

Pemanfaatan Yolo11 dan ByteTrack untuk Penghitungan Telur Berbasis Visi Komputer pada Konveyor Secara *Real-time*

Nurmahendra Harahap*, Jhoni Hidayat, Paula Risten Butarbutar, Akbar Dandi Aljaba

Fakultas Sains dan Teknologi, Teknik Elektro, Universitas Tjut Nyak Dhien, Medan, Indonesia

Email: ^{1*}nurmahendrasahap@gmail.com, ²hidayat.jhoni@gmail.com, ³paulabutarbutar63@gmail.com, ⁴Akbaraljaba@gmail.com.

Email Penulis Korespondensi: nurmahendrasahap@gmail.com

Submitted 27-03-2026; Accepted 30-04-2026; Published 30-06-2026

Abstrak

Penghitungan telur merupakan tahap penting dalam industri peternakan ayam petelur karena berpengaruh langsung terhadap produktivitas dan efisiensi pengelolaan hasil produksi. Metode penghitungan secara manual masih memiliki beberapa keterbatasan, di antaranya memerlukan waktu relatif lama, membutuhkan tenaga kerja dalam jumlah besar, serta memiliki tingkat kesalahan yang cukup tinggi akibat faktor manusia. Kondisi tersebut menunjukkan perlunya penerapan teknologi otomatis yang mampu meningkatkan kecepatan, ketepatan, dan konsistensi dalam proses penghitungan telur. Penelitian ini bertujuan untuk mengembangkan sistem penghitung telur otomatis berbasis visi komputer dengan mengintegrasikan Yolo11 sebagai metode deteksi objek dan ByteTrack sebagai metode pelacakan objek. Dataset telur dikumpulkan melalui proses pengambilan gambar pada berbagai kondisi, kemudian dilakukan proses anotasi dan augmentasi data sebelum digunakan pada tahap pelatihan model menggunakan Google Colab. Hasil pengujian menunjukkan bahwa sistem yang dikembangkan mampu mendeteksi dan menghitung telur dengan tingkat akurasi yang tinggi, di mana nilai presisi dan recall melampaui 0,90 serta rata-rata akurasi penghitungan telur mencapai 94,4% pada berbagai kondisi pengujian. Faktor utama yang memengaruhi kesalahan sistem adalah kecepatan konveyor yang tinggi yang menyebabkan motion blur serta variasi pencahayaan yang dapat menimbulkan false positive maupun false negative. Kontribusi utama penelitian ini adalah menghasilkan rancangan sistem penghitungan menggunakan kamera, sehingga mengurangi kesalahan dalam menghitung jumlah telur. Hasil penelitian ini menunjukkan bahwa integrasi Yolo11 dan ByteTrack berpotensi meningkatkan efisiensi proses penghitungan telur secara otomatis serta memberikan kontribusi terhadap pengembangan teknologi visi komputer dalam industri peternakan modern.

Kata Kunci: Bytrack; Deteksi Objek; Konveyor; Visi Komputer; Yolo11

Abstract

Egg counting is an important stage in the laying hen farming industry because it directly affects productivity and the efficiency of production management. Manual counting methods still have several limitations, including relatively long processing time, the need for a large workforce, and a fairly high error rate due to human factors. These conditions indicate the need for the implementation of automated technology capable of improving speed, accuracy, and consistency in the egg counting process. This study aims to develop an automatic egg counting system based on computer vision by integrating YOLO11 as the object detection method and ByteTrack as the object tracking method. The egg dataset was collected through image acquisition under various conditions, followed by annotation and data augmentation processes before being used in the model training stage using Google Colab. The test results show that the developed system is capable of detecting and counting eggs with a high level of accuracy, where precision and recall values exceed 0.90, and the average counting accuracy reaches 94.4% under various testing conditions. The main factors affecting system errors are high conveyor speed, which causes motion blur, and lighting variations that can lead to false positives and false negatives. The main contribution of this research is the development of a camera-based counting system design, thereby reducing errors in counting the number of eggs. The results of this study indicate that the integration of YOLO11 and ByteTrack has the potential to improve the efficiency of automated egg counting processes and contribute to the advancement of computer vision technology in the modern poultry industry.

Keywords: Bytrack; Computer Vision; Conveyor; Object Detection; Yolo11

1. PENDAHULUAN

Penghitungan telur merupakan tahapan krusial dalam industri peternakan ayam petelur yang secara langsung memengaruhi produktivitas, pengendalian inventaris, dan kualitas keluaran secara keseluruhan. Metode penghitungan telur secara manual yang masih umum digunakan saat ini memiliki berbagai keterbatasan. Proses ini membutuhkan banyak tenaga kerja, memakan waktu yang signifikan, dan sangat rentan terhadap kesalahan manusia [1]. Penelitian menunjukkan bahwa kurangnya ketelitian dalam identifikasi telur selama penyortiran manual sering kali menyebabkan pengategorian yang tidak tepat, yang pada gilirannya mengurangi akurasi dan kualitas produk akhir. Produktivitas yang rendah, dengan rata-rata kurang dari 1000 telur per jam yang dapat disortir secara manual, juga berkontribusi pada tingginya biaya tenaga kerja [2]. Keterbatasan ini secara langsung berdampak pada profitabilitas peternakan ayam petelur. Bahkan dalam lingkungan penelitian yang terkontrol, penghitungan manual terbukti tidak sepenuhnya andal. Studi menunjukkan bahwa peneliti berpengalaman pun dapat mengalami kelelahan dan melakukan kesalahan klasifikasi saat menghitung telur secara manual, yang berpotensi menyebabkan kesalahan dalam fenotipe [3]. Sistem penghitung analog yang saat ini digunakan dalam industri telur juga tidak sepenuhnya bebas dari masalah. Sistem ini memerlukan investasi yang besar untuk instalasi dan pemeliharaan, dan sering kali rentan terhadap kesalahan [4].

Dari pemaparan masalah-masalah yang ada, penelitian bertujuan untuk membuat sebuah perangkat yang mampu menghitung telur ayam otomatis secara *real-time* untuk mempermudah industri peternakan khususnya industri ayam petelur. Terdapat beberapa sistem yang telah dikerjakan pada penelitian yang telah lalu, beberapa diantaranya sistem penghitung telur secara *real-time* menggunakan kamera Raspberry Pi dan operasi morfologi. Gambar diproses dengan

filter dan konversi biner, lalu telur dihitung dengan matriks kontur. Hasilnya ditampilkan di website dengan akurasi 100% [5]. Sistem penghitung telur digital berbasis konveyor dan sensor inframerah untuk meningkatkan akurasi dan efisiensi dalam produksi komersial. Sistem ini menggunakan Arduino-Uno sebagai pemroses data dari sensor inframerah, yang mendeteksi dan menghitung telur tanpa kontak fisik [6]. Pengembangan sistem visi computer menggunakan algoritma *Faster R-CNN* telah dilakukan untuk menghitung ayam secara otomatis dalam kandang baterai di peternakan unggas komersial [7]. Xiangheng Wangdkk, membahas pengembangan sistem berbasis kecerdasan buatan untuk mendeteksi noda kotor pada cangkang telur menggunakan YOLOv4 yang dipercepat dengan TensorRT dan diimplementasikan pada Jetson Nano[8]. Selanjutnya terdapat pula pengembangan sistem penghitung telur otomatis berbasis sensor optik jarak pada konveyor. Fokus utama penelitian adalah perumusan deskripsi matematis untuk algoritma deteksi objek menggunakan metode pencarian maksimum lokal [9].

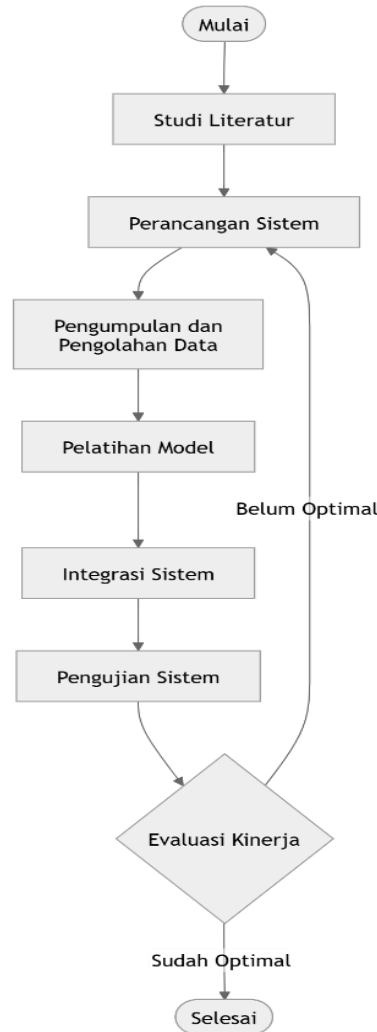
Terdapat beberapa studi yang menggunakan model yolo untuk mendeteksi objek lain untuk tugas berkaitan dengan telur, seperti penggunaan YOLOv8 untuk deteksi telur merpati (YOLOv8-PG) dengan peningkatan untuk penerapan *lightweight*[10]. YOLOv4 telah digunakan untuk pengelompokan ukuran telur secara *real-time* [11]. Keberhasilan penggunaan YOLOv4 untuk mengklasifikasikan telur berdasarkan ukuran menunjukkan kemampuan kerangka kerja YOLO untuk menganalisis dimensi telur. YOLOv5 juga telah diterapkan dalam aplikasi pertanian seperti mendeteksi telur bebek yang pecah[12]. Rekam jejak YOLO yang terbukti dalam konteks pertanian menjadikan Yolo11 kandidat yang menjanjikan untuk mengatasi tantangan deteksi telur. Penelitian[13] menggunakan YOLOv3 untuk deteksi telur di konveyor. Pada dataset citra bergerak resolusi rendah, YOLOv3 unggul dengan efektivitas deteksi-pelacakan ~93%. Penelitian ini membandingkan juga template matching dan Faster R-CNN; hasilnya YOLOv3 paling cepat dan akurat (93%). Terdapat pula penelitian yang memperkenalkan *Tailored-YOLO* (YOLOv5 ringan) yang dipangkas parameternya, dipadukan dengan DeepSORT untuk menghitung paket di jalur logistic hasilnya mAP mencapai 99.50% dan akurasi Hitung mencapai 99.80%[14]. Xavier [15] memaparkan sistem YOLOv5 untuk deteksi balok jerami dan kotak produk pada lini produksi. Dengan perangkat keras khusus, mereka melaporkan akurasi penghitungan mencapai 99,91% jauh melebihi hitung manual. Meskipun demikian, sistem ini diuji pada satu jenis konveyor industri; generalisasi ke objek lain atau perubahan latar belum diuji.

Terdapat pula aplikasi *smart egg counting* yang menggunakan sensor foto listrik untuk menghitung jumlah telur yang melewati konveyor[16]. Suyanto [17], mendesain sistem penghitungan jumlah telur ayam berbasis IoT menggunakan sensor *photodiode*. Penelitian ini mengusulkan implementasi sistem penghitungan telur otomatis yang didukung oleh teknologi visi computer. Berdasarkan beberapa ulasan penelitian yang telah diulas, kebaruan penelitian ini terletak pada integrasi Yolo11 dengan ByteTrack untuk menghitung telur pada sistem konveyor. Pemilihan Yolo11 didasarkan pada kemampuannya yang terbukti dalam memberikan kecepatan dan akurasi tinggi, serta efisiensinya dalam pemrosesan data visual[18], sementara pemilihan algoritma ByteTrack dikarenakan terbukti lebih akurat dan efisien dibandingkan dengan SORT dan DeepSORT dalam melacak objek[19]. ByteTrack akan melacak setiap telur yang terdeteksi saat bergerak di sepanjang konveyor. Kontribusi utama penelitian ini terletak pada tiga hal, yaitu: (1) perancangan arsitektur sistem penghitung telur yang menggabungkan deteksi dan pelacakan objek secara *real-time*; (2) pengujian sistem pada variasi kecepatan konveyor dan kondisi pencahayaan; serta (3) evaluasi kinerja menggunakan precision, recall, F1-score, mean Average Precision, dan error rate sebagai dasar penilaian akurasi penghitungan. Dengan demikian, penelitian ini diharapkan dapat memberikan solusi yang lebih efektif untuk otomasi penghitungan telur pada lingkungan produksi. Keberhasilan sistem yang dikembangkan akan dievaluasi berdasarkan metrik utama seperti akurasi deteksi (*precision*, *recall* dan *F1-Score*) akurasi penghitungan.

2. METODOLOGI PENELITIAN

2.1 Tahapan Penelitian

Penelitian ini menggunakan pendekatan eksperimen berbasis pengembangan sistem (*experimental research and system development*) untuk merancang dan menguji alat penghitung telur otomatis pada konveyor. Pendekatan ini dipilih karena memungkinkan evaluasi langsung terhadap kinerja model deteksi dan pelacakan objek dalam kondisi nyata. Sistem yang dikembangkan memanfaatkan algoritma Yolo11 untuk deteksi objek dan ByteTrack untuk pelacakan objek secara *real-time*. Kedua metode tersebut dipilih karena memiliki performa yang tinggi dalam tugas deteksi dan tracking objek bergerak. Tahapan penelitian disusun secara sistematis yang meliputi studi literatur, perancangan sistem, pengumpulan dan pengolahan data, pelatihan model, integrasi sistem, serta pengujian dan evaluasi kinerja seperti yang ditampilkan pada Gambar 1.



Gambar 1. Tahapan-tahapan Penelitian

Tahap awal penelitian dilakukan melalui studi literatur untuk mengidentifikasi permasalahan, menentukan kebutuhan sistem, serta memilih metode yang sesuai. Penelusuran pustaka komprehensif sangat penting untuk memahami konteks penelitian, karena perumusan pertanyaan penelitian yang baik bergantung pada *knowledge* literatur terkini[20]. Literatur yang dikaji mencakup penelitian terkait deteksi objek berbasis YOLO, algoritma pelacakan ByteTrack, serta sistem penghitungan objek pada konveyor. Berdasarkan kajian tersebut, ditentukan spesifikasi perangkat keras dan perangkat lunak yang digunakan, yaitu kamera sebagai alat akuisisi citra, laptop sebagai unit pemrosesan utama, dan konveyor sebagai media pengujian. Selain itu, dilakukan perancangan sistem yang mencakup penentuan area pengamatan (*region of interest*), jalur pergerakan objek, serta mekanisme penghitungan berbasis lintasan objek.

Pada tahap pengembangan dataset, Pengumpulan dan pelabelan data yang sistematis diakui sebagai tahapan krusial dalam pengembangan model visi komputer; misalnya, Roh dkk. mencatat bahwa proses pengumpulan data umumnya mencakup akuisisi citra dan anotasi yang memadai[21]. Pada tahap ini citra telur dikumpulkan dari berbagai sudut pandang dan kondisi pencahayaan untuk menghasilkan data yang representative seperti pada Gambar 2. Dataset kemudian diproses menggunakan Roboflow melalui tahap preprocessing berupa pelabelan data, *auto-orientation* dan *resizing* menjadi resolusi 640×640 piksel. Selanjutnya dilakukan augmentasi data untuk meningkatkan variasi dan memperkuat kemampuan generalisasi model. Teknik augmentasi yang digunakan meliputi rotasi horizontal dan vertikal serta cropping dengan rentang 0% hingga 20%. Hasil dari proses ini menghasilkan total 868 citra yang kemudian dibagi menjadi 783 data latih dan 85 data validasi. Proses pelatihan model dilakukan menggunakan Google Colab dengan memanfaatkan GPU untuk mempercepat komputasi.

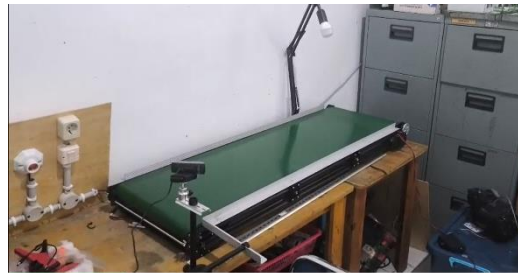




Gambar 2. Gambar Telur yang Diampil Dari Berbagai Sudut

Model Yolo11 yang telah dilatih kemudian diintegrasikan dengan algoritma ByteTrack untuk melakukan pelacakan objek secara konsisten. Integrasi ini bertujuan untuk menghindari perhitungan ganda serta memastikan setiap objek hanya dihitung satu kali ketika melewati area pengamatan. Sebelum diterapkan pada sistem nyata, model diuji dalam lingkungan simulasi untuk memastikan stabilitas dan akurasi deteksi. Target awal yang ditetapkan dalam penelitian ini adalah mencapai tingkat akurasi deteksi minimal sebesar 90%.

Tahap pengujian dilakukan dengan mengimplementasikan sistem pada perangkat konveyor yang dilengkapi kamera dan laptop seperti pada Gambar 3. Pengujian dilakukan dengan beberapa skenario, yaitu variasi kecepatan konveyor dan variasi kondisi pencahayaan. Setiap hasil penghitungan sistem dibandingkan dengan penghitungan manual sebagai acuan (*ground truth*). Perbedaan antara hasil sistem dan penghitungan manual digunakan untuk menghitung tingkat kesalahan (*error rate*). Selain itu, dilakukan analisis terhadap kesalahan yang terjadi, seperti *false positive*, *false negative*, dan pengaruh *motion blur* akibat pergerakan objek. Sebagaimana dicatat Zhang dkk., efek *ghosting* akibat gerakan kamera (*motion blur*) dapat secara signifikan menurunkan akurasi deteksi objek[22].



Gambar 3. Perangkat Penghitung Telur Otomatis

Evaluasi kinerja sistem dilakukan menggunakan beberapa parameter, yaitu *precision*, *recall*, F1-score, dan *mean Average Precision (mAP)*. Selain itu, digunakan juga parameter *error rate* untuk menilai akurasi sistem dalam menghitung jumlah telur. Target akhir penelitian ini adalah menghasilkan sistem dengan tingkat kesalahan kurang dari 5% dibandingkan metode manual. Untuk meningkatkan kinerja sistem, dilakukan optimasi melalui penyesuaian *threshold* deteksi, penerapan *non-maximum suppression*, serta penggunaan *region of interest* sebagai pembatas area penghitungan. Dengan metode yang dirancang, sistem diharapkan mampu bekerja secara akurat, efisien, dan dapat diimplementasikan pada lingkungan industri secara nyata. Metrik evaluasi standar seperti *precision*, *recall*, dan *accuracy* umum digunakan untuk menilai performa model deteksi objek dan memberikan gambaran kinerja sistem secara kuantitatif[23].

3. HASIL DAN PEMBAHASAN

3.1 Pembahasan Hasil Pelatihan Model

Berdasarkan data pelatihan selama 120 epoch, model menunjukkan pola konvergensi yang cukup jelas. Secara umum, performa model meningkat pesat pada fase awal pelatihan, lalu cenderung stabil pada fase pertengahan hingga akhir. Hal ini terlihat dari kenaikan metrik evaluasi seperti *precision*, *recall*, mAP50, dan mAP50-95, yang diikuti oleh penurunan nilai *loss* pada data pelatihan maupun validasi. Pola ini mengindikasikan bahwa model berhasil mempelajari karakteristik objek dengan baik dan mampu melakukan generalisasi secara cukup konsisten.

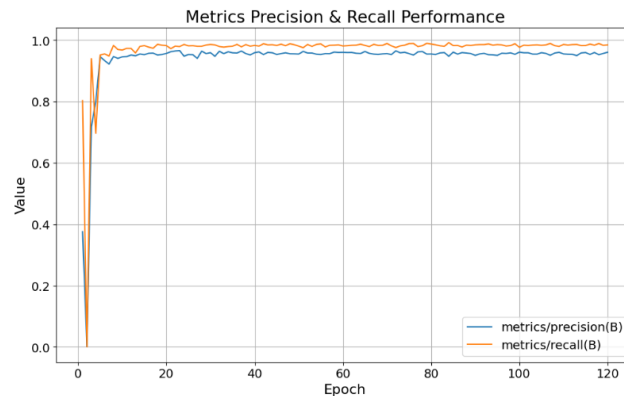
Pada epoch awal, performa model masih fluktuatif. Epoch pertama menunjukkan *precision* sebesar 0,37603, *recall* 0,80310, mAP50 sebesar 0,32682, dan mAP50-95 sebesar 0,24419. Nilai ini masih relatif rendah karena model baru mulai menyesuaikan bobot awal terhadap karakteristik dataset. Pada epoch kedua, hampir seluruh metrik turun sangat tajam, bahkan *precision* menjadi 0,00111, *recall* 0,00046, mAP50 0,00056, dan mAP50-95 0,00045. Penurunan drastis seperti ini umumnya terjadi akibat proses *warm-up*, penyesuaian *learning rate*, atau dinamika awal optimasi yang belum stabil. Dengan kata lain, kondisi tersebut bukan berarti model gagal, melainkan masih berada pada fase adaptasi awal.

Setelah fase sangat awal tersebut, model mulai menunjukkan peningkatan yang sangat cepat. Pada epoch ketiga, *precision* naik menjadi 0,71864, *recall* menjadi 0,90911, mAP50 menjadi 0,73036, dan mAP50-95 menjadi 0,56991. Lonjakan ini menunjukkan bahwa model mulai mengenali pola objek dengan lebih baik. Memasuki epoch keempat dan kelima, performa meningkat lebih jauh. Pada epoch kelima, *precision* telah mencapai 0,94651, *recall* 0,96169, mAP50 0,98380, dan mAP50-95 0,80526. Artinya, hanya dalam lima epoch setelah fase awal, model sudah mampu mencapai

tingkat deteksi yang sangat tinggi pada *threshold* IoU 0,5. Ini menunjukkan dataset cukup informatif dan model arsitektural yang digunakan mampu belajar dengan efisien.

3.2 Perkembangan *Precision* dan *Recall*

Precision merupakan indikator seberapa akurat prediksi yang dihasilkan model, yaitu proporsi prediksi positif yang benar terhadap seluruh prediksi positif. Dari data pelatihan yang ditampilkan pada Gambar 4, *precision* meningkat dari 0,37603 pada epoch pertama menjadi 0,96097 pada epoch terakhir. Nilai *precision* tertinggi tercapai pada epoch ke-23, yaitu 0,96627. Ini menunjukkan bahwa di pertengahan pelatihan, model mampu menghasilkan prediksi dengan tingkat kesalahan *false positive* yang sangat rendah.



Gambar 4. Performa Metrik Presisi Dan *Recall*

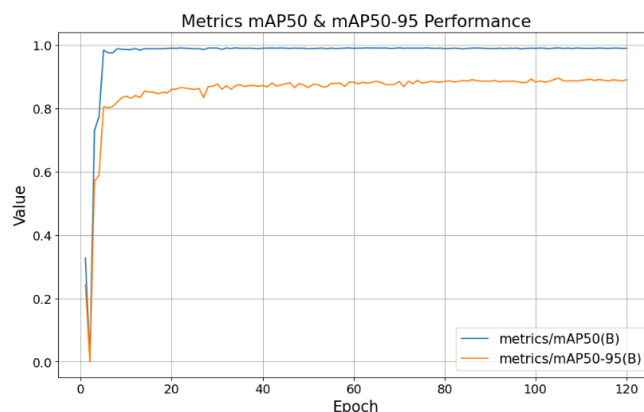
Peningkatan *precision* yang cukup cepat ini memperlihatkan bahwa model tidak hanya belajar mendeteksi objek, tetapi juga belajar membedakan objek target dari latar belakang secara efektif. Dalam konteks deteksi objek, *precision* yang tinggi sangat penting karena menunjukkan bahwa sebagian besar kotak prediksi yang dihasilkan benar-benar merepresentasikan objek yang dimaksud[24]. Dengan *precision* yang mendekati 0,97, model dapat dikatakan cukup selektif dan tidak terlalu banyak menghasilkan deteksi yang salah.

Sementara itu, *recall* menggambarkan kemampuan model dalam menemukan seluruh objek yang seharusnya terdeteksi. *Recall* pada epoch pertama sudah relatif tinggi, yaitu 0,80310, lalu naik menjadi 0,99225 pada epoch ke-84, yang merupakan nilai tertinggi selama pelatihan. Pada epoch terakhir, *recall* tetap sangat tinggi, yakni 0,98496. Nilai *recall* yang sangat tinggi ini menandakan bahwa model hampir selalu berhasil menemukan objek target ketika objek tersebut memang ada pada citra atau video.

Jika *precision* menilai ketepatan prediksi, *recall* menilai kelengkapan deteksi. Kombinasi *precision* dan *recall* yang sama-sama tinggi menunjukkan model berada pada kondisi seimbang. Tidak banyak objek yang terlewat, dan tidak banyak pula prediksi yang salah. Ini merupakan karakteristik yang sangat baik, terutama untuk penelitian deteksi objek pada skenario nyata, karena sistem harus mampu mendeteksi objek secara akurat sekaligus lengkap.

3.3 Perkembangan mAP50 dan mAP50-95

Metrik utama yang paling penting dalam evaluasi deteksi objek adalah mAP. Sesuai data pada Gambar 5, mAP50 mencapai nilai tertinggi sebesar 0,99112 pada epoch ke-70. Pada epoch terakhir, nilai mAP50 tetap tinggi di 0,98987. Angka ini menunjukkan bahwa model memiliki kemampuan deteksi yang sangat baik ketika menggunakan *threshold* IoU 0,5. Dalam praktik evaluasi YOLO, mAP50 yang mendekati 1,0 menandakan bahwa prediksi *bounding box* hampir selalu berada pada posisi yang sangat dekat dengan *ground truth* pada batas toleransi yang lebih longgar.



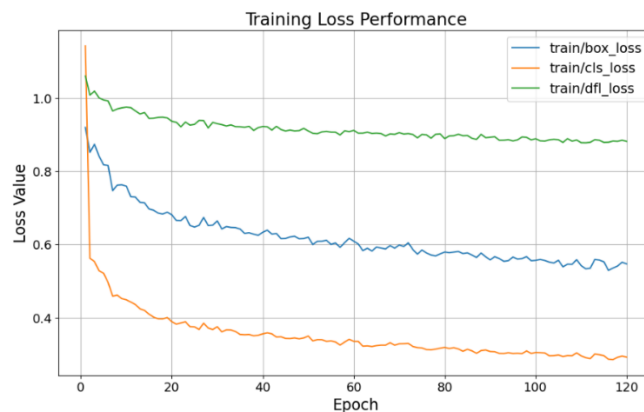
Gambar 5. Performa Metrik mAP50 & mAP50-95

Namun, ketika evaluasi menjadi lebih ketat melalui mAP50-95, performa model tentu lebih rendah dibanding mAP50, dan ini merupakan hal yang normal. Nilai mAP50-95 tertinggi pada data ini adalah 0,89525 yang dicapai pada epoch ke-105, sedangkan nilai pada epoch terakhir adalah 0,89065. Angka ini tetap sangat tinggi, karena mAP50-95 menguji kualitas deteksi pada berbagai ambang IoU dari 0,5 hingga 0,95. Dengan kata lain, model tidak hanya baik pada IoU rendah, tetapi juga cukup presisi ketika batas kesesuaian kotak pembatas diperketat.

Selisih antara mAP50 dan mAP50-95 sekitar 0,10 poin menunjukkan bahwa model memang lebih kuat pada level deteksi umum dibanding level lokalisasi yang sangat presisi. Tetapi selisih ini masih tergolong wajar dan sehat. Jika selisih terlalu besar, itu bisa berarti model hanya bagus pada IoU rendah tetapi kurang presisi pada posisi *bounding box* [25]. Dalam kasus ini, karena mAP50-95 tetap berada di sekitar 0,89, model masih dapat dinilai sangat stabil dan presisi. Ini menunjukkan hasil pelatihan yang matang dan tidak hanya mengandalkan keberhasilan klasifikasi objek, tetapi juga ketepatan lokalisasi.

3.4 Analisis Loss Training dan Validation

Selain metrik evaluasi, penurunan *loss* juga penting untuk menilai keberhasilan pelatihan. Dari data yang ditampilkan pada Gambar 6, *train/box_loss* menurun dari 0,91958 pada epoch pertama menjadi 0,54670 pada epoch terakhir, dengan nilai terendah 0,52922 pada epoch ke-116. Penurunan ini menandakan bahwa kemampuan model dalam memprediksi lokasi *bounding box* semakin baik. Artinya, kesalahan dalam menentukan posisi dan ukuran kotak deteksi makin kecil seiring bertambahnya epoch.

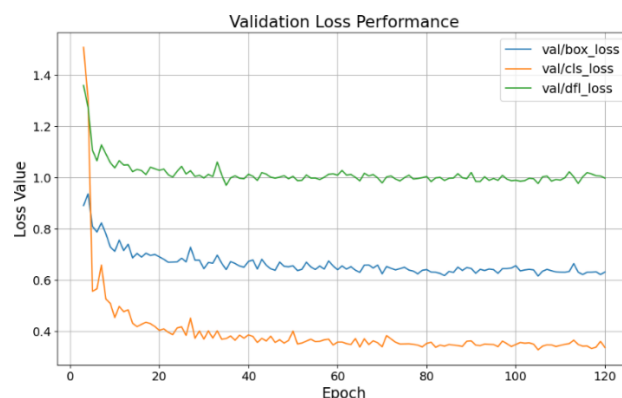


Gambar 6. Training Loss Selama Beberapa Epochs

Untuk *train/cls_loss*, penurunannya lebih tajam, dari 1,14239 menjadi 0,29210, dengan nilai minimum 0,28528 pada epoch ke-117. Ini memperlihatkan bahwa model semakin baik dalam mengklasifikasikan objek ke kelas yang benar. Dalam deteksi objek, penurunan *cls loss* yang signifikan biasanya menunjukkan model berhasil memahami perbedaan antar kelas, atau jika hanya satu kelas, model berhasil membedakan objek dari *background* dengan lebih akurat.

Sementara itu, *train/df_l_loss* juga menurun dari 1,05990 menjadi 0,88137, dengan nilai minimum 0,87764 pada epoch ke-111. *Distribution Focal Loss* berhubungan dengan presisi regresi *bounding box*. Penurunan *loss* ini menguatkan kesimpulan bahwa model tidak hanya lebih tepat mengenali objek, tetapi juga semakin presisi dalam membentuk batas kotaknya.

Pada data validasi seperti yang tertera di Gambar 7, tren juga menunjukkan perbaikan. *val/box_loss* mencapai nilai minimum 0,61512 pada epoch ke-105, *val/cls_loss* minimum 0,32706 pada epoch ke-105, dan *val/df_l_loss* minimum 0,96943 pada epoch ke-35. Fakta bahwa *validation loss* ikut menurun menunjukkan model memiliki generalisasi yang baik terhadap data yang tidak dilihat saat training. Meskipun pada fase akhir ada sedikit variasi naik-turun, nilai validasi secara umum tetap stabil dan tidak menunjukkan gejala *overfitting* yang berat.



Gambar 7. Validation Loss Selama Beberapa Epoch

3.5 Keseimbangan Antara *Training* dan *Validation*

Salah satu hal penting dari data ini adalah adanya keseimbangan yang cukup baik antara *training loss* dan *validation loss*. Jika sebuah model mengalami overfitting berat, biasanya *training loss* akan terus turun tajam, sedangkan *validation loss* justru naik atau tidak stabil [26]. Pada data ini, hal tersebut tidak terlihat secara ekstrem. *Training loss* menurun, *validation loss* juga menurun, dan metrik evaluasi tetap tinggi sampai akhir.

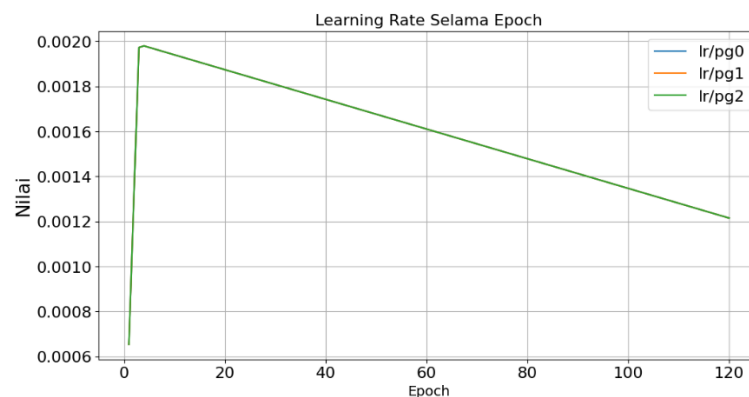
Rata-rata 10 epoch pertama menunjukkan kondisi model yang masih belajar dasar-dasar dataset. Pada fase ini, *precision* rata-rata sekitar 0,753, *recall* 0,822, mAP50 0,773, dan mAP50-95 0,631. Sebaliknya, rata-rata 10 epoch terakhir meningkat menjadi *precision* 0,956, *recall* 0,985, mAP50 0,990, dan mAP50-95 0,889. Perbandingan ini menegaskan bahwa model benar-benar mengalami perbaikan performa yang signifikan dari awal hingga akhir pelatihan.

Dari sisi *loss*, rata-rata *training loss* pada 10 epoch pertama juga jauh lebih tinggi dibanding 10 epoch terakhir. Ini berarti proses pembelajaran berjalan efektif. Model tidak hanya menghafal data, tetapi semakin mampu melakukan generalisasi. Karena itu, jika hasil ini digunakan dalam konteks penelitian, dapat dikatakan bahwa arsitektur model, kualitas data, serta skema pelatihan yang digunakan telah mendukung performa deteksi yang tinggi.

3.6 Pengaruh *Learning Rate* dan Konvergensi

Pada Gambar 8 terlihat nilai *learning rate* untuk tiga parameter group yang bergerak secara seragam. Nilai *learning rate* naik pada tahap awal dan kemudian menurun secara bertahap sampai epoch akhir. Pola ini umumnya digunakan untuk membantu model belajar cepat di awal, lalu melakukan penyesuaian yang lebih halus pada tahap akhir. Strategi seperti ini sangat membantu dalam mencapai konvergensi yang stabil.

Learning rate yang menurun secara perlahan memungkinkan model keluar dari kondisi optimasi yang kasar dan masuk ke tahap *fine-tuning*. Hal ini terlihat pada metrik mAP50-95 yang terus membaik hingga epoch ke-105. Dengan kata lain, walaupun mAP50 sudah mencapai titik sangat tinggi sejak sekitar epoch 70, model masih tetap memperbaiki kualitas lokalisasi bounding box pada epoch-epoch berikutnya. Ini merupakan tanda bahwa proses training berjalan dengan baik dan menghasilkan model yang matang.



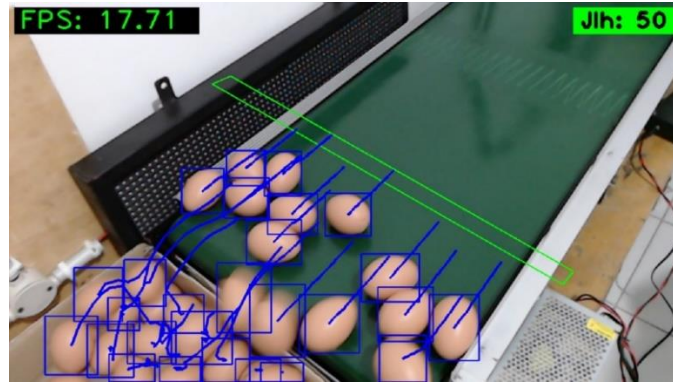
Gambar 8. Tingkat Pembelajaran (lr/pg0,lr/pg1,lr/pg2) Selama Epoch

3.7 Interpretasi Umum Hasil Model

Berdasarkan seluruh data, model yang dilatih dapat dikatakan memiliki performa sangat baik. Nilai mAP50 sebesar 0,99112 dan mAP50-95 sebesar 0,89525 menunjukkan bahwa model sangat kuat dalam mendeteksi objek pada *threshold* umum maupun pada pengujian yang lebih ketat. *Precision* dan *recall* yang sama-sama tinggi juga menegaskan bahwa model tidak hanya sering benar saat mendeteksi objek, tetapi juga jarang melewatkan objek yang seharusnya terdeteksi. Jika dilihat dari bentuk kurva pelatihan secara konseptual, model mencapai fase stabil setelah beberapa epoch awal. Metrik utama meningkat cepat, lalu menetap pada level tinggi dengan fluktuasi kecil. Kondisi ini ideal untuk penelitian deteksi objek karena menunjukkan bahwa model sudah cukup belajar dan tidak mengalami instabilitas yang serius.

3.8 Pengujian Sistem Secara *Real-time*

Setelah melakukan *training data*, dintegrasikanlah sistem dan algoritma Bytrack dengan hasil *training data* tersebut, yang kemudian dilanjutkan dengan uji coba perhitungan telur seperti terlihat pada Gambar 5.



Gambar 9. Tampilan *Output* Peghitung Telur Otomatis

Tabel 1 mengungkapkan pola yang jelas mengenai faktor-faktor yang memengaruhi keandalannya. Secara umum, sistem ini menunjukkan robustisitas yang tinggi, namun kinerjanya terdegradasi pada kondisi operasional spesifik yang berkecepatan tinggi.

Tabel 1. Ringkasan Jumlah Kesalahan Deteksi pada Setiap Kondisi Pengujian

| Kondisi Pengujian | | Jumlah Percobaan | Total Kesalahan | | Akurasi (%) | Presisi (%) | Recall (%) |
|----------------------|--------------|------------------|-----------------|----|-------------|-------------|------------|
| Kecepatan (cm/detik) | Cahaya (LUX) | | FP | FN | | | |
| 4,4 | 57 | 9 | 0 | 0 | 100 | 100 | 100 |
| 4,4 | 105 | 9 | 0 | 0 | 100 | 100 | 100 |
| 6,79 | 57 | 9 | 1 | 0 | 100,28 | 99,73 | 100 |
| 6,79 | 105 | 9 | 0 | 0 | 100 | 100 | 100 |
| 9,74 | 57 | 9 | 0 | 1 | 99,72 | 100 | 99,72 |
| 9,74 | 105 | 9 | 1 | 0 | 100,37 | 99,64 | 100 |

Data menunjukkan bahwa alat sangat akurat dan handal dalam melakukan perhitungna jumlah objek, hal ini dapat dilihat dari kegagalan dalam menghitung objek. Alat berhasil menghitung 51 dari 54 percobaan (tingkat keberhasilan 94,4%). Kecepatan konveyor adalah variabel paling berpengaruh terhadap stabilitas sistem. Pada kecepatan terendah (4,4 cm/detik), sistem mencapai kinerja sempurna 100% tanpa satu pun kesalahan dalam 18 percobaan di kedua kondisi pencahayaan. Sebaliknya, seluruh tiga insiden kesalahan (100% dari total kesalahan) terjadi secara eksklusif pada kecepatan yang lebih tinggi, yaitu 6,79 cm/detik dan 9,74 cm/detik.

Peningkatan kecepatan secara langsung mengurangi waktu akuisisi gambar, sehingga meningkatkan risiko *motion blur* (gambar buram karena gerakan) yang dapat mengaburkan detail objek [27]. Hal ini menyulitkan algoritma segmentasi dan klasifikasi untuk bekerja secara akurat.

Meskipun kecepatan adalah pemicu utama, kondisi pencahayaan menentukan jenis kesalahan yang muncul. Seperti pada Cahaya Redup (57 LUX), kondisi ini terbukti paling problematis karena menjadi satu-satunya kondisi yang menghasilkan dua jenis kesalahan berbeda, satu *False Positive* (FP) pada 6,79 cm/detik dan satu *False Negative* (FN) pada 9,74 cm/detik. Hal ini dimung-kinkan karena dalam kondisi redup, noise ikut meningkat karena rendahnya SNR, jika kontras rendah atau noise terlalu tinggi, objek bisa tidak terdeteksi (FN), atau *noise* dianggap objek (FP) [28].

3.9 Perbandingan dengan Penelitian Terdahul

Hasil akurasi penghitungan telur yang diperoleh dari penelitian ini sebesar 94,4% dengan algoritma YOLO11 + ByteTrack (presisi dan recall masing-masing >90%). Hasil ini sebanding atau lebih baik dibanding literatur sebelumnya. Misalnya, Geffen et al, hanya mencapai 89,6% akurasi (MAE 2,5 ayam per sangkar) pada sistem deteksi dan perhitungan ayam [7]. Lika et al, melaporkan *F1-score* 0,89 untuk pengelompokan ukuran telur menggunakan YOLOv4 [11], sedangkan Jiang et al, dengan model YOLOv8 meningkatkan F1, mAP hingga 99%+, menunjukkan presisi deteksi yang sangat tinggi [10]. Perbedaan mencolok terutama terletak pada metode: studi ini menggunakan CV berbasis YOLO di lingkungan konveyor bergerak, sedangkan banyak studi lain memakai sensor jarak (Ogurtsov 2024) atau arsitektur berbeda (Faster R-CNN oleh Geffen 2020). Faktor kendala seperti blur gerak (kecepatan konveyor tinggi) dan variasi pencahayaan juga dibahas. Beberapa penelitian menekankan bahwa penanganan blur atau peningkatan citra (mis. Sayed & Brostow 2021, Yoon & Cho 2023) dapat meningkatkan deteksi hingga puluhan persen [27][28]. Secara keseluruhan, hasil kita konsisten dengan atau lebih baik dari literatur (akurasi tinggi melebihi 90%), namun terdapat ruang untuk memperbaiki kondisi pengujian (seperti optimasi sudut pandang kamera, pencahayaan) agar kesalahan penghitungan lebih diminimalisir.

4. KESIMPULAN

Berdasarkan hasil penelitian, dapat disimpulkan bahwa sistem penghitung telur otomatis berbasis visi komputer yang mengintegrasikan Yolo11 dan ByteTrack berhasil menjawab permasalahan utama dalam penghitungan telur secara manual, yaitu lambat, rentan kesalahan, dan kurang konsisten. Hasil pelatihan menunjukkan performa model yang sangat baik, ditunjukkan oleh *precision* dan *recall* yang sama-sama tinggi, mAP50 yang mencapai 0,99112, serta mAP50-95 sebesar 0,89525, sehingga model tidak hanya mampu mendeteksi objek dengan akurat pada ambang IoU yang umum, tetapi juga cukup presisi pada evaluasi yang lebih ketat. Pada pengujian nyata, sistem juga mampu mencapai tingkat keberhasilan penghitungan sebesar 94,4%, yang memperlihatkan bahwa integrasi deteksi dan pelacakan objek dapat bekerja secara efektif pada konveyor. Selain itu, hasil analisis menunjukkan bahwa kecepatan konveyor dan kondisi pencahayaan merupakan faktor yang paling memengaruhi kesalahan sistem, terutama karena motion blur pada kecepatan tinggi dan penurunan kualitas citra pada cahaya redup. Dengan demikian, penelitian ini membuktikan bahwa sistem yang dikembangkan layak digunakan sebagai dasar pengembangan teknologi penghitungan telur otomatis pada lingkungan industri. Namun, penelitian ini masih memiliki keterbatasan, yakni pengujian dilakukan pada kondisi dan skenario yang masih terbatas, sehingga generalisasi sistem pada variasi objek, latar, serta kondisi operasional yang lebih beragam masih perlu diuji lebih lanjut. Perbaikan pada penelitian berikutnya dapat diarahkan pada perluasan dataset, peningkatan ketahanan terhadap perubahan pencahayaan dan blur, serta optimasi sistem agar lebih stabil pada kecepatan konveyor yang lebih tinggi.

REFERENCES

- [1] A. Vinod, D. Mohanty, A. John, and B. Depuru, "Application of Artificial Intelligence in Poultry Farming - Advancing Efficiency in Poultry Farming By Automating The Egg Counting Using Computer Vision System," 2023. doi: 10.21203/rs.3.rs-3266412/v1.
- [2] Advantech, "Revolutionizing Egg Production: An AI-Based Optical Inspection System." [Online]. Available: <https://www.advantech.com/en/resources/case-study/revolutionizing-egg-production-an-ai-based-optical-inspection-system>
- [3] S. P. Fraher *et al.*, "A comparison of three automated root-knot nematode egg counting approaches using machine learning, image analysis, and a hybrid model," *Plant Dis.*, vol. 108, no. 9, pp. 2625–2629, 2024.
- [4] D. S. Smith and G. H. Hoggard, "Counting Eggs With AI," 2022. [Online]. Available: <https://www.sfmagazine.com/articles/2022/february/counting-eggs-with-ai/>
- [5] I. Kanjanasurat, W. Krungseanmuang, V. Chaowalittawin, and B. Purahong, "Egg-Counting System Using Image Processing and a Website for Monitoring," in *2021 7th International Conference on Engineering, Applied Sciences and Technology (ICEAST)*, 2021, pp. 101–104. doi: 10.1109/ICEAST52143.2021.9426295.
- [6] J. B. Ajewole *et al.*, "Development of A Digital Egg Counting System for Eggs on A Conveyor Belt Using Sensor," in *2024 International Conference on Science, Engineering and Business for Driving Sustainable Development Goals (SEB4SDG)*, 2024, pp. 1–10. doi: 10.1109/SEB4SDG60871.2024.10629932.
- [7] O. Geffen, Y. Yitzhaky, N. Barchilon, S. Druyan, and I. Halachmi, "A machine vision system to detect and count laying hens in battery cages," *animal*, vol. 14, no. 12, pp. 2628–2634, 2020, doi: 10.1017/S1751731120001676.
- [8] X. Wang, X. Yue, H. Li, and L. Meng, "A high-efficiency dirty-egg detection system based on YOLOv4 and TensorRT," in *2021 International Conference on Advanced Mechatronic Systems (ICAMechS)*, 2021, pp. 75–80. doi: 10.1109/ICAMechS54019.2021.9661509.
- [9] S. Ogurtsov, E. Karikov, and D. Bushuev, "DEVELOPMENT OF AN AUTOMATIC EGG COUNTING SYSTEM BASED ON LOCAL MAXIMUM SEARCH FOR EGGS TRANSPORTED ON A CONVEYOR," *Acad. J. Manuf. Eng.*, vol. 22, pp. 29–33, 2024.
- [10] T. Jiang *et al.*, "Improved YOLOv8 model for lightweight pigeon egg detection," *Animals*, vol. 14, no. 8, p. 1226, 2024.
- [11] C. Lika, M. Sandy, A. Husna, and R. A. Rizal, "Real Time Chicken Egg Size Classification Using Yolov4 Algorithm," vol. 4, no. 2, pp. 577–584, 2024.
- [12] Y. Luo, Y. Huang, Q. Wang, K. Yuan, Z. Zhao, and Y. Li, "An improved YOLOv5 model: Application to leaky eggs detection," *LWT*, vol. 187, p. 115313, 2023, doi: 10.1016/j.lwt.2023.115313.
- [13] M. Ulaszewski, R. Janowski, and A. Janowski, "Application of computer vision to egg detection on a production line in real time.," *ELCVIA Electron. Lett. Comput. Vis. Image Anal.*, vol. 20, pp. 113–143, Feb. 2022, doi: 10.5565/rev/elcvia.1390.
- [14] S. Xie, Y. Zhou, I. Zhong, W. Yan, and Q. Zhang, "A package auto-counting model based on tailored YOLO and DeepSort techniques," *MATEC Web Conf.*, vol. 355, p. 2054, Jan. 2022, doi: 10.1051/mateconf/202235502054.
- [15] R. J. Xavier, C. F. O. Viegas, B. C. Costa, and R. P. Ishii, "A Deep Learning Approach for Automatic Counting of Bales and Product Boxes in Industrial Production Lines," in *International Conference on Computational Science and Its Applications*, 2022, pp. 619–633.
- [16] F. L. Afriansyah, F. E. Purnomo, and B. Prasetyo, "Sistem Hitung Cepat Aplikasi Smart Egg Counting Untuk Mengoptimalkan Produksi Telur Dan Mengetahui Recording Hen Day Pada Peternakan Ayam Petelur: Quick Counting System Smart Egg Counting Application To Optimize Egg Production And Knowing Hen Day Recordin," *J-Dinamika J. Pengabd. Masy.*, vol. 7, no. 3, pp. 398–403, 2022.
- [17] S. Suryanto, E. Priatna, and F. M. S. Nursuwars, "Sistem Penghitungan Jumlah Telur Ayam Berbasis Internet Of Things," *J. Energy Electr. Eng.*, vol. 3, no. 2, 2022.
- [18] A. Vina, "All You Need to Know About Ultralytics YOLO11 and Its Applications." [Online]. Available: <https://www.ultralytics.com/blog/all-you-need-to-know-about-ultralytics-yolo11-and-its-applications>
- [19] M. Abouelyazid, "Comparative Evaluation of SORT, DeepSORT, and ByteTrack for Multiple Object Tracking in Highway Videos," *International J. Sustain. Infrastruct. Cities Soc.*, vol. 8, no. 11, pp. 42–52, 2023.
- [20] E. Barroga and G. J. Matanguihan, "A practical guide to writing quantitative and qualitative research questions and hypotheses in scholarly articles," *J. Korean Med. Sci.*, vol. 37, no. 16, 2022.

- [21] Y. Roh, G. Heo, and S. E. Whang, "A survey on data collection for machine learning: a big data-ai integration perspective," *IEEE Trans. Knowl. Data Eng.*, vol. 33, no. 4, pp. 1328–1347, 2019.
- [22] H. Zhang, Y. Wang, F. Li, and H. Wang, "Motion Blur Robust Wheat Pest Damage Detection with Dynamic Fuzzy Feature Fusion," *arXiv Prepr. arXiv2601.03046*, 2026.
- [23] T. Schlosser, M. Friedrich, T. Meyer, and D. Kowerko, "A consolidated overview of evaluation and performance metrics for machine learning and computer vision," *Tobias Schlosser, Michael Friedrich, Trixy Meyer, Danny Kowerko--Junior Prof. Media Comput. Chemnitz Univ. Technol.*, vol. 9107, 2024.
- [24] J. ROMERO RIVEROS, "A metric evaluation framework for object detection," 2018.
- [25] Z. Cai and N. Vasconcelos, "Cascade R-CNN: High quality object detection and instance segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 5, pp. 1483–1498, 2019.
- [26] B. Ghogh and M. Crowley, "The theory behind overfitting, cross validation, regularization, bagging, and boosting: tutorial," *arXiv Prepr. arXiv1905.12787*, 2019.
- [27] M. Sayed and G. Brostow, "Improved handling of motion blur in online object detection," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 1706–1716.
- [28] S. Yoon and J. Cho, "Low-light image contrast enhancement with adaptive noise attenuator for augmented vehicle detection," *Electronics*, vol. 12, no. 16, p. 3517, 2023.