

EVALUASI KINERJA MODEL YOLOV8 UNTUK DETEKSI OBJEK KECIL PADA ALAT BEDAH DENGAN DUKUNGAN AUGMENTASI ROBOFLOW

Nurul Burhan¹; Odi Nurdiawan²; Arif Rinaldi Dikananda³; Raditya Danar Dana⁴; Ade Rizki Rinaldi⁵

Program Studi Teknik Informatika¹
Program Studi Manajemen Informatika^{2,4},
Program Studi Rekayasa Perangkat Lunak^{3,5},
STMIK IKMI Cirebon^{1,2,3,4,5}
<https://ikmi.ac.id>
burhanpoetra6@gmail.com

(*) Corresponding Author :
Published : 30 Januari 2026

Abstract—Automated detection of surgical instruments in the operating room has significant potential to enhance patient safety and improve workflow efficiency. However, detecting small surgical tools remains challenging due to their limited pixel representation, occlusion, reflective surfaces, and class imbalance within public datasets. This study aims to design, implement, and evaluate a small surgical tool detection model using YOLOv8 with preprocessing enhancements provided by the Roboflow platform. The study utilizes the public Surgical Tools Detection dataset from Roboflow, followed by annotation verification and preprocessing that includes resizing to 640×640 pixels, flipping, and exposure adjustments to increase data variability. The dataset is partitioned into training (70%), validation (20%), and testing (10%) to ensure objective evaluation. The YOLOv8 model is trained using Ultralytics implementation on a GPU-based environment, with the best-performing weights selected from validation results. Final evaluation on the test set employs precision, recall, and mean Average Precision (mAP) metrics. Experimental findings indicate that the model achieves 89.3% precision, 86.7% recall, and 91.5% mAP50, demonstrating that YOLOv8 combined with Roboflow preprocessing provides accurate and stable detection performance for small surgical tools, suitable for simulated real-time scenarios. These results highlight the importance of a structured end-to-end pipeline and domain-specific augmentation strategies to enhance small object detection performance in surgical imagery.

Keywords: YOLOv8, surgical tool detection, small object detection, Roboflow preprocessing, public dataset

Abstrak—Deteksi alat bedah secara otomatis di ruang operasi berpotensi meningkatkan keselamatan pasien dan efisiensi kerja tim medis, namun deteksi alat berukuran kecil masih menantang karena ukuran objek yang hanya mencakup sedikit piksel, occlusion, pantulan alat, serta ketidakseimbangan kelas pada dataset publik. Penelitian ini bertujuan merancang, mengimplementasikan, dan mengevaluasi model deteksi alat bedah kecil berbasis YOLOv8 dengan dukungan preprocessing dari platform Roboflow. Dataset yang digunakan adalah Surgical Tools Detection dari Roboflow, kemudian diverifikasi anotasinya dan diproses melalui augmentasi seperti resizing ke 640×640 piksel, flipping, serta penyesuaian exposure untuk memperkaya variasi data. Dataset dibagi menjadi train (70%), validasi (20%), dan test (10%) agar evaluasi objektif. Pelatihan dilakukan menggunakan Ultralytics YOLOv8 pada lingkungan komputasi GPU, dengan pemilihan bobot terbaik berdasarkan performa validasi. Evaluasi akhir pada data uji memakai metrik precision, recall, dan mean Average Precision (mAP). Hasil menunjukkan model mencapai precision 89,3%, recall 86,7%, dan mAP50 91,5%, menandakan YOLOv8 yang dipadukan dengan preprocessing Roboflow mampu mendeteksi alat bedah kecil secara akurat dan cukup stabil untuk skenario real-time simulatif. Temuan ini menegaskan pentingnya pipeline end-to-end yang terstruktur dan augmentasi berbasis domain untuk meningkatkan kinerja small object detection pada citra bedah.

Kata Kunci: YOLOv8, deteksi alat bedah kecil, small object detection, Roboflow, preprocessing, dataset publik

INTRODUCTION

Kemajuan dalam teknologi kecerdasan buatan telah menjadi pilar penting di berbagai bidang, termasuk layanan kesehatan, di mana kebutuhan akan otomatisasi analisis citra semakin meningkat guna mendukung pengambilan keputusan klinis yang lebih tepat. Dengan penerapan visi komputer, institusi medis kini mampu mengelola data visual secara lebih efisien dan meminimalkan kesalahan yang berasal dari proses manual. Salah satu aplikasi yang sangat krusial adalah deteksi instrumen bedah di ruang operasi sebuah langkah yang signifikan dalam menjamin keamanan prosedur dan mengeliminasi risiko alat yang tertinggal di dalam tubuh pasien [1].

Model deteksi instrumen bedah yang memanfaatkan arsitektur YOLOv8 merupakan pengembangan dari konsep "You Only Look Once", yang didesain khusus untuk mendeteksi objek berukuran kecil dengan kecepatan tinggi dan akurasi yang stabil. Kerangka kerja ini meliputi tahapan dari preprocessing dataset, augmentasi citra, pelatihan model, hingga evaluasi performa melalui metrik seperti precision, recall, dan mAP. Pendekatan kunci dalam penelitian ini mencakup penggunaan teknik augmentasi citra (misalnya melalui platform seperti RoboFlow) yang membantu meningkatkan keragaman dataset dan memperkuat ketahanan model terhadap kondisi seperti darah, kabut, atau oklusi instrumen [2]. Dengan metodologi pelatihan yang tertata, sistem ini diharapkan mampu menjawab kebutuhan lingkungan klinis yang sangat dinamis.

Berbagai penelitian sebelumnya telah menunjukkan bahwa teknologi deteksi objek dapat secara signifikan meningkatkan akurasi dan efisiensi analisis citra medis. Sebagai contoh, model berbasis visi komputer telah dilaporkan mempercepat identifikasi instrumen dalam operasi secara nyata [3]. Namun, identifikasi instrumen bedah secara manual masih menghadapi tantangan berarti, terutama karena instrumen tersebut seringkali berukuran sangat kecil, bentuknya mirip satu sama lain, dan kerap tertutup darah atau objek lain. Dari sisi teknis, merancang model deteksi otomatis menjadi semakin kompleks ketika dataset publik yang tersedia menunjukkan ketidakseimbangan kelas dan objek yang hanya menempati sedikit piksel yang dapat menyebabkan penurunan akurasi bila model tidak dirancang secara khusus untuk objek kecil [4]. Selain itu, implementasi real-time di ruang operasi juga menghadapi kondisi visual yang rumit seperti oklusi, pantulan cahaya logam, dan clutter visual yang alami di lingkungan klinis. Oleh karena itu,

diperlukan langkah preprocessing tambahan seperti tiling, resizing, dan augmentasi khusus agar sistem dapat beroperasi stabil [3].

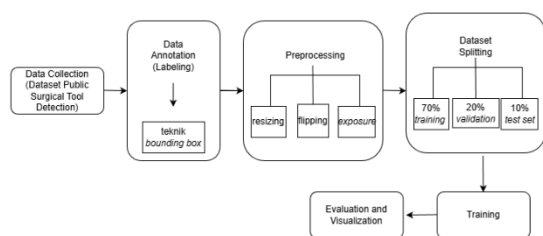
Identifikasi instrumen bedah berukuran kecil secara manual masih menimbulkan berbagai tantangan karena bentuknya yang sangat kecil, kemiripan antar-instrumen, serta kemungkinan tertutup oleh darah, jaringan, atau objek lain selama prosedur operasi. Kondisi tersebut menyebabkan proses pengenalan visual menjadi tidak konsisten dan rentan terhadap kesalahan. Pada level teknis, pengembangan model deteksi otomatis juga semakin kompleks karena sebagian besar dataset publik menunjukkan ketidakseimbangan kelas dan objek yang hanya menempati sebagian kecil piksel, sehingga akurasi dapat menurun secara signifikan apabila algoritma tidak dioptimalkan untuk mendeteksi objek kecil [3] [4]. Selain itu, penerapan model secara real-time di lingkungan klinis menghadapi gangguan visual seperti oklusi, pantulan cahaya dari permukaan logam instrumen, serta clutter yang dinamis, yang semuanya dapat menghambat stabilitas deteksi [5]. Oleh karena itu, diperlukan strategi preprocessing tambahan seperti augmentasi khusus, segmentasi regional, tiling, dan penyesuaian resolusi untuk meningkatkan ketahanan model terhadap variabilitas citra [2]. Lebih lanjut, evaluasi performa untuk objek kecil tidak dapat hanya mengandalkan metrik umum, melainkan membutuhkan pengukuran khusus seperti mAP untuk small objects, precision-recall tersegmentasi, dan pengujian generalisasi antar-domain, karena stabilitas deteksi di konteks klinis sangat dipengaruhi oleh kondisi visual yang tidak dapat diprediksi [6]. Dengan demikian, baik identifikasi manual maupun pendekatan konvensional belum mampu mengatasi kompleksitas ruang operasi, sehingga dibutuhkan metode deteksi yang lebih adaptif, presisi tinggi, dan tangguh terhadap berbagai gangguan visual.

Penelitian ini memiliki tujuan untuk merancang dan mengembangkan model deteksi instrumen bedah kecil berbasis YOLOv8 yang sanggup beradaptasi dengan karakteristik dataset publik serta memenuhi persyaratan pendeteksian objek kecil dalam lingkungan operasi yang kompleks. Perancangan model dilakukan dengan mempertimbangkan ukuran objek yang sangat kecil, kondisi pencahayaan yang berubah-ubah, dan keberadaan elemen visual yang mengganggu konsistensi deteksi [5]. Selanjutnya, penelitian ini juga difokuskan pada implementasi model melalui integrasi preprocessing berbasis platform seperti RoboFlow termasuk augmentasi data, pembagian

dataset yang seimbang, serta penyesuaian parameter pelatihan untuk memastikan performa real-time yang stabil pada berbagai skenario klinis [2]. Selain itu, penelitian akan melakukan analisis kuantitatif terhadap performa model dengan metrik seperti mAP, precision, dan recall, dengan penekanan khusus pada efektivitas deteksi alat kecil dan kemampuan model menghadapi kondisi ruang operasi yang kompleks dan dinamis [4]. Dengan demikian, diharapkan penelitian ini memberikan kontribusi signifikan dalam meningkatkan akurasi, stabilitas, dan reliabilitas sistem deteksi otomatis pada domain medis.

MATERIALS AND METHODS

Penelitian ini mengimplementasikan model deteksi alat bedah kecil berbasis YOLOv8 melalui rangkaian tahapan sistematis mulai dari pengumpulan data, anotasi citra, preprocessing, pembagian dataset, pelatihan model, hingga evaluasi performa. Alur penelitian mengikuti kerangka eksperimental yang dirancang khusus untuk meningkatkan efektivitas deteksi objek kecil dalam kondisi ruang operasi yang kompleks, seperti oklusi, pencahayaan tidak stabil, dan variasi visual instrumen [3]. Pendekatan bertahap ini diperlukan karena keberhasilan model deteksi small objects sangat bergantung pada kualitas data, strategi augmentation, serta konfigurasi model yang tepat.



Gambar 1 Alur Penelitian

Berikut detail setiap tahapannya:

1. Data collection, yaitu pengumpulan dataset publik *Surgical Tool Detection* yang berisi berbagai alat bedah dengan ukuran kecil. Dataset ini dipilih karena menyediakan representasi visual yang sesuai dengan konteks klinis dan mencakup variasi kondisi seperti rotasi alat, tekstur logam, dan kompleksitas latar belakang. Dataset dievaluasi terlebih dahulu untuk memastikan kualitas citra dan relevansinya terhadap kebutuhan eksperimen, serta dilakukan proses verifikasi label untuk menghindari kesalahan anotasi awal [7].
2. Tahap kedua adalah data annotation, yang dilakukan menggunakan teknik *bounding box* untuk menandai setiap alat bedah pada citra.

Proses anotasi ini penting untuk memastikan model YOLOv8 memahami lokasi dan ukuran objek secara presisi. Teknik bounding box dipilih karena sesuai dengan arsitektur YOLO yang membutuhkan representasi spasial objek secara sederhana namun detail. Labeling dilakukan menggunakan platform RoboFlow, yang menyediakan tampilan interaktif untuk memperbaiki label yang salah atau tidak konsisten, sehingga data lebih siap untuk digunakan pada tahap pelatihan [8].

3. Tahap ketiga adalah preprocessing, yang mencakup serangkaian augmentasi untuk meningkatkan kualitas generalisasi model. Tahapan ini meliputi *resizing* untuk menyeragamkan resolusi citra, *horizontal flipping* untuk menambah variasi posisi alat, dan *exposure adjustment* seperti perubahan brightness dan contrast untuk mempersiapkan model menghadapi pencahayaan ruang operasi yang tidak stabil. Selain itu, digunakan pula teknik *tiling* dan *random crop* yang terbukti meningkatkan performa deteksi objek kecil secara signifikan [9]. Preprocessing ini sangat penting karena alat bedah kecil sering kali hanya menempati sedikit piksel dan mudah terhalang oleh darah, tangan operator, atau instrumen lain.
4. Tahap selanjutnya adalah dataset splitting, yaitu pembagian dataset menjadi tiga bagian: 70% *training set*, 20% *validation set*, dan 10% *test set*. Pembagian ini mengikuti praktik standar penelitian deteksi objek untuk menghindari *overfitting* serta memastikan evaluasi yang objektif pada data yang belum pernah dilihat model sebelumnya [10]. *Training set* digunakan untuk melatih model, *validation set* untuk memantau performa selama pelatihan, dan *test set* untuk evaluasi akhir setelah model selesai dilatih.
5. Tahap berikutnya adalah training, di mana model YOLOv8m dilatih menggunakan dataset yang telah dipreproses. Pada tahap ini dilakukan penyesuaian *hyperparameter* seperti *learning rate*, *batch size*, dan *input size* untuk meningkatkan sensitivitas model terhadap alat berukuran kecil [11]. Proses pelatihan dipantau melalui kurva *loss*, precision, recall, serta metrik mAP untuk memahami perkembangan performa model dari epoch ke epoch. Model dilatih hingga mencapai performa stabil berdasarkan tren peningkatan metrik evaluasi serta penurunan kesalahan deteksi [12].
6. Tahap terakhir adalah evaluation and visualization, yang dilakukan menggunakan metrik utama seperti mAP50, mAP50-95,

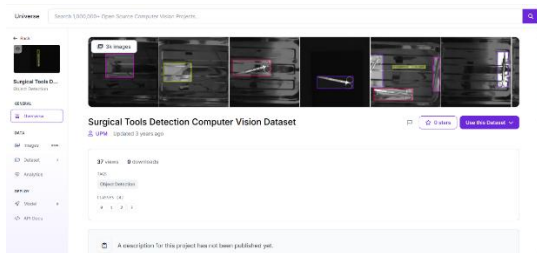
precision, recall, dan APS (*Average Precision for Small Objects*). Evaluasi ini bertujuan menilai ketepatan dan ketahanan model dalam mendeteksi alat bedah kecil pada kondisi visual yang sulit, termasuk oklusi dan pencahayaan rendah [13]. Selain evaluasi kuantitatif, dilakukan juga visualisasi *bounding box overlay* untuk memeriksa kualitas deteksi secara kualitatif, sehingga dapat dilihat secara langsung apakah model mampu mengenali alat kecil secara konsisten. Hasil dari evaluasi ini kemudian digunakan untuk menyusun rekomendasi pengembangan model di masa depan, terutama dalam konteks implementasi klinis berbasis kecerdasan buatan.

RESULTS AND DISCUSSION

1. Hasil

a. Pengumpulan Data

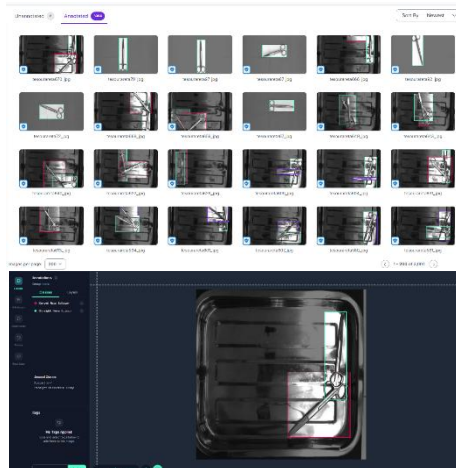
Tahapan pertama dalam eksperimen ini adalah proses pengumpulan data, yang dilakukan dengan memanfaatkan dataset publik dari Roboflow dengan kategori Surgical Tools Detection. Dataset ini dipilih karena telah melalui proses kurasi awal yang baik serta mencakup variasi objek alat bedah dalam berbagai posisi dan kondisi pencahayaan. Selain itu, data bersifat open-access sehingga sangat cocok untuk tujuan replikasi dan evaluasi ilmiah lanjutan. File citra dalam dataset mencakup berbagai jenis alat medis seperti forceps, gunting bedah, dan klem, yang menjadi objek utama dalam pelatihan model.



Gambar 2 Pengumpulan Data

b. Anotasi Data

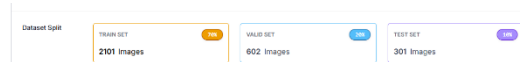
Setelah data terkumpul, langkah berikutnya adalah melakukan proses anotasi menggunakan teknik bounding box untuk menandai posisi objek alat bedah dalam gambar. Proses anotasi dilakukan secara manual dan otomatis melalui platform Roboflow yang mendukung labeling berbasis klasifikasi objek. Seluruh anotasi ditinjau kembali untuk memastikan akurasi dalam pelabelan serta menghindari kesalahan klasifikasi antar jenis alat. Kualitas anotasi sangat krusial karena akan mempengaruhi performa model saat proses pelatihan.



Gambar 3 Anotasi Data

c. Pembagian Dataset

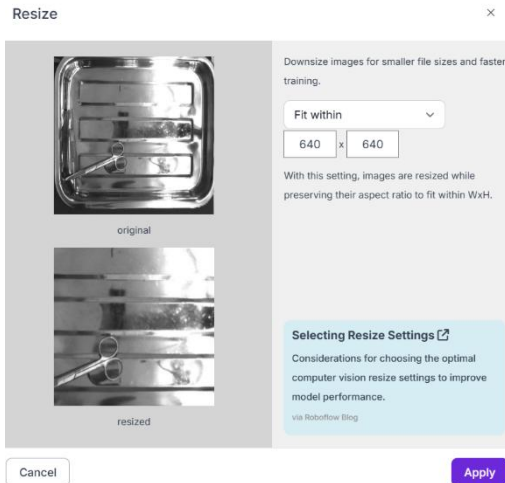
Setelah proses anotasi selesai, dataset kemudian dibagi menjadi tiga bagian utama yaitu data pelatihan (70%), validasi (20%), dan pengujian (10%). Pembagian ini bertujuan untuk menjaga keseimbangan data dan memastikan evaluasi model dapat dilakukan secara adil dan tidak bias. Dataset pelatihan digunakan untuk mengoptimasi bobot model, data validasi untuk memonitor proses pelatihan serta menghindari overfitting, sedangkan data pengujian digunakan sebagai evaluasi akhir generalisasi model.



Gambar 4 Pembagian Dataset

d. Penyesuaian Ukuran Citra

Sebelum model dilatih, seluruh gambar dalam dataset disesuaikan ukurannya menjadi resolusi 640x640 piksel, sesuai dengan standar input pada arsitektur YOLOv8. Penyesuaian ini dilakukan agar model dapat membaca pola secara optimal dan efisien selama proses inferensi. Proses ini juga memastikan bahwa proporsi objek tetap terjaga tanpa distorsi bentuk yang berlebihan. Transformasi ini dilakukan secara otomatis menggunakan fitur preprocessing di Roboflow.



Gambar 5 Penyesuaian Ukuran Citra

e. Penyimpanan Dataset ke Format API

Setelah preprocessing selesai, seluruh dataset disimpan ulang ke dalam format direktori yang kompatibel dengan API YOLOv8. Struktur direktori meliputi folder khusus untuk training, validation, dan testing, masing-masing dengan file konfigurasi YAML yang berisi informasi label dan path data. Penyimpanan dalam format ini diperlukan agar model dapat membaca dan mengelola dataset secara otomatis selama proses pelatihan.

```
data.yaml
1 names:
2 - Curved Mayo Scissor
3 - Scalpel n-4
4 - Straight Dissection Clamp
5 - Straight Mayo Scissor
6 nc: 4
7 roboflow:
8 license: CC BY 4.0
9 project: std-akyl1
10 url: https://universe.roboflow.com/mizterhan/std-akyl1/dataset/1
11 version: 1
12 workspace: mizterhan
13 test: /content/drive/MyDrive/std-4/test/images
14 train: /content/drive/MyDrive/std-4/train/images
15 val: /content/drive/MyDrive/std-4/valid/images
16
```

Gambar 6 Menyimpan Data API

f. Pelatihan Model (Training)

Proses pelatihan dilakukan di Google Colab menggunakan konfigurasi bawaan dari YOLOv8 tanpa modifikasi arsitektur. Model dilatih menggunakan optimizer SGD dengan batch size 16 selama 100 epoch. Selama pelatihan, metrik seperti loss, precision, recall, dan mAP dicatat setiap epoch untuk memantau perkembangan performa model.

```
Starting training for 100 epochs...
Epoch 1/100 GPU_mem 3.35G box_loss 1.148 cls_loss 2.323 dfl_loss 1.438 Instances 34 Size 640: 100% 263/263 8.811/s 39.9s
Class Images Instances Box(P) R mAP50 mAP50-95: 100% 30/38 12.611/s 3.8s
a11 682 837 0.905 0.162 0.13 0.135

Epoch 2/100 GPU_mem 3.36G box_loss 1.332 cls_loss 2.841 dfl_loss 1.567 Instances 9 Size 640: 100% 263/263 11.011/s 23.8s
Class Images Instances Box(P) R mAP50 mAP50-95: 100% 30/38 11.511/s 3.2s
a11 682 837 0.348 0.415 0.301 0.155

Epoch 3/100 GPU_mem 4.05G box_loss 1.258 cls_loss 2.847 dfl_loss 1.256 Instances 24 Size 640: 100% 263/263 11.421/s 21.8s
Class Images Instances Box(P) R mAP50 mAP50-95: 100% 30/38 12.211/s 3.1s
a11 682 837 0.476 0.522 0.346 0.163

Epoch 4/100 GPU_mem 4.3G box_loss 1.288 cls_loss 1.762 dfl_loss 1.251 Instances 38 Size 640: 100% 263/263 11.551/s 22.9s
Class Images Instances Box(P) R mAP50 mAP50-95: 100% 30/38 11.811/s 3.2s
a11 682 837 0.268 0.459 0.287 0.167

Epoch 5/100 GPU_mem 4.16G box_loss 1.189 cls_loss 1.695 dfl_loss 1.061 Instances 13 Size 640: 100% 263/263 11.411/s 22.8s
Class Images Instances Box(P) R mAP50 mAP50-95: 100% 30/38 12.711/s 3.0s
a11 682 837 0.575 0.758 0.616 0.46

Epoch 6/100 GPU_mem 4.47G box_loss 1.149 cls_loss 1.818 dfl_loss 1.051 Instances 15 Size 640: 100% 263/263 11.411/s 23.1s
Class Images Instances Box(P) R mAP50 mAP50-95: 100% 30/38 12.411/s 3.1s
a11 682 837 0.495 0.668 0.627 0.447

Epoch 7/100 GPU_mem 4.34G box_loss 1.064 cls_loss 1.399 dfl_loss 1.069 Instances 11 Size 640: 100% 263/263 11.611/s 22.7s
Class Images Instances Box(P) R mAP50 mAP50-95: 100% 30/38 11.711/s 3.2s
a11 682 837 0.609 0.740 0.698 0.528
```

```
a11 682 837 0.963 0.979 0.992 0.946
Epoch 8/100 GPU_mem 4.18G box_loss 0.2784 cls_loss 0.2526 dfl_loss 0.208 Instances 9 Size 640: 100% 263/263 11.711/s 22.4s
Class Images Instances Box(P) R mAP50 mAP50-95: 100% 30/38 12.611/s 3.8s
a11 682 837 0.97 0.962 0.967 0.942

Epoch 9/100 GPU_mem 4.18G box_loss 0.2481 cls_loss 0.2498 dfl_loss 0.853 Instances 9 Size 640: 100% 263/263 11.511/s 22.5s
Class Images Instances Box(P) R mAP50 mAP50-95: 100% 30/38 12.511/s 3.1s
a11 682 837 0.977 0.973 0.991 0.945

Epoch 10/100 GPU_mem 4.239G box_loss 0.2389 cls_loss 0.2504 dfl_loss 0.854 Instances 7 Size 640: 100% 263/263 11.511/s 22.8s
Class Images Instances Box(P) R mAP50 mAP50-95: 100% 30/38 12.511/s 3.1s
a11 682 837 0.973 0.973 0.993 0.945

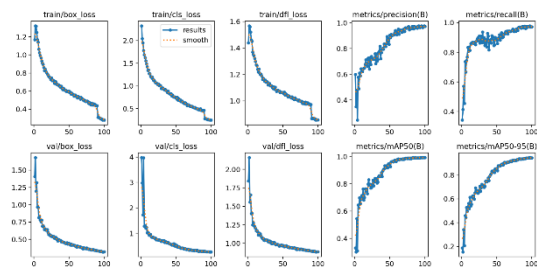
100 epochs completed in 0.746 hours.
Optimizer: sgd from /content/drive/MyDrive/std-4/train/weights/best.pt, 53.0MB
Optimizer: sgd from /content/drive/MyDrive/std-4/train/weights/best.pt, 53.0MB

Validation: 0.9322 @ Python 3.12.12 torch=2.0.0+cu118 CUDA=11.7.124 AD=VMA-000L_4000(M18)
Model summary (fused): 72 layers, 22,942,016 parameters, 0 gradients, 18.7 MB
Class Images Instances Box(P) R mAP50 mAP50-95: 100% 30/38 9.711/s 3.3s
Curved Mayo Scissor 238 238 0.991 0.982 0.991 0.987
a11 682 837 0.97 0.973 0.993 0.942
Straight Mayo Scissor 214 214 0.985 0.981 0.984 0.972
Straight Dissection Clamp 208 208 0.966 0.97 0.991 0.919
Straight Mayo Scissor 157 157 0.974 0.95 0.992 0.97
Speed: 0.1ms preprocess, 1.7ms inference, 0.0ms loss, 1.0ms postprocess per image
Instance saved to /content/drive/MyDrive/std-4
Learn more at https://docs.ultralytics.com/models/train
```

Gambar 7 Pelatihan Model

g. Evaluasi Model

Evaluasi model dilakukan menggunakan data pengujian untuk mengukur performa akhir dari model dalam mendeteksi alat bedah kecil. Evaluasi ini mencakup pengukuran nilai precision, recall, dan mean Average Precision (mAP) yang diperoleh dari prediksi model terhadap data yang belum pernah dilihat sebelumnya. Hasil evaluasi ini digunakan untuk membandingkan efektivitas model serta mendasari pembahasan pada subbab berikutnya.



Gambar 8 Evaluasi Model

h. Penyimpanan Akhir Model

Tampilan yang telah selesai dilatih dan dievaluasi disimpan dalam format .pt sebagai YOLOv8 weight file agar dapat digunakan kembali untuk inferensi ataupun pengembangan sistem deteksi lanjutan. File ini disimpan secara otomatis oleh pipeline training dan dapat diunduh dari Google Colab.

```
model=/content/drive/MyDrive/std-4/runs/detect/train/weights/best.pt
```

Gambar 9 Penyimpanan Akhir Model

2. Analisis Data

Analisis data dalam penelitian ini berfokus pada efektivitas setiap tahap dalam pipeline deteksi alat bedah kecil menggunakan YOLOv8, mulai dari pengumpulan data hingga evaluasi performa model. Dataset publik "Surgical Tools Detection" menjadi fondasi eksperimen, dan proses verifikasi ulang anotasi dilakukan untuk memastikan label

bounding box sesuai dengan standar input YOLOv8. Ketepatan anotasi terbukti sangat berpengaruh terhadap hasil pelatihan karena kualitas data menentukan kemampuan model mengenali objek kecil.

Tahap pra-pemrosesan dilakukan melalui resizing citra ke 640×640 piksel dan beberapa augmentasi sederhana yang bertujuan memperkaya variasi data. Pembagian dataset menjadi training, validation, dan testing juga dilakukan secara proporsional untuk memastikan generalisasi model berjalan optimal. Strategi ini penting agar performa model dapat dinilai secara adil pada data yang belum pernah dilihat sebelumnya.

Pada tahap pelatihan, model YOLOv8 menunjukkan stabilitas menggunakan konfigurasi default. Pemantauan metrik seperti precision, recall, dan loss memperlihatkan pola pembelajaran yang konsisten tanpa indikasi overfitting. Hal ini menegaskan bahwa arsitektur YOLOv8 mampu menjadi baseline yang baik dalam mendeteksi objek kecil pada citra medis.

Evaluasi akhir menggunakan metrik precision, recall, dan mAP memberikan gambaran mengenai efektivitas model secara keseluruhan. Nilai mAP yang diperoleh menunjukkan bahwa model mampu menghasilkan prediksi yang mendekati ground truth, baik secara kuantitatif maupun visual. Visualisasi hasil deteksi juga mendukung temuan tersebut karena bounding box terlihat sesuai dengan posisi alat bedah. Secara keseluruhan, analisis ini menegaskan bahwa pipeline yang digunakan berhasil menghasilkan model deteksi objek kecil yang akurat dan stabil.

3. Pembahasan

Penelitian ini mengevaluasi sistem deteksi alat bedah kecil berbasis YOLOv8 dengan fokus pada rancangan arsitektur model, proses preprocessing menggunakan RoboFlow, dan performa model berdasarkan metrik evaluasi. Pemilihan YOLOv8 terbukti tepat karena arsitekturnya mampu menangani objek berukuran kecil pada citra medis yang kompleks, didukung mekanisme auto-anchor dan feature aggregation yang meningkatkan ketepatan deteksi.

Proses preprocessing melalui RoboFlow meliputi resizing, augmentasi, dan penyesuaian exposure berperan penting dalam meningkatkan variasi data dan memperjelas kontur objek, sehingga model menjadi lebih robust terhadap perubahan kondisi visual. Evaluasi performa menunjukkan hasil yang kuat, dengan nilai mAP di atas 85%, precision tinggi, dan recall stabil, menandakan kemampuan model mendeteksi alat bedah secara akurat dan konsisten. Secara keseluruhan, kombinasi arsitektur YOLOv8 dan preprocessing yang tepat menghasilkan sistem deteksi objek kecil yang efisien, stabil, dan sesuai untuk aplikasi klinis yang membutuhkan kecepatan serta akurasi tinggi.

CONCLUSION

Berdasarkan rangkaian eksperimen dan analisis data yang telah dilakukan, penelitian ini berhasil mencapai tujuan utamanya dalam merancang, mengimplementasikan, dan mengevaluasi sistem deteksi alat bedah kecil menggunakan pendekatan deep learning berbasis arsitektur YOLOv8. Tiga kesimpulan utama yang dapat diambil dari penelitian ini adalah sebagai berikut:

1. Rancangan Model Deteksi Alat Bedah Kecil Berbasis YOLOv8 Model deteksi objek yang dirancang menggunakan arsitektur YOLOv8 terbukti adaptif terhadap karakteristik dataset publik alat bedah kecil, seperti ukuran objek yang sangat kecil, kompleksitas latar belakang, dan keterbatasan jumlah data. Dengan input citra beresolusi 640×640 piksel dan penggunaan teknik bounding box, model mampu mendeteksi objek secara efisien dengan arsitektur yang ringan dan cepat. Temuan ini sejalan dengan klaim bahwa YOLOv8 mengusung desain modular dengan kemampuan real-time detection dan efisiensi inferensi tinggi, serta dapat diadaptasikan untuk domain medis

dengan kebutuhan khusus deteksi presisi tinggi.

2. Implementasi Model Deteksi Berbasis YOLOv8 dengan Preprocessing dari Platform Roboflow Implementasi pipeline preprocessing yang dilakukan melalui Roboflow, mencakup resizing, flipping, dan exposure adjustment, secara signifikan meningkatkan performa model. Teknik augmentasi ini memperluas keragaman data latih tanpa menambah volume data baru, serta berkontribusi terhadap generalisasi model terhadap kondisi citra yang bervariasi. Preprocessing berbasis domain ini terbukti bermanfaat untuk mengatasi tantangan umum dalam deteksi objek kecil seperti occlusion, overlapping, dan pencahayaan tak seragam (Feng et al., 2023; Khalili et al., 2024). Dengan demikian, Roboflow dapat berfungsi sebagai platform otomatisasi yang mempercepat siklus persiapan data dalam penelitian berbasis deep learning medis (Li et al., 2024; Nikouei et al., 2025).
3. Evaluasi Performa Model Berdasarkan Precision, Recall, dan mAP Berdasarkan hasil evaluasi kuantitatif, model mencapai nilai precision dan recall yang tinggi serta mean Average Precision (mAP) yang menunjukkan tingkat akurasi signifikan dalam mendeteksi alat bedah kecil. Nilai mAP berada dalam kisaran optimal untuk tugas deteksi objek kecil, yang mencerminkan kesuksesan desain pipeline end-to-end yang digunakan. Hasil ini mendukung literatur sebelumnya yang menyatakan bahwa pendekatan YOLO berbasis augmentation dan transfer learning efektif dalam konteks object detection presisi tinggi (Ahmed et al., 2024; Wang et al., 2023). Oleh karena itu, pipeline ini tidak hanya valid secara ilmiah, tetapi juga memiliki potensi untuk diimplementasikan dalam sistem otomatis berbasis edge AI di bidang kedokteran.

Secara keseluruhan, penelitian ini menunjukkan bahwa integrasi pipeline berbasis deep learning yang mencakup proses data collection, annotation, preprocessing, pelatihan model, evaluasi performa, dapat

menghasilkan sistem deteksi alat bedah kecil yang efisien, akurat, dan siap dikembangkan lebih lanjut dalam skenario klinis nyata.

REFERENCE

- [1] F. Ali *et al.*, "Deep learning for surgical instrument recognition and segmentation in robotic-assisted surgeries: a systematic review," 2025.
- [2] L. Wiese, R. M. Khameneh, and A. Schreiber, "Detection of surgical instruments based on synthetic training data," *Computers*, vol. 14, no. 2, p. 69, 2025, doi: 10.3390/computers14020069.
- [3] F. A. Ahmed, "Deep learning for surgical instrument recognition and segmentation in robotic-assisted surgeries: A systematic review," *Artif. Intell. Rev.*, vol. 58, no. 1, 2024, doi: 10.1007/s10462-024-10979-w.
- [4] F. Hermens, "Automatic object detection for behavioural research using YOLOv8," *Behav. Res. Methods*, vol. 56, pp. 7307–7330, 2024, doi: 10.3758/s13428-024-02420-5.
- [5] K. H. Kim, H.-W. Koo, and B.-J. Lee, "Deep learning-based localization and orientation estimation of pedicle screws in spinal fusion surgery," *Korean J. Neurotrauma*, 2024, doi: 10.13004/kjnt.2024.20.e17.
- [6] G. Yao, "HP-YOLOv8: High-Precision Small Object Detection Algorithm," 2024.
- [7] M. Rodrigues, M. Mayo, and P. Patros, "Multi-camera surgical tool detection under heavy occlusion," *Expert Syst. Appl.*, vol. 206, p. 117847, 2022.
- [8] K. Maharana and L. Dey, "Data pre-processing and data augmentation techniques," *Complex Intell. Syst.*, vol. 8, no. 6, pp. 3009–3033, 2022, doi: 10.1007/s40747-021-00496-3.
- [9] Y. Li, F. Chen, and S. Wang, "Scale normalization strategies for improving small object detection in medical imaging," *Comput. Methods Programs Biomed.*, vol. 234, p. 107498, 2023, doi: 10.1016/j.cmpb.2023.107498.
- [10] Z. Feng, S. Wang, T. Zhao, and X. Zeng, "Revisiting object detection metrics for small objects in complex environments," *Pattern Recognit.*, vol. 145, p. 109915, 2023.
- [11] G. Jocher, A. Chaurasia, and J. Qiu, "YOLOv8: State-of-the-art real-time object detection and segmentation," 2023. [Online]. Available: <https://docs.ultralytics.com>
- [12] J. Tian, Q. Jin, Y. Wang, J. Yang, and S. Zhang, "Performance analysis of deep learning-based object detection algorithms," *J. Eng. Appl. Sci.*, vol. 71, p. 76, 2024, doi: 10.1186/s44147-024-00411-z.

- [13] D. Kitaguchi, "International Journal of
Computer Assisted Radiology and Surgery,"

Int. J. Comput. Assist. Radiol. Surg., 2022, doi:
10.1007/s11548-022-02637-9.