

ANALISIS KINERJA DETEKSI GERAKAN DAN PENGENALAN OBJEK PRODUK RITEL BERBASIS YOLOv8

PERFORMANCE ANALYSIS OF MOTION DETECTION AND OBJECT RECOGNITION FOR RETAIL PRODUCT BASED ON YOLOV8

Luthfia Azzahra Kusumawardhani¹, Dara Kusumawati Ramadani Yasir²,
Ida Bagus Indrabudhi Kusuma³, Suryo Adhi Wibowo^{4*}, Rissa Rahmania⁵, Asep Rahmat Hidayat⁶

^{1,2,4}School of Electrical Engineering, Telkom University, Bandung, Indonesia

³School of Computing, Telkom University, Bandung, Indonesia

^{3,4}Center of Excellence Artificial Intelligence for Learning and Optimization, Telkom University, Bandung, Indonesia

⁵Computer Science Department, School of Computer Science, Bina Nusantara University, Bandung Campus, Jakarta 11480

⁶National Research and Innovation Agency, BRIN, Indonesia

¹luthfiaazzahra@student.telkomuniversity.ac.id, ²daraavsr@student.telkomuniversity.ac.id,
³indrabudhikusuma@student.telkomuniversity.ac.id, ⁴suryoadhiwibowo@telkomuniversity.ac.id,
⁵rissa.rahmania@binus.ac.id

Abstrak

Teknologi yang saat ini paling umum digunakan oleh industri ritel untuk mengidentifikasi produk adalah *barcode*. Karena keterbatasan *barcode*, QR (*quick response*) *code* lalu diusulkan. Namun, tantangan dari QR *code* adalah tidak semua produk memiliki QR *code* untuk dipindai pada mesin transaksi. Banyaknya variasi kode kemudian memicu penelitian untuk penggunaan teknologi visi komputer untuk mengenali sebuah produk. Berbagai teknologi *deep learning* telah diterapkan untuk mengenali produk, diantaranya adalah *Faster R-CNN*, *Mask R-CNN*, FCIS, *RetinaNet*, dan YOLO. Teknologi YOLO pada penelitian sebelumnya menggunakan versi YOLOv2 dan mampu mengenali produk pada *dataset* VOC 2012 dengan nilai mAP sebesar 78,2%. Penelitian ini bertujuan untuk mengimplementasikan dan menganalisis YOLO versi terbaru yaitu YOLOv8 untuk mengenali dan mendeteksi arah gerak produk ritel. Data yang digunakan terdiri dari 987 gambar dari 10 produk. Hasil pengujian pada proses pengenalan produk secara umum diperoleh nilai mAP₅₀ sebesar 98% dan mampu mendeteksi arah gerak produk dengan baik. Berdasarkan hasil pengujian tersebut dapat disimpulkan bahwa penggunaan YOLOv8 secara signifikan dapat mendeteksi arah dan mengenali produk ritel dengan baik.

Katakunci: Deteksi Gerakan, Pengenalan Produk Ritel, YOLOv8

Abstract

The most common technology used in retail industries to identify retail products is barcode recognition. Due to barcode recognition limitations, QR (quick response code) is proposed. However, the challenge of QR codes is that not all products have QR codes to be scanned on transaction machines. The large number of code variations then triggered research for the use of computer vision to recognize a product. Various deep learning approaches have been applied to recognize products, including *Faster R-CNN*, *Mask R-CNN*, FCIS, *RetinaNet*, and YOLO. YOLO in previous research used the YOLOv2 and was able to recognize products on the 2012 VOC dataset with mAP score of 78.2%. This research aims to implement and analyze the latest version of YOLO, YOLOv8, to recognize and detect the direction of retail products motion. The data used consists of 987 images of 10 products. The results on the product recognition generally obtained mAP₅₀ score of 98% and was able to detect the direction of product's motion properly. Based on the test results, it can be inferred that the use of YOLOv8 can significantly detect the direction and recognize retail products well.

Keywords: Motion Detection, Retail Product Recognition, YOLOv8.

1. PENDAHULUAN

Saat ini, pengenalan *barcode* menjadi teknologi yang paling umum digunakan oleh industri ritel untuk mengidentifikasi suatu produk. *Barcode* merupakan representasi dari sebuah data yang ditampilkan dalam bentuk satu dimensi yang berupa beberapa garis dengan lebar tertentu yang disusun dengan spasi secara paralel. Umumnya, tiap produk ritel memiliki *barcode* yang tertera di kemasan produk tersebut [1]. Pemindaian *barcode* juga umum digunakan untuk transaksi antara kasir dengan konsumen di supermarket atau industri ritel lainnya. Seiring dengan bertambahnya jumlah produk yang beredar di masyarakat dan keterbatasan kapasitas dari *barcode* dalam menyimpan data, maka *quick response (QR) code* diusulkan.

QR code banyak digunakan salah satunya untuk transaksi dengan mudah dan cepat. Dengan banyaknya pengguna *smartphone* saat ini yang didukung dengan pemindai *QR code*, akses terhadap informasi dan transaksi semakin cepat. Tantangan yang dihadapi oleh industri ritel saat ini seperti supermarket yaitu tidak semua produk memiliki *QR code* sehingga metode untuk mengidentifikasi produk menjadi bervariasi dan dapat membutuhkan waktu yang lama untuk melakukan transaksi saat konsumen berada di kasir. Sementara itu, menurut survei [2] algoritma kecerdasan buatan yang berkaitan dengan deteksi objek dan pengenalan citra pada penelitian sebelumnya telah sampai pada penggunaan *deep learning* seperti YOLOv2 (*You Look Only Once v2*) [3-5], FCIS [6], *Mask R-CNN* [7], *Faster R-CNN* [8], dan *RetinaNet* [9]. Pendekatan *Faster R-CNN*, *Mask R-CNN*, FCIS, dan *RetinaNet* menggunakan *dataset* D2S dalam penelitiannya. *Dataset* D2S [10] adalah *dataset* pertama yang menjadi *benchmark* yang menyediakan anotasi piksel pada tingkat *instance* dengan tujuan untuk mengakomodasi setiap aplikasi dari sistem pembayaran otomatis, inventaris, atau gudang. *Dataset* ini terdiri atas 21,000 gambar dengan resolusi tinggi yang terdiri atas 60 kategori produk. Dengan *dataset* ini diperoleh nilai mAP₅₀ yang tertinggi diantara pendekatan tersebut yaitu 90.3% pada *Faster R-CNN*. Sedangkan pada YOLO yang masih menggunakan versi 2, memperoleh nilai mAP sebesar 78,2% pada *dataset* VOC 2012 yang memiliki 20 kategori produk [10].

Lebih lanjut terkait pengembangan YOLO yang telah sampai pada YOLOv8 dalam dua tahun terakhir [11], YOLOv6 yang dirilis pada 2022 memiliki beberapa kelebihan salah satunya memiliki strategi penyulingan sendiri untuk tugas regresi dan klasifikasi. Pada tahun yang sama, YOLOv7 juga dirilis dengan perubahan arsitektur didalamnya, yaitu memiliki *extended efficient layer aggregation network* (E-ELAN) dan penskalaan model untuk model berbasis penggabungan. Pembaruan pada arsitektur YOLOv7 ini mampu meningkatkan akurasi tanpa mempengaruhi kecepatan inferensi. Sedangkan pada YOLOv8 yang dirilis pada 2023 menggunakan *backbone* yang sama dengan YOLOv5 dengan menambah pembaruan pada CSPLayer yang nantinya disebut sebagai modul C2f, memiliki model segmentasi yang dinamakan YOLOv8-Seg model, dan menggunakan CIoU dan *loss functions* dari DFL untuk *bounding box loss* dan *binary cross-entropy* untuk *classification loss*. Pembaruan tersebut meningkatkan performa pendeteksian objek YOLOv8, khususnya apabila berurusan dengan objek yang lebih kecil. Model segmentasi YOLOv8 juga telah mencapai hasil yang mutakhir pada berbagai *benchmark* deteksi objek dan segmentasi semantik dengan tetap mempertahankan kecepatan dan efisiensi yang tinggi.

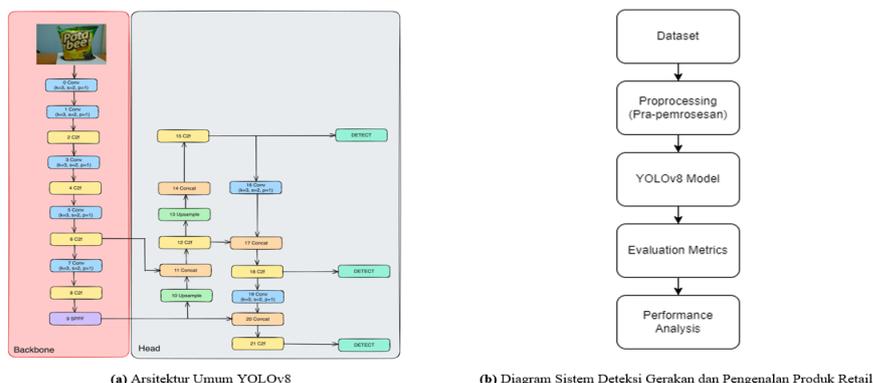
Oleh karena itu, untuk meningkatkan keakuratan pengenalan produk ritel, diperlukan teknologi visi komputer yang lebih baik. Kontribusi dari penelitian ini bertujuan untuk merancang dan membangun model kecerdasan buatan berbasis YOLOv8 yang mampu mengenali produk retail dan mendeteksi arah gerak dari produk yang dimasukkan ke dalam troli belanja konsumen. Lebih lanjut, struktur dari penulisan paper ini sebagai berikut: pada Bagian 2 membahas terkait

metodologi penelitian. Bagian 3 membahas terkait hasil dari pengujian terhadap *dataset* yang telah dibangun. Bagian 4 membahas terkait kesimpulan dari penelitian ini.

2. PERANCANGAN SISTEM

2.1 Arsitektur YOLOv8

Arsitektur YOLOv8 secara umum adalah versi terbaru dari serangkaian model YOLO, yang dikenal karena kemampuannya dalam deteksi objek secara *real-time* seperti yang terlihat pada Gambar 1 (a). YOLOv8 memperkenalkan beberapa perubahan dan peningkatan dari versi sebelumnya untuk meningkatkan kinerja dan keakuratannya. Arsitektur YOLOv8 terdiri dari beberapa komponen utama, termasuk *Convolutional Neural Network (CNN)*, *Grid Cell Division*, *Bounding Box Prediction*, dan *Non-max Suppression*. YOLOv8 juga dapat memiliki penyesuaian arsitektur tertentu, seperti penambahan atau penghapusan lapisan, peningkatan resolusi input, atau optimasi lainnya untuk meningkatkan kinerja dan akurasi deteksi objek. Dengan arsitektur yang dioptimalkan dan fitur-fitur baru, YOLOv8 menawarkan solusi yang kuat dan efisien untuk deteksi objek dalam berbagai aplikasi, termasuk di lingkungan ritel, yang memungkinkan deteksi objek secara *real-time* dengan tingkat akurasi yang tinggi [12].



Gambar 1. Arsitektur YOLOv8 (a) dan Diagram Sistem (b)

2.2 Diagram Sistem

Sistem yang diusulkan untuk analisis kinerja deteksi gerakan dan pengenalan gambar produk ritel berbasis YOLOv8 terdiri dari beberapa komponen utama. Seperti pada Gambar 1 (b) dapat dilihat komponen pertama, *dataset* digunakan sebagai basis untuk pelatihan dan evaluasi model. Data ini kemudian melalui tahap pra-pemrosesan, di mana gambar-gambar dari *dataset* disiapkan dengan melakukan operasi seperti segmentasi, pengurangan *noise*, dan augmentasi data. Setelah itu, model YOLOv8 diterapkan untuk melakukan deteksi gerakan dan pengenalan gambar pada produk ritel. Evaluasi kinerja model kemudian dilakukan menggunakan berbagai metrik kinerja seperti *Precision*, *Recall*, dan *mean Average Precision (mAP)*. Hasil evaluasi tersebut kemudian dianalisis secara menyeluruh untuk mengidentifikasi kekuatan dan kelemahan model dalam tugas deteksi dan pengenalan. Selain itu, analisis kinerja juga dapat melibatkan perbandingan dengan metode lain atau versi model YOLO sebelumnya. Melalui rangkaian komponen ini, sistem dapat memberikan pemahaman yang mendalam tentang kinerja deteksi gerakan dan pengenalan gambar untuk produk ritel berbasis YOLOv8.

2.3 Dataset

Implementasi YOLO (*You Only Look Once*) v8 dalam deteksi objek melibatkan beberapa langkah. Pertama, model YOLO v8 harus dilatih menggunakan *dataset* yang sesuai dengan hasil yang diinginkan. Pada tabel 1 dijelaskan bahwa *dataset* yang digunakan terbagi menjadi tiga bagian berdasarkan pelatihan (*train*), validasi (*valid*), dan pengujian (*test*). *Training dataset* ini berfungsi untuk melatih model mesin. Dilakukan tiga kali pelatihan model untuk mengamati hasil akurasi dari masing-masing model. Setelah pelatihan selesai, model-model tersebut dievaluasi menggunakan *dataset* validasi atau uji untuk membandingkan akurasi. Berikut pada tabel 1 merupakan *dataset* untuk jumlah citra dan label dalam penelitian ini:

Tabel 1. *Dataset*

Bagian	Jumlah Citra	Jumlah Label / Anotasi
<i>Train</i>	2094	698
<i>Valid</i>	193	193
<i>Test</i>	96	96

Dataset terbagi menjadi *subset* utama yang memiliki peran atau fungsi masing, yang pertama *dataset* bagian *train* yang berisi sebagian data yang digunakan untuk melatih model untuk menyesuaikan parameter agar sesuai dengan pola yang terdapat dalam *dataset*. Bagian *dataset valid* berisi *dataset* yang sudah di ketahui labelnya tetapi tidak digunakan selama proses pelatihan. *Dataset valid* digunakan untuk mengukur kinerja model pada setiap iterasi pelatihan untuk memonitor kemajuan pelatihan dan mencegah *overfitting* atau *underfitting*. Bagian *dataset test* berisi data yang tidak pernah dilihat oleh model selama proses pelatihan atau validasi. Data ini digunakan untuk mengevaluasi kinerja model secara obyektif pada data baru yang belum pernah dilihat sebelumnya. Berikut pada tabel 2 terdapat daftar *dataset* yang dilengkapi dengan kode produk dan nama produk dari *dataset*.

Dataset yang digunakan pada penelitian ini terdiri dari produk retail yang dibagi menjadi 10 kelas berbeda. Setiap produk memiliki label kelas yang didasarkan pada kode produk yang tercetak pada setiap kemasannya. Label kelas untuk setiap produk didasarkan pada kode produk yang tercetak pada kemasannya, memungkinkan identifikasi dan klasifikasi yang akurat. Pada tabel 2 merupakan *dataset* yang mencakup nama produk beserta kode produk yang terkait:

Tabel 2. Daftar *Dataset*

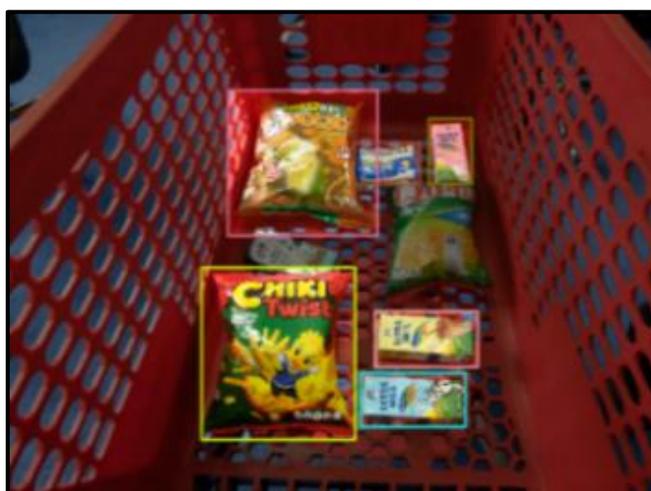
No	Kode Produk	Nama Produk
1	4902430563864	Pantene Sachet Shampoo Anti Ketombe
2	89686723021	Chiki Twist Jagung Bakar
3	8993200666836	Cimory FM Almond
4	8996006853387	Fruit Tea Apel
5	8997032680510	Taronet 3D Potato Jungle Chicken
6	8998009010552	Ultra Milk Vanilla
7	8998009010569	Ultra Milk Chocolate
8	8998009010576	Ultra Milk Strawberry
9	8998866201490	Potabee Nori
10	8999999048167	Sunsilk Shampoo Sachet Black Shine

2.4 Metrik Evaluasi Kinerja

Beberapa komponen yang digunakan dalam deteksi objek menggunakan YOLO V8 yaitu ada *Bounding Box Prediction*, *Class Prediction*, dan *Loss Prediction*.

2.4.1 *Bounding Box Prediction*

Setiap *grid cell* dalam gambar input memprediksi beberapa *bounding boxes* (kotak pembatas) yang mungkin mengandung objek. Setiap *bounding box* diprediksi dengan menggunakan koordinat relatif terhadap *grid cell* tempat *bounding box* itu berada, serta skor kepercayaan (*confidence score*) bahwa *bounding box* tersebut berisi objek yang sesungguhnya. Seperti pada gambar 2 *bounding box* berada pada objek yang sesuai berisi produk yang terdeteksi.



Gambar 2. *Bounding Box*

2.4.2 *Class Prediction*

Selain prediksi *bounding box*, setiap *grid cell* juga memprediksi probabilitas kelas objek yang mungkin ada di dalam *bounding box*. Biasanya menggunakan fungsi *softmax* untuk memperkirakan probabilitas kelas. Mengacu pada kemampuan model untuk memprediksi kelas objek yang mungkin hadir dalam sebuah *bounding box* yang telah dideteksi. Setiap *grid cell* dalam gambar *input* memprediksi probabilitas untuk berbagai kelas objek yang mungkin ada dalam *bounding box* yang diprediksi oleh *grid cell* tersebut.

2.4.3 *Loss Function*

YOLOv8 menggunakan *loss function* gabungan yang menggabungkan kesalahan prediksi *bounding box* (seperti *Intersection over Union*), kesalahan prediksi kelas objek, dan skor kepercayaan (*confidence score*) dari *bounding box*. *Loss function* ini digunakan untuk melatih model agar dapat menghasilkan prediksi yang lebih akurat.

2.5 Parameter

Akurasi (*Accuracy*): Akurasi mengukur seberapa baik model YOLOv8 dapat mengidentifikasi objek dengan benar dalam gambar. Ini dihitung dengan membandingkan jumlah objek yang terdeteksi dengan benar dengan jumlah total objek yang sebenarnya dalam *dataset* uji. Persamaan umum untuk menghitung akurasi yaitu (1):

$$\text{Akurasi} = \frac{\text{Jumlah prediksi benar}}{\text{Total jumlah data}} \quad (1)$$

Presisi (*Precision*): Presisi mengukur seberapa banyak dari objek yang diprediksi sebagai positif oleh model yang benar-benar positif. Dalam deteksi objek, presisi mengacu pada berapa banyak dari *bounding box* yang terdeteksi oleh model yang benar-benar mengandung objek tersebut. Perhitungan presisi dalam konteks deteksi objek memiliki persamaan (2):

$$\text{Presisi} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}} \quad (2)$$

True Positives yaitu jumlah objek yang diprediksi dengan benar oleh model sebagai positif sedangkan *False Positives* yaitu jumlah objek yang diprediksi oleh model sebagai positif tetapi sebenarnya tidak ada.

Recall: *Recall* (sensitivitas) mengukur seberapa banyak dari objek yang benar-benar positif dalam *dataset* uji yang berhasil terdeteksi oleh model. Dalam konteks deteksi objek, ini berarti seberapa banyak dari seluruh objek yang ada yang berhasil diidentifikasi oleh model. Perhitungan *recall* dalam konteks deteksi objek memiliki persamaan (3):

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}} \quad (3)$$

True Positives adalah jumlah objek yang di prediksi dengan benar sedangkan *False Negative* adalah jumlah objek yang tidak terdeteksi oleh model, yang sebenarnya objek tersebut ada.

3.HASIL DAN PENGUJIAN SISTEM

Pada jurnal ini membahas implementasi dan hasil eksperimen terkait mengenai pengenalan objek, khususnya dalam konteks penggunaan model YOLOv8 untuk mendeteksi objek pada produk retail. Dalam penelitian ini, sebanyak 987 gambar produk digunakan, dengan 698 gambar untuk pelatihan dan 96 gambar untuk pengujian setelah augmentasi. Model *deep learning* yang digunakan adalah *Convolutional Neural Network* (CNN). YOLOv8 digunakan untuk melakukan pemrosesan citra *end-to-end*, memberikan prediksi dalam bentuk kotak pembatas (*bounding box*) yang mengidentifikasi kelas objek dalam citra. Selain itu, eksplorasi juga dilakukan terhadap kombinasi dengan *motion detection* untuk meningkatkan kemampuan deteksi objek dalam konteks pergerakan objek di lingkungan retail.

3.1 Hasil Pelatihan Dataset

Pada tahap pelatihan, proses berlangsung selama 54.42 menit dengan *dataset* pelatihan terdiri dari 987 gambar produk. Model dilatih selama 100 *epoch* dengan menggunakan gambar berukuran 640x640 piksel sebagai *input*. Untuk mencegah *overfitting*, digunakan parameter *early stopping* [13] yang akan menghentikan pelatihan jika tidak ada peningkatan pada kinerja model pada data validasi selama 25 *epoch* berturut-turut. Proses ini dilakukan dengan menggunakan 10 gambar produk secara berulang-ulang. Hasil dari tahap pelatihan adalah sebuah model yang disimpan dalam file *.weights* yang akan digunakan untuk langkah deteksi objek selanjutnya. Tingkat keakuratan dari deteksi objek bervariasi untuk setiap produk. Berikut merupakan *capture* dari proses pelatihan:

```
100 epochs completed in 0.907 hours.
Optimizer stripped from C:\Users\SMART CART\smart_pisun\ultralitics\smart_pisun\ultralitics\runs\detect\train13\weights\last.pt, 22.5MB
Optimizer stripped from C:\Users\SMART CART\smart_pisun\ultralitics\smart_pisun\ultralitics\runs\detect\train13\weights\best.pt, 22.5MB

Validating C:\Users\SMART CART\smart_pisun\ultralitics\smart_pisun\ultralitics\runs\detect\train13\weights\best.pt...
Ultralytics YOLOv8.1.22 Python-3.10.9 torch-2.1.1+cu118 CUDA:0 (NVIDIA GeForce RTX 2060, 6144MiB)
Model summary (fused): 168 layers, 11129454 parameters, 0 gradients

```

Class	Images	Instances	Box(P)	R	mAP50	mAP50-95)
all	193	272	0.986	0.982	0.987	0.846
4902430563864	193	36	0.998	0.944	0.965	0.868
89586723021	193	28	0.983	1	0.995	0.854
8993200666836	193	23	0.985	1	0.995	0.894
8996006853387	193	22	0.984	1	0.995	0.774
8997832680510	193	25	0.974	1	0.995	0.9
8998009810552	193	26	0.961	0.944	0.969	0.78
8998009810569	193	31	0.988	0.968	0.976	0.837
8998009810576	193	30	1	0.986	0.995	0.852
8998866201490	193	23	0.985	1	0.995	0.842
8999999848167	193	28	1	0.98	0.995	0.863

```
Speed: 0.4ms preprocess, 4.2ms Inference, 0.8ms loss, 1.0ms postprocess per image
Results saved to C:\Users\SMART CART\smart_pisun\ultralitics\smart_pisun\ultralitics\runs\detect\train13
Ultralytics YOLOv8.1.22 Python-3.10.9 torch-2.1.1+cu118 CUDA:0 (NVIDIA GeForce RTX 2060, 6144MiB)
Model summary (fused): 168 layers, 11129454 parameters, 0 gradients
val: Scanning C:\Users\SMART CART\smart_pisun\ultralitics\smart_pisun\ultralitics\runs\detect\train13\weights\best.pt...

```

Class	Images	Instances	Box(P)	R	mAP50	mAP50-95)
all	193	272	0.986	0.982	0.987	0.846
4902430563864	193	36	0.998	0.944	0.965	0.868
89586723021	193	28	0.983	1	0.995	0.854
8993200666836	193	23	0.985	1	0.995	0.894
8996006853387	193	22	0.984	1	0.995	0.774
8997832680510	193	25	0.974	1	0.995	0.9
8998009810552	193	26	0.961	0.944	0.969	0.78
8998009810569	193	31	0.988	0.968	0.976	0.837
8998009810576	193	30	1	0.986	0.995	0.852
8998866201490	193	23	0.985	1	0.995	0.842
8999999848167	193	28	1	0.98	0.995	0.863

```
Speed: 0.7ms preprocess, 11.0ms Inference, 0.0ms loss, 1.1ms postprocess per image
Results saved to C:\Users\SMART CART\smart_pisun\ultralitics\smart_pisun\ultralitics\runs\detect\train132
```

Gambar 3. Capture Proses Pelatihan Dataset

Dari gambar 3, diperoleh nilai *Mean Average Precision* (mAP) dari semua produk mencapai 98,7%. Hal ini menunjukkan tingkat akurasi yang sangat tinggi dalam deteksi objek untuk seluruh kelas produk yang ada. Tingkat akurasi yang tinggi seperti ini sangat memperkuat kepercayaan terhadap kinerja dan kemampuan model dalam mengidentifikasi objek dengan tepat.

3.2 Hasil Pengujian Dataset

Pada proses pelatihan *dataset*, jumlah citra yang digunakan pada proses pelatihan yaitu 2094 citra dan untuk proses pengujian menggunakan 96 citra. Setelah proses pelatihan pada *dataset* selesai, hasil file *.weights* dari tahap tersebut digunakan untuk menguji *dataset*. Dalam tahap pengujian ini, sistem diterapkan dengan memanfaatkan kamera untuk mengarahkan produk yang akan dideteksi ke perangkat. Dengan menggunakan model yang telah dilatih, sistem kemudian secara otomatis menganalisis *input* visual dari kamera untuk mengidentifikasi dan menempatkan kotak pembatas pada produk yang muncul dalam gambar. Berikut merupakan hasil dari tahap pengujian *dataset* yang telah diujikan:

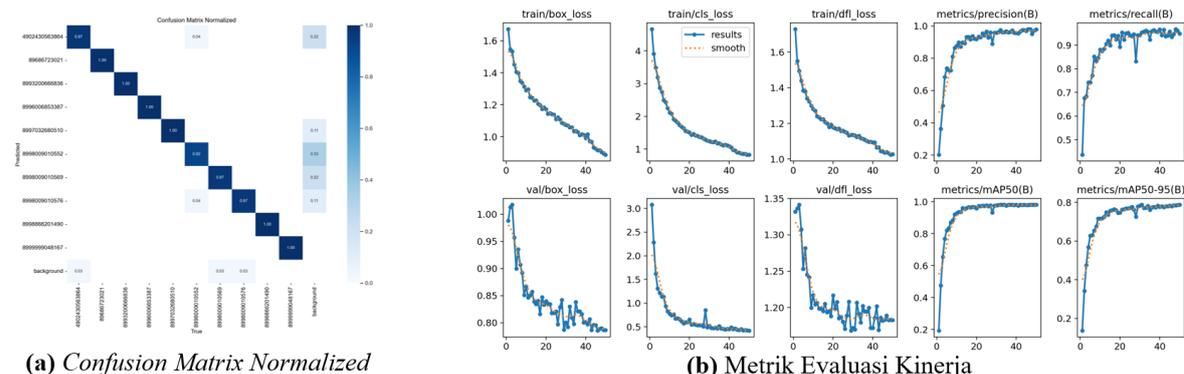


Gambar 4. Hasil Pengujian YOLOv8

Hasil pengujian pada gambar 4 menggunakan YOLOv8 menunjukkan bahwa semua barang yang terdeteksi sesuai dengan kode produk yang terdaftar dalam *dataset*. Hal ini menunjukkan tingkat keakuratan yang baik dalam identifikasi produk menggunakan model yang telah dikembangkan.

3.3 Hasil Parameter

Hasil dari tahap pelatihan menunjukkan adanya *confusion matrix* yang telah dinormalisasi dan hasil dari metrik evaluasi kinerja seperti yang terlihat pada gambar 5. *Confusion matrix normalized* memberikan gambaran yang lebih rinci tentang seberapa baik model dapat mengklasifikasikan setiap kelas produk. Dengan melihat *confusion matrix normalized*, kita dapat mengevaluasi proporsi prediksi yang benar dan salah untuk setiap kelas secara lebih terperinci, yang membantu dalam mengidentifikasi area di mana model dapat ditingkatkan.



Gambar 5. Hasil Pelatihan *Dataset*

Selain itu, hasil akurasi dari model juga dapat dilihat dari *confusion matrix normalized* pada tabel 3. Secara umum, nilai akurasi yang tinggi menunjukkan bahwa model memiliki kemampuan yang baik dalam mengklasifikasikan berbagai kelas produk. Meskipun demikian, adanya beberapa kebingungan antara kelas produk menandakan bahwa terdapat ruang untuk peningkatan kinerja model, terutama dalam mengidentifikasi kelas-kelas tersebut dengan lebih tepat.

Tabel 3. Nilai *Confusion Matrix Normalized*

No	Kode Produk	Akurasi
1	4902430563864	0.94
2	89686723021	1.00
3	8993200666836	1.00
4	8996006853387	1.00
5	8997032680510	1.00
6	8998009010552	0.96
7	8998009010569	0.97
8	8998009010576	1.00
9	8998866201490	1.00
10	8999999048167	1.00

Dari hasil metrik evaluasi kinerja pada gambar 5(b) terlihat bahwa model menunjukkan penurunan *train loss* seiring waktu, sementara *validation loss* mengalami peningkatan. Selain itu, grafik *metrics* pada data validasi juga menunjukkan tren yang menurun atau stagnan setelah beberapa *epoch*. Pada proses pelatihan pada *epoch* ke-100, mAP terbaik diperoleh untuk seluruh kelas mencapai 98,7%. Namun, penurunan kinerja pada data validasi dapat mengindikasikan bahwa model tidak mampu melakukan generalisasi dengan baik pada data baru atau data yang belum pernah dilihat sebelumnya. Hal ini dapat disebabkan oleh model terlalu memfokuskan detail-detail

kecil atau *noise* dalam data pelatihan. Pada gambar 6 terdapat produk Cimory Almond tidak terdeteksi ke dalam kelas tertentu.



Gambar 6. Model tidak dapat mengenali produk Cimory Almond

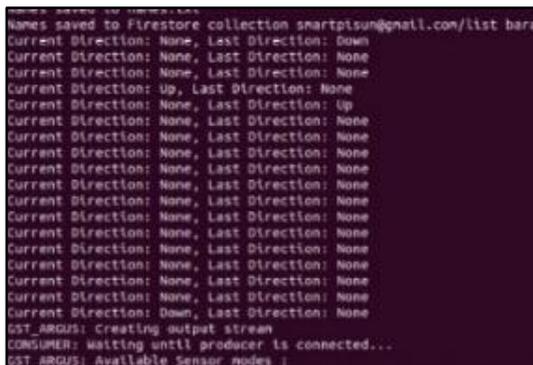
3.4 Kombinasi *Motion Detection* dengan *Object Recognition*

Motion Detection pada penelitian ini untuk memastikan kehandalan dan responabilitas sistem dalam mendeteksi pergerakan. Pengujian ini dimulai dengan simulasi pergerakan pada lingkungan kontrol, di mana sensitivitas dan keakuratan dari deteksi gerakan dievaluasi dengan memanfaatkan berbagai skenario dan intensitas gerakan yang mungkin akan terjadi. Selain itu, uji coba dilakukan dalam berbagai kondisi pencahayaan pada tabel 4 untuk memastikan bahwa sistem dapat beroperasi optimal pada situasi nyata yang dapat bervariasi.

Tabel 4. Kondisi Pencahayaan

No	Tingkat Pencahayaan	Intensitas Pencahayaan (Lux)	Waktu Deteksi Rata-Rata	Jumlah Deteksi yang Tepat	Jumlah Total Deteksi	Akurasi
1	Terang	>1000	4,9 detik	19	20	0,95
2	Sedang	100 – 1000	5,3 detik	11	20	0,55
3	Redup	<100	5,5 detik	7	20	0,35

Dari gambar 7 terlihat bahwa *Motion Detection* berhasil menangkap gerakan yang terjadi di kamera dengan efektif pada Tingkat pencahayaan yang terang. Ketika produk muncul di bidang pandang kamera, gerakan akan terdeteksi, baik dari atas ke bawah maupun sebaliknya, sesuai dengan arah gerakan produk tersebut. Di sisi lain, jika tidak ada barang yang terdeteksi, *motion detection* akan memberikan output “none”. Selanjutnya, ketika ada produk yang terdeteksi di kamera, sistem akan menampilkan jumlah produk yang terdeteksi beserta kode produknya sesuai dengan *dataset* yang telah terdaftar. Berikut uji coba yang telah dilakukan pada perangkat kamera untuk *motion detection*:



Gambar 7. Uji coba *motion detection*

Lebih lanjut, penelitian ini masih terbuka untuk penelitian kedepannya, diantaranya terkait tantangan dalam mendeteksi barang yang tertumpuk dalam keranjang belanja, masalah keamanan sistem belanja apabila barang yang dimasukkan tidak sesuai jumlahnya dengan yang dideteksi oleh sistem, dan terkait kebutuhan sosial masyarakat ketika bertransaksi yang membutuhkan interaksi, untuk itu pengembangan sistem berikutnya disarankan dapat memiliki sebuah agent untuk berinteraksi dengan konsumen dalam bertransaksi.

4.KESIMPULAN

Berdasarkan perancangan dan pengujian yang telah dilakukan pada model YOLOv8 untuk deteksi arah gerak dan pengenalan objek pada produk ritel, diperoleh hasil yang signifikan. Model dilatih menggunakan gambar produk yang melewati tahap pengujian dan tahap augmentasi. Model yang dihasilkan dari tahap pelatihan dapat digunakan untuk langkah deteksi objek selanjutnya. Tingkat keakuratan deteksi objek bervariasi antar produk, hal ini menunjukkan variasi performa deteksi objek pada produk yang berbeda dalam *dataset*.

Hasil pengujian menggunakan metoda YOLOv8 menunjukkan bahwa semua barang yang terdeteksi sesuai dengan kode produk yang terdaftar dalam *dataset*. Hasil dari *confusion matrix* menunjukkan performa yang baik dalam mengklasifikasikan berbagai kelas produk, dengan sebagian besar nilai akurasi di atas 80% dan bahkan mencapai 100%. Berdasarkan dari hasil *train* pada data yang diperoleh secara umum produk dapat diprediksi dengan nilai mAP₅₀ 98,7% dan mAP₅₀₋₉₅ sebesar 84,6% dengan proses deteksi memerlukan waktu rata-rata sekitar 5 detik setiap gambar.

5.ACKNOWLEDGEMENT

Penelitian ini didukung oleh program pendanaan RIIM Batch 3 yang didanai oleh LPDP Kementerian Keuangan Republik Indonesia dan BRIN (No. 81/IV/KS/05/2023).

DAFTAR PUSTAKA

- [1] R. Wudhikarn, P. Charoenkwan and K. Malang, "Deep Learning in Barcode Recognition: A Systematic Literature Review," in IEEE Access, vol. 10, pp. 8049-8072, 2022, doi: 10.1109/ACCESS.2022.3143033.
- [2] Yuchen Wei, Son Tran, Shuxiang Xu, Byeong Kang, Matthew Springer, "Deep Learning for Retail Product Recognition: Challenges and Techniques", Computational Intelligence and Neuroscience, vol. 2020, Article ID 8875910, 23 pages, 2020. <https://doi.org/10.1155/2020/8875910>

- [3] J. Redmon and A. Farhadi, "Yolov3: an incremental improvement," 2018, <https://arxiv.org/abs/1804.02767>
- [4] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: unified, real-time object detection," in Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 2016.
- [5] J. Redmon and A. Farhadi, "YOLO9000: better, faster, stronger," in Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 2017.
- [6] Y. Li, H. Qi, J. Dai, X. Ji, and Y. Wei, "Fully convolutional instance-aware semantic segmentation," in Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 2017.
- [7] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in Proceedings of the 2017 IEEE International Conference on Computer Vision, Venice, Italy, 2017.
- [8] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: towards real-time object detection with region proposal networks," Advances in Neural Information Processing Systems, pp. 91–99, MIT Press, Cambridge, MA, USA, 2015.
- [9] M. Klasson, C. Zhang, and H. Kjellstrom, "A hierarchical " grocery store image dataset with visual and semantic labels," in Proceedings of the 2019 IEEE Winter Conference on Applications of Computer Vision (WACV), IEEE, Waikoloa Village, HI, USA, 2019.
- [10] P. Follmann, T. Bottger, P. Hartinger, R. Konig, and M. Ulrich, "MVTec D2S: densely segmented supermarket dataset," in Proceedings of the 2018 European Conference on Computer Vision (ECCV), Munich, Germany, 2018.
- [11] Terven, Juan, Diana-Margarita Córdova-Esparza, and Julio-Alejandro Romero-González, "A Comprehensive Review of YOLO Architectures in Computer Vision: From YOLOv1 to YOLOv8 and YOLO-NAS," in Machine Learning and Knowledge Extraction 5, no. 4: 1680-1716, 2023, <https://doi.org/10.3390/make5040083>
- [12] Reis, D., Kupec, J., Hong, J., & Daoudi, A. "Real-Time Flying Object Detection with YOLOv8", 2023, <http://arxiv.org/abs/2305.09972>
- [13] E. Y. B. H. Y. Y. J. L. Y. M. G. N. T. L. Yingbin Bai, "Understanding and Improving Early Stopping for Neural Networks Training" dalam 35th Conference on Neural Information Processing Systems (NeurIPS 2021), 2021.