

Implementasi *Artificial Intelligence* untuk Mendeteksi Kantong Pecah di *Cement Packing House*

Abdussalam Darmaatmaja^{1*}, Guntoro²

¹PT Indocement Tunggul Prakarsa, Citereup, Bogor, Jawa Barat

²Politeknik Semen Indonesia, Kebomas, Gresik, Jawa Timur

*Koresponden email: abdussalamd68@gmail.com

Diterima; 21 Januari 2026

Disetujui : 05 Februari 2026

Abstract

Bag cleanliness is one of the major concerns in most of cement factories especially in PT Indocement Tunggul Prakarsa. Typically, during cement packaging when a bag is broken the operator will stop the belt conveyor immediately and ordered the worker to evacuate the bag. However, there have been a lot of cases where the operator was late in stopping the conveyor. This will cause material to spill on conveyor and contaminate it. To resolve this problem, a fast response system capable of stopping the conveyor when it detects a broken bag is needed. The system needs to differentiate broken bag from normal ones, and one of the ways to do that is by machine vision. The model used for the machine vision is called YOLO (*Ultralytics Documentation*). In this research, the model was trained using 1741 images of bag conditions in a cement packing house. The model is then connected with plant's control system using Snap7 library from Python (*Snap7 Documentation*). With another 471 images, the model was able to successfully detect broken bag with a 68.07% accuracy.

Keywords: *Bag cleanliness, machine vision, YOLO, Snap7, Python*

Abstrak

Kebersihan kantong semen merupakan salah satu masalah yang menjadi perhatian utama di pabrik semen terutama di PT Indocement Tunggul Prakarsa. Secara umum, dalam proses pengantongan semen, saat kantong pecah operator akan langsung menghentikan *belt conveyor* dan menginstruksikan pekerja untuk mengevakuasi kantong. Namun terdapat banyak kejadian di mana operator akan terlambat menghentikan *belt conveyor*. Hal ini menyebabkan material tumpah ke conveyor dan mengkontaminasinya. Untuk mengatasi masalah ini, dibutuhkan sistem yang bereaksi dengan cepat dan mampu menghentikan conveyor saat mendeteksi kantong pecah. Sistem ini harus mampu membedakan kantong pecah dari kantong normal, dan salah satu cara untuk membuat sistem tersebut adalah *machine vision*. Model yang digunakan untuk ini disebut YOLO (*Dokumentasi Ultralytics*). Dalam penelitian ini, model dilatih menggunakan 1741 gambar kondisi kantong di *Cement Packing House*. Model ini akan dikoneksikan dengan system control pabrik menggunakan Snap7 library dari Python (*Dokumentasi Snap7*). Dengan 471 gambar lain, model telah mampu mendeteksi kantong pecah dengan akurasi 68,07%

Kata Kunci: *Kebersihan kantong, machine vision, YOLO, Snap7, Python*

1. Pendahuluan

Proses produksi semen mempunyai 5 tahapan utama penambangan yaitu, *grinding, burning, cooling, dan packing* [1]. Dalam setiap tahapan ini, terdapat potensi untuk menghasilkan kontaminan udara berupa debu halus yang dapat mengotori kantong semen dan juga lingkungan sekitar. Tahapan yang penting diawasi karena berhubungan langsung dengan pembeli adalah tahapan packing. Terdapat beberapa sumber kontaminan di tahapan ini, antara lain: Saat mengisi semen ke kantong di rotary packer, saat menutup kantong semen yang terisi, dan saat proses transportasi kantong semen menuju truk pengiriman. Saat proses transportasi, jika terjadi kantong pecah dan tidak segera dievakuasi secepatnya oleh operator maka dapat berdampak fatal, dikarenakan *belt conveyor* yang dipakai akan tertumpahi material semen, hingga tersebar sepanjang *belt conveyor* jika tidak dihentikan secara dini. Hal ini akan menyebabkan, kantong semen yang kotor dan dapat mengurangi estetika produk semen.

Untuk meningkatkan kecepatan evakuasi kantong pecah oleh operator di proses packing semen, maka dibuat suatu system yang dapat menggantikan operator memantau *belt conveyor* secara langsung selama 24 jam. Sistem ini akan mengontrol jalannya *belt conveyor* dengan memperhatikan ada atau tidaknya kantong pecah di *belt conveyor*. Sistem ini akan menggunakan kamera Hikvision sebagai alat penerima data (sensor), dan menggunakan program yang dibuat menggunakan bahasa python sebagai

controller yang memberikan data berupa bit kepada PLC (Programmable Logic Controller) yang mana akan dikoneksikan dengan system DCS (Distributed Control System) pabrik sehingga program mampu memberikan peringatan maupun menghentikan laju belt conveyor.

Program akan menggunakan model yang dibuat oleh YOLO dari Ultralytics [2]. Model ini banyak digunakan oleh praktisi dan akademisi untuk mengembangkan model AI (Artificial Intelligence) untuk memproses gambar dan mendeteksi benda di gambar. Model ini juga bebas digunakan tanpa biaya apa pun, sehingga model mudah digunakan dari segi teknik maupun legalitas. Selain itu, seiring waktu dengan bertambahnya data yang dikumpulkan dapat meningkatkan akurasi model AI hingga dapat menyamai penglihatan manusia.

Oleh karena itu, dengan menggunakan model AI untuk mendeteksi kantong pecah di proses pengantongan semen dapat memudahkan peranan operator dalam menjalankan mesin. Model AI dapat memantau jalannya system pengantongan semen, dan melakukan intervensi langsung ke sistem saat terdapat kondisi abnormal (Kantong pecah). Hal ini dapat mengurangi kotornya belt conveyor karena keterlambatan operator dalam mengevakuasi dan mendeteksi kantong pecah.

2. Metode Penelitian

Secara umum, dalam mengembangkan model AI berbasis neural network terdapat 4 tahapan utama [3] antara lain pengumpulan data, pelatihan model, perakitan sistem kontrol dan komisioning sistem. Alur dari tahapan penelitian ini dapat diperhatikan pada **Gambar 1** berikut ini



Gambar 1. Flowchart Penelitian
Sumber: Penulis (2025)

2.1. Pengumpulan Data

Kualitas dari model AI sangat bergantung pada jumlah data dan kualitas data yang digunakan untuk melatih model tersebut. Model yang akan dibuat bertujuan untuk mendeteksi kantong pecah yang terjadi di belt conveyor sehingga data yang dibutuhkan untuk melatih model adalah gambar kantong pecah yang diambil oleh CCTV yang terpasang. Contoh data yang diambil untuk melatih model ini dapat dilihat pada **Gambar 2**.

Berdasarkan penelitian sebelumnya [4] didapatkan bahwa model mampu meningkatkan akurasi dengan banyaknya data training yang digunakan untuk melatih model, namun peningkatan akurasi ini hanya meningkat secara logarithmic berdasarkan *power law* di statistika. Sehingga, untuk training mungkin hanya diperlukan kurang dari 1000 data saja, penambahan lebih dari seribu hanya akan menambah akurasi secara minimal dikarenakan hubungan data set dengan akurasi yang mengikuti *power law* (terjadi secara logarithmic) [5]. Karena itu, untuk sistem deteksi objek ini akan dikumpulkan data sebanyak 1741 data, dengan 1009 data bag normal, 50 data kantong pecah, dan sisa data merupakan data di mana belt conveyor tidak mendeteksi kantong apapun. Alasan mengapa hanya diambil 50 data untuk kantong pecah dikarenakan keterbatasan kesempatan dalam mengambil data tanpa mengganggu proses produksi semen.



Gambar 2. Sampel Data Broken Bag
Sumber: Penulis (2025)

2.2. Pelatihan Model

Dengan 1781 data yang telah diperoleh, model YOLO dilatih dengan syntax kode yang dibuat pada **Gambar 3** berikut.

```
Train(): #if you're using dedicated GPU

#load the Model
model = YOLO("C:\\Users\\Inspection_P011\\anaconda\\venv\\AI\\Project\\Models\\Used\\yolo11x.pt")

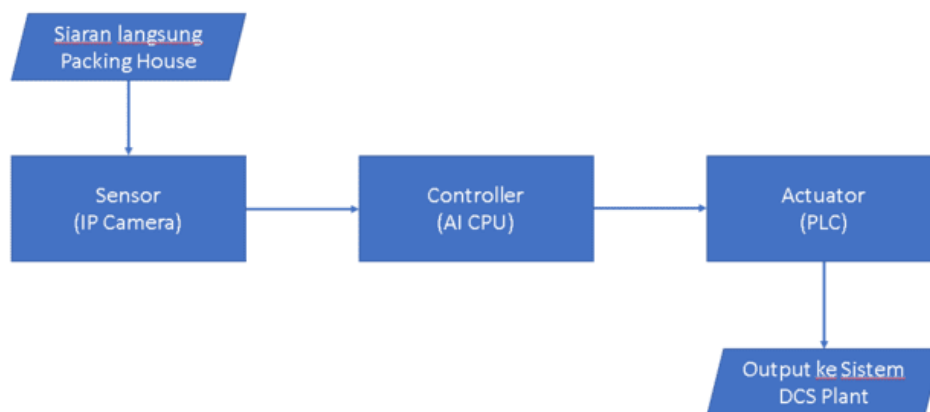
#assigning save directories
savedir = "C:\\Users\\Inspection_P011\\anaconda\\venv\\AI\\Project\\Models\\Used\\Trained_Model\\Real_Training_Results_Final\\"
#train the model
result = model.train(data = "C:\\Users\\Inspection_P011\\Downloads\\AI_Data\\Broken_Bag_P01\\Real_Camera_Broken_Bag_P01.yolo11\\data.yaml", epochs = 100, device = 0, save
```

Gambar 3. Kode untuk Pelatihan Model YOLO
Sumber: Penulis (2025)

Hasil output dari kode di atas merupakan model YOLO yang siap untuk dipakai. Model YOLO menggunakan convolutional neural network [6] dimana model akan memprediksi gambar dengan memberikan bounding boxes pada gambar yang diterima

2.3. Perakitan Sistem Kontrol

Sama dengan system control pada umumnya [7], system ini terdiri dari 3 bagian utama: CPU AI model sebagai controller, kamera sebagai sensor, dan PLC sebagai actuator (PLC akan memberikan perintah untuk menghentikan laju belt). Konfigurasi system dapat dilihat pada **Gambar 4** berikut



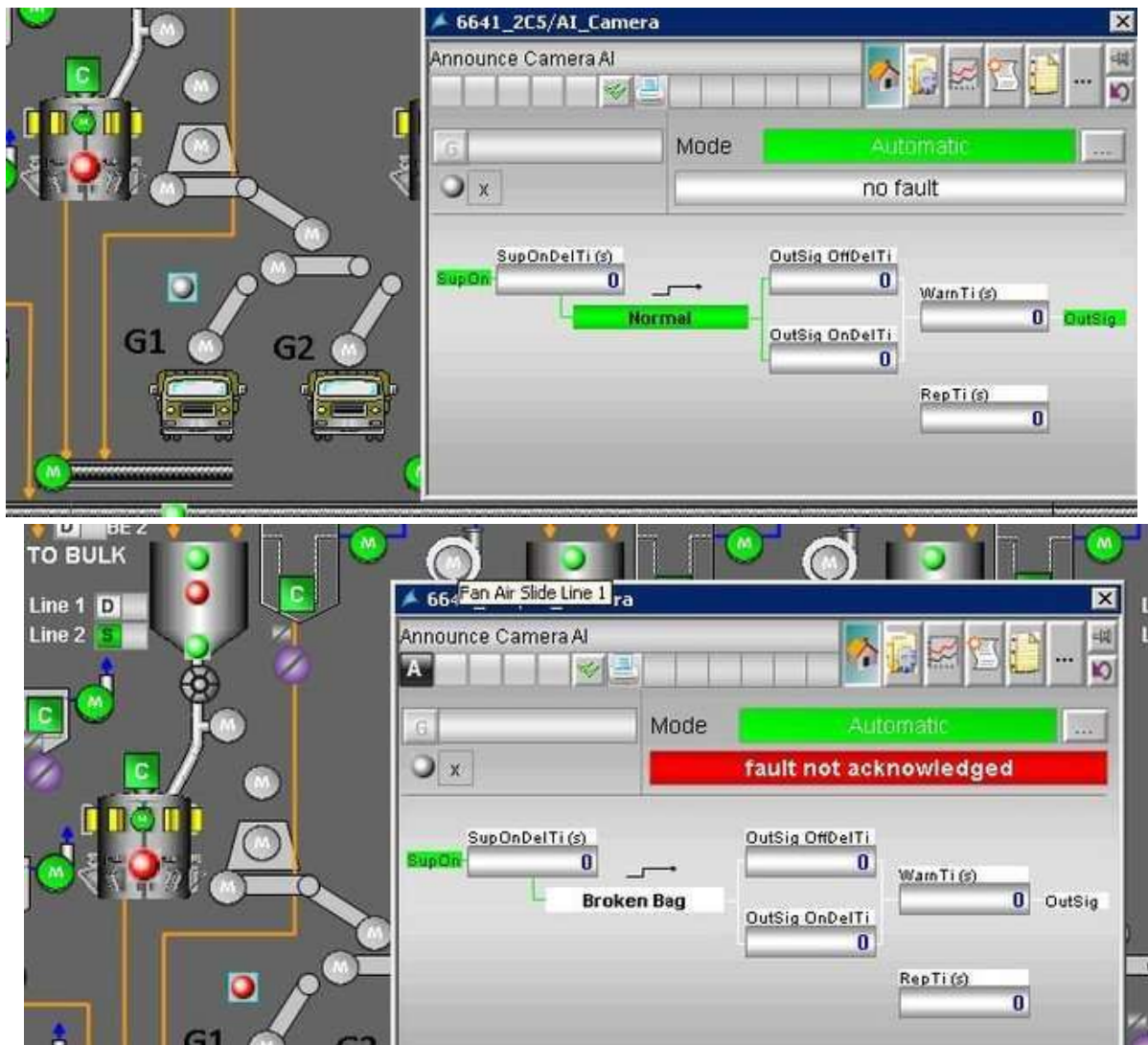
Gambar 4. Konfigurasi Sistem Kontrol
Sumber: Penulis (2025)

CPU yang berupa sebuah PC (Personal Computer) di mana model AI akan mengolah data gambar dan video yang didapat secara real time bertindak sebagai kontroler yang akan memberikan perintah pada actuator di system DCS untuk menghentikan belt conveyor. Controller akan memberikan perintah melalui perantara PLC, sementara Controller juga menerima data rekaman secara langsung melalui kamera yang bekerja sebagai sensor. PC tidak disambungkan langsung dengan system DCS di plant untuk memenuhi peraturan dari Heidelberg group material mengenai program eksternal yang digunakan di system control plant [8]. Dalam pemberian perintah dari PC menuju ke PLC akan digunakan library khusus dari python yang dinamakan Snap7 [9]. Output yang dikirim akan berupa bit data yang akan diterima oleh IO DCS

plant. Bit ini akan diterima sebagai Digital Input oleh DCS dan akan diproses untuk menghentikan laju belt conveyor.

2.4. Komisioning Sistem

Setelah model dilatih menggunakan data yang ada, dengan 471 gambar yang berbeda dengan data pelatihan, model akan diuji apakah mampu mendeteksi kantong pecah. Tahapan validasi ini termasuk tahapan komisioning system. Dalam tahapan ini, system yang telah dibuat juga akan diuji dengan system DCS plant. Dalam system DCS indikasi berhentinya belt conveyor dapat dilihat pada **Gambar 5** berikut.



Gambar 5. Indikasi Stop di Sistem DCS Plant
Sumber: Penulis (2025)

Pada **Gambar 5**, warna merah menunjukkan bahwa system mendeteksi kantong pecah sehingga akan menghentikan laju *belt conveyor*. Sementara, warna abu-abu menunjukkan sistem normal dan tidak mendeteksi kantong pecah

3. Hasil dan Pembahasan

Dalam penelitian ini, implementasi model AI yang telah dibuat akan diuji saat mesin packer berjalan dan terdapat pesanan semen. Sistem akan diuji apakah mendeteksi kantong pecah dan mampu membedakannya dengan kantong normal

3.1. Hasil Training Model

Setelah model dilakukan training didapatkan hasil yang ditunjukkan dengan beberapa metrics yang sudah pernah dilakukan oleh penelitian lain [10], hasil dapat dilihat pada **Tabel 1** berikut.

Tabel 1. Hasil Pelatihan Model YOLO

Class	Precision	Recall	mAP50	mAP50-95
Normal	91.1%	75%	98.7%	93.4%
Broken	52%	84.6%	74.8%	43.1%
Miring	71.6%	99.1%	79.3%	75%

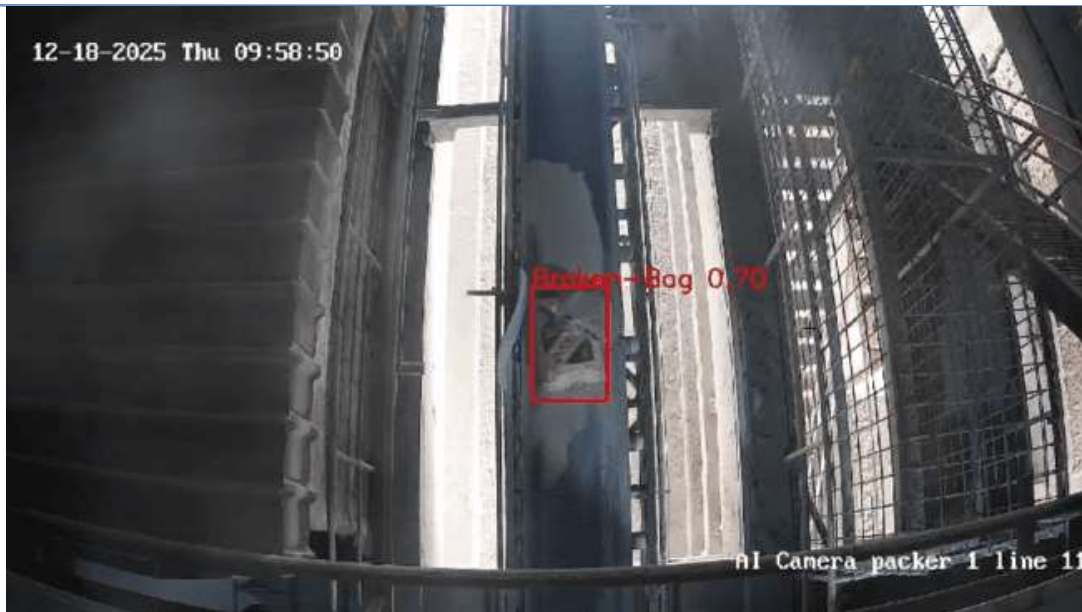
Sumber: Penulis (2025)

Tabel 1 menunjukkan hasil pelatihan model YOLO setelah dilakukan validasi dengan 471 data baru di luar data pelatihan model. Dari table 1 dengan nilai parameter untuk bag broken sangat kecil dibandingkan dengan nilai parameter untuk bag normal. Selain itu nilai Precision yang sangat kecil dibandingkan nilai Recall juga menunjukkan kinerja model yang kurang, ini dikarenakan Precision yang berarti kemampuan model untuk menghasilkan prediksi yang sama, sementara recall merupakan kemampuan model menemukan objek deteksi yang benar [11]. Karena precision lebih kecil dari recall, menunjukkan kalau model tidak bisa konsisten dalam mendeteksi kantong pecah

3.2. Pengujian Sistem

Sistem diuji saat rotary packer berjalan normal hasil deteksi system dapat dilihat pada **Gambar 6** di bawah





Gambar 6. Deteksi Model saat Packer Machine Normal
Sumber: Penulis (2025)

Setelah melakukan pengujian, data pengujian yang didapat dianalisa untuk melihat hasil prediksi dari kamera. **Tabel 2** berikut merupakan hasil pengujian yang didapatkan

Tabel 2. Hasil Pengujian Sistem

No. Prediction	True Positive (Images)	False Positive (Images)	True Negative (Images)	False Negative (Images)
1 st	5	0	0	17
2 nd	206	0	0	154
3 rd	5	0	0	190
4 th	0	189	0	0

Sumber: Penulis (2025)

Tabel 2 menunjukkan seberapa banyak objek yang dideteksi secara benar oleh model dalam kondisi normal. True positive menunjukkan seberapa banyak model mampu memprediksi objek secara benar [12]. Dengan menggunakan persamaan untuk menghitung nilai presisi, akurasi dan recall [13] didapatkan **Gambar 7** dengan hasil pada **Tabel 3** berikut:

$$Recall = \frac{TP}{RP} = \frac{A}{A + C}$$

$$Precision = \frac{TP}{PP} = \frac{A}{A + B}$$

$$Accuracy = \frac{TP + TN}{N}$$

+R	-R				
tp	fp	pp	+P	A	A+B
fn	tn	pn	-P	C	C+D
rp	rn	1		A+C	B+D
					N

Gambar 7: Persamaan Presisi, Recall, dan Akurasi [13]
Sumber: D.M.W, Powers (2011)

Pada **Gambar 7**, TP adalah nilai True Positive, RP adalah jumlah True Positive dan False Negative,

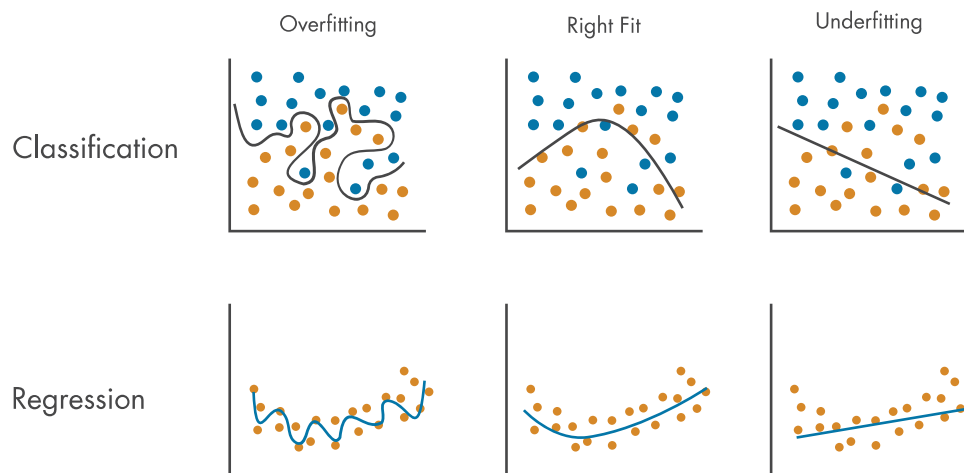
PP adalah jumlah True Positive dan False Positive, sementara N adalah jumlah total dari sampel.

Tabel 3. Hasil Perhitungan Presisi, Recall dan Akurasi

Class	Precision	Recall	Akurasi
Broken	100%	53.02%	68.07%

Sumber: Penulis (2025)

Tabel 3 diatas menunjukkan bahwa jika dibandingkan dengan nilai yang didapat pada tabel 1 cukup berbeda dikarenakan jumlah sampel yang berbeda saat pelatihan dengan pengujian di lapangan. Selain itu model juga terlihat mengalami overfitting [14] ditandai dengan nilai presisi yang sangat lebih besar dari nilai akurasi model, hal ini diakibatkan model yang terlalu terbiasa dengan dataset tertentu sehingga mengurangi akurasi. Gambar model overfitting dapat dilihat pada **Gambar 8** di bawah



Gambar 8: Overfit, Right Fit, Underfit [15]

Sumber: Matlab (2026)

Model overfit menunjukkan kalau model terlalu mengikuti pola tertentu data sehingga tidak mampu memprediksi objek secara umum. Model underfit menunjukkan kalau model kurang mengenali pola yang ada dari data sehingga tidak mampu mendeteksi objek. Tujuan dari setiap pelatihan model AI adalah mendapatkan model yang right fit sehingga mampu memprediksi objek secara optimal.

4. Kesimpulan

Setelah model YOLO yang dibuat selesai diuji dan data yang diterima dianalisis didapatkan bahwa model memiliki akurasi sebesar 68,07% dengan presisi 100%. Nilai ini menunjukkan kemampuan model untuk mendeteksi objek secara konsisten namun nilai akurasi rendah menunjukkan model mempunyai potensi salah deteksi. Nilai akurasi yang kecil dibanding dengan nilai presisi yang besar setelah dilakukan tes terhadap model menunjukkan model memiliki gejala overfitting. Di mana, model terlalu terbiasa dengan suatu kelas dataset sehingga akurasi model menjadi kurang optimal. Model yang telah dibuat juga terkoneksi dengan system control pabrik sehingga mampu mengendalikan laju belt conveyor secara otomatis. Saat model mendeteksi kantong pecah, model akan memberikan sinyal ke system control pabrik untuk menghentikan belt conveyor.

5. Saran

Kinerja model dapat ditingkatkan dengan menggunakan data lebih banyak lagi untuk pelatihan model. Saat ini hanya terdapat 50 data kantong pecah yang dimasukkan ke model, sehingga model masih kurang optimal dalam mendeteksi kantong pecah tersebut. Penambahan data ini juga bertujuan untuk menghilangkan overfitting dari model. Karena itu data kantong pecah harus ditambahkan sebanyak 1004 data hingga seimbang dengan kantong normal menjadi 1009 data. Selain itu, saat ini model hanya mampu mendeteksi 1 objek dalam 1 frame kamera. Hal ini dapat ditingkatkan dengan membuat model mendeteksi lebih dari 1 objek dalam 1 frame kamera.

6. Daftar Pustaka

- [1] Panagoda, L.P.S.S. 2023. "Cement Manufacturing Process and Its Environmental Impact". Nugegodha, Sri Lanka. Journal of Research Technology & Engineering ISSN: 2714 – 1837 | p161-168
- [2] Cong, Xiaohan. 2023. "A Review of YOLO Object Detection Algorithms based on Deep Learning". Tianjin, China. Frontiers in Computing and Intelligent System ISSN: 2832-6024 | Vol 4, No2
- [3] Saputra, Haris Dede. 2023. Object Detection untuk Mendeteksi Citra Buah Buahan Menggunakan Metode YOLO. Mataram, Indonesia. Jurnal Kecerdasan Buatan dan Teknologi Informasi
- [4] Hestness, Joel. 2017. "Deep Learning Scaling is Predictable, Empirically". New York, USA. Arxiv, Cornell University
- [5] Sun, Chen. 2017. "Revisiting Unreasonable Effectiveness of Data in Deep Learning Era". New York, USA. Arxiv, Cornell University
- [6] Ghosh, Aniruda. 2020. "Fundamental Concepts of Convolutional Neural Network". West Bengal, India. Springer
- [7] Ogata, Katsuhiko. 2010. "Modern Control Engineering 5th edition". New Jersey, USA. Prentice Hall
- [8] Sterr, Roland. 2024. "AI Policy". Heidelberg, Jerman. Group Legal & Compliance, Heidelberg Materials
- [9] Cook, M. Marco. 2023. "PLCPrint: Fingerprinting Memory Attacks in Programmable Logic Controllers". Glasgow, United Kingdom. University of Glasgow
- [10] Azam Adeel, Muhammad. 2022. "Deep Learning Applied to White Light and Narrow Band Imaging Videolaryngoscopy: Toward Real-Time Laryngeal Cancer Detection". New Jersey, USA. Laryngoscope, Wiley
- [11] Zhang, Peng. 2012. "Statistical Inference on Recall, Precision and Average Precision under Random Selection". Zhejiang, China. 9th International Conference on FSKD (Fuzzy Systems and Knowledge Discovery)
- [12] Sathyanarayan, S. 2024. "Confusion Matrix-Based Performance Evaluation Metrics". Karnataka, India. African Journal of Biomedical Research
- [13] Powers, D. M. W. 2011. "Evaluation: From Precision, Recall and F-Measure to ROC, Informedness, Markedness & Correlation". South Australia, Australia. Journal of Machine Learning Technologies
- [14] Montesinos Lopez, Osval Antonio. 2022. "Multivariate Statistical Machine Learning Methods for Genomic Prediction. Cham, Switzerland. Springer Nature Switzerland AG
- [15] Ying, Xue. 2019. "An Overview of Overfitting and its Solutions". Beijing, China. IOP Publishing: Journal of Physics