, "Pengklasifikasian Grade Telur Ayam Negeri menggunakan Klasifikasi K-Nearest Neighbor berbasis Android," *ELKOMIKA*, vol. 6, no. 2, pp. 288-302, 2018.
- [8] R. R. M. Akbar, F. Rizal dan W. J. Shudiq, "Implementasi Algoritma Convolutional Neural Network (CNN) Untuk Deteksi Kesegaran Telur Berbasis Android," *Jusikom : Jurnal Sistem Komputer Musi Rawas*, vol. 8, no. 1, pp. 1-10, 2023.
- [9] K. He, X. Zhang dan J. Sun, "Deep Residual Learning for Image Recognition," *EE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 13, no. 2, pp. 45-52, 2021.