
YOLOv4 dan Mask R-CNN Untuk Deteksi Kerusakan Pada Karung Komoditi

Eka Rahayu Setyaningsih^{1*}, Muhamad Sarwo Edy²

^{1,2}Program Studi S2 Teknologi Informasi, Institut Sains dan Teknologi Terpadu Surabaya, Jawa Timur
Email: ^{1*}eka@stts.edu, ²sarwoedy152@gmail.com

(Naskah masuk: 22 Okt 2021, direvisi: 16 Feb 2022, 08 Mar 2022, diterima: 11 Mar 2022)

Abstrak

Industri pertanian memiliki berbagai cara dalam memproses produk mulai dari proses bertani hingga ke pemasaran produk. Salah satu proses yang terlibat di dalamnya adalah penyimpanan produk yang belum dipasarkan melalui beberapa media, contohnya karung untuk menyimpan komoditi. Karung memiliki ketahanan tertentu namun juga dapat memiliki kerusakan, baik ketika digunakan secara berlebihan ataupun ketika berada dalam kondisi penyimpanan secara bertumpuk. Guna membantu mengobservasi keadaan karung komoditi pada suatu penyimpanan, penelitian ini menggagas konsep program deteksi objek berupa karung dan lubang kerusakan yang ada pada suatu *frame* gambar. Penelitian ini mengusung dua metode sebagai perbandingan berupa *You Only Look Once* (YOLO) versi 4 dan *Mask R-CNN*. YOLOv4 merupakan metode deteksi objek yang menekankan konsep *one-stage object detection* dengan keunggulan pada waktu pemrosesan dan akurasi yang dihasilkan. Keluaran YOLOv4 berupa *bounding box* yang teridentifikasi pada objek. *Mask R-CNN* menggunakan konsep *two-stage object detection* dimana selain identifikasi objek dengan menghasilkan *bounding box*, terdapat *masking* terhadap objek yang merepresentasikan segmentasi terhadap objek yang dideteksi. *Dataset* yang digunakan merupakan hasil akuisisi kamera dengan jumlah yang disamakan pada masing-masing algoritma senilai 700 data *training*, 100 data validasi. Kelas objek yang dideteksi pada penelitian ini yaitu karung dan lubang pada karung. Pengujian terhadap kedua algoritma tersebut dilakukan terhadap sejumlah 20 data uji dengan perbandingan terhadap kalkulasi secara manual oleh mata manusia. Hasilnya didapatkan bahwa model dari YOLOv4 dapat memberikan performa lebih baik dengan akurasi 96,8%, sedangkan model *Mask R-CNN* mengalami kinerja yang kurang dapat diandalkan dengan akurasi 65,78% pada data uji yang sama.

Kata Kunci: Karung, Lubang Karung, Pertanian, Deteksi Objek, YOLOv4, Mask R-CNN, Pengolahan Citra Digital, *Computer Vision*

YOLOv4 and Mask R-CNN for Defect Detection On Commodity Sack

Abstract

The agricultural industry has various ways of processing products varying from farming processes to product marketing. One of the processes involved is storing products that have not been marketed through several media, for example sacks to store commodity. Sacks have a certain resistance but can also be damaged both when used excessively and when stored in stacks. In order to help observe the condition of commodity sacks in a warehouse, this study initiated the concept of an object detection program to perform detection of sacks and its holes in a picture frame. This study uses two methods as a comparison of You Only Look Once (YOLO) version 4 and Mask R-CNN. YOLOv4 is an object detection method which is a one-stage object detection where in contrast to the resulting object, there is a masking of the object that represents the segmentation of the detected object. The dataset used is the result of camera acquisition with an equal number of each in accordance with 700 training data and 100 data validation. The class of objects detected in this study consist of two classes as the sacks itself and holes in sacks. Testing on each algorithm was carried out on a number of 20 test data with comparisons to manual calculation. The results show that the model from YOLOv4 can perform better accuracy of 96.8%, while the Mask R-CNN less reliable performance with an accuracy of 65.78% on the same data.

Keywords: *Sacks, Sack's Hole, Agricultural, Object Detection, YOLOv4, Mask R-CNN, Digital Image Processing, Computer Vision*

I. PENDAHULUAN

Industri pertanian memiliki berbagai cara dalam memproses produk mulai dari proses bertani hingga ke pemasaran produk. Salah satu proses yang terlibat di dalamnya adalah menyimpan produk yang belum dipasarkan melalui beberapa media, contohnya karung untuk menyimpan beras. Bahan pembuat karung untuk menyimpan beras dapat beraneka ragam yang disesuaikan dengan kapasitas penyimpanan hingga tingkat maksimal [1]. Namun dalam beberapa kasus, pelaku industri pertanian dapat memanfaatkan penyimpanan pada karung beras yang melebihi batas maksimal. Lebih lagi cara pelaku industri mengelola produk yang mereka simpan dalam bentuk karung beras menggunakan cara dengan ditumpuk untuk mengoptimalkan ruang kapasitas pada suatu tempat misalnya gudang. Dengan cara mengelola karung beras seperti itu, menimbulkan kerugian tersendiri terhadap kemasan karung berupa kerusakan seperti robekan, lubang, dan sebagainya. Terlebih jika ada faktor lain seperti hewan pengerat yang juga dapat merusak kemasan karung ketika berada di dalam gudang penyimpanan. Hal tersebut membuat pihak pengelola harus rajin melakukan pemeriksaan terhadap setiap tumpukan karung beras dan melakukan tindakan terhadap karung dengan kemasan yang memiliki kerusakan sehingga mempengaruhi kadar air, suhu, dan kelembapan relatifnya [2].

Dengan adanya perkembangan teknologi saat ini khususnya di bidang kecerdasan buatan, masalah pada industri pertanian seperti gambaran di atas dapat dibantu dengan pendekatan teknologi yang sesuai. Salah satunya dalam bentuk deteksi objek dimana objek yang diharapkan dalam berupa lubang kerusakan pada kemasan karung beras. Deteksi objek dilakukan dengan memanfaatkan gambar dari kamera dan menganalisa *frame* dengan suatu metode untuk menemukan objek yang diharapkan. Sederhananya deteksi objek bekerja dengan mendeteksi instansi suatu objek berdasarkan kelas yang ada pada gambar. Pembuatan teknologi deteksi objek dilakukan dengan implementasi pembelajaran mesin pada salah satu sub-konsep yaitu *Deep Learning* [3]. Pada konsep *deep learning*, komputer akan berusaha memetakan suatu fitur dari kumpulan gambar yang melalui proses pelatihan untuk mendapatkan pola tertentu misalnya kanal warna, ukuran objek dan sebagainya guna membantu skema deteksi objek.

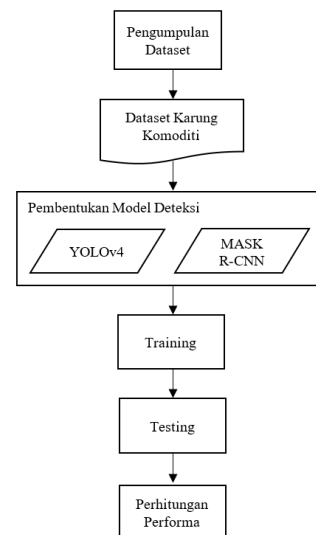
Teknologi yang berkembang untuk deteksi objek saat ini memiliki berbagai acuan parameter untuk dijalankan meliputi akurasi deteksi, kecepatan deteksi, dan ukuran model yang dihasilkan [4]. *You Only Look Once version 4* (YOLOv4) [5][6] dan *Mask R-CNN* [7] merupakan algoritma deteksi objek yang populer dengan masing-masing memiliki kelebihan dan kekurangan. Kedua algoritma tersebut sama-sama harus diimplementasikan dengan proses pelatihan atau *training* terhadap data gambar dengan kelas objek yang diharapkan. Komputasi komputer terhadap dua algoritma tersebut dapat berjalan memanfaatkan piranti pemrosesan utama komputer atau dengan bantuan kartu grafis guna sama-sama menghasilkan model untuk deteksi objek.

II. TINJAUAN PUSTAKA

Peneliti juga melakukan studi literatur pada masing-masing algoritma untuk mengimplementasikan deteksi objek. Hatab, et.al menggunakan jaringan YOLO untuk mendeteksi cacat pada permukaan baja [8]. Dalam penelitian yang dimaksud cacat pada baja berupa goresan dan tambalan. *Dataset* yang digunakan berisi 6 jenis cacat permukaan seperti digulung, tambalan, permukaan tidak beraturan, permukaan berlubang, inklusi dan goresan sejumlah masing-masing 300 data gambar. *Backbone* yang digunakan yaitu *Darknet-53* yang juga implementasikan oleh Xin, et.al untuk deteksi cacat pada papan sirkuit dipilih sebagai model awal pembobotan [9]. Berdasarkan laporan yang dibuat melewati beberapa skenario YOLO mampu mencapai nilai mAP sebesar 70.66% dalam mendeteksi cacat pada permukaan baja. Disisi lain pada deteksi kerusakan papan sirkuit berupa lubang hilang, gigitan tikus, sirkuit terbuka, korsleting, sirkuit dan tembaga tiruan mendapatkan nilai mAP sebesar 96.88%.

Penelitian terkait deteksi objek menggunakan *Mask R-CNN* juga dilakukan Hao Wu, et.al [10] dan Gou, et.al [11] menggunakan *backbone resnet-101*. Wu yang mengimplementasikan deteksi objek sambungan solder pada papan sirkuit digunakan *pretrained model* dari *mscoco dataset*. Data pelatihan terdiri atas gambar komponen yang salah, komponen benar, komponen diganti, komponen terlepas dan tanpa komponen. Performa *Mask R-CNN* dalam deteksi sambungan solder mendapatkan nilai 100%. Sedangkan Jun-Nian Gou yang melakukan deteksi objek berupa kerusakan hasil cetak pada gambar hasil *computer tomography* atau CT. Data gambar yang digunakan terdiri atas beberapa jenis kerusakan seperti gelembung, kerak dan patahan. Adapun faktor *noise* yang terdapat pada cetakan biasanya mempunyai kontras rendah, derau garis keabu-abuan dan warna kabur pada tepi citra yang mempersulit identifikasi objek pada cetakan. Pengujian terhadap data gambar cetakan CT mencapai nilai mAP 98% untuk kemampuan deteksi objeknya.

III. METODOLOGI PENELITIAN



Gambar 1. Metodologi Penelitian

Metode penelitian yang digunakan terdiri dari 5 tahapan yaitu pengumpulan *dataset*, pembentukan model deteksi, pelatihan model deteksi, pengujian dan evaluasi model. Berdasarkan 5 tahapan itu peneliti akan memperjelas alur metodologi penelitian yang digunakan melalui Gambar 1.

A. Pengumpulan *Dataset*

Penelitian ini menggunakan *dataset* yang diambil melalui akuisisi kamera digital. Jumlah *dataset* yang digunakan sebanyak 800 gambar dan terbagi menjadi 700 gambar untuk data *training* dan 100 gambar untuk data *testing*. *Dataset* ini pada dasarnya terdapat dua jenis karung yaitu karung tebal dan karung tipis. Dimana Gambar 2 merupakan contoh karung tebal dan Gambar 3 merupakan contoh karung tipis. Pada suatu *frame* data dapat memuat banyak objek dari kedua kelas yang digagas yaitu karung dan lubang.



Gambar 2. Contoh Data Gambar Karung Tebal



Gambar 3. Contoh Data Gambar Karung Tipis

Pada Gambar 2 dan 3 merupakan jenis karung dengan bahan plastik dengan ukuran lebar 75 cm dan panjang 115 cm. Variasi ukuran lubang yang dikumpulkan untuk dideteksi berkisar 1-5 cm. Data diambil menggunakan kamera digital yang menghasilkan ukuran 2000 x 1333 piksel dengan ukuran berkas berkisar 2-4 MB. Besarnya ukuran data gambar yang dihasilkan dari proses akuisisi akan diproses dengan mengecilkan setiap ukuran berkas menjadi kisaran 600 hingga 800 KB namun tetap pada ukuran 2000 x 1333 piksel. Setelah gambar selesai diproses menjadi ukuran yang lebih kecil, maka dilakukan pelabelan pada gambar. Proses pelabelan untuk YOLOv4 dan *Mask R-CNN* mempunyai teknik dan format hasil pelabelan yang berbeda. Pelabelan pada YOLOv4 peneliti menggunakan *labelimg* dimana cara pelabelannya hanya menandai objek dengan pembatas berbentuk kotak sesuai lebar dan tinggi objek. Bentuk format data label pada YOLOv4 berekstensi *.txt*. berbeda dari YOLOv4, algoritma *Mask R-CNN* mempunyai format data berbentuk JSON yang peneliti buat dengan bantuan aplikasi *Labelme*. Cara kerja pelabelan objek untuk *Mask R-CNN* ini berbentuk *polygon* sehingga mengikuti bentuk objek.

B. Pembentukan Model Deteksi

Pada tahapan ini dilakukan konfigurasi berbagai parameter yang digunakan dari algoritma YOLOv4 dan *Mask R-CNN*. Adapun pada tahapan ini juga peneliti akan memaparkan pemilihan *pre-trained model* dan *backbone* yang digunakan pada masing-masing algoritma.

(1) YOLOv4

Konfigurasi parameter yang digunakan untuk pembentukan model YOLOv4 pada Tabel 1 akan disesuaikan dengan jumlah kelas objek yang akan dideteksi dan kemampuan kartu grafis dalam menangani proses komputasi yang dimiliki oleh peneliti. Adapun parameter yang dilibatkan seperti *batch* dengan nilai 64 dan *subdivision* dengan nilai 16. Parameter tersebut diartikan bahwa dalam satu proses *step* mampu melakukan komputasi terhadap 64 data dengan pembagian 16 data.

Tabel 1. Konfigurasi Model YOLOv4

Parameter	Nilai
<i>Batch</i>	64
<i>Subdivision</i>	16
<i>Width</i>	416
<i>Height</i>	416
<i>Channels</i>	3
<i>Momentum</i>	0,949
<i>Decay</i>	0,0005
<i>Angle</i>	0
<i>Saturation</i>	1,5
<i>Exposure</i>	1,5
<i>Hue</i>	0,1
<i>Learning rate</i>	0,001
<i>Burn_in</i>	1000
<i>Max_batches</i>	6000
<i>Policy</i>	Steps
<i>Steps</i>	4800; 5400
<i>Scales</i>	0,1; 0,1

Beberapa pemrosesan citra terjadi pada konfigurasi parameter diatas seperti pengubahan saturasi warna, nilai *exposure* dan *hue*. Merujuk pada penelitian yang dilakukan oleh Matija Buric, et.al [12] ukuran gambar yang diberikan pada model Yolo berukuran 416x416 piksel. Pada proses pembentukan model, peneliti menetapkan parameter ukuran piksel gambar dialokasikan 416x416 piksel. Konfigurasi ini merupakan ukuran piksel yang sesuai bagi kartu grafis yang mempunyai memori berkapasitas 8 GB untuk melakukan proses komputasi algoritma YOLOv4.

(2) *Mask R-CNN*

Konfigurasi parameter yang digunakan untuk pembentukan model *Mask R-CNN* dalam penelitian ini akan mengacu pada kemampuan komputasi dan besaran memori pada kartu grafis. Hal ini dilakukan agar kartu grafis mampu memproses gambar sebab beban komputasi dengan *Mask R-CNN* ini memerlukan *resource* yang tinggi. Pada penelitian

yang sama pula Matija Buric, et.al [12] menggunakan model Mask R-CNN untuk mendeteksi objek bola. Matija menerapkan ukuran 1024x1024 piksel sebagai gambar masukan yang diberikan kepada model. Selanjutnya pembentukan model dilakukan oleh peneliti dengan konfigurasi parameter yang dijelaskan pada tabel di bawah ini.

Tabel 2. Konfigurasi Model Mask R-CNN

Parameter	Nilai
<i>Backbone</i>	Resnet101
<i>Backbone_strides</i>	[4,8,16,32,64]
<i>Batch_size</i>	1
<i>Detection_min_confidence</i>	0,9
<i>Detection_nms_threshold</i>	0,3
<i>FPN_fc_layer_size</i>	1024
<i>Image_per_gpu</i>	1
<i>Image_max_dim</i>	1024
<i>Image_min_dim</i>	800
<i>Learning_rate</i>	0,001
<i>Num_classes</i>	3
<i>RPN_nms_threshold</i>	0,7
<i>Steps_per_epoch</i>	100
<i>Validation_steps</i>	50
<i>Weight_decay</i>	0,0001

C. Pelatihan Model Deteksi

Pada tahapan pelatihan awal, masing-masing algoritma menggunakan *pre-trained model* dari mscoco. Peneliti memilih model awal dari mscoco karena memiliki jumlah data yang besar dengan 80 kelas objek yang baik untuk memulai pembobotan awal. Pada algoritma YOLOv4 akan digunakan *backbone* dengan varian CSP-Darknet53. YOLOv4 akan menggunakan CSP-Darknet53 untuk melakukan ekstraksi fitur. Gambar yang akan masuk ke jaringan ekstraksi fitur itu akan dirubah ukurannya menjadi 416x416 piksel sesuai dengan konfigurasi model. Adapun penelitian yang dilakukan oleh Malbog [13] untuk ekstraksi fiturnya digunakan Resnet-101. Meskipun pelatihan model Mask R-CNN banyak memiliki varian *backbone* yang dapat digunakan namun pada penelitian ini dipilih Resnet-101.

Hasil berkas pembobotan dari kedua algoritma pada penelitian ini disimpan dengan cara dan format yang berbeda. YOLOv4 akan menyimpan hasil pembobotan setiap 1000 *steps* dengan ekstensi *.weight*. Mask R-CNN akan menyimpan hasil training dari setiap *epoch* dengan ekstensi *.h5*.

D. Pengujian Model

Pada tahapan ini dilakukan pengujian dilakukan dengan memilih model yang memberikan akurasi terbaik saat akhir proses pelatihan. Setelah didapatkan model yang memiliki performa yang baik dan stabil akan dimuat kembali pada proses pengujian. Selanjutnya model akan diberikan gambar masukan untuk memulai proses pengujiannya.

Pengujian yang dilakukan pada algoritma YOLOv4 akan memberikan hasil deteksi pada pada gambar berupa kotak penanda dan label objek berdasarkan kelasnya. Kemudian pengujian pada Mask R-CNN akan memberikan hasil deteksi

berupa kotak penanda beserta lapisan *masking* yang melapisi objek sesuai kelasnya. Pada akhir proses pengujian deteksi pada gambar, masing-masing algoritma memberikan hasil perhitungan objek yang terdeteksi sesuai kelasnya.

E. Evaluasi Model

Merujuk pada penelitian yang dilakukan oleh Selorm Garfo, et.al [14] tentang deteksi cacat pada produk cetak tiga dimensi dan beton bahwa dalam penelitian itu hanya terdapat 2 kelas objek yang dideteksi. Selorm Garfo kemudian menyajikan hasil evaluasi model terhadap data pelatihan berupa grafik nilai *loss* menggunakan *tensorboard*. Penelitian lain yang dilakukan oleh Hansen Liu, et.al [15] tentang deteksi pesawat tanpa awak menyajikan hasil evaluasi model terhadap data pelatihan berupa nilai akhir mAP. Maka peneliti menyajikan hasil evaluasi model terhadap data pelatihan YOLOv4 berupa visualisasi grafik mAP sedangkan pada Mask R-CNN akan disajikan berupa grafik nilai *loss*. Pada tahapan ini juga dilakukan proses perhitungan performa dari tahapan pengujian pada masing-masing algoritma. Peneliti menggunakan metode evaluasi *mean average precision* atau mAP. Metode evaluasi tersebut dipilih untuk membandingkan jumlah objek yang sebenarnya dan jumlah objek yang benar terdeteksi.

$$Precision = \frac{TP}{TP+FP} \quad (1)$$

Disajikan pada persamaan (1) adalah rumus perhitungan dari *precision*. Diketahui bahwa *precision* dihitung dengan cara membagi nilai prediksi yang benar sebagai *True Positive* (TP) dengan total dari *True Positive* (TP) dan nilai prediksi yang salah sebagai *False Positive* (FP).

$$Recall = \frac{TP}{TP+FN} \quad (2)$$

Disajikan pada persamaan (2) adalah rumus perhitungan dari *recall*. Diketahui bahwa *recall* dihitung dengan cara membagi antara nilai prediksi yang benar sebagai *True Positive* (TP) dengan total *True Positive* (TP) dan nilai objek yang gagal terdeteksi sebagai *False Negative* (FN). Pada tahapan ini dilakukan pengujian dilakukan dengan memilih model yang memberikan akurasi terbaik saat akhir proses pelatihan.

IV. HASIL DAN PEMBAHASAN

Pada tahapan ini akan dijelaskan mengenai hasil pembentukan *dataset*, inisialisasi perangkat yang digunakan dalam penelitian, hasil pelatihan model deteksi, hasil uji coba model dan evaluasi model. Peneliti akan menyajikan dalam beberapa poin di bawah ini.

A. Hasil Pembentukan Dataset

Contoh hasil anotasi untuk *dataset* format YOLO berupa kotak penanda di sekeliling objek. Hasil anotasi akan

menentukan kualitas data pelatihan dan performa model dalam mendeteksi objek.



Gambar 4. Contoh Hasil Anotasi Format YOLO

Dari Gambar 4 diketahui bahwa dalam satu *frame* dapat memuat banyak *bouding box*. Setiap *bouding box* memiliki titik koordinat berupa $xmin$, $xmax$, $ymin$, $ymax$ yang disimpan dalam berkas *.txt*. Kemudian pada pembentukan *dataset* untuk *Mask R-CNN* teknik anotasinya berbentuk garis *polygon* yang mengelilingi objek.



Gambar 5. Contoh Hasil Anotasi Format COCO

Anotasi terhadap *dataset* yang disiapkan untuk *Mask R-CNN* pada Gambar 5 di atas memiliki ekstensi berupa *json* (*javascript object notation*) dengan menggunakan konsep format pada *COCO dataset*. Konsep anotasi ini merupakan kumpulan titik yang mengelilingi objek dan membentuk sebagai lapisan di area objek.

B. Inisialisasi Perangkat

Pada penelitian ini, peneliti melakukan serangkaian tahapan penelitian dengan spesifikasi perangkat keras dan perangkat lunak sehingga bisa melakukan implementasi. Perangkat keras dan lunak yang dipaparkan peneliti meliputi spesifikasi CPU, sistem operasi, dan *library*.

(1) Spesifikasi Perangkat Keras

Perangkat keras berikut didasarkan pada kebutuhan saat melakukan proses *training* dan proses menjalankan program

dengan acuan model YOLOv4 dan *Mask R-CNN*. Peneliti berhasil menyelesaikan seluruh proses penelitian dan berjalan lancar dengan spesifikasi pada Tabel 3 berikut:

Tabel 3. Alokasi Spesifikasi Perangkat Keras

Komponen	Spesifikasi
CPU	AMD Ryzen 5 3600
Kartu Grafis	NVIDIA RTX 2070 8 GB

Dengan spesifikasi CPU pada tabel diatas, memiliki 6 *core* dan 12 *thread*. Lalu untuk kartu grafis yang digunakan varian memori 8 GB. Pada saat proses training untuk memperbesar alokasi memorinya dilakukan *swap* pada media penyimpanan SSD.

(2) Spesifikasi Perangkat Lunak

Perangkat Lunak berikut didasarkan pada kebutuhan saat melakukan proses training dan menjalankan program baik pada YOLOv4 dan *Mask R-CNN*. Perangkat lunak yang dipaparkan peneliti pada saat penelitian dapat berjalan baik dengan rincian pada Tabel 4 sebagai berikut:

Tabel 4. Alokasi Spesifikasi Perangkat Lunak

Komponen	Spesifikasi
Sistem Operasi	Linux Ubuntu 20.04
Platform Environment	Anaconda
Library	Maskrcnn by matterport [16], yolo4 by alaxeyAb [17]
Annotation Tools	Labelimg, Labelme
IDE Editor	VSCode

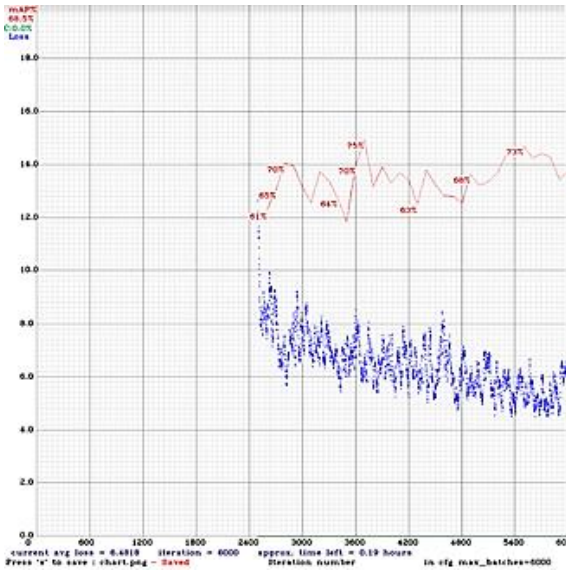
Diketahui peneliti memilih *Anaconda* sebagai *environment* untuk memasang paket-paket pemrograman *python* yang berjalan pada *Ubuntu 20.04*. Dan masing-masing aplikasi pelabelan yang digunakan *library maskrcnn* adalah *labelme* sedangkan untuk YOLOv4 menggunakan *labelimg*.

C. Hasil Pelatihan Evaluasi Model Deteksi

Hasil dari pelatihan model YOLOv4 akan dihasilkan representasi berupa grafik evaluasi dalam bentuk grafik yang diproses pembaharuannya setiap 1000 step. YOLOv4 juga akan menandai berkas dengan hasil pembobotan terbaik dengan nama *best.weight* dan *last.weight* untuk pembobotan pada *step* terakhir dimana proses pelatihan berjalan. Grafik yang dihasilkan oleh aktifitas pelatihan YOLOv4 hanya akan bisa dilihat pada akhir proses pelatihan seperti terlihat pada Gambar 6.

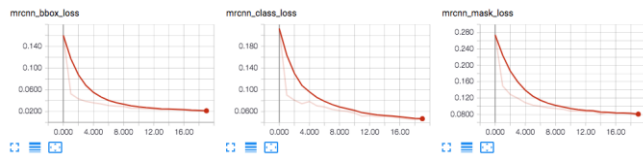
Proses pelatihan YOLOv4 yang diobservasi peneliti diketahui sumber daya yang dipakai lebih menitik beratkan pada penggunaan memori GPU dan tidak menggunakan alokasi sumber daya lain seperti CPU. Penggunaan CPU yang digunakan peneliti jumlah *thread* yang terpakai hanya 20-45% dari 24 *thread* sedangkan untuk kartu grafis aktifitasnya mencapai 88% dari total alokasi memori GPU. Nilai akurasi evaluasi dari proses pelatihan YOLOv4 terhadap objek karung

dan lubang mendapatkan 68,50%. Tahapan pelatihan YOLOv4 selama 28 jam 21 menit.



Gambar 6. Grafik Evaluasi Pelatihan YOLOv4

Pada alokasi sumber daya yang sama pula, pelatihan *Mask R-CNN* dijalankan dengan bantuan *tools* berupa *Tensorflow* dan *Keras*. *Mask R-CNN* menggunakan nilai *loss* untuk melakukan evaluasi pada model pelatihannya. Evaluasi dari proses pelatihan *Mask R-CNN* dihasilkan berupa grafik nilai *loss* pada *tensorboard* pada Gambar 7 berikut.



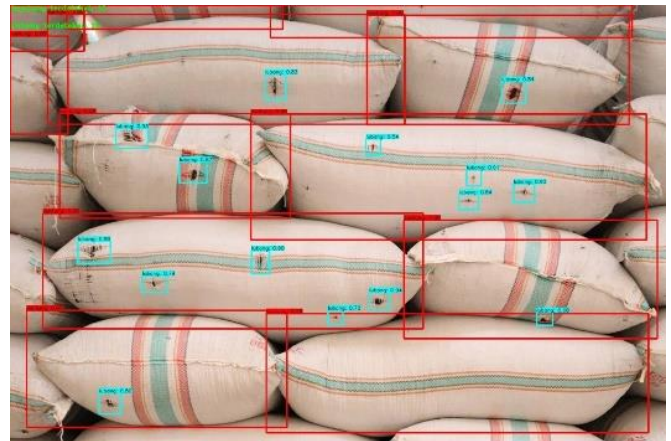
Gambar 7. Grafik Evaluasi Pelatihan pada *Tensorboard*

Berdasarkan masing-masing grafik nilai *loss* yang ditunjukkan pada *tensorboard* diketahui memberikan rata-rata nilai *loss* sebesar 0,1254. Peneliti akan menggunakan model dengan hasil akurasi terbaik dan nilai *loss* terendah yang dihasilkan dari proses pelatihan untuk dimuat pada program pengujian. Diketahui penggunaan CPU terpakai 80-92% dari 24 *thread*, alokasi memori kartu grafis mencapai 92% dari total memori dan konsumsi RAM dengan kapasitas 16 GB mencapai 80-90%. Tercatat pada akhir proses pelatihan *Mask R-CNN* selama 8 jam 42 menit.

D. Uji Coba dan Evaluasi

Tahapan uji coba dilakukan peneliti dengan memuat berkas pembobotan yang dihasilkan dari proses pelatihan dimana yang mempunyai akurasi terbaik. Proses menjalankan program deteksi jauh berbeda dibanding proses pelatihan terkait alokasi sumber daya. Waktu yang dibutuhkan bagi YOLOv4 untuk mendeteksi sebuah gambar berkisar 10-15 detik. Uji coba deteksi dilakukan pada 20 data gambar yang sama untuk

masing-masing algoritma. Berikut merupakan contoh gambar hasil deteksi menggunakan YOLOv4.



Gambar 8. Hasil Deteksi Objek menggunakan YOLOv4

Pada Gambar 8 diketahui bahwa model YOLOv4 berhasil membedakan masing-masing kelas objek dengan representasi *bounding box* yang menandai objek. Adapun penyajian data pengujian dalam tabel diketahui notasi seperti karung (K), lubang (L), *true positive* (TP), *false positive* (FP), *false negative* (FN), *precision* (P), *recall* (R), jumlah aktual (ACT), jumlah prediksi (PRED). Hasil terhadap pengujian menggunakan 20 data uji menggunakan YOLOv4 disajikan pada Tabel 5 berikut:

Tabel 5. Pengujian YOLOv4

Data	ACT		PRED		K		L		
	K	L	K	L	P	R	P	R	
1.png	15	1	5	0	1	0,83	1	1	
2.png	15	3	10	2	1	0,66	1	0,66	
3.png	6	17	4	12	1	0,66	1	0,7	
4.png	8	18	8	18	1	0,75	1	1	
5.png	5	13	4	11	1	0,8	1	0,84	
7.png	5	10	7	6	1	1	1	0,6	
8.png	6	8	4	6	1	0,66	1	0,75	
9.png	14	10	10	11	1	0,57	1	1	
10.png	16	18	7	15	1	0,43	1	0,83	
11.png	20	14	10	12	1	0,5	1	0,85	
12.png	18	16	7	11	1	0,38	1	0,68	
13.png	11	13	4	13	1	0,36	1	1	
14.png	10	17	10	13	1	1	1	0,76	
15.png	9	16	7	10	1	0,77	1	0,62	
16.png	9	16	9	15	1	1	1	0,93	
17.png	9	8	8	5	1	0,88	1	0,62	
18.png	22	50	9	41	1	0,4	1	0,82	
19.png	8	9	2	10	1	0,28	0,8	0,9	
20.png	10	28	7	21	1	0,7	100	0,75	
					AP	95,0	61,0	98,0	79,0
						%	%	%	%
					mAP	96,5			
						%			

Berdasarkan tabel pengujian YOLOv4 sebanyak 20 data gambar dengan masing-masing kelas. Mendapatkan *average precision* sebesar 95,0% untuk objek karung dan 98,6% untuk objek lubang sehingga mendapatkan nilai *mean average precision* tercapai 96,8%. Apabila dianalisis, YOLOv4 dapat mendeteksi objek yang hanya ditemukan sedikit kesalahan deteksi dan ditemukan juga dalam skenario pengujian pada gambar uji tidak mampu mendeteksi jumlah objek sesuai dengan jumlah aktual objek dalam gambar. Selanjutnya pengujian pada gambar dengan *Mask R-CNN* diawali dengan memuat berkas pembobotan dengan nilai *loss* terendah. Hasil keluaran proses uji coba deteksinya menghasilkan *region proposal* berupa *bounding box* dengan empat koordinat disertai lapisan atau *layer* pada area objek yang terdeteksi. Waktu yang dibutuhkan *Mask R-CNN* dalam memproses sebuah gambar berkisar 3-5 detik. Berikut pada Gambar 9 contoh keluaran dari proses uji cobanya.



Gambar 9. Hasil Deteksi Objek menggunakan *Mask R-CNN*

Hasil pengujian deteksi direpresentasikan pada gambar diatas diketahui terdapat kotak penanda berupa *bounding box* beserta lapisan yang berada di area objek. Peneliti juga menyajikan hasil pengujian deteksi terhadap 20 gambar uji menggunakan *Mask R-CNN* pada Tabel 6 berikut:

Tabel 6. Pengujian *Mask R-CNN*

Data	ACT		PRED		K		L	
	K	L	K	L	P	R	P	R
1.png	15	1	14	0	1	0,73	0	0
2.png	15	3	14	4	0,5	0,87	0,75	1
3.png	6	17	10	24	0,7	1	1	0,58
4.png	8	18	10	7	0,85	1	0,8	0,28
5.png	5	13	5	7	0,66	0,66	1	0,25
6.png	6	12	11	46	0	0	0,71	1
7.png	5	10	9	1	0	1	1	0,11
8.png	6	8	6	9	0	0	0,5	0,5
9.png	14	10	17	9	0,42	0,66	0,8	0,44
10.png	16	18	2	4	0	0	1	0,3
11.png	20	14	14	1	0,42	0,85	1	0,5
12.png	18	16	17	3	0,56	0,9	1	0,33
13.png	11	13	10	1	0,66	0,5	1	0,14

14.png	10	17	15	25	0,38	0,71	0,83	1	
15.png	9	16	6	2	1	0,83	1	0,15	
16.png	9	16	16	0	1	1	0	0	
17.png	9	8	17	14	0,56	0,81	0,42	0,5	
18.png	22	50	25	15	0,66	0,94	0,51	0,9	
19.png	8	9	3	23	0,75	1	1	0,66	
20.png	10	28	31	2	0,93	1	1	0,06	
					AP	55,56%	63,15%	76,0%	43,0%
					mAP	65,78%			

Pengujian juga dilakukan pada *Mask R-CNN* terhadap 20 data gambar uji tidak merepresentasikan hasil yang baik dari model *Mask R-CNN*. Hasil deteksinya diamati memiliki rentang yang beragam terbagi menjadi jumlah aktual dan jumlah yang jauh dari aktual. Beragam hasil pengujian dari gambar ditemui kesalahan deteksi maupun gagal deteksi pada objek. Hasil akhir nilai *average precision* untuk objek karung mencapai 55,56% sedangkan untuk objek lubang mencapai 76,00% sehingga dihasilkan nilai *mean average precision* sebesar 65,78%.

V. KESIMPULAN

Berdasarkan hasil uji coba yang dilakukan dapat disimpulkan bahwa YOLOv4 dan *Mask R-CNN* dapat berjalan dan diimplementasikan untuk deteksi objek pada gambar. Serangkaian variasi data pengujian pada data penelitian ini menjelaskan bahwa YOLOv4 memberikan performa deteksi yang lebih baik untuk diterapkan sebagai deteksi objek dibandingkan dengan model *Mask R-CNN*. Diketahui dalam mendeteksi semua kelas objek pada gambar, YOLOv4 mampu mencapai nilai *mean average precision* sebesar 98,60% sedangkan *Mask R-CNN* memberikan performa sebesar 65,78%. Dalam beberapa skenario gambar pengujian yang beragam dari sudut dan gelap terang, YOLOv4 lebih mampu menangani dalam kemampuan deteksi objek sesuai pada tabel hasil pengujian

Rekomendasi untuk penelitian selanjutnya adalah dengan melakukan data augmentasi diluar proses algoritma untuk memperkaya variasi data dan banyaknya jumlah data untuk proses pelatihan. Perbaikan tahapan *preprocessing* seperti pelabelan dan perbaikan citra dapat dipertimbangkan untuk meningkatkan kualitas data penelitian agar meningkatkan performa deteksi. Besarnya alokasi sumber daya perangkat keras dan lama waktu komputasi yang dibutuhkan kedua algoritma patut mencoba alternatif *framework*, *library* maupun model yang lebih ringan dari segi arsitektur dan beban komputasinya.

REFERENSI

- [1] A. Naufal, A. Azhar, and A. Nugroho, "Analisis Sistem Pemasaran cengkeh (*syzygium aromaticum*) di kecamatan Lhoknga Kabupaten Aceh Besar," *Jurnal*

- Ilmiah Mahasiswa Pertanian*, vol. 3, no. 4, pp. 518–524, 2018.
- [2] W. Dewayani, Arafah, N. Razak, and A. Darmawidah, “Efek Jenis Kemasan Terhadap Kualitas Gabah dan Beras Varietas Cigeulis,” *Jurnal Pengkajian dan Pengembangan Teknologi Pertanian*, vol. 16, no. 1, pp. 8–19, 2013.
- [3] A. Zhang, Z. C. Lipton, M. Li, and A. J. Smola, Dive into Deep Learning. *arXiv preprint*, 2021.
- [4] P. A. Flach and M. Kull, “Precision-Recall-Gain Curves: PR Analysis Done Right,” *Advances in Neural Information Processing Systems*, 2015.
- [5] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, “Yolov4: Optimal Speed and accuracy of object detection,” *arXiv.org*, 23-Apr-2020. [Online]. Available: <https://arxiv.org/abs/2004.10934>. [Accessed: 11-Mar-2022].
- [6] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, “Scaled-YOLOv4: Scaling Cross Stage Partial Network,” *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13029–13038, 2021.
- [7] K. He, G. Gkioxari, P. Dollar, and R. Girshick, “Mask R-CNN,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 2, pp. 386–397, 2020.
- [8] M. Hatab, H. Malekmohamadi, and A. Amira, “Surface defect detection using Yolo Network,” *Advances in Intelligent Systems and Computing*, pp. 505–515, 2020.
- [9] H. Xin, Z. Chen, B. Wang. “PCB Electronic Component Defect Detection Method Based on Improved Yolov4 Algorithm,” *Journal of Physics: Conference Series*, vol. 1827, no. 1, p. 012167, 2021.
- [10] H. Wu, W. Gao, and X. Xu, “Solder joint recognition using mask R-CNN method,” *IEEE Transactions on Components, Packaging and Manufacturing Technology*, vol. 10, no. 3, pp. 525–530, 2020.
- [11] J. N. Gou, X. Y. Wu, and L. Liu, “Detection and Segmentation of Defects in Industrial CT Images Based on Mask R-CNN,” *Journal of Computers*, vol. 31, no. 6, pp. 141–154, 2020.
- [12] M. Buric, M. Pobar, and M. Ivacic-Kos, “Ball detection using Yolo and mask R-CNN,” 2018 International Conference on Computational Science and Computational Intelligence (CSCI), 2018.
- [13] M. A. Malbog, “Mask R-CNN for pedestrian crosswalk detection and instance segmentation,” *2019 IEEE 6th International Conference on Engineering Technologies and Applied Sciences (ICETAS)*, 2019.
- [14] S. Garfo, M. A. Muktadir, and S. Yi, “Defect detection on 3D print products and in concrete structures using image processing and convolution neural network,” *Journal of Mechatronics and Robotics*, vol. 4, no. 1, pp. 74–84, 2020.
- [15] H. Liu, K. Fan, Q. Ouyang, and N. Li, “Real-time small drones detection based on pruned Yolov4,” *Sensors*, vol. 21, no. 10, p. 3374, 2021.
- [16] Matterport, “Matterport/MASK_RCNN: Mask R-CNN for object detection and instance segmentation on Keras and tensorflow,” *GitHub*, 2017. [Online]. Available: https://github.com/matterport/Mask_RCNN. [Accessed: 30-Jul-2021].
- [17] AlexeyAB, “Alexeyab/Darknet: Yolov4 / scaled-yolov4 / yolo - neural networks for object detection (windows and linux version of darknet),” *GitHub*, 2016. [Online]. Available: <https://github.com/AlexeyAB/darknet>. [Accessed: 30-Jul-2021].