

PENERAPAN ALGORITMA CNN MENGGUNAKAN *FRAMEWORK* YOLO UNTUK DETEKSI OBJEK PRODUK DI PERUSAHAAN MANUFAKTUR

Asep Maulana ^{1*}; Maman Suherman ²; Anis Fitri Nur Masruriyah ³; Hilda Yulia Novita ⁴

Teknik Informatika ^{1,2,3,4}

Universitas Buana Perjuangan ^{1,2,3,4}

<https://ubpkarawang.ac.id> ^{1,2,3,4}

if20.asepmaulana@mhs.ubpkarawang.ac.id ^{1*}, if20.mamansuherman@mhs.ubpkarawang.ac.id ²,
anis.masruriyah@ubpkarawang.ac.id ³, hilda.yulia@ubpkarawang.ac.id ⁴



Abstract—Component products used for manufacturing a machine in manufacturing companies have two types of products, type A and B. The problem that often occurs in the industry is product sorting errors due to the traditional sorting process, using human labor. The disadvantages are limited human labor so that fatigue can occur, causing errors in sorting products and losses for the company. Many studies discuss object detection, Industrial problems in the checking process can be approached with the help of this technology. Object detection works to analyze frames with the method of finding objects. There are methods in digital image processing, CNN algorithms which include methods in computer vision. The growing framework makes the CNN algorithm more powerful. YOLO includes a framework based on the CNN algorithm. YOLOv5 detects objects by taking into account the object's confidence value, the output of the detected object is a bounding box on the object. The problem in the industry in the checking process can be approached with the help of this technology. For this reason, this research aims to create a model for product object detection in manufacturing companies. The process carried out is data collection, image annotation, training, testing, evaluation. The images collected were 137 for training data and 34 for validation data totaling 171 image data. The results of the model using YOLOv5 with epoch 1000 get a precision value of 100%, recall 100% and mAP 99%, the product detection results get an average value of 100%.

Keywords: convolutional neural network, detection object, manufacturing, product, YOLO.

Abstrak—Produk komponen yang digunakan untuk pembuatan mesin di perusahaan manufaktur terdapat dua tipe produk tipe A dan B. Masalah yang sering terjadi pada industri yaitu kesalahan sortir produk yang disebabkan proses sortir masih tradisional, menggunakan tenaga manusia. Kekurangannya terbatasnya tenaga manusia sehingga kelelahan dapat terjadi dan menyebabkan kesalahan dalam penyortiran produk dan kerugian bagi perusahaan. Banyak penelitian yang membahas deteksi objek. Deteksi objek bekerja untuk menganalisa frame untuk menemukan objek. Terdapat metode dalam pengolahan citra digital, algoritma CNN yang termasuk metode dalam visi komputer. Kerangka kerja yang terus berkembang menjadikan algoritma CNN lebih kuat, YOLO termasuk kerangka kerja yang berbasis algoritma CNN. YOLOv5 mendeteksi objek dengan memperhitungkan nilai keyakinan objek, luaran objek yang terdeteksi berupa kotak pembatas pada objek. Masalah pada industri dalam proses pengecekan dapat dilakukan pendekatan dengan bantuan teknologi ini. Untuk itu penelitian ini bertujuan membuat model untuk deteksi objek produk di perusahaan manufaktur. Proses yang dilakukan pengumpulan data, anotasi citra, pelatihan, pengujian, evaluasi. Citra yang dikumpulkan sebanyak 137 untuk data latih dan 34 untuk data validasi total 171 data Citra. Hasil model menggunakan YOLOv5 yang dibangun dengan epoch 1000 mendapatkan nilai presisi 100%, recall 100% dan mAP 99%, hasil deteksi produk mendapatkan nilai rata-rata 100%.

Kata kunci: jaringan syaraf konvolusi, deteksi objek, manufaktur, produk, YOLO.

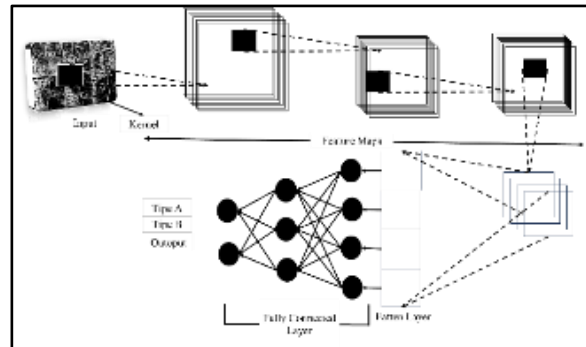
PENDAHULUAN

Perusahaan manufaktur adalah perangkat kegiatan dan operasi yang saling berhubungan yang mencakup desain, pemilihan material, perencanaan, produksi, penjaminan mutu, manajemen dan

pemasaran (Sobron, 2021). Dalam Undang-undang (UU) No. 3 Tahun 2014 industri manufaktur yaitu suatu kegiatan ekonomi, yang mengolah bahan baku, barang setengah jadi, barang jadi, menjadi barang dengan nilai yang lebih tinggi dalam penggunaannya. Masalah yang sering terjadi pada

industri yaitu kesalahan sortir produk yang disebabkan proses sortir masih tradisional dengan cara menggunakan tenaga manusia terutama menggunakan mata untuk melihat bentuk produk. Kekurangannya terbatasnya tenaga manusia sehingga kelelahan dapat terjadi. Hal tersebut menyebabkan kesalahan dalam penyortiran produk dan berakibat fatal karena dapat menyebabkan kerugian bagi perusahaan. Kemudian setiap produk yang tidak sesuai standar harus ditolak dan dimulai ulang dari awal ini pasti akan berhubungan bertambahnya biaya operasional (Simbolon, 2021).

Banyak penelitian yang membahas deteksi objek, namun belum ada penelitian mengenai deteksi objek produk di perusahaan manufaktur. Deteksi objek adalah salah satu teknik *computer vision* (Duhayyim, 2022) (Chiu, 2020), masalah pada industri dalam proses pengecekan dapat dilakukan pendekatan dengan bantuan teknologi ini. Salah satunya dapat dilakukan dengan deteksi objek dengan menggunakan gambar atau video melalui kamera (Zhao, 2019). Deteksi objek bekerja untuk menganalisa *frame* dengan metode menemukan objek sesuai dengan kelasnya berdasarkan model yang dibuat. Salah satu pengolahan citra digital adalah algoritma *Convolutional Neural Network* (CNN) (Li, 2022). Algoritma CNN adalah suatu jaringan syaraf tiruan dengan menggunakan metode *feed-forward* (Raj, 2021). Arsitektur CNN meliputi input citra, *convolutional layer*, *pooling layer* atau *max pooling*, *activation function* dan *fully connected layer* (Bhatt, 2021)(Yedla, 2020). *Convolutional layer* merupakan tahap awal dari CNN untuk melakukan konvolusi kemudian dilanjut *pooling layer* berfungsi untuk mereduksi ukuran matriks dari input citra. Kemudian fungsi aktivasi untuk menentukan neuron diaktivasi atau tidak. *fully connected layer* untuk menghubungkan semua fungsi aktivasi dan lapisan konvolusi. CNN dapat mengenali dan mengklasifikasikan gambar melalui pola citra yang terdeteksi dengan akurasi yang baik (Narvekar, 2020) (Raj, 2021). Secara keseluruhan CNN tidak berbeda jauh dengan *neural network* biasa. CNN memiliki neuron dengan bobot, bias, dan fungsi aktivasi. Lapisan konvolusi dalam CNN juga terdiri dari neuron yang disusun sedemikian rupa sehingga membentuk filter dengan ukuran panjang dan lebar (*pixel*). CNN sering digunakan untuk mengenali objek, melakukan deteksi, atau segmentasi objek. Arsitektur CNN terbagi menjadi dua bagian, yaitu *feature learning* dan *classification*. Gambar masukan diproses melalui CNN melalui proses *feature learning*, dan hasilnya kemudian digunakan untuk proses klasifikasi hingga menghasilkan prediksi akhir (Kurniasari, 2021) arsitektur CNN dapat dilihat pada gambar 1.

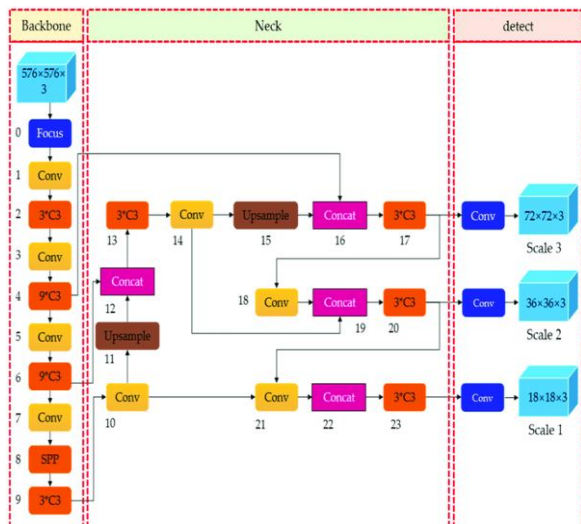


Sumber: (Agustian, 2019)

Gambar 1. Arsitektur CNN

Kerangka kerja yang terus berkembang menjadikan algoritma CNN lebih *powerfull*, YOLO termasuk kerangka kerja yang berbasis algoritma CNN (Zhang, 2020) (Gothane, 2021). YOLO singkatan dari *You Only Look Once* artinya dapat mendeteksi multi objek dalam 1 citra, dan dapat digunakan untuk deteksi objek secara *real time* (Strbac, 2020)(Fang, 2020)(Yi, 2019). Proses deteksi YOLO membagi citra kedalam grid $S \times S$, setiap *grid cell* memprediksi objek dengan *bounding box* (B) dan *confidence* (C). Informasi pada *bounding box* terdapat 5 nilai yaitu x, y, w, h dan c . x dan y adalah titik pusat koordinat kotak pembatas, w dan h adalah lebar dan tingginya kotak pembatas, c adalah nilai *confidence bounding box* tersebut. sehingga output prediksi YOLO yaitu $S \times S \times (B \times 5 + C)$ (Li et al., 2019) (Xiao, 2021) (Yedla, 2020). *You Only Look Once version 5* merupakan *framework* deteksi objek yang tercepat dan akurasi tinggi dalam implementasi di Pytorch dengan menggunakan bahasa pemrograman Python (Kundu, 2021) (Thuan, 2021). YOLO versi 5 dirilis semenjak bulan april tahun 2020 merupakan model generasi kelima untuk deteksi objek. *Framework* ini dikembangkan oleh komunitas yang bernama ultralytics. Untuk implementasi harus menentukan kotak pembatas pada objek di dalam citra, yang memiliki informasi x, y, w, h . x dan y adalah titik pusat kotak pembatas w dan h adalah lebar dan tinggi kotak pembatas. Kemudian proses pelatihan terhadap data citra dengan kelas objek. Komputasi komputer terhadap *framework* tersebut dapat berjalan memanfaatkan piranti kartu grafis guna menghasilkan model untuk deteksi objek. Arsitektur YOLO versi 5 terdiri dari 3 bagian yaitu *backbone*, *neck*, *head* dimana terdapat proses konvolusi dari setiap bagiannya (Glucina, 2023). YOLOv5 merupakan salah satu model pembelajaran yang paling populer di ranah deteksi objek untuk memprediksi beberapa kotak pembatas dan probabilitas kelas untuk kotak-kotak pembatas tersebut. YOLO melatih seluruh bagian gambar dan

mengoptimalkan kinerja pendeteksian secara langsung dan dapat digunakan untuk *real time detection*. Gambar 2 merupakan arsitektur YOLOv5.



Sumber: (Sary, 2023)

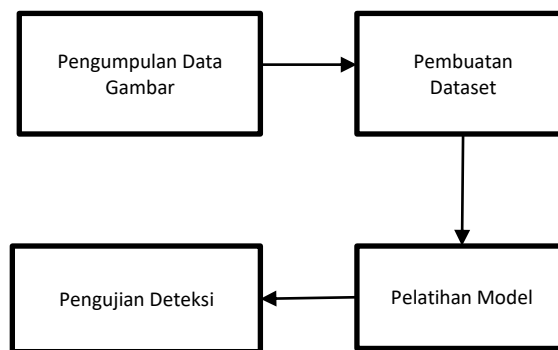
Gambar 2. Arsitektur YOLOv5

Banyak peneliti yang menggunakan YOLO untuk mendeteksi objek dan klasifikasi objek dalam penelitian yang dilakukan oleh Hatab yaitu mendeteksi cacat pada permukaan baja dengan menggunakan arsitektur YOLO (Hatab, 2021). Cacat permukaan baja yang dimaksud adalah goresan dan tambalan. Data yang digunakan terdapat 6 kelas cacat permukaan diantaranya digulung, tambalan, permukaan tidak beraturan, lubang di permukaan, inklusi dan goresan. Data gambar yang digunakan masing-masing kelas sebanyak 300 gambar, setelah dilakukan pelatihan dengan menggunakan arsitektur YOLO mencapai mAP sebesar 70.66%. Dalam penelitian Wang menggunakan YOLO versi 5 untuk mendeteksi tomat di industri tomat, merubah *anchor value* preisi yang didapat sebesar 97.8% dan *recall* sebesar 96.7% dan model sebesar 6.33 MB (Wang, 2023). Peneliti berikutnya Leamsaard menggunakan algoritma *deep learning* dengan YOLOv5 untuk mendeteksi masker wajah (Leamsaard, 2021). Pada penelitian ini dilakukan dengan variasi *epoch* yaitu 20, 50, 100, 300, dan 500. Dengan menggunakan total 682 foto yang terbagi menjadi tiga kelas, yaitu menggunakan masker, tanpa masker dan penggunaan masker yang salah. Pekerjaan YOLOv3 terbaik didapat pada 300 *epoch* mencapai hasil mAP sebesar 0,738. Pada penelitian ini menggunakan data validasi sebesar 85 foto dan hasilnya dilakukan *testing* pada 86 foto. Dengan hasil mAP yang telah disebutkan sebelumnya dan hasil metrik lainnya berupa presisi dan *recall*, maka performansi model ini mendapatkan akurasi sebesar 96.5%.

Berdasarkan penjelasan yang diuraikan di atas, memiliki suatu tujuan yaitu membuat model untuk deteksi produk dengan menggunakan CNN dan *framework* YOLO. Kemudian mencoba beberapa parameter *epoch* dan ukuran gambar input untuk mengetahui performa model. Selain itu versi YOLO sampai saat ini masih berkembang akan menggunakan 2 versi yaitu versi 5 dan versi 8 untuk membandingkan akurasi dari setiap model. Kemudian hasil akhir dilakukan evaluasi dengan melakukan pengujian deteksi produk dari model yang telah dibangun menggunakan *framework* YOLO.

BAHAN DAN METODE

Dalam penelitian ini dimulai dengan akuisisi data yaitu mengumpulkan data gambar dari setiap tipe produk menggunakan kamera digital dengan ukuran 640 x 480. Kemudian melakukan *preprocessing* yaitu membuat dataset dengan menentukan kotak pembatas dan memberi label pada setiap gambar. Proses selanjutnya yaitu pelatihan model dengan menggunakan CNN dengan kerangka kerja YOLOv5. Kemudian pengujian deteksi untuk mengevaluasi performa model. Metode penelitian yang digunakan dalam penelitian ini dapat diidentifikasi dan dipahami lebih lanjut dengan merujuk pada Gambar 3.



Sumber: (Maulana et al., 2024)

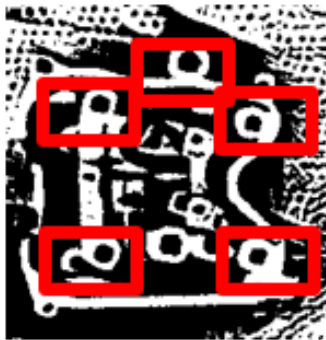
Gambar 3. Alur Metode Penelitian

Pengumpulan data gambar untuk bahan penelitian diambil dari dua tipe produk diantaranya produk tipe A dan tipe B. Produk tipe A dan B memiliki perbedaan pada lubang produk di mana produk tipe A hanya memiliki 4 lubang seperti pada Gambar 4. Sedangkan produk tipe B memiliki 5 lubang seperti pada Gambar 5. Pengambilan gambar dilakukan menggunakan kamera digital dengan ukuran gambar 640 x 480. Proses pengambilan gambar dilakukan dengan cara mengambil gambar dari posisi atas kemudian merotasi objek untuk mendapatkan beberapa sudut atau *angle* produk.

Gambar yang diambil sebanyak 171 gambar produk tipe A dan tipe B, data ini akan dijadikan dataset dimana membagi 80% atau 137 data latih dan 20% atau 34 untuk data uji.



Sumber: (Maulana et al., 2024)
Gambar 4. Tipe A



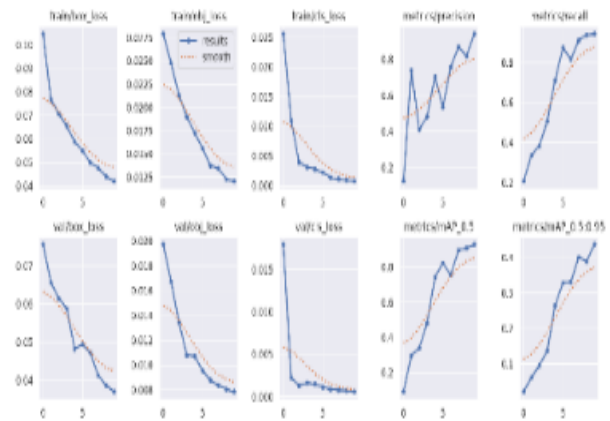
Sumber: (Maulana et al., 2024)
Gambar 5. Tipe B

Pembuatan dataset yang dilakukan secara online melalui situs makesense.ai, kemudian memberi label dengan menentukan kotak pembatas atau *bounding box* pada setiap objek dan menentukan kelasnya. Setelah semua gambar di beri label kemudian meng-*export* label, file yang akan didapat berupa file txt. Di dalam file txt terdapat informasi kelas objek dan nilai x,y,w,h, dimana x dan y adalah titik pusat kotak pembatas w dan h adalah lebar dan tinggi kotak pembatas.

Pelatihan yang dilakukan dengan cara membagi dataset dimana 80% atau 137 data latih dan 20% atau 34 untuk validasi. Pelatihan model yang dilakukan melalui Google Collabs, karena prosesnya membutuhkan memori yang besar, untuk mengaksesnya harus terhubung internet yang stabil. Google collabs memiliki spesifikasi GPU Tesla T4 dan memori 16 GB. Metode pelatihan akan menggunakan *epoch* 10, 100, 1000 dan ukuran gambar 320, 480, 640. Hasil setiap *epoch* akan menjadi bahan perbandingan dalam menentukan model terbaik.

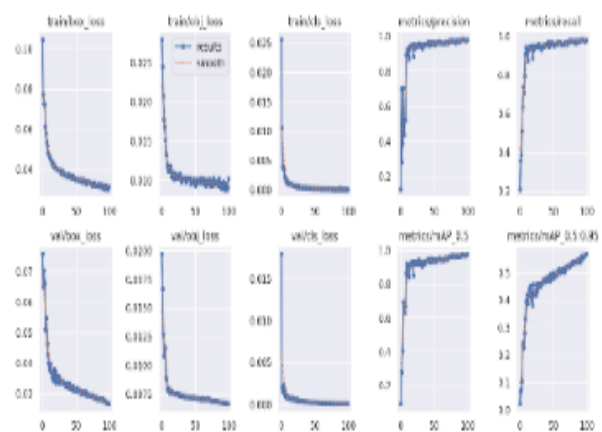
HASIL DAN PEMBAHASAN

Hasil Pelatihan menggunakan model YOLO versi 5 melalui *google collabs* dengan *epoch* atau iterasi sebanyak 10 kali membutuhkan waktu selama 2 menit. Hasil yang didapatkan diantaranya presisi sebesar 0.942 *recall* sebesar 0.942 mAP sebesar dapat dilihat pada Gambar 6.



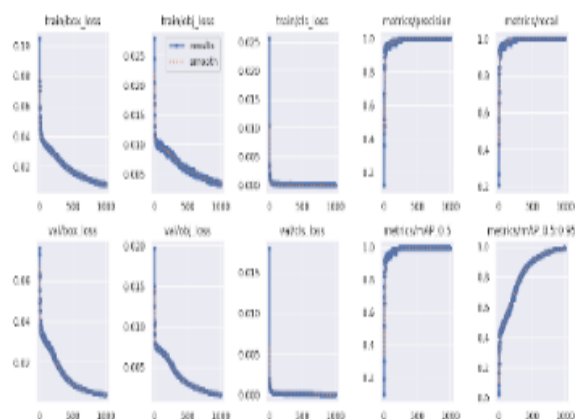
Sumber: (Maulana et al., 2024)
Gambar 6. Hasil *Training* 10 *Epoch*

Pelatihan dengan *epoch* 100 membutuhkan waktu selama 22 menit dan hasil yang didapatkan diantaranya presisi sebesar 0.979 *recall* sebesar 0.980 mAP sebesar 0.977 dapat dilihat pada Gambar 7.



Sumber: (Maulana et al., 2024)
Gambar 7. Hasil *Training* 100 *Epoch*

Pelatihan dengan *epoch* 1000 membutuhkan waktu selama 154 menit dan hasil yang didapatkan diantaranya presisi sebesar 1.00 *recall* sebesar 1.00 mAP sebesar 0.995 dapat dilihat pada Gambar 8.



Sumber: (Maulana et al., 2024)
 Gambar 8. Hasil *Training* 1000 Epoch

Sebelum menggunakan model untuk deteksi yaitu membandingkan model dengan tujuan mencari performa model terbaik. Perbandingan berdasarkan ukuran gambar input, *epoch*, *precision*, *recall*, mAP. Pelatihan menggunakan *epoch* 10, dengan ukuran citra 640 mendapatkan presisi 0.94. Kemudian dengan ukuran citra 480 mendapatkan presisi 0.93. Kemudian ukuran citra 320 mendapatkan presisi 0.93. Pelatihan menggunakan *epoch* 100, dengan ukuran citra 640 mendapatkan presisi 0.97. Kemudian dengan ukuran citra 480 mendapatkan presisi 0.96. Kemudian ukuran citra 320 mendapatkan presisi 0.97. Pelatihan menggunakan *epoch* 1000, dengan ukuran citra 640 mendapatkan presisi 1. Kemudian dengan ukuran citra 480 mendapatkan presisi 1. Kemudian ukuran citra 320 mendapatkan presisi 1. Hasil perbandingan dapat dilihat pada Tabel 1.

Tabel 1. Perbandingan *Precision*

No	Ukuran gambar	Precision		
		10 epoch	100 epoch	1000 epoch
1	640	0.942	0.979	1.000
2	480	0.937	0.967	1.000
3	320	0.932	0.971	1.000

Sumber: (Maulana et al., 2024)

Pelatihan menggunakan *epoch* 10, dengan ukuran citra 640 mendapatkan *recall* 0.94. Kemudian dengan ukuran citra 480 mendapatkan presisi 0.94. Kemudian ukuran citra 320 mendapatkan presisi 0.93. Pelatihan menggunakan *epoch* 100, dengan ukuran citra 640 mendapatkan *recall* 0.98. Kemudian dengan ukuran citra 480 mendapatkan presisi 0.96. Kemudian ukuran citra 320 mendapatkan presisi 0.97. Pelatihan menggunakan *epoch* 1000, dengan ukuran citra 640 mendapatkan *recall* 1. Kemudian dengan ukuran citra 480 mendapatkan *recall* 1. Kemudian ukuran

citra 320 mendapatkan *recall* 1. Hasil perbandingan dapat dilihat pada Tabel 2.

Tabel 2. Perbandingan *Recall*

No	Ukuran gambar	<i>Recall</i>		
		10 epoch	100 epoch	1000 epoch
1	640	0.942	0.980	1.000
2	480	0.942	0.968	1.000
3	320	0.935	0.970	1.000

Sumber: (Maulana et al., 2024)

Pelatihan menggunakan *epoch* 10, dengan ukuran citra 640 mendapatkan mAP_{0.5} 0.92. Kemudian pelatihan dengan ukuran citra 480 mendapatkan mAP_{0.5} 0.93. Kemudian pelatihan dengan ukuran citra 320 mendapatkan mAP_{0.5} 0.91. Pelatihan menggunakan *epoch* 100, dengan ukuran citra 640 mendapatkan mAP_{0.5} 0.97. Kemudian pelatihan dengan ukuran citra 480 mendapatkan mAP_{0.5} 0.95. Kemudian pelatihan dengan ukuran citra 320 mendapatkan mAP_{0.5} 0.91. Pelatihan menggunakan *epoch* 1000, dengan ukuran citra 640 mendapatkan mAP_{0.5} 0.99. Kemudian pelatihan dengan ukuran citra 480 mendapatkan mAP_{0.5} 0.99. Kemudian pelatihan dengan ukuran citra 320 mendapatkan mAP_{0.5} 0.99. Hasil perbandingan dapat dilihat pada Tabel 3.

Tabel 3. Perbandingan mAP_{0.5}

No	Ukuran gambar	mAP _{0.5}		
		10 epoch	100 epoch	1000 epoch
1	640	0.929	0.977	0.995
2	480	0.932	0.958	0.995
3	320	0.912	0.912	0.995

Sumber: (Maulana et al., 2024)

Berdasarkan tabel perbandingan presisi bahwa pada setiap ukuran gambar dan jumlah *epoch* yang berbeda, presisi cenderung tinggi, dengan nilai diatas 0.9. Hal ini menunjukkan bahwa model memiliki kemampuan yang baik. Kemudian *Recall* cenderung tinggi pada setiap ukuran gambar dan jumlah *epoch* yang berbeda, dengan nilai diatas 0.9. Hal ini menunjukkan bahwa model memiliki kemampuan yang baik. Kemudian mAP_{0.5} menunjukkan angka yang tinggi pada setiap ukuran gambar dan jumlah *epoch* yang berbeda, dengan nilai diatas 0.9. Hal ini menunjukkan bahwa model memiliki kemampuan yang baik. Pada tabel 1, 2 dan 3 menunjukkan bahwa model memiliki tingkat presisi, *recall*, dan mAP_{0.5} yang cukup tinggi di berbagai ukuran gambar dan jumlah *epoch* yang berbeda. Berdasarkan hasil *training* dan perbandingan pada tabel 1, 2 dan 3 bahwa dengan

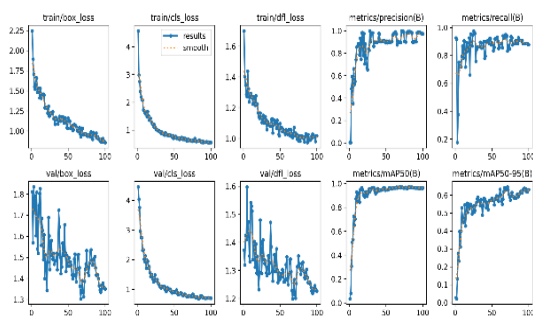
ukuran gambar 640 *epoch* 1000 adalah model terbaik yang akan digunakan untuk deteksi.

Setelah pelatihan model YOLO, model akan dievaluasi menggunakan data validasi. Hasilnya berupa deteksi dan klasifikasi, luaran dari evaluasi adalah *bounding box* atau kotak pembatas dan nama setiap produk pada dari kelas objek pada titik yang sudah ditentukan sebelumnya. Hasil evaluasi menggunakan YOLOv5 menggunakan *input* citra ukuran 640 dan *epoch* 1000 dapat dilihat pada gambar 9. Berdasarkan evaluasi model YOLO mampu mendeteksi dan membedakan setiap kelas objek.



Sumber: (Maulana et al., 2024)
Gambar 9. Hasil Validasi Model

Selain membandingkan *epoch* pelatihan menggunakan YOLOv5 dalam penelitian ini membandingkan dengan model lain yaitu YOLOv8. Hasil yang didapatkan pelatihan menggunakan YOLOv8 dengan *epoch* 100 mendapatkan presisi 1, recall 0.832 dan mAP 0.97. dapat dilihat pada gambar 10. Meskipun hasil dari masing-masing model terdapat kelebihan dan kekurangan namun dari kedua versi mendapatkan hasil yang cukup baik.



Sumber: (Maulana et al., 2024)
Gambar 10. Hasil *training* YOLOv8

Setelah dilakukannya *training* model menggunakan YOLOv5 dan dan YOLOv8, dari setiap model mendapatkan hasil yang cukup baik. Pada YOLOv5 presisi sedikit lebih rendah dibandingkan dengan YOLOv8. Kemudian *recall* pada YOLOv5 lebih tinggi dibandingkan dengan YOLOv8. Kemudian mAP pada masing-masing versi hasil tidak ada perbedaan. Perbandingan dari setiap versi YOLO dapat dilihat pada Tabel 4.

Tabel 4. Perbandingan YOLOv5 dan YOLOv8

YOLO	Precision	Recall	mAP	Epoch	Size
V5	0.979	0.98	0.97	100	640
V8	1	0.83	0.97	100	640

Sumber: (Maulana et al., 2024)

Pada tahap ini akan dilakukan pengujian deteksi dengan menggunakan citra baru dengan menggunakan model terbaik yang dipilih. Model yang dipilih yaitu model yang dilatih dengan *epoch* 1000. Pengujian deteksi menggunakan model *yolo.pt* sebagai *weight custom*, menentukan nilai *threshold* yaitu sebesar 0.7 atau 70% sebagai ambang batas deteksi. Untuk pengujian deteksi model menggunakan data gambar baru. Sample gambar yang diambil dari setiap tipe sebanyak 5 gambar dengan berbagai sudut total 10 gambar 5 gambar tipe A dan 5 gambar tipe B. Hasil uji deteksi gambar A1.jpg menunjukkan nilai *confidence* sebesar 0.8 atau 80% dan label menunjukkan class A. Hasil uji deteksi gambar A2.jpg menunjukkan nilai *confidence* sebesar 0.83 atau 83% dan label menunjukkan class A. Hasil uji deteksi gambar A3.jpg menunjukkan nilai *confidence* sebesar 0.85 atau 85% dan label menunjukkan class A. Hasil uji deteksi gambar A4.jpg menunjukkan nilai *confidence* sebesar 0.85 atau 85% dan label menunjukkan class A. Hasil uji deteksi gambar A5.jpg menunjukkan nilai *confidence* sebesar 0.86 atau 86% dan label menunjukkan class A. Hasil uji deteksi gambar B1.jpg menunjukkan nilai *confidence* sebesar 0.95 atau 95% dan label menunjukkan class B. Hasil uji deteksi gambar B2.jpg menunjukkan nilai *confidence* sebesar 0.94 atau 94% dan label menunjukkan class B. Hasil uji deteksi gambar B3.jpg menunjukkan nilai *confidence* sebesar 0.95 atau 95% dan label menunjukkan class B. Hasil uji deteksi gambar B4.jpg menunjukkan nilai *confidence* sebesar 0.94 atau 94% dan label menunjukkan class B. Hasil uji deteksi gambar B5.jpg menunjukkan nilai *confidence* sebesar 0.94 atau 94% dan label menunjukkan class B. Sample gambar uji deteksi untuk dijadikan asumsi akurasi seperti pada Tabel 5.

Tabel 5. Pengujian Deteksi Objek

No	Tipe	Gambar	Confidence	Label
1	A	A1.jpg	0.8	A
2	A	A2.jpg	0.83	A
3	A	A3.jpg	0.85	A
4	A	A4.jpg	0.85	A
5	A	A5.jpg	0.86	A
6	B	B1.jpg	0.95	B
7	B	B2.jpg	0.94	B
8	B	B3.jpg	0.95	B
9	B	B4.jpg	0.94	B
10	B	B5.jpg	0.94	B

Sumber: (Maulana et al., 2024)

Diambil 10 gambar sebagai sampel uji dengan 5 gambar tipe A, 5 gambar tipe B dan 1 gambar yang di dalamnya terdapat dua tipe. Hasil yang diperoleh mendapatkan nilai *confidence* lebih dari 0.8 dan label menunjukan sesuai dengan tipe produk. Masing-masing tipe terdeteksi dengan benar sehingga disimpulkan akurasi adalah:

$$Akurasi = \frac{5}{5} \times 100\%$$

Hasil yang didapatkan untuk deteksi produk disimpulkan berhasil karena nilai uji deteksi yang didapat 100% dan saat 2 tipe dicampur algoritma mampu membedakan dengan benar sesuai kelasnya.

KESIMPULAN

Dengan menerapkan algoritma CNN menggunakan *framework* YOLO pada deteksi objek mendapatkan presisi pada pelatihan cukup baik Pelatihan dengan *epoch* 10 membutuhkan waktu selama 32 menit dan hasil yang didapatkan diantaranya presisi sebesar 0.942 *recall* sebesar 0.942 mAP sebesar 0.929. Kemudian hasil pelatihan dengan *epoch* 100 membutuhkan waktu selama 182 menit dan hasil yang didapatkan diantaranya presisi sebesar 0.979 *recall* sebesar 0.980 mAP sebesar 0.977. Kemudian hasil pelatihan dengan *epoch* 1000 membutuhkan waktu selama 654 menit dan hasil yang didapatkan diantaranya presisi sebesar 1.00 *recall* sebesar 1.00 mAP sebesar 0.995. Kemudian pada uji deteksi produk tipe A dan B mendapatkan nilai *confidence* yang cukup baik dan output label sesuai dengan kelas objek sehingga akurasi yaitu sebesar 100%, maka metode ini ampuh dalam mengatasi masalah dalam penelitian ini.

REFERENSI

Al Duhayyim, M., Alabdulkreem, E., Tarmissi, K., Aljebreen, M., El Khier, B. S. I. A., Zamani, A. S., Yaseen, I., & I. Eldesouki, M. (2022). *Aquila Optimization with Transfer Learning Based Crowd Density Analysis for Sustainable Smart Cities*. Applied Sciences (Switzerland), 12(21). <https://doi.org/10.3390/app12211187>

Agustian, I., Hadi, F., Khairul, M., & Rosa, A. (2019). *Pre-Diagnosis Gangguan Ginjal Melalui Citra Iris Mata Menggunakan Raspberri PI Dengan Metode Convolutional Neural Network (CNN)*. In Jurnal Amplifier Mei (Vol. 9, Issue 1).

Bhatt, D., Patel, C., Talsania, H., Patel, J., Vaghela, R., Pandya, S., Modi, K., & Ghayvat, H. (2021). *Cnn variants for computer vision: History, architecture, application, challenges and future scope*. In Electronics (Switzerland) (Vol. 10, Issue 20). MDPI. <https://doi.org/10.3390/electronics10202470>

Chiu, Y.-C., Tsai, C.-Y., Ruan, D., Shen, G.-Y., & Lee, T.-T. (2020). *Mobilenet-SSDv2: An Improved Object Detection Model for Embedded Systems*. International Conference on System Science and Engineering (ICSSE).

Fang, W., Wang, L., & Ren, P. (2020). *Tinier-YOLO: A Real-Time Object Detection Method for Constrained Environments*. IEEE Access, 8, 1935–1944. <https://doi.org/10.1109/ACCESS.2019.2961959>

Glucina, M., Andelic, N., Lorencin, I., & Car, Z. (2023). *Detection and Classification of Printed Circuit Boards Using YOLO Algorithm*. Electronics (Switzerland), 12(3). <https://doi.org/10.3390/electronics12030667>

Gothane, Dr. S. (2021). *A Practice for Object Detection Using YOLO Algorithm*. International Journal of Scientific Research in Computer Science, Engineering and Information Technology, 268–272. <https://doi.org/10.32628/cseit217249>

Hatab, M., Malekmohamadi, H., Amira, A. (2021). *Surface Defect Detection Using YOLO Network*. In: Arai, K., Kapoor, S., Bhatia, R. (eds) Intelligent Systems and Applications. IntelliSys 2020. Advances in Intelligent Systems and Computing, vol 1250. Springer, Cham. https://doi.org/10.1007/978-3-030-55180-3_37

Kundu, N., Rani, G., & Dhaka, V. S. (2021). *Seeds Classification and Quality Testing Using Deep Learning and YOLO v5*. ACM International

- Conference Proceeding Series, 153–160. <https://doi.org/10.1145/3484824.3484913>
- Kurniasari, N., & Sugiono, J. P. (2021). *Deteksi Jalur Yang Terputus Pada Rangkaian Listrik Dalam Pcb Menggunakan Metode Convolutional Neural Network (CNN)*. In Surabaya Jurnal Sistem Cerdas dan Rekayasa (*JSCR*) (Vol. 3, Issue 1).
- Leamsaard, J., Charoensook, S. N., & Yammen, S. (2021). Deep Learning-based Face Mask Detection Using *YoloV5*. International Electrical Engineering Congress, *IEECON 2021*, 428–431. <https://doi.org/10.1109/IEECON51072.2021.9440346>
- Li, W., Shen, Z., & Li, P. (2019). *Crack Detection of Track Plate Based on YOLO*. *Proceedings - 2019 12th International Symposium on Computational Intelligence and Design, ISCID 2019*, 15–18. <https://doi.org/10.1109/ISCID.2019.10086>
- Li, Z., Liu, F., Yang, W., Peng, S., & Zhou, J. (2022). *A Survey of Convolutional Neural Networks: Analysis, Applications, and Prospects*. *IEEE Transactions on Neural Networks and Learning Systems*, 33(12), 6999–7019. <https://doi.org/10.1109/TNNLS.2021.3084827>
- Narvekar, C., & Rao, M. (2020). *Flower classification using CNN and transfer learning in CNN-Agriculture Perspective*. *Proceedings of the 3rd International Conference on Intelligent Sustainable Systems, ICISS 2020*, 660–664. <https://doi.org/10.1109/ICISS49785.2020.9316030>
- Raj, R., Nagaraj, S. S., Ritesh, S., Thushar, T. A., & Aparanji, V. M. (2021). *Fruit Classification Comparison Based on CNN and YOLO*. *IOP Conference Series: Materials Science and Engineering*, 1187(1), 012031. <https://doi.org/10.1088/1757-899x/1187/1/012031>
- Sary, I. P., Armin, E. U., Andromeda, S. (2023). *Performance Comparison of YOLOv5 and YOLOv8 Architectures in Human Detection Using Aerial Images*. *Ultima Computing: Jurnal Sistem Komputer*, Vol. 15, No.1
- Simbolon, R., & Santoso, S. (2021). *Product And Service Quality Development In Manufacturing: A Study Of Optical Lens Manufacturing Indonesia*. *Media Ekonomi Dan Manajemen*, 36(1). <https://doi.org/10.24856/mem.v36i1.1726>
- Sobron, M., Lubis Bidang, Y., Manufaktur, T., Teknik, P., Jurusan, M., Industri, T., & Kunci, K.-K. (2021). *Implementasi Artificial Intelligence Pada System Manufaktur Terpadu*. *SEMNASSTEK UISU: 978-623-7297-39-0*
- Strbac, B., Gostovic, M., Lukac, Z., & Samardzija, D. (2020). *YOLO Multi-Camera Object Detection and Distance Estimation*. In 2020 Zooming Innovation in Consumer Technologies Conference (ZINC).
- Thuan, Do. (2021). *Evolution Of Yolo Algorithm And Yolov5: The State-Of-The-Art Object Detection Algorithm*. Oulu: Oulu University of Applied Sciences
- Undang-undang Republik Indonesia Nomor 3 tentang Perindustrian 2014. Jakarta.
- Wang, X., Wu, Z., Jia, M., Xu, T., Pan, C., Qi, X., & Zhao, M. (2023). *Lightweight SM-YOLOv5 Tomato Fruit Detection Algorithm for Plant Factory*. *Sensors (Basel, Switzerland)*, 23(6). <https://doi.org/10.3390/s23063336>
- Xiao, J. (2021). *ExYOLO: A small object detector based on YOLOv3 Object Detector*. *Procedia CIRP*, 188, 18–25. <https://doi.org/10.1016/j.procs.2021.05.048>
- Yedla, S. K., Manikandan, V. M., & Panchami, V. (2020). *Real-Time Scene Change Detection with Object Detection for Automated Stock Verification*. *ICDCS 2020 - 2020 5th International Conference on Devices, Circuits and Systems*, 157–161. <https://doi.org/10.1109/ICDCS48716.2020.243571>
- Yi, Z., Yongliang, S., & Jun, Z. (2019). *An improved tiny-yolov3 pedestrian detection algorithm*. *Optik*, 183, 17–23. <https://doi.org/10.1016/j.ijleo.2019.02.038>
- Zhang, S., Wu, Y., Men, C., & Li, X. (2020). *Tiny yolo optimization oriented bus passenger object detection*. *Chinese Journal of Electronics*, 29(1), 132–138. <https://doi.org/10.1049/cje.2019.11.002>
- Zhao, Z. Q., Zheng, P., Xu, S. T., & Wu, X. (2019). *Object Detection with Deep Learning: A Review*. In *IEEE Transactions on Neural Networks and Learning Systems* (Vol. 30, Issue 11, pp. 3212–3232). Institute of Electrical and Electronics Engineers Inc. <https://doi.org/10.1109/TNNLS.2018.2876865>