

Sistem Inspeksi Visual Penempatan Label Produk *Lip Cream Line* Menggunakan Metode *Deep Learning*

Moch Deny Triatmaja^{1*}, Lie Jasa²

[Submission: 10-06-2023, Accepted: 24-07-2023]

Abstract—SBL (Single Bottom Labeller) machine which was still not precise for placing the bottom label batches of lip cream products. Postal operators must then check one by one to ensure product quality control so that there are bottlenecks. Therefore, this research uses the YOLOv5 deep learning method to solve this problem. The YOLOv5 algorithm uses the idea of regression, making it easier to learn generalizations, target characteristics and solve speed problems. The YOLO algorithm uses a one-stage neural network to complete the localization and classification of detected objects in real time. The core idea of YOLO is to take the entire image as network input and directly return the bounding box position and bounding box class in the output. In YOLO, each bounding box is predicted by the features of the entire image, and each bounding box contains the five predictions and a confidence, which is relative to the grid cell in the middle of the bounding box. The results of research using the YOLOv5 method can classify into three classification categories of bottom label placement, namely accept, reject and no label. In addition to classifying, this study will also send triggers to the controller if the bottom label is said to be rejected and no label to do product sorting to assist QC operators in sorting lip cream production defects.

Keywords— Deep Learning, Lip cream, Quality Control, You Only Look Once v5

Intisari—Mesin SBL (*Single Bottom Labeller*) yang masih belum presisi untuk penempatan *bottom label batch* dari produk *lip cream*. Operator pos selanjutnya harus memeriksa satu persatu untuk memastikan *quality control* produk sehingga terdapat *bottleneck*. Maka dari itu dalam penelitian ini menggunakan metode *deep learning YOLOv5* untuk memecahkan masalah tersebut. Algoritma YOLOv5 menggunakan ide regresi, sehingga lebih mudah untuk mempelajari generalisasi, menargetkan karakteristik dan memecahkan masalah kecepatan. Algoritma YOLO menggunakan jaringan saraf *one stage* untuk menyelesaikan lokalisasi dan klasifikasi objek yang terdeteksi secara *real time*. Ide inti dari YOLO adalah menggunakan seluruh gambar sebagai input jaringan dan langsung mengembalikan posisi kotak pembatas dan kelas kotak pembatas pada output. Di YOLO, setiap kotak pembatas diprediksi oleh fitur dari keseluruhan gambar, dan setiap kotak pembatas berisi lima prediksi dan *confidence*, yang relatif terhadap sel *grid* di tengah kotak pembatas. Hasil dari penelitian menggunakan metode YOLOv5 dapat mengklasifikasi dalam tiga kategori klasifikasi penempatan *bottom label* yaitu *Accept*, *Reject* dan *No Label*. Selain melakukan klasifikasi, penelitian ini juga akan mengirimkan *trigger* ke *controller*

apabila *bottom label* dikatakan *Reject* dan *No Label* untuk dilakukan sortir produk guna membantu operator QC dalam penyortiran *lip cream* cacat produksi.

Kata Kunci— *Deep Learning*, *Lip cream*, *Quality Control*, *You Only Look Once v5*

I. PENDAHULUAN

Proses QC yang dilakukan pada produk *lip cream* masih manual, tentunya proses ini memiliki kekurangan yaitu terjadinya *human error* yang meliputi beberapa faktor diantaranya ketelitian dan ketajaman mata manusia yang berbeda-beda dikarenakan kelelahan sehingga menurunkan akurasi dan sensitifitas operator. Proses QC (*Quality Control*) yang dilakukan pada produk *lip cream line VSN 03* berguna untuk mencapai standar yang telah di tentukan, proses tersebut merupakan langkah perusahaan untuk menentukan apakah produk akan layak untuk dijual atau tidak. QC yang terdapat pada line VSN 03 masih menggunakan tenaga manusia, tentunya proses ini memiliki kekurangan yaitu terjadinya *human error* yang meliputi beberapa faktor diantaranya ketelitian dan ketajaman mata manusia yang berbeda-beda dikarenakan kelelahan sehingga menurunkan akurasi dan sensitifitas operator. Permasalahan tersebut mempengaruhi kualitas QC sehingga terdapat beberapa produk cacat yang lolos inspeksi. Berkembangnya teknologi dibidang kecerdasan buatan yang dapat membantu manusia sebagai operator QC dan meminimalisir *human error*. Maka dibuat perancangan sistem inspeksi visual kecacatan label produk *lip cream* guna membantu operator QC dalam penyortiran *lip cream* cacat produksi.

Ada beberapa referensi yang pernah digunakan untuk melakukan QC dengan menggunakan kamera contohnya Dalam sebuah jurnal yang berjudul "Rancang Bangun Sistem Deteksi Label Kardus Berbasis Metode Kecerdasan Buatan YOLO dan EasyOCR Serta ESP32-CAM", membahas tentang alat pendeteksi label kardus yang cepat dan akurat. ESP32-CAM memberikan respon pemrosesan sebesar 0,762 detik. Model YOLO yang telah dilatih dengan 40 iterasi pelatihan menggunakan set data kustom menghasilkan loss deteksi objek sebesar 0,00997 dan mean Average Precision (mAP) sebesar 0,983. Hasil pendeteksian objek kardus dengan model YOLO menghasilkan rata-rata nilai *confidence* sebesar 58,002%. Model YOLO menggunakan threshold sebesar 0,5 dengan rasio akurasi: error sebesar 19:1 dan deteksi objek non-kardus sebanyak 2/20 atau 10%. Setelah model YOLO mendeteksi kardus berlabel. [1]

Berikutnya membahas deteksi kerusakan pada karung komoditi dengan menggagas konsep program deteksi objek berupa karung dan lubang kerusakan pada suatu frame gambar

p-ISSN:1693 – 2951; e-ISSN: 2503-2372

¹,Departemen Teknik ELEktro Otomasi, Institut Teknologi Sepuluh Nopember Surabaya, Jawa Timur, Email: denyt920@gmail.com:

² Teknik Elektro, Fakultas Teknik, Universitas Udayana, Bali email : liejasa@unud.ac.id



mengusung dua metode sebagai perbandingan, yaitu You Only Look Once (YOLO) versi 4 dan Mask R-CNN. Hasilnya menunjukkan bahwa model YOLOv4 dapat memberikan performa lebih baik dengan akurasi mencapai 96,8%, sedangkan model Mask R-CNN memiliki kinerja yang kurang dapat diandalkan dengan akurasi 65,78%. [2]

Sementara itu, dalam penelitian lain yang berjudul "Deteksi dan Pengenalan Objek Dengan Model Machine Learning: Model YOLO", mengulas tentang bagaimana model YOLO mampu meningkatkan efisiensi dan kecepatan pemrosesan pendeteksian gambar yang tinggi dengan akurasi yang mampu menyaingi model-model algoritma yang ada sebelumnya. Penelitian ini bertujuan untuk membantu para pengembang baru dalam memahami dan mengembangkan model YOLO dengan mudah, sehingga dapat bersaing secara lebih kompetitif dengan model lain di berbagai bidang aplikasi, seperti pengenalan dan pendeteksian objek pada Computer Vision. [3][4][5][6]

Pada penelitian kali ini dibuatlah sebuah sistem sortir berdasarkan kategori klasifikasi menggunakan metode YOLOv5. Label yang disortir memiliki tiga kategori yaitu *Accept* dan *reject*. YOLO mendeteksi objek dengan menggunakan *unifield model* dimana sebuah *single convolutional network* memprediksi beberapa *bounding boxes*. Hasil dari parameter bobot kemudian dicocokkan dengan tangkapan gambar dari kamera untuk dicocokkan informasi bobotnya dengan model yang telah dilatih sebelumnya. Sistem deteksi kecacatan pada label produk *lip cream* menggunakan YOLO diharapkan dapat membantu proses *quality control*, meminimalisir produk yang lolos *quality control*.

II. TINJAUAN PUSTAKA

Peneliti melakukan studi literatur pada metode yang digunakan dalam menyelesaikan penelitian. algoritma yang digunakan untuk mengimplementasikan deteksi objek.

2.1 YOLOv5

YOLOv5 termasuk dalam keluarga dari YOLO (*You Only Look Once*) dan dibagi menjadi empat model yang dengan ukuran arsitektur berbeda, diantaranya adalah YOLOv5s, YOLOv5m, YOLOv5l, dan YOLOv5x. Jaringan struktur YOLOv5 masih dekat dengan YOLOv4 [7], ilustrasinya dapat dilihat pada Gambar 1. YOLOv5 menggunakan *Cross Stage Paritial Network* (CSPNet) [8] sebagai *backbone* untuk menghasilkan kombinasi gradien sekaligus mengurangi proses komputasi. Pada bagian *neck*, arsitektur ini menggunakan *Spatial Pyramid Pooling-Fast* (SPPF) *block* dan struktur dari PANet (*Path Aggregation Network*) dengan fitur piramida berdasarkan FPN (*Feature Pyramid Network*) *layer*. Bagian *head* YOLOv5 mengeluarkan prediksi berupa vektor yang berisi prediksi kelas dan koordinat *bounding box* beserta *confidence score*-nya.

2.2 Hyperparameter tuning

Proses tuning hyperparameter melibatkan perancangan model machine learning yang ideal melalui pengaturan hyperparameter optimal [9],[10],[11],[12]. Hyperparameter bertujuan untuk mengonfigurasi model machine learning dan menetapkan algoritma yang diterapkan untuk meminimalisir kerugian [10],[13],[14]. Tuning hyperparameter sangat penting dalam pengembangan model machine learning yang

efektif dan setiap model memiliki jenis hyperparameter yang berbeda. YOLOv5, misalnya, memiliki parameter seperti ukuran gambar input, epochs, batch size, learning rate, momentum, dan fungsi aktivasi yang bisa disesuaikan untuk meningkatkan performa model.

1) *Learning Rate* ialah parameter pelatihan yang penting pada model *machine learning* yang berfungsi dalam menghitung pengaturan bobot yang dibenarkan pada waktu pelatihan. Disebut *Learning Rate* (LR), hal ini dapat dikatakan bahwa LR menentukan jumlah step yang digunakan pada setiap iterasi untuk mencapai minimum lokal. Memilih nilai LR memiliki dampak besar terhadap keakuratan dan kecepatan pelatihan pada model [15],[16],[17]

2) *Batch Size* adalah penentu berapa banyak sampel yang akan diberi dalam training kepada jaringan sebelum parameter model diperbaharui. Setiap akhir batch, loss akan dihitung dan parameter model [18],[19] akan diperbaharui dengan nilai tersebut untuk mencapai peningkatan performa model. *Batch size* akan mempengaruhi indikator dalam akurasi dan waktu komputasi [19],[20].

2.3 Evaluasi Model

Mean Average Precision (mAP) merupakan salah satu indikator yang sering digunakan untuk mengevaluasi kinerja suatu model deteksi objek. Fungsi mAP sebagai pengukur performa bertujuan untuk menilai seberapa handalnya suatu model. Selain itu, mAP juga mengkombinasikan antara nilai presisi dan *recall* sehingga dapat memberikan hasil yang optimal dari kedua metrik tersebut.

1) *Mean Average Precision: Precision* adalah kemampuan model dalam mendeteksi objek yang relevan atau presentase dari prediksi positif yang benar. Sedangkan *Average Precision* (AP) diambil dari nilai *precision* dan *recall*. Perhitungan AP ditujukan untuk masing-masing kelas. Pada dataset yang besar, dibutuhkan metrik yang mampu mempresentasikan keakuratan model dalam mendeteksi semua kelas. Dalam kasus ini, mAP yang merupakan rata-rata AP dari semua kelas, menjadi metrik yang dibutuhkan. Perhitungan mAP dan AP didefinisikan pada Persamaan 1.

$$mAP = \frac{1}{N} \sum_{i=1}^N AP \quad (1)$$

Pada mAP, N merupakan jumlah kelas dan AP merupakan *Average Precision* dari tiap kelas. Pada AP, Rn dan Pn merupakan *recall* dan *precision* pada saat *threshold* ke-n. Sedangkan, nilai *recall* dan *precision* dihitung berdasarkan nilai *true positive* (TP), *false positive* (FP), dan *false negative* (FN). Perhitungan dari *recall* dan *precision* ditunjukkan pada Persamaan 2 dan 3

$$Precision = \frac{TP}{TP+FP} \quad (2)$$

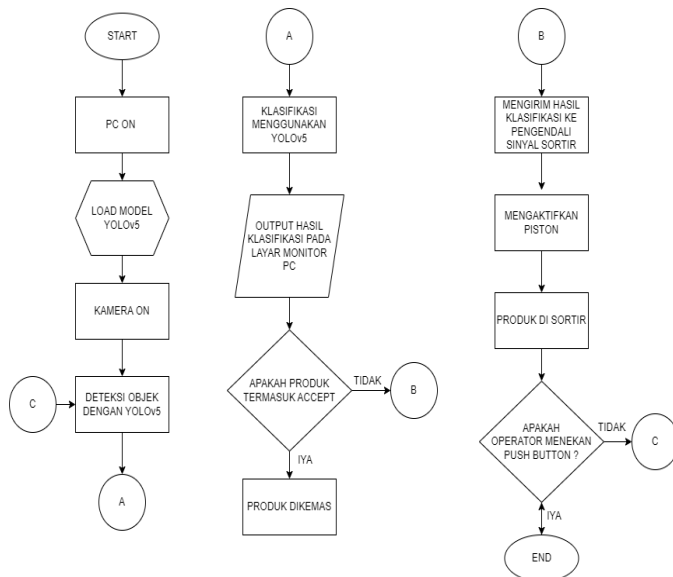
$$Recall = \frac{TP}{TP+FN} \quad (3)$$

III. METODOLOGI

3.1 Diagram Alir Penelitian

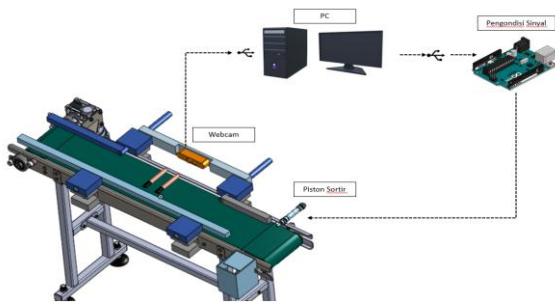
Diagram alir penelitian ini menjadi panduan bagi penulis dalam merancang sistem inspeksi visual penempatan label produk lip cream, pada dataset dilakukan dengan proses *image*

menggunakan model *Neural Network*. Kemudian, dataset tersebut menjadi masukan untuk melatih YOLOv5 yang dioptimalkan dengan proses *hyperparameter tuning*. Diagram alir dapat dilihat pada Gambar 1.



Gambar 1 Diagram Alir Penelitian

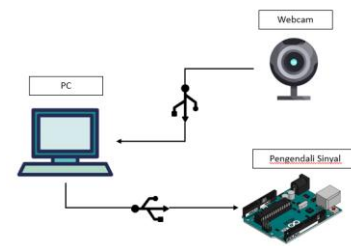
3.2 Arsitektur Sistem



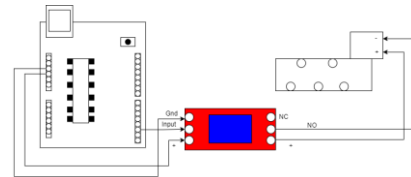
Gambar 2 Arsitektur Sistem

Desain arsitektur keseluruhan sistem merupakan gambaran umum dari keseluruhan sistem yaitu mesin sortir penempatan label berdasarkan citra agar mudah dipahami. Pada Gambar 2 dapat dilihat tiga bagian utama dari arsitektur mesin sortir. Dua bagian tersebut memiliki satu plant yaitu konveyor dengan *lip cream* yang berjalan di atasnya. *lip cream* tersebut kemudian melewati kamera yang terhubung dengan PC. Kamera mengirimkan citra dari *lip cream* yang dideteksi, dengan kelas atau kategori label dibagi menjadi tiga yaitu *accept*, *reject*, dan *no label*. Saat *lip cream* tersebut dideteksi dan diklasifikasi berdasarkan kategori, maka deteksi akan tampil pada layar monitor, *lip cream* yang dideteksi *reject* dan *no label* akan mentrigger pengendali sinyal menggunakan kabel serial. Pengendali sinyal akan mengontak relay untuk mengaktifkan piston untuk memsortir produk *lip cream* kedalam *box reject*. Wiring sistem, wiring sortir dan perancangan P&ID dapat dilihat pada Gambar 3 hingga Gambar 5.

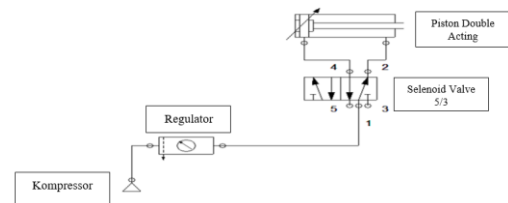
Moch Deny Triatmaja : Sistem Inspeksi Visual Penempatan ...



Gambar 3 Wiring sistem



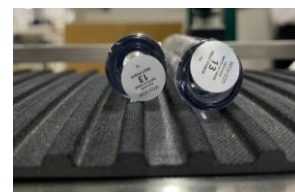
Gambar 4 Wiring Sortir



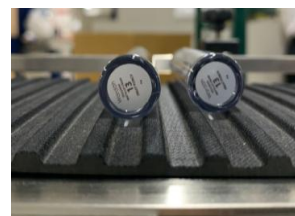
Gambar 5 Perancangan P&ID

3.3 Pengumpulan Dataset

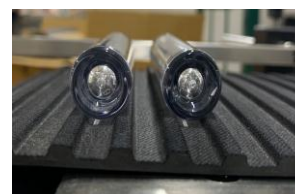
Pengumpulan dataset ini merupakan proses pengambilan gambar *bottom label*, dilakukan menggunakan *program capture* untuk memudahkan proses pengumpulan dataset. Proses pengambilan gambar *bottom label*, gambar yang diambil masing-masing kelasnya sebanyak 50 gambar, total dataset yang diambil yaitu sebanyak 150 gambar *lip cream*.



Gambar 6 Produk Reject



Gambar 7 Produk Accept



Gambar 8 Produk No Label



3.4 Anotasi Dataset

Produk *accept* dalam dataset ini didefinisikan dengan *bottom label* yang tepat ditengah. Produk *reject* memiliki ciri label yang berada dibawah miring dan tidak tepat berada ditengah packaging. Produk yang didefinisikan *no label* pada dataset ini memiliki ciri yaitu tidak terdapat *bottom label* pada packaging produk *lip cream*. Gambar dataset dapat dilihat pada Gambar 6 dan Gambar 7

Anotasi dataset merupakan melakukan pelabelan, penentuan koordinat, dan kotak batas (*bounding box*), anotasi ini dilakukan dengan menggunakan aplikasi labelImg dengan format YOLO, untuk memudahkan proses training. Hasil dari pelabelan tersebut disimpan dalam bentuk *file .txt*, yang didalamnya merupakan informasi dari kelas (anotasi id) dan koordinat (x,y,w,h)

Kelas pada YOLO berbentuk format *file .txt* dimulai dengan angka 0, untuk sistem deteksi ini menggunakan 3 kelas. Setelah melakukan anotasi dataset langkah selanjutnya adalah split dataset ini menjadi dua yaitu *train* dan *test* tujuannya untuk memperkuat proses pelatihan model yang akan kita buat. Pembagian *train* dan *test* dataset ini bisa dibuat rasio 80% untuk *train* dan 20% untuk *test*.

3.5 Training YOLOv5

Langkah terakhir untuk mendapatkan model deteksi adalah *training model*. *Training model* dilakukan dengan cara *transfer learning* dari *pre-trained* model YOLOv5 menggunakan *framework* Darknet-53 yang sudah di-*custom* sesuai dengan kebutuhan. Proses training dilakukan dengan cara memasukkan dataset yang sudah dilakukan proses *pre-processing* sebagai konfigurasi untuk melakukan training terhadap *pre-trained* model *framework* Darknet-53. Proses training ini akan diulang secara terus-menerus sampai sesuai dengan konfigurasi yang ditentukan. *Output* dari training ini merupakan merupakan beberapa bobot data satu dimensi yang akan disatukan pada jaringan syaraf tiruan, bobot-bobot ini akan dijumlahkan dan dijadikan parameter untuk mengenali suatu objek. Penjumlahan parameter bobot tersebut akan menghasilkan hasil training berbentuk format *.weights* yang nantinya akan tersimpan sebagai memori bobot yang telah dilatih, dengan format tersebut kita dapat memanggil kembali parameter ini dan konfigurasi yang nantinya akan dijadikan acuan deteksi dan klasifikasi pada proses *real-time object detection* pada sistem deteksi penempatan label.

3.6 Spesifikasi Perangkat

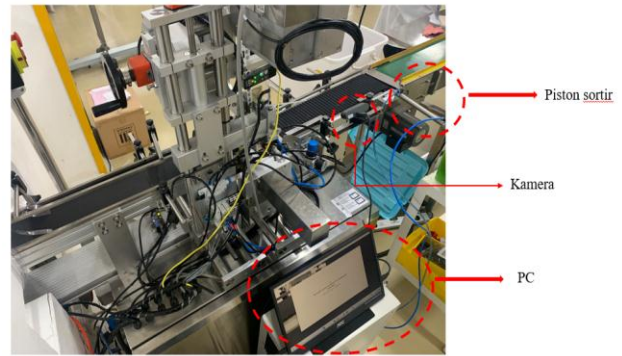
Pada penelitian perangkat keras yang digunakan pada penelitian ini merupakan PC yang memiliki spesifikasi pada tabel dibawah ini :

TABEL I. Spesifikasi Perangkat

Prosesor	RAM	Kartu Grafis	Penyimpanan
Intel(R) Core TM i5-7600 CPU @ 3.50 GHz	16 GB	NVIDIA GeForce GTX 1660 SUPER	SSD 239 GB

IV. PENGUJIAN DAN ANALISIS

4.1 Hasil Perakitan Hardware



4.2 Analisis Perbandingan Kecepatan Konveyor

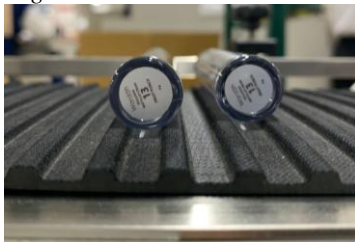
TABEL II. Data Statistik Percobaan Kecepatan Konveyor

	13.08 cm/s	16.75 cm/s	21.99 cm/s
Maximal Confidance	81%	81%	81%
Minimal Confidance	25%	26%	29%
Rata- Rata Confidance	61.40%	45.06%	58.06%
Standart Devisiasi	16.85%	16.75%	15.93%

Dari Tabel 2 dapat dilihat semakin besar kecepatan dari semakin cepat waktu deteksi obyek. Pada kecepatan 2 yang berjalan 13.08 cm/s mencapai nilai rata rata pendeteksian 0.84 detik dimana ini diperoleh dari obyek yang terlalu lama pada posisi *angle* kamera yang pas. Pada kecepatan 13.08 cm/s juga memiliki *standard* devisiasi yang paling besar dibandingkan dengan dua kecepatan lainnya. Pada kecepatan 3 yang berjalan 16.75 cm/s mencapai nilai rata-rata pendeteksian 0.62 detik, nilai ini lebih cepat dibandingkan oleh rata-rata kecepatan 2. Pada kecepatan 16.75 cm/s juga memiliki *standard* devisiasi yang besar dibandingkan dengan kecepatan 21.99cm/s dan lebih kecil daripada kecepatan 13.08 cm/s. Sedangkan pada kecepatan 4 yang berjalan 21.99 cm/s mencapai nilai rata-rata pendeteksian 0.50 detik, nilai ini lebih cepat dibandingkan dengan kecepatan 2 dan kecepatan 3 sehingga kecepatan 4 dapat mengimbangi ppm mesin *labeller*. Pada kecepatan 21.99 cm/s juga memiliki *standard* devisiasi yang paling kecil dibandingkan dengan dua kecepatan lainnya.

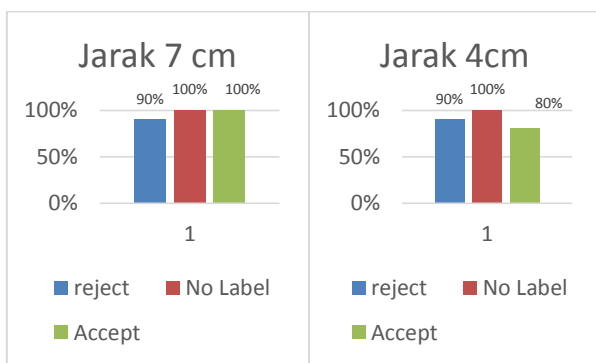
4.3 Pengujian Terhadap Jarak Obyek Dengan Kamera

Pengujian Terhadap Jarak Obyek Dengan Kamera Sistem deteksi penempatan label dalam proses penangkapan citra label produk mengandalkan kamera. Pada kamera jarak obyek sangat mempengaruhi hasil tangkapan citra, hal tersebut juga mempengaruhi proses deteksi label. Pada sistem ini untuk pendeteksian label telah dibuat kondisi deteksi tetap dengan meletakkan kamera pada samping konveyor untuk mendeteksi label produk. Kondisi deteksi dibuat tetap dikarenakan label yang dideteksi bertempat di bawah dari packaging produk *lip cream*. Pengujian yang dilakukan diantaranya dibagi menjadi dua kondisi yaitu pengujian deteksi dengan jarak 7cm dari kamera, dan dengan jarak 4 cm dari kamera.



Gambar 9 Obyek yang dideteksi

Hasil dari pengujian dengan jarak 4 cm dengan kamera deteksi penempatan label dengan kondisi *Reject* sebanyak sepuluh kali, dari sepuluh kali percobaan tersebut menghasilkan akurasi deteksi 80 %, karena dari sepuluh percobaan deteksi terdapat 2 deteksi salah. Jarak 4cm fokus dari lensa kamera kurang meskipun akurasi pendeteksian 80% namun pembacaan penempatan labelnya juga membutuhkan waktu yang sedikit lama dibandingkan dengan jarak obyek 7 cm.



Gambar 10 Bagan Hasil Pengujian Jarak

Hasil keseluruhan pengujian terhadap jarak obyek dari kamera dapat dilihat pada gambar 4.6 dan gambar 4.7 bahwa jarak *bottom label* ke kamera juga berpengaruh terhadap waktu dan juga hasil deteksi. Jarak 4 cm yang terlalu dekat dengan kamera pembacaan *autofocus* dari kamera memerlukan waktu yang cukup lama dan akurasinya lebih rendah dibandingkan dengan jarak 7 cm.

4.4 Pengujian kecepatan 13.08 cm/s dengan jarak 7cm

Pada Tabel 3 dapat dilihat hasil dari klasifikasi *accept* mendapatkan 30 *true positive* (TP), 0 *false positive* (FP), dan 0 *false negative* (FN) yang menghasilkan *average precision* (AP) sebesar 100%. Sedangkan untuk hasil klasifikasi *reject* mendapatkan 29 *true positive* (TP), 0 *false positif* (FP), dan 1 *true negative* (TN) menghasilkan *average precision* sebesar 96.66%. *No label* mendapatkan hasil klasifikasi 26 *true positive* (TP), 4 *false positive* (FP) dan 0 *false negative* (FN) menghasilkan *average precision* 86.66%. Hasil dari percobaan menggunakan kecepatan konveyor 2 menghasilkan *average precision* sebesar 94.44% yang memiliki nilai yang tinggi dibandingkan kecepatan 3 dan lebih rendah dari percobaan kecepatan 4. Namun demikian meskipun nilai *average precision* nya cukup tinggi namun kecepatan dari konveyor kurang mengejar production per menit dari line VSN03 yang cukup tinggi

Moch Deny Triatmaja : Sistem Inspeksi Visual Penempatan ...

TABEL III : Hasil Pengujian Kecepatan 13.08 cm/s

Deteksi Bottom Label		Hasil
Accept	TP	30
	TN	0
	FP	0
	FN	0
AP		100%
Reject	TP	29
	TN	1
	FP	0
	FN	0
AP		96.66%
No Label	TP	26
	TN	0
	FP	4
	FN	0
AP		86.66%
Precision		0.955
Recall		1
mAP		94.44%

4.5 Pengujian Kecepatan 16.75 cm/s Dengan Jarak 7cm

TABEL III : Hasil Pengujian Kecepatan 16.75 cm/s

Deteksi Bottom Label		Hasil
Accept	TP	27
	TN	0
	FP	0
	FN	3
AP		90%
Reject	TP	30
	TN	0
	FP	0
	FN	0
AP		100%
No Label	TP	26
	TN	0
	FP	4
	FN	0
AP		86.66%
Precision		0.954
Recall		0.965
mAP		92.22%

Pada Tabel 4 dapat dilihat hasil dari klasifikasi *accept* mendapatkan 27 *true positive* (TP), 0 *false positive* (FP), dan 3 *false negative* (FN) yang menghasilkan *average precision* (AP) sebesar 90%. Sedangkan untuk hasil klasifikasi *reject* mendapatkan 30 *true positive* (TP), 0 *false positif* (FP), dan 0 *false negative* (FN) menghasilkan *average precision* sebesar 100%. *No label* mendapatkan hasil klasifikasi 26 *true positive* (TP), 4 *false positive* (FP) dan 0 *false negative* (FN) menghasilkan *average precision* 86.66%. Hasil dari percobaan menggunakan kecepatan konveyor 4 menghasilkan *average precision* sebesar 92.22% yang memiliki nilai yang paling rendah dari percobaan menggunakan kecepatan 13.08 cm/s dan kecepatan 21,99 cm/s.

4.6 Pengujian Kecepatan 21.99 cm/s Dengan Jarak 7cm



TABEL III : Hasil Pengujian Kecepatan 16.75 cm/s

Deteksi Bottom Label		Hasil
Accept	TP	30
	TN	0
	FP	0
	FN	0
	AP	100%
Reject	TP	29
	TN	0
	FP	1
	FN	0
	AP	96.66%
No Label	TP	28
	TN	0
	FP	2
	FN	0
	AP	93.33%
Precision		0.966
Recall		1
mAP		96.66%

Pada Tabel 4.23 dapat dilihat hasil dari klasifikasi *accept* mendapatkan 30 *true positive* (TP), 0 *false positive* (FP), dan 0 *false negative* (FN) yang menghasilkan *average precision* (AP) sebesar 100%. Sedangkan untuk hasil klasifikasi *reject* mendapatkan 29 *true positive* (TP), 1 *false positif* (FP), dan 0 *false negative* (FN) menghasilkan *average precision* sebesar 96.66%. *No label* mendapatkan hasil klasifikasi 28 *true positive* (TP), 2 *false positive* (FP) dan 0 *false negative* (FN) menghasilkan *average precision* 93.33%. Hasil dari percobaan menggunakan kecepatan konveyor 4 menghasilkan *average precision* sebesar 96.66% yang memiliki nilai yang lebih tinggi dari percobaan menggunakan kecepatan 2 dan kecepatan 3.

V. KESIMPULAN

Berdasarkan pengujian dan analisa yang telah dilakukan pada sistem inspeksi visual penempatan label, sistem berhasil mendeteksi penempatan label secara otomatis dengan menggunakan metode *deep learning you only look once* (YOLOv5). Deteksi optimal apabila sistem tersebut berada dalam ruangan produksi dengan pencahayaan yang stabil yaitu berkisar antara 250 – 300 lumen. Sistem dapat mendeteksi dan mengklasifikasi tiga kategori penempatan label yaitu *Accept*, *Reject*, dan *No Label*. Sistem dapat berjalan efektif jika menggunakan kecepatan konveyor 4 dengan jarak *object* dari kamera sejauh 7cm karena mendapatkan *average precision* yang lebih tinggi dibandingkan dengan pengujian yang lainnya.

Hasil dari pengujian kecepatan 13.08 cm/s memiliki rata rata presisi di setiap kategori label 94.44%, Kecepatan 16.75 cm/s sebesar 92.22% dan kecepatan 21.99 cm/s sebesar 96.66%. jadi Ketika ingin menggunakan sistem insoeksi visual penempatan label makan konveyor harus di kecepatan 21.99 cm/s. Pada gambar 5.2 merupakan hasil perbandingan persentase deteksi dengan jarak obyek sejauh 7 cm dan 4 cm. Dapat dilihat dari Gambar 4.5 diatas jarak 4cm dan jarak 7cm memiliki nilai rata-rata persentase yang berbeda-beda. Jarak 4 cm memiliki nilai rata-rata 90 % dan jarak 7 cm memiliki nilai rata-rata 93.33%. Jarak 7cm akan menjadi acuan jarak obyek dari kamera dikarenakan memiliki nilai rata-rata persentase yang lebih besar.

Jadi kesimpulan yang dihasilkan sistem dapat mendeteksi dan mengklasifikasi tiga kategori penempatan label yaitu *Accept*, *Reject*, dan *No Label* dengan kecepatan konveyor 4 yang berjalan 21.99 cm/s dan jarak sekitar 7cm dari kamera dan mempunyai *rate acceptance* sebesar 98.88%

REFERENSI

- [1] Stefanus Adhie Nugroho, Nur Kholis, Endryansyah and Farid Baskoro, "Rancang Bangun Sistem Deteksi Label Kardus Berbasis Model Kecerdasan Buatan YOLO dan EasyOCRserta ESP32-CAM," 2022
- [2] E. R. Setyaningsih and M. S. Edy, "YOLOv4 dan Mask R-CNN Untuk Deteksi Kerusakan Pada Karung Komoditi," *Teknika*, vol. 11, no. 1, pp. 45–52, Mar. 2022, doi: 10.34148/teknika.v11i1.419.
- [3] Q. Aini, N. Lutfiani, H. Kusumah, and M. S. Zahran, "Deteksi dan Pengenalan Objek Dengan Model Machine Learning: Model Yolo," *CESS J. Comput. Eng. Syst. Sci.*, vol. 6, no. 2, p. 192, Jul. 2021, doi: 10.24114/cess.v6i2.25840.
- [4] D. R. Wilson and T. R. Martinez, "The need for small learning rates on large problems," in *IJCNN'01. International Joint Conference on Neural Networks. Proceedings (Cat. No. 01CH37222)*, vol. 1. IEEE, 2001, pp. 115–119.
- [5] P. M. Radiuk, "Impact of training set batch size on the performance of convolutional neural networks for diverse datasets," 2017.
- [6] Baudet, N., Maire, J.C., and Pillet, M., 2012. *The Visual Inspection of Product Surface. Food Quality and Preference*.
- [7] Ramos, M., Valdes, A., and Garrigos, M.A., 2016. *Packaging for Drinks. Reference Modul in Food Science: Analytical Chemistry, Nutrition & Food Science. University of Alicante, Spain*.
- [8] Wang, J., Fu, P., and Gao, R.X., 2019. *Machine Vision Intelligence for Product Defect Inspection Based on Deep Learning and Hough Transform. Journal of Manufacturing System*, 51: 52-60.
- [9] Liang, Q., Zhu, W., Sun, W., Yu, Z., Wang, Y., and Zhang, D., 2019. *In Line Inspection Solution for Codes on Complex Background for The Plastic Container Industry, Journal of Measurement*, 148, 106965.
- [10] Perez, H., Joseph, H., Tah, M. and Mosavi, A., 2019. *Deep Learning for Detecting Building Defects Using Convolutional Neural Network, MDPI Journal*.
- [11] Panella, F., Boehm, J., Loo, Y., Kaushik, A., and Gonzales, D., 2018. *Deep Learning and Image Processing for Automated Crack Detection and Defect Measurement in Underground Structures, Journal of Remote Sensing and Spatial Information Sciences, Volume: XLII-2*.
- [12] Costa, A.Z., Hugo, E.H.F., and Fracarolli, J.A., 2020. *Computer Vision Based Detection of External Defects on Tomatoes using Deep Learning, Journal of Biosystem Engineering*, 190: 131-144.
- [13] Rokhana, R., Priambodo, J., Karlita, T., Sunarya, I.M.G., Yuniarto, E.M., Purnama, I.K.E., dan Purnomo, M.H., 2019. *Convolutional Neural Network untuk Pendektaksian Patah Tulang Femur pada Citra Ultrasonik B-Mode, JNTETI, Volume 8, No.1*.
- [14] LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P., 1998. *Gradient-Based Learning Applied to Document Recognition, Proceeding of the IEEE, Vol.86, No.11, 2278-2324*.
- [15] Simonyan, K., and Zisserman, A., 2014. *Very Deep Convolutional Neural Network for Large-Scale Image Recognition, ArXiv140091556Cs*.
- [16] Zhang, X., Zhou, X., Lin, M., and Sun, J., 2018. *ShuffleNet: An Extremely Efficient Convolutional Neural Network for Mobile Devices, CVF Conference on Computer Vision and Pattern Recognition, IEEE*.
- [17] Rezende, E., Ruppert, G., Carvalho, T., Ramos, F., and Geus, P.D., 2017. *Malicious Software Classification using Transfer Learning of ResNet-50 Deep Neural Network, Conference on Machine Learning & Applications, 16th*.
- [18] Wang, J., May, Y., Zhang, I., Gao, R.X., and Wu, D., 2018. *Deep Learning for Smart Manufacturing: Methods and Applications, Journal of Manufacturing System*, 48: 144-16.
- [19] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A., 2015. *Going Deeper with Convolutional, IEEE, 978-1-4673-6964-0/15*.
- [20] He, K., Zhang, X., Ren, S., and Sun, J. *Deep Residual Learning for Image Recognition. Conference on Computer Vision and Pattern Recognition, IEEE*.