

Klasifikasi Benda Organik dan Anorganik Dengan Metode YOLOv3 dan ResNet50

Kevin Reynaldi Tanjung, Liliana, Hans Juwiantho
Program Studi Informatika Fakultas Teknologi Industri Universitas Kristen Petra
Jl. Siwalankerto 121 – 131 Surabaya 60236
Telp. (031) – 2983455, Fax. (031) - 8417658

E-mail: kevinreynaldiwork@gmail.com, lilian@petra.ac.id, hans.juwiantho@petra.ac.id

ABSTRAK

Masih banyak masyarakat yang membuang sampah tidak pada tempatnya. Salah satu penyebabnya adalah masih banyak masyarakat Indonesia yang masih susah dalam memilah benda organik dan anorganik. Oleh karena itu pengenalan akan benda organik dan anorganik sangatlah penting dan diperlukan sesuatu yang mampu membantu dalam pemilahan benda organik dan anorganik. Dengan mengetahui perbedaan benda organik dan anorganik, masyarakat dapat memilah sampah organik dan anorganik.

Metode yang digunakan adalah menggunakan YOLOv3 untuk mendapatkan objek sampah dari sebuah gambar atau video. Objek yang dideteksi akan di potong dan hasilnya akan diolah oleh Convolutional Neural Network dengan model arsitektur ResNet50 untuk klasifikasi. Dalam proses pelatihan YOLOv3 dan ResNet50 dilakukan penyetelan untuk menemukan parameter yang mendapatkan akurasi terbaik.

Penelitian ini akan melakukan klasifikasi objek pada objek sampah dalam gambar atau video. Mean average precision yang didapatkan YOLOv3 adalah 45% dan average loss 91%. Sedangkan untuk ResNet50 terdapat rule of thumb dimana dengan menggunakan input size 416x416 dan semakin rendah angka learning rate dapat meningkatkan akurasi. Ketika digabungkan, ResNet50 mampu meningkatkan akurasi jenis objek yang dideteksi oleh YOLOv3.

Kata Kunci: Organik, Anorganik, YOLOv3, Convolutional Neural Network, ResNet50, Klasifikasi Objek

ABSTRACT

There are still many Indonesian people throw waste in the wrong place. One of the reasons is that there are still many Indonesian people who still find it difficult to sort organic and inorganic objects. Therefore, the introduction of organic and inorganic objects is very important and we need something that can help in sorting organic and inorganic objects. By knowing the difference between organic and inorganic objects, people can sort out organic and inorganic waste.

The methods used are You Only Look Once to get waste objects from an images or videos. The detected object will be cut and the results will be processed by the Convolutional Neural Network with the ResNet50 architectural model for classification. In the YOLOv3 and ResNet50 training process, adjustments are made to find parameters to get best accuracy

This research will classify objects on waste objects in images or videos. The Mean Average Precision obtained by YOLOv3 is 45% and the average loss is 91%. For ResNet50 there is rule of thumb where when using input size 416x416 and the lower the number of learning rates can increase accuracy. When combined, ResNet50

is able to increase the accuracy of the detected object types by YOLOv3.

Keywords: Organic, Inorganic, YOLOv3, Convolutional Neural Network, ResNet50, Classify Object.

1. PENDAHULUAN

Masih banyak masyarakat yang membuang sampah tidak pada tempatnya. Salah satu penyebabnya adalah masih banyak masyarakat Indonesia yang masih susah dalam memilah benda organik dan anorganik. Oleh karena itu pengenalan akan benda organik dan anorganik sangatlah penting dan diperlukan sesuatu yang mampu membantu dalam pemilahan benda organik dan anorganik.

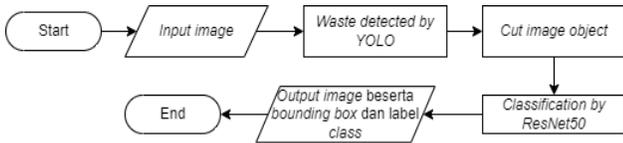
Pada tahun 2019, terdapat penelitian mengenai penerapan model Convolutional Neural Network untuk pemilah sampah dan dari penelitian ini dapat disimpulkan bahwa ResNet50 memiliki performa lebih baik dari metode lain[8]. Tetapi dari penelitian tersebut memiliki kekurangan dimana peneliti tidak membagi sampah plastik menjadi beberapa macam sampah plastik. Pada tahun 2019 pula, ada penelitian mengenai waste object detection and classification dimana peneliti menggunakan Hybrid Training yang menyebabkan akurasi rendah karena merusak kinerja dari klasifikasi gambar[5]. Adapula penelitian mengenai penggunaan Object Detection yang didasari dengan ResNet Network. Hasil yang diberikan sangat memuaskan dimana dengan menggunakan Object Detection dengan dasar ResNet Network memperoleh akurasi yang mencolok dan baik dalam mendeteksi objek[3].

Pada penelitian ini, akan menentukan sampah organik dan anorganik dengan metode Object Detection YOLOv3 sebagai pendeteksi dan menggunakan ResNet50 sebagai pembantu dalam klasifikasi. Jalan kerjanya adalah dengan memasukan sebuah gambar yang berisi beberapa sampah yang di campur dalam satu input, dengan menggunakan YOLOv3 sebagai pendeteksi dan ResNet50 untuk klasifikasi, dapat membantu membedakan sampah mana organik dan sampah mana yang anorganik dengan membuat frame atau kotak. Selain itu dengan menggunakan IP Webcam, dapat menginputkan objek secara real time.

2. LANDASAN TEORI

2.1 Tinjauan Pustaka

Metode yang digunakan dalam penelitian ini terbagi menjadi detection dan classification. Setelah dideteksi, gambar objek akan dilanjutkan kedalam ResNet50 untuk proses classification. Alur dari proses ini dapat dilihat pada flowchart pada Gambar 1.



Gambar 1. Flowchart alur proses detection dan classification

Sistem akan menerima *input* berupa gambar atau video yang berisikan objek benda organik atau anorganik. Video akan dipecah menjadi *frame*. Didalam *frame* atau gambar tersebut terdapat benda yang akan dideteksi lalu dipotong dan dimasukkan kedalam *ResNet50* untuk diklasifikasi. *Output* dari proses ini akan berupa video atau gambar yang disertai dengan *bounding box*, label *class*, dan angka akurasi yang menunjukkan lokasi objek, *class*, dan angka akurasi dari objek dalam gambar atau video.

2.1.1 Object Detection

Object Detection adalah teknik penglihatan komputer yang memungkinkan kita untuk mengidentifikasi dan menemukan objek dalam gambar atau video. Dengan jenis identifikasi dan lokalisasi ini, deteksi objek dapat digunakan untuk menghitung objek dalam suatu pemandangan dan menentukan serta melacak lokasi tepatnya, sambil memberi label secara akurat

Cara kerja dari *Object Detection* adalah *encoder* mengambil gambar sebagai masukan dan menjalankannya melalui serangkaian blok dan lapisan yang belajar mengekstrak fitur statistik yang digunakan untuk mencari dan memberi label objek. Keluaran dari *encoder* kemudian diteruskan ke *decoder*, yang memprediksi kotak pembatas dan label untuk setiap objek. *The Intersection of Union (IoU)* digunakan sebagai batas untuk menentukan apakah hasil yang diprediksi adalah *true positive* atau *false positive*. *IoU* adalah jumlah tumpang tindih antara *bounding box* disekitar objek yang di prediksi dan *bounding box* disekitar *data ground truth*.

2.1.2 YOLOv3

YOLOv3 merupakan salah satu *object detector* yang memiliki akurasi relatif baik dan mampu mendeteksi objek secara *real-time*. Algoritma dari *YOLOv3* merupakan penyempurnaan dari *YOLOv1* dan *YOLOv2* karena memiliki kelebihan dalam bidang akurasi, penentuan posisi dan kecepatan. Kelebihan dari *YOLOv3* dibandingkan dengan jaringan lain adalah jauh lebih cepat dan tetap menjaga akurasi.

YOLOv3 dibuat pada tahun 2018 dimana diimplementasikan menggunakan deep learning libraries yaitu *Keras* dan *OpenCV*. Sistem klasifikasi objek dilakukan dengan AI (Artificial Intelligence) untuk melihat objek tertentu. Sistem mengurutkan objek dalam gambar ke dalam kelompok di mana objek dengan karakteristik serupa ditempatkan bersama, sementara yang lain diabaikan kecuali diprogram untuk melakukan sebaliknya.[10]

Cara kerja dari *YOLOv3* adalah dengan menerapkan jaringan syaraf tunggal pada keseluruhan gambar. Jaringan akan membagi gambar menjadi wilayah – wilayah kemudian memprediksi kotak pembatas dan probabilitas. Untuk setiap kotak wilayah pembatas ditimbang probabilitasnya untuk mengklasifikasi sebagai objek atau bukan. Terakhir akan dipilih kotak pembatas dengan nilai paling tinggi untuk dijadikan pemisah antar objek.[1]. Sebagai *localizer*, *YOLO* memiliki *mAP (mean Average Precision)* dan *loss function*. *mAP* merupakan rata rata dari *AP (Average Precision)* tiap *class*. *AP* didapatkan dari rata rata *bounding box* prediksi yang mendekati *bounding box* sebenarnya[10]. *Loss function* sangatlah penting dimana dapat mengukur jarak antara *output model* dan nilai target (*truth*). *Loss function* bekerja ketika model memberikan *error*

yang perlu diperhatikan. *Loss function* yang baik adalah *Loss function* yang menghasilkan *error* yang paling rendah. *Loss function YOLO* dibagi menjadi 3 bagian yaitu yang bertanggung jawab untuk menemukan koordinat *bounding-box*, skor prediksi *bounding-box*, dan prediksi *class-score*. Semuanya adalah *Mean-squared error lost* dan dimodulasi oleh beberapa meta-parameter atau skor *IoU* antara prediksi dan *ground truth*. Formula untuk *loss function YOLO* terdapat gambar pada Persamaan 1.[2][6]

$$\begin{aligned}
 Loss_{yolo} = & \lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^B 1_{ij}^{obj} [(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2] + \\
 & \lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^B 1_{ij}^{obj} \left[(\sqrt{w_i} - \sqrt{\hat{w}_i})^2 + (\sqrt{h_i} - \sqrt{\hat{h}_i})^2 \right] + \sum_{i=0}^{S^2} \sum_{j=0}^B 1_{ij}^{obj} (C_i - \\
 & \hat{C}_i)^2 + \lambda_{noobj} \sum_{i=0}^{S^2} \sum_{j=0}^B 1_{ij}^{noobj} (C_i - \hat{C}_i)^2 + \sum_{i=0}^{S^2} 1_{ij}^{obj} \sum_{c \in classes} (p_i(c) - \\
 & \hat{p}_i(c))^2
 \end{aligned} \quad (1)$$

2.1.3 ResNet50

ResNet50 merupakan singkatan dari *Residual Network*. *ResNet50* adalah jaringan saraf konvolusional dengan kedalaman 50 layer. *ResNet50* memiliki 48 lapisan Konvolusi bersama dengan 1 MaxPool dan 1 lapisan Pool Rata-rata. Ini memiliki operasi titik mengambang $3,8 \times 10^9$ [4].

Dalam mencegah terjadinya *vanishing gradient problem ResNet* memperkenalkan *identity shortcut connection* atau yang disebut dengan *residual block* dimana dapat melewati 1 lapisan atau *layer*.

Didalam arsitektur *ResNet50* memiliki konvolusi yang membentuk *kernel* dengan ukuran 7×7 , *filter* sebanyak 64 buah dan *stride* 2. Pada tahapan pembuatan *filter*, model belum memasuki modul *residual*. Setelah tahap ini, gambar memasuki *pooling layer* yang memiliki *stride* 2. *Pooling Layer* digunakan untuk merangkum informasi yang dihasilkan oleh *convolution* dengan salah satu caranya yaitu mengurangi dimensi. *Pooling layer* juga dapat mengurangi *noise* yang ada pada gambar. Terdapat 2 jenis *pooling* yaitu *max pooling* dan *average pooling*.

Pada konvolusi berikutnya ada *kernel* $1 \times 1,64$ diikuti dengan *kernel* $3 \times 3,64$ dan akhirnya *kernel* $1 \times 1,256$, dimana ketiga lapisan ini diulang total 3 kali sehingga memberi kita 9 lapisan baru. Konvolusi berikutnya adalah *kernel* $1 \times 1,128$ diikuti dengan *kernel* $3 \times 3,128$ dan akhirnya *kernel* $1 \times 1,512$. Langkah ini diulang 4 kali sehingga memberikan 12 lapisan baru pada langkah ini. Konvolusi berikutnya adalah *kernel* $1 \times 1,256$ dan dua *kernel* lagi dengan $3 \times 3,256$ dan $1 \times 1,1024$ dan ini diulang 6 kali memberikan total 18 lapisan. Konvolusi berikutnya sekali lagi memasuki *kernel* $1 \times 1,512$ dengan dua $3 \times 3,512$ dan $1 \times 1,2048$ dan ini diulang 3 kali sehingga menghasilkan total 9 lapisan[4].

Hasil dari setiap *filter* akan melewati *average pooling* dan masuk ke dalam jaringan *fully connected* dengan fungsi *softmax* untuk mengetahui hasil klasifikasi.

$$softmax(x)_i = \frac{\exp(x_i)}{\sum_{j=1}^n \exp(x_j)} \quad (2)$$

Softmax adalah suatu fungsi yang menghitung probabilitas dari setiap target class atas semua target class yang memungkinkan dan membantu menentukan kelas target *input* yang diberikan. Keuntungan dari *softmax* adalah rentang probabilitas *output* dengan nilai 0 hingga 1 dan jumlah semua probabilitas akan sama dengan 1. Fungsi ini biasanya digunakan di ujung *layer* dari *fully connected* layer yang digunakan pada *CNN* untuk menghasilkan nilai

probabilitas suatu objek terhadap *class* yang ada. Rumus *softmax* dapat dilihat pada Persamaan 2 dimana menghitung nilai eksponen dari x_i terhadap jumlah dari seluruh nilai eksponen yang ada.[7]

$$\text{sigmoid}(x) = \left(\frac{1}{1+\exp^{-x}}\right) \quad (3)$$

Sigmoid terkadang disebut sebagai fungsi logistik. Fungsi sigmoid digunakan untuk memprediksikan probabilitas output dalam bentuk klasifikasi biner. Rumus Sigmoid dapat dilihat pada Persamaan 3 Jumlah probabilitas dari Sigmoid tidak harus 1.[9]

$$CE = - \sum t_i \log(s_i) \quad (4)$$

Untuk memperbaiki kesalahan pada saat *training*, model menggunakan fungsi *cross entropy loss* Persamaan 4. Fungsi ini akan menghitung kelas *actual* t_i berbanding dengan s_i yang akan menghitung jumlah *log* negatif dari hasil perkalian nilai prediksi dikali dengan kelas asli lalu dibagi dengan jumlah kelas[7].

2.2 Tinjauan Studi

Penelitian ini akan menggunakan beberapa penelitian lain terkait yang telah dilakukan sebelumnya sebagai tinjauan studi. Berikut adalah penelitian-penelitian yang telah dilakukan terhadap *YOLOv3* dan *ResNet50*.

Terdapat penelitian mengenai membandingkan model *transfer learning ResNet50, VGG*, dan lain lain. Tiap model di *test* dengan dilakukan dua kali *cross validation*[8]. Dari penelitian milik tersebut, *Resnet50* merupakan model terbaik dibandingkan model lainnya dimana validasi akurasi milik *Resnet50* adalah paling tinggi yaitu 80%. Selain itu, *Resnet50* juga memiliki performa yang lebih baik dibandingkan *VGG 16* berdasarkan waktu, jumlah *epoch* dan param[8]. Penelitian ini memiliki kelemahan dimana ketika mengklasifikasi botol plastik, hasil yang diberikan adalah kaca dan kurangnya pembagian kelas seperti plastic dibagi menjadi botol plastik dan tempat makan plastic.

Terdapat juga penelitian mengenai *waste object detection and classification*. Dari penelitian tersebut, peneliti menggunakan pendekatan *Hybrid* untuk *Transfer Learning*. Penulis memulai dengan model yang dilatih sebelumnya pada *imagenet*. Kemudian tambahkan *layer Global Average Pooling*, diikuti dengan *Batch Normalization*, lalu *Dense output layer*. Keuntungan dari pendekatan ini adalah tidak mengubah bobot lapisan dasar secara drastis. Tetapi metode ini susah di *train* dan masih ada lapisan yang belum stabil sehingga memberikan hasil yang kurang memuaskan[5]. Selain itu peneliti juga menggunakan metode *Gaussian-Poisson Generative Adversarial Network (GP-GAN)* dimana metode ini menggunakan cara optimasi gabungan antara gradien dan informasi warna gambar. Tetapi cara ini memiliki masalah dimana ketika menggunakan metode ini, terjadi pengkaburan fitur gambar, dan merusak kinerja[5].

Terdapat penelitian mengenai *Object Detection* menggunakan *VGG ResNet*. Disini memperkenalkan *Object Detection* yang ditingkatkan modeny dengan menggunakan *ResNet*. Penggunaan *ResNet* memberikan dampak yang besar bagi *Object Detection* karena metode ini mengurangi *computation cost* dan *layers dimension* yang membantu *network* memperoleh kinerja yang signifikan untuk mendeteksi objek. Selain itu, penggunaan *ResNet* dalam *Object Detection* juga mengurangi kesalahan dalam pelatihan[3]. Metode yang digunakan digunakan mencapai akurasi rata – rata 85.8 mAP dengan specific object mAP memiliki kisaran akurasi yang cukup tinggi yaitu antara 80.3 % hingga 93.7%.[3].

3. DATASET

Dataset yang digunakan dalam penelitian ini adalah *TACO dataset* untuk *object detection, Waste classification* data dan *Garbage classification* untuk *ResNet50*.

3.1 Dataset TACO

Dataset ini terdiri atas 1500 gambar dimana total objek sampah adalah 4784 sampah. Selain gambar, terdapat *file annotation json* dimana didalam *file annotation.json* terdapat anotasi dari objek sampah di dalam gambar. Anotasi merupakan informasi tentang letak dan *class* dari sebuah objek dalam gambar. *File txt* akan dibuat sebanyak jumlah gambar dimana 1 gambar memiliki 1 *file txt*.

3.2 Dataset ResNet50

Dataset yang digunakan untuk *ResNet50* berasal penggabungan dataset *Waste Classification Data, Garbage Classification*, dan gambar dari *Google* dimana *dataset* ini akan dibagi menjadi 2 *class* yaitu organik dan anorganik. Pada awalnya dalam dataset ini, gambar kardus termasuk ke dalam anorganik sehingga dilakukan perbaikan isi data dengan memindahkan gambar gambar yang berada di *folder* kelas yang salah ke *folder* kelas yang benar sehingga selisih jumlah gambar antar kelas adalah 8724 gambar. Setelah itu dilakukan penghapusan gambar *noise* dan menambah gambar pembantu sehingga totalnya 24772 dan selisihnya 446. Dikarenakan ketika melakukan deteksi pada objek oleh *YOLOv3* dimana *YOLOv3* akan memberikan *bounding box* yang berisikan gambar 1 objek bukan objek berkelompok, dilakukan penghapusan gambar yang memiliki objek berkelompok dalam dataset dengan tujuan untuk menyesuaikan dengan hasil prediksi *bounding box* oleh *YOLOv3*. Setelah itu dilanjutkan dengan penambahan gambar bungkus makanan yang memiliki aneka macam warna dan bentuk untuk mengetahui apakah dengan penambahan gambar bungkus makanan yang memiliki aneka ragam warna dan bentuk mempengaruhi akurasi dari dataset sebelumnya.

4. PENGUJIAN

Pengujian akan dilakukan dalam beberapa tahap. Pertama akan dilakukan percobaan terhadap *YOLOv3* sebagai *object detector* dalam menemukan parameter yang menghasilkan *mAP* terbaik. Setelah menemukan parameter untuk *YOLOv3*, dilakukan pengujian pada *ResNet50* sebagai *object classification* untuk menemukan dataset dan parameter tiap dataset yang menghasilkan akurasi diatas 80% dan *val_loss* yang rendah.

4.1 Pengujian Terhadap Object Detector

Pembuatan model *Object Detector* menggunakan *YOLOv3* sebagai model untuk mendeteksi objek. Pengujian model dilakukan dengan menghitung nilai *mAP* (*mean Average Precision*) dan nilai rata – rata *loss* pada saat *training*. Pengujian dilakukan dengan mengganti jumlah *class* dan *learning rate*. Tujuannya adalah untuk mendapatkan *model* dengan hasil terbaik yang akan digunakan untuk proses *testing* dan yang akan digabungkan dengan *model ResNet50*.

5.1.1 Pengujian Terhadap Jumlah Class dan Learning Rate

Pengujian ini dilakukan dengan tujuan untuk menemukan jumlah *class* dan *learning rate* yang paling sesuai untuk mendapatkan nilai *mAP* yang baik dan nilai rata – rata *loss* yang rendah. *mAP* adalah rata rata dari *AP* (*Average Precision*) tiap *class*. *AP* didapatkan dari rata rata *bounding box* prediksi yang mendekati *bounding box* sebenarnya.

Untuk rata – rata *loss* telah dijelaskan pada bab 4.2 dimana ditujukan memprediksi jenis kelas dari objek yang terdeteksi. Pengujian dilakukan dengan cara mengganti jumlah *class* yaitu 2 dan 1. Selain jumlah *class*, dilakukan pengujian dengan cara mengganti *learning rate* 0.001 dan 0.0001. Sedangkan untuk *batch* adalah 64, *subdivision* adalah 16, *width* dan *height* adalah 416.

Pada tahap pengujian ini dilakukan pengujian terhadap *dataset TACO* yang berisikan 1500 gambar. Didalam *dataset* ini terdapat 59 jenis objek dimana untuk *class* pertama berisikan objek berbahan kertas, *tissue*, objek berbahan karton, dan *food waste*. Sedangkan untuk sisanya akan masuk ke *class* kedua. Selain di bagi menjadi 2 *class*, dilakukan pengujian dengan menggabungkan semua jenis objek menjadi 1 sehingga 1 *class* berisikan 59 jenis objek.

Tabel 1. Tabel Pengujian Object Detector 1 class dan 2 class

Jumlah class	Learning Rate	Avg training loss	Best mAP	Last mAP
1	0.001	46.62%	42%	42%
1	0.005	38.04%	42%	41%
1	0.0001	91.00%	45%	45%
1	0.0005	49.44%	43%	40%
2	0.001	53.09%	33%	32%
2	0.005	53.51%	34%	31%
2	0.0001	95.17%	33%	30%
2	0.0005	67.50%	37%	37%

Best mAP adalah persentasi *mAP* tertinggi dari sebuah model. Untuk *last mAP* adalah persentasi *mAP* terakhir dari *model* yang di *train*. Berdasarkan Tabel 1, dengan *dataset* berjumlah 1500 gambar *best mAP* dan *last mAP* terbaik terjadi dengan parameter jumlah *class* adalah 1 dan *learning rate* adalah 0.0001 dengan akurasi masing masing adalah 45%. Sedangkan untuk rata - rata *training loss* paling rendah adalah milik jumlah *class* 1 dan *learning rate* 0.005 dengan persentasi 38.04%. Untuk jumlah *class* 2, *best mAP* dan *last mAP* terbaik adalah 37% dimana *learning ratenya* adalah 0.0005 dan untuk rata – rata *training loss* terendah adalah 53.09%.

Selain itu juga dilakukan pengujian mengurangi jumlah gambar dari 1500 gambar menjadi 491 gambar dan hanya memiliki 55 jenis objek. Hal ini ditujukan untuk mengetahui apakah jumlah gambar berpengaruh terhadap tingkat *mAP* dan rata – rata *training loss*.

Tabel 2. Tabel Pengujian Object Detector 3

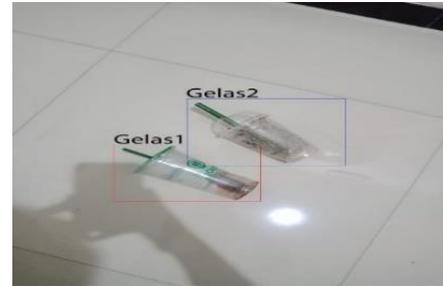
Jumlah class	Learning Rate	Avg training loss	Best mAP	Last mAP
2	0.001	22.40%	33%	28%
2	0.0001	53.28%	38%	38%

Berdasarkan hasil Tabel 2, *best mAP* dan *last mAP* terbaik terjadi pada *learning rate* 0.0001 dimana persentasinya adalah 38%. Sedangkan untuk rata – rata *training loss* terbaik terjadi pada *learning rate* 0.001 yaitu 22.40%. Pengujian ini memiliki rata – rata *training loss* yang lebih rendah karena memiliki jumlah jenis objek yang lebih sedikit dibandingkan dengan pengujian pertama. Selain itu jumlah gambar juga mempengaruhi hasil *mAP* dan rata – rata

training loss karena semakin sedikit data semakin kurang model untuk belajar.

5.2.2 Pengujian Dengan Menggunakan Gambar Dari Luar Dataset

Setelah dilakukan pengujian untuk mendapatkan nilai dari rata – rata *loss*, *mAP* terbaik dan *mAP* terakhir, dilakukan pengujian dengan menggunakan gambar yang diambil secara nyata dengan kamera. Untuk symbol V artinya adalah muncul *bounding box*, X tidak muncul *bounding box* dan untuk “?” adalah 2 gambar atau lebih yang dideteksi menjadi 1 *bounding box*.



Gambar 2. Gambar Testing 1

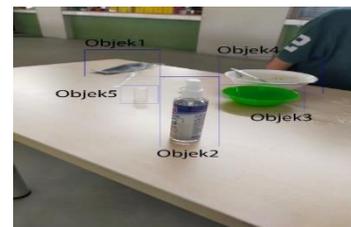
Tabel 3. Hasil pengujian Object Detector 1 class dan 2 class pada gambar testing 1

Class	Learning rate	Gelas 1	Gelas 2
1	0.001	V	V
1	0.005	V	V
1	0.0001	V	V
1	0.0005	V	V
2	0.001	V	V
2	0.005	V	V
2	0.0001	V	V
2	0.0005	V	V

Tabel 4. Hasil pengujian Object Detector 491 gambar pada gambar testing 1

Learning rate	Gelas 1	Gelas 2
0.001	V	V
0.0001	V	V

Gambar 2 merupakan gambar contoh 1 yang digunakan untuk pengujian. Berdasarkan hasil dari Tabel 3 dan Tabel 4, semua *object detector* dapat mendeteksi semua objek didalam gambar.



Gambar 3. Gambar Testing 2

Tabel 5. Hasil pengujian *Object Detector 1 class dan 2 class* pada gambar *testing 2*

Class	Learning rate	Objek 1	Objek 2	Objek 3	Objek 4	Objek 5
1	0.001	X	V	?	?	V
1	0.005	V	V	?	?	X
1	0.0001	V	X	V	V	X
1	0.0005	X	V	V	V	X
2	0.001	X	V	V	V	X
2	0.005	V	V	?	?	X
2	0.0001	X	V	V	V	X
2	0.0005	X	V	V	V	X

Tabel 6. Hasil pengujian *Object Detector 491 gambar* pada gambar *testing 2*

Learning rate	Objek 1	Objek 2	Objek 3	Objek 4	Objek 5
0.001	X	V	V	V	X
0.0001	X	V	V	V	X

Gambar 3 merupakan gambar yang digunakan untuk pengujian. Berdasarkan hasil dari Tabel 5 dan Tabel 6, *object detector 491* gambar hanya dapat mendeteksi objek 2,3, dan 4 dimana *detector* tidak dapat mendeteksi objek berbentuk kecil transparan dan jauh. Sedangkan untuk *object detector 1 class dan 2 class* terkadang *bounding box* menjadikan 2 objek menjadi 1 jika berdekatan. Selain itu tiap *epoch* memberikan hasil prediksi yang berbeda tetapi hanya 1 *class* dan *learning rate* 0.001 yang dapat mendeteksi objek 5 yang merupakan objek kecil dan transparan. Untuk objek 1 sedikit susah untuk dideteksi karena memiliki jarak yang sedikit jauh.

4.2 Pengujian *Object Classification* dengan *ResNet50*

Pengujian ini ditujukan untuk menemukan jumlah dan jenis gambar dalam setiap *class, epoch, learning rate, dan input size* yang paling sesuai untuk *ResNet50*. Hal ini ditujukan untuk mendapatkan nilai akurasi yang tinggi karena semakin tinggi nilai akurasi semakin akurat model mengklasifikasikan objek yang dideteksi. Pengujian pertama adalah pengujian terhadap dataset pertama yang berisi 21986 gambar. Dalam pengujian dataset ini menggunakan *batch size 8, activation function sigmoid* dan dengan *image preprocessing rescale, rotate, shear, dan horizontal flip*. Dalam pelaksanaan *training* juga dilakukan pemberhentian *training* pada *epoch* tertentu karena akurasi tidak mengalami perubahan yang signifikan dan adanya terjadi *overfitting*.

Tabel 7. Tabel pengujian *Object Classification dataset 21986 gambar learning rate 0.001*

Epoch	Ukuran	Acc	Val loss	Val Acc	Avg F1 Score
170	100x100	95.45%	37.55%	87.45%	87%
170	224x224	93.20%	28.61%	88.65%	89%
130	416x416	96.37%	54.60%	85.04%	85%

Berdasarkan Tabel 7 dan Tabel 8, untuk dataset 21986 terdapat *rule of thumb* dimana untuk mendapatkan akurasi dan *validation accuracy* yang tinggi adalah menggunakan *input size* 416x416 dan *learning rate* 0.0001. Selain itu berdasarkan hasil pada Tabel 7 dan

Tabel 8, semakin kecil angka *learning rate*, semakin tinggi angka akurasi dan *validation accuracy* yang didapatkan untuk dataset 21986. Setelah pengujian dataset 21986 gambar, dilakukan pengujian pada dataset selisih 4446 gambar.

Tabel 8. Tabel pengujian *Object Classification dataset 21986 gambar learning rate 0.0001*

Epoch	Ukuran	Acc	Val loss	Val Acc	Avg F1 Score
140	100x100	96.69%	48.58%	87.54%	88%
160	224x224	97.12%	40.41%	88.68%	89%
170	416x416	97.64%	42.64%	89.77%	90%

Tabel 9. Tabel pengujian *Object Classification selisih 4446 gambar learning rate 0.001*

Epoch	Ukuran	Acc	Val loss	Val Acc	Avg F1 Score
180	100x100	95.13%	43.73%	86.84%	87%
170	224x224	90.64%	46.50%	88.54%	89%
170	416x416	95.21%	32.81%	91.21%	91%

Berdasarkan Tabel 9 dan Tabel 10, terdapat *rule of thumb* dimana untuk mendapatkan akurasi dan *validation accuracy* yang tinggi adalah menggunakan *learning rate* 0.0001. Untuk mendapatkan akurasi yang tinggi menggunakan *input size* 416x416 sedangkan untuk *validation accuracy* menggunakan *input size* 224x224.

Tabel 10. Tabel pengujian *Object Classification selisih 4446 gambar learning rate 0.0001*

Epoch	Ukuran	Acc	Val loss	Val Acc	Avg F1 Score
170	100x100	98.10%	50.12%	88.44%	88%
140	224x224	97.37%	52.88%	86.68%	87%
190	416x416	98.14%	29.72%	91.74%	92%

Setelah pengujian pada dataset selisih 4446 gambar, dilakukan perbandingan antara dataset 21980 gambar dan dataset 4446 gambar dengan tujuan untuk mengetahui apakah selisih jumlah gambar dalam dataset mempengaruhi tingkat akurasi dan *validation accuracy*.

Tabel 11. Tabel perbandingan akurasi antara dataset 21986 dan dataset selisih 4446 gambar

Jenis Dataset	Learning Rate	Ukuran	Epoch	Accuracy
Dataset 21986 gambar	0.0001	416x416	170	97.64%
Dataset beda 4446 gambar	0.0001	416x416	190	98.14%

Berdasarkan Tabel 11, dapat dilihat bahwa dataset beda 4446 gambar memiliki tingkat akurasi yang lebih tinggi dibandingkan dengan dataset 21986 gambar. Hal ini menandakan semakin kecil selisih jumlah dataset pada tiap *class* semakin tinggi akurasi yang

didapatkan. Setelah perbandingan antara akurasi dataset 21986 dan dataset beda 4446 gambar dilakukan perbandingan antara *validation accuracy*.

Tabel 12. Tabel perbandingan *validation accuracy* antara dataset 21986 dan dataset selisih 4446 gambar

Jenis Dataset	Learning Rate	Ukuran	Epoch	Val Acc
Dataset 21986 gambar	0.0001	416x416	170	89.77%
Dataset beda 4446 gambar	0.0001	416x416	190	91.74%

Berdasarkan Tabel 12, dapat dilihat bahwa *validation accuracy* milik dataset beda 4446 lebih tinggi dibandingkan dataset 21986. Hal ini menunjukkan bahwa semakin kecil selisih jumlah dataset pada *class* semakin tinggi angka *validation accuracy*.

Tabel 13. Tabel pengujian *Object Classification dataset 1 objek learning rate 0.001*

Epoch	Ukuran	Acc	Val loss	Val Acc	Avg F1 Score
160	100x100	97.30%	44.78%	89.53%	90%
150	224x224	95.93%	24.60%	91.62%	92%
170	416x416	97.69%	50.49%	87.43%	87%

Tabel 14. Tabel pengujian *Object Classification dataset 1 objek learning rate 0.0001*

Epoch	Ukuran	Acc	Val loss	Val Acc	Avg F1 Score
120	100x100	98.23%	28.21%	91.27%	91%
150	224x224	92.61%	29.14%	88.48%	88%
160	416x416	98.67%	36.64%	92.63%	93%

Berdasarkan Tabel 13 dan Tabel 14, terdapat *rule of thumb* untuk dataset 1 objek dimana untuk mendapatkan angka akurasi dan *validation accuracy* yang tinggi adalah menggunakan *input size* 416x416 dan *learning rate* 0.0001. Selain itu berdasarkan Tabel 13 dan Tabel 14, semakin kecil angka *learning rate*, semakin tinggi angka akurasi dan angka *validation accuracy* yang didapatkan.

Tabel 15. Tabel pengujian *Object Classification dataset 1 objek tambah bungkus makanan learning rate 0.001*

Epoch	Ukuran	Acc	Val loss	Val Acc	Avg F1 Score
130	100x100	96.53%	74.86%	77.25%	77%
180	224x224	96.46%	44.92%	87.96%	88%
170	416x416	96.68%	77.59%	85.08%	85%

Tabel 16. Tabel pengujian *Object Classification dataset 1 objek tambah bungkus makanan learning rate 0.0001*

Epoch	Ukuran	Acc	Val loss	Val Acc	Avg F1 Score
170	100x100	97.39%	56.76%	88.54%	89%
120	224x224	98.45%	35.63%	92.09%	92%
170	416x416	98.57%	87.33%	90.19%	90%

Berdasarkan Tabel 15 dan Tabel 16, terdapat *rule of thumb* untuk dataset 1 objek ditambah bungkus makanan dimana untuk mendapatkan angka akurasi yang tinggi adalah dengan menggunakan parameter *input size* 416x416 dan *learning rate* 0.0001. Menurut Tabel 15 dan Tabel 16 untuk mendapatkan *validation accuracy* yang tinggi untuk dataset 1 objek ditambah bungkus makanan adalah menggunakan parameter *input size* 224x224 dan *learning rate* 0.0001. Selain itu menurut Tabel 15 dan Tabel 16 semakin kecil angka *learning rate*, semakin tinggi angka akurasi dan angka *validation accuracy* yang didapatkan.

Tabel 17. Tabel perbandingan akurasi dataset 1 objek dan dataset 1 objek ditambah bungkus makanan

Jenis Dataset	Learning Rate	Ukuran	Epoch	Accuracy
Dataset 1 objek	0.0001	416x416	160	98.67%
Dataset objek satuan ditambah bungkus makanan	0.0001	416x416	170	98.57%

Berdasarkan Tabel 17, angka akurasi milik dataset 1 objek ditambah bungkus makanan lebih rendah dibandingkan dengan akurasi milik dataset 1 objek. Hal ini menandakan bahwa menambahkan gambar bungkus makanan yang memiliki aneka ragam warna dan bentuk mempengaruhi menjadi lebih rendah dibandingkan dengan tanpa menambahkan gambar bungkus makanan aneka ragam warna dan bentuk.

Tabel 18. Tabel perbandingan *validation accuracy dataset 1 objek dan dataset 1 objek ditambah bungkus makanan*

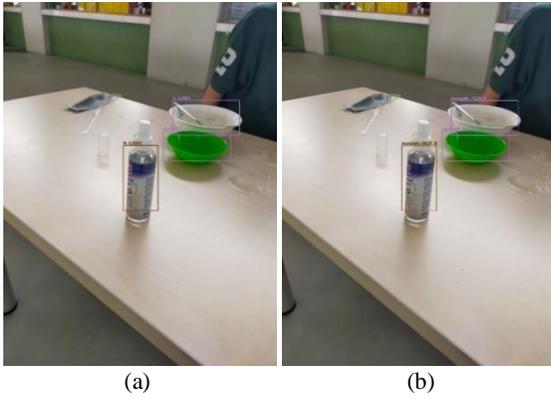
Jenis Dataset	Learning Rate	Ukuran	Epoch	Val Accuracy
Dataset 1 objek	0.0001	416x416	160	92.63%
Dataset objek satuan ditambah bungkus makanan	0.0001	224x224	120	92.09%

Berdasarkan Tabel 18, angka *validation accuracy* tertinggi didapatkan oleh dataset 1 objek dengan *input size* 416x416, *learning rate* 0.0001 dan 160 *epoch*. Hal ini juga menandakan bahwa menambahkan gambar bungkus makanan yang memiliki

aneka ragam warna dan bentuk mempengaruhi menjadi lebih rendah dibandingkan dengan tanpa menambahkan gambar bungkus makanan aneka ragam warna dan bentuk.

4.3 Pengujian akurasi YOLOv3 dan ResNet50

Tujuan dari pengujian ini adalah untuk menguji perbandingan kelas dan *score* dari objek yang di klasifikasi dengan YOLOv3 dan ResNet50. Pengujian dilakukan dengan menggunakan *weight YOLOv3* 1500 gambar dengan parameter 2 class dan *learning rate* 0.0005 dan untuk ResNet50 menggunakan *weight* dari ResNet50 dataset berjumlah 21.986.



Gambar 4. Hasil pengujian YOLOv3 ResNet50

Tabel 19. Tabel pengujian YOLOv3 dan ResNet50 pada gambar testing YOLOv3 salah memberikan class

	Objek 2	Objek 3	Objek 4
YOLO	R 83.89%	O 82.21%	O 58.84%
ResNet50	Anorganik 96.27%	Anorganik 55.73%	Anorganik 73.92%

Berdasarkan hasil dari Tabel 19 dan Gambar 4 YOLOv3 dengan 1500 gambar dengan parameter 2 class dan *learning rate* 0.0001 melakukan kesalahan dalam prediksi objek 3 dan objek 4. YOLOv3 mendeteksi objek 3 dan 4 sebagai organik dan hal tersebut salah. ResNet50 mampu memperbaiki hal tersebut dengan memberikan class anorganik meskipun akurasi kurang baik tetapi class prediksi yang diberikan benar.

5. KESIMPULAN & SARAN

5.1 Kesimpulan

Hasil pengujian yang dilakukan, jumlah class untuk YOLOv3 memberikan pengaruh cukup signifikan terhadap *mAP* dan *average loss* dimana semakin kecil jumlah class semakin tinggi *mAP* yang diberikan. Hasil pengujian yang dilakukan, jenis objek dan jumlah gambar berpengaruh dalam memberikan *mAP* dan *average loss*.

Untuk ResNet50, dapat disimpulkan bahwa kualitas dataset memberikan pengaruh besar terhadap akurasi dan *val_loss* pada ResNet50. Selain itu bentuk objek gambar dan warna gambar berpengaruh pada hasil akurasi ResNet50. Berdasarkan percobaan, akurasi tertinggi untuk ResNet50 didapatkan ketika menggunakan *image size* 416x416 dan *learning rate* 0.0001. Selain itu semakin kecil angka *learning rate*, semakin meningkat angka akurasi yang didapatkan. Untuk *validation accuracy*, tiap dataset memiliki rule

of thumb masing masing dimana ada yang menggunakan *image size* 224x224 untuk mendapatkan angka *validation accuracy* yang tinggi dan ada pula yang menggunakan *input size* 416x416 untuk mendapatkan *validation accuracy* yang tinggi. Semakin kecil selisih jumlah data dalam dataset semakin besar akurasi dan *validation accuracy* yang didapatkan.

Dalam pengujian YOLOv3 yang di gabung dengan ResNet50, dapat disimpulkan bahwa ResNet50 mampu memperbaiki kesalahan dalam memberikan label class dan menaikkan akurasi.

5.2 Saran

Saran yang dapat diberikan adalah dapat menggunakan versi YOLO yang lebih tinggi. Jumlah dataset sangat berpengaruh dalam meningkatkan akurasi dan *validation accuracy*. Dapat menggunakan *pretrain* untuk meningkatkan akurasi ResNet50 dan mengurangi jumlah *epoch* untuk mempercepat *training*.

6. DAFTAR PUSTAKA

- [1] Divvala, S., Redmon, J., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 779-788). DOI: 10.1109/CVPR.2016.91.
- [2] Farhadi, A. & Redmon, J. (2018). Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*.
- [3] Haque, M. F., Lim, H. Y., & Kang, D. S. (2019, January). Object detection based on VGG with ResNet network. In *2019 International Conference on Electronics, Information, and Communication (ICEIC)* (pp.1-3). IEEE. DOI: 10.23919/ELINFOCOM.2019.8706476.
- [4] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770-778). DOI: 10.1109/CVPR.2016.90.
- [5] Kulkarni, H. N., & Raman, N. K. S. (2019). Waste object detection and classification. In *CS230 Stanford*.
- [6] Liu, Xinchao & Gan, Haiyun & Yan, Ying. (2021). Study on Improvement of YOLOv3 Algorithm. *Journal of Physics: Conference Series*. 1884. 012031. 10.1088/1742-6596/1884/1/012031. DOI : 1088/1742-6596/1884/1/012031
- [7] Mahmud, K. H., Adiwijaya, A., & Al Faraby, S. (2019). Klasifikasi Citra Multi-Kelas Menggunakan Convolutional Neural Network. *eProceedings of Engineering*, 6(1). ISSN : 2355-9365
- [8] Mulyadi, S., & Budiraharjo, R. (2019, July 24). PENERAPAN MODEL CONVOLUTIONAL NEURAL NETWORK UNTUK PEMILAHAN SAMPAH. *Proceedings of Conference on Information Technology and Electrical Engineering*. ISSN : 2085-6350.
- [9] Nwankpa, C., Ijomah, W., Gachagan, A., & Marshall, S. (2018). Activation functions: Comparison of trends in practice and research for deep learning. *arXiv preprint arXiv:1811.033*
- [10] Van Etten, A. (2019, January). Satellite imagery multiscale rapid detection with windowed networks. In *2019 IEEE winter conference on applications of computer vision (WACV)* (pp. 735-743). IEEE. *arXiv preprint arXiv:1809.09978*.