Deteksi Sampah Plastik Menggunakan Algoritma Yolov5 (You Only Look Once Version 5)

1st Fauzi Ananta
Fakultas Teknik Elekro
Universitas Telkom
Bandung, Indonesia
fauziananta@student.telkomuniversity.ac.id

2nd Meta Kallista
Fakultas Teknik Elekro
Universitas Telkom
Bandung, Indonesia
metakallista@telkomuniversity.ac.id

3rd Casi Setianingsih
Fakultas Teknik Elekro
Universitas Telkom
Bandung, Indonesia
Setiacasie@telkomuniversity.ac.id

***Abstrak* — Masalah sampah sangat sulit untuk diselesaikan, sampah seperti botol plastik dan keresek menjadi salah satu sampah yang sering dibuang oleh masyarakat, salah satu cara untuk mempermudah pemilihan sampah adalah dengan membuat pendeteksi sampah menggunakan algoritma YOLOv5 (You Only Look Once Version 5). YOLO (You Only Look Once) merupakan salah satu model deep learning yang dapat mendeteksi objek sesuai dengan dataset yang telah dimasukan dan dipelajari oleh YOLO tersebut. Dataset yang digunakan untuk melatih algoritma YOLO merupakan dataset yang dibuat oleh kami sendiri yang dimana dataset memiliki 2004 dataset yang terdiri dari 4 kelas sesuai dengan objek sampah yang dibuang oleh masyarakat. Dari total dataset yang dibuat, dataset dibagi menjadi 3 partisi, diantaranya 1749 *Data Training*, 159 *Data Validation*, dan 96 *Data Testing*. Dengan nilai mAP (*Mean Average Precision*) = 76.4%, *Precision* = 92.3%, dan *Recall* = 71.1%. Dengan melihat matrix tersebut, pendeteksian memiliki ke akuratan yang cukup baik untuk mendeteksi sampah yang sering dibuang oleh masyarakat. Pengimplementasian dan pengujian pada projek ini berhasil mendeteksi sampah yang dapat membantu dalam pemilihan sampah yang dibuang oleh Masyarakat.**

Kata kunci— deteksi sampah, sampah, YOLO, YOLOv5.

I. PENDAHULUAN

Sampah berserakan sudah menjadi masalah yang termasuk sulit untuk dihilangkan. Masih banyak masyarakat yang masih bermalas-malasan untuk membuang sampah pada tempatnya. Begitupun di sungai yang ada di negara kita. Masih banyak sungai yang tercemar oleh sampah, terutama di bagian air sungai. Banyak masyarakat yang tidak memperdulikan sampah di sungai yang resikonya dapat mencemari area sekitar serta mengganggu aliran sungai. Contoh, biasanya masyarakat yang membuang sampah dari jembatan lalu melempar sampah tersebut ke area sungai. Sampah di sungai juga memiliki tingkatan banyaknya sampah yang ada di sekitar sungai tersebut dan sampahnya pun bervariasi, mulai dari sampah plastik, kaleng, styrofoam, botol dsb. Semua sampah dapat aliran air sungai, air sungai akan terhambat sehingga dapat mengakibatkan air meluap ke atas, dan terjadi banjir.

II. KAJIAN TEORI

A. Pengumpulan Dataset

Dataset yang akan digunakan sebagai data merupakan gambar objek yang akan dideteksi. Objek yang akan dideteksi pada pendeteksi sampah pada Sungai Cikapundung ini diantaranya Botol plastik, Kantong plastik, Styrofoam dan Kaleng. Dataset dikumpulkan dengan cara mengunduh gambar melalui Google dan pemotretan langsung menggunakan kamera *Smartphone*. Dataset yang akan dibutuhkan relatif banyak agar sistem dapat mendeteksi secara akurat.

B. Pembuatan Dataset

Pembuatan dataset dilakukan setelah bahan dataset terkumpul. Pembuatan dataset akan dilakukan pada platform Roboflow. Dataset Ada beberapa tahap untuk melakukan pembuatan dataset pada platform Roboflow, diantaranya anotasi objek, *Split* dataset, *Pre-processing* dataset, Augmentasi dataset, dan *Generate* dataset.

C. Konfigurasi Jaringan Algoritma YOLO

Konfigurasi jaringan YOLO dilakukan untuk menjalankan *training* dataset dan pendeteksian pada output kamera CCTV. Algoritma YOLO perlu dikonfigurasi agar algoritma YOLO dapat mengenali objek apa saja yang terdapat pada dataset.

D. Training Dataset

Training dataset dilakukan setelah memiliki dataset yang sudah dibuat sebelumnya, dan algoritma YOLO sudah terkonfigurasi ke Visual Studio Code. Dataset yang telah dibuat kemudian ditraining pada algoritma YOLO agar algoritma YOLO dapat melatih dirinya untuk mengenali objek yang telah diberi *bounding box* dan *label.*

E. Pengujian Algoritma YOLO dan RTSP *(Real Time Streaming Protocol)*

Pengujian algoritma YOLO akan dilakukan dengan mengimplementasikan sistem ke kamera CCTV menggunakan algoritma RTSP (*Real Time Streaming Protocol*). Ouput akan menghasilkan tangkapan dari kamera CCTV di sekitar Sungai Cikapundung, dan menghasilkan deteksi dari Algoritma YOLO.

III. METODE

Berdasarkan permasalah yang telah dijelaskan sebelumnya, perancanga sistem pendeteksi sampah ini bertujuan untuk mendeteksi sampah yang terdapat pada Sungai Cikapundung dengan menggunakan algoritma YOLO dan OpenCV. Perancangan sistem ini akan dibantu dengan web yang dapat memudahkan penjaga kebersihan untuk dapat mengakses dan menjalankan pendeteksi sampah ini. Perancangan sistem ini bertujuan untuk memudahkan para penjaga kebersihan untuk dapat memantau sampah dari jauh

dengan bantuan kamera CCTV. Sehingga para penjaga kebersihan dapat langsung mengetahui informasi keaadaan sungai dengan melihat output dari CCTV dan algoritma YOLO.

A. Object Detection

deteksi objek adalah pendeteksian berbagai jenis objek dan menugaskannya ke kelas; ini adalah salah satu tugas mendasar dari visi computer [1]. Permasalahan yang terdapat pada deteksi objek adalah bagaimana cara membuat mesin dapat mengenali beberapa objek dan menentukan posisi objek tersebut yang terdapat pada sebuah gambar. Konsep deteksi objek merupakan konsep yang sederhana, yaitu dimana objek deteksi melakukan *scanning* pada keseluruhan gambar dan menentukan objek mana yang akan dideteksi dan objek mana yang bukan untuk dideteksi, deteksi objek pun merupakan proses menemukan *instance* objek dari kelas tertentu seperti sampah, pohon, wajah, mobil, motor, dan sebagiannya dari gambar atau video.

B. YOLO (*You Only Look Once*)

YOLO (*You Only Look Once*) adalah jaringan saraf cerdas untuk deteksi waktu nyata [2]. YOLO menerapkan jaringan saraf tunggal ke seluruh gambarYOLO juga merupakan salah satu algoritma yang dapat mendeteksi objek dengan kecepatan proses dan tingkat akurasi yang tinggi. Algoritma ini dijalankan di dua framework yaitu Darknet dan Darkflow yang didukung oleh *Graphics Processing Unit* (GPU). Algoritma YOLO perlu dataset untuk dijadikan bahan ajar bagi sistem agar sistem dapat mengenali objek-objek yang akan dideteksi. Dataset yang dibutuhkan merupakan gambar yang sudah dianotasi menggunakan *Bounding Box* dan *Labeling* untuk output pendeteksian. Algoritma YOLO akan dilatih sampai sistem dapat mengenali objek sesuai dengan dataset yang dibuat. Sistem akan membagi sebuah dataset menjadi beberapa grid dan memprediksi bagian grid yang terdapat memiliki *Bounding Box*. Setelah menemukan objek yang dideteksi, maka sistem akan mengeluarkan output beserta dengan label kelas sesuai dengan objek yang dideteksi. Cara menggunakan yolo dapat dilihat pada gambar 1 berikut

GAMBAR 1
melakukan *cloning respository* dan menginstall *depedencies*

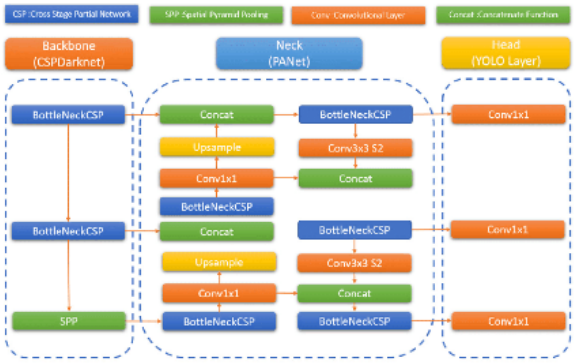
Pada gambar 1 berikut ini merupakan cara yang harus dilakukan untuk menggunakan YOLO, pertama kita memanggil YOLO dengan mengcloning YOLO dari *respotory* dan menginstall *depedencies* yang dibutuhkan oleh YOLO tersebut

C. OpenCV (*Open-Source Computer Vision Library)*

OpenCV (*Open-Source Computer Vision Library*) merupakan *library open source* yang tujuannya dikhususkan untuk lekakukan pengolahan citra [3]. Program OpenCV ini adalah perangkat lunak yang merupakan *open-source* yang dapat digunakan secara bebas oleh pengguna (*user*).

D. Arsitektur Sistem

Arsitektur Sistem pada YOLO (*You Only Look Once*) dapat dibagi menjadi 3 komponen penting yaitu *backbone, neck,* dan *head.* Gambaran dari arsitektur YOLO dapat dilihat pada gambar 2 berikut.



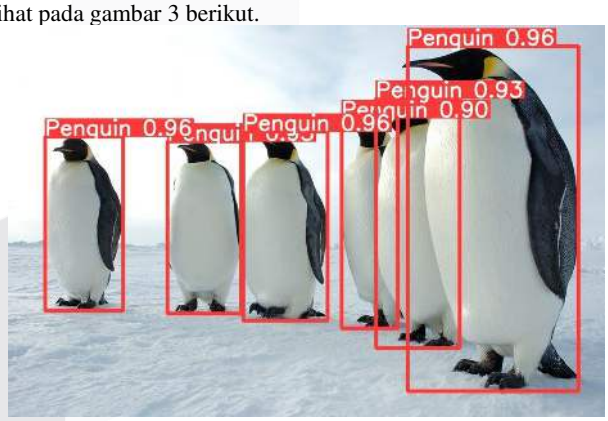
GAMBAR 2
Arsitektur YOLO

Pada gambar 2, ketiga komponen YOLO dapat disimpulkan menjadi:

1. *Backbone:* Model *backbone* digunakan untuk membantu mengurangi resolusi spasial pada gambar dan memperbesar gambar serta meningkatkan resolusi fitur tersebut.
2. *Neck:* Model *neck* digunakan untuk membantu model dalam melakukan menyamaratakan (*generalize*) dengan baik untuk objek pada skala dan ukuran yang berbeda-beda.
3. *Head:* Model *head* digunakan untuk membantu melakukan operasi tahap akhir pada arsitektur YOLO. Model *head* ini akan menerapkan *anchor boxes* pada *maps* (peta fitur) dan akan menampilkan hasil akhir seperti: *Class,Confidence score,* dan *bounding box.*

E. Cara Kerja Sistem

Algoritma YOLO adalah sebuah *Neural Network* pada sebuah gambar dataset dan akan membagi input gambar pada dataset tersebut agar menjadi beberapa grid. Setelah input gambar dibagi menjadi beberapa grid, sistem akan memprediksi *bounding box* serta probabilitas untuk masing-masing grid sesuai dengan dataset yang telah dibuat untuk YOLO tersebut. Ilustrasi *bounding box* dapat dilihat pada gambar 3 berikut.



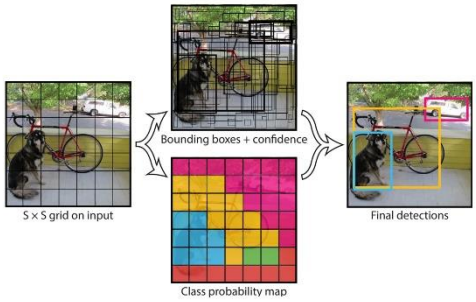
GAMBAR 3
Ilustrasi *bounding box*

Pada gambar 3 berikut ini merupakan contoh ilustrasi dari sebuah *bounding box* untuk sebuah kelas penguin. Untuk mendapatkan sebuah *output bounding box,* sistem akan melakukan proses konvolusi pada gambar yang di-*input* (dimasukan), sehingga hasil akhirnya akan diperoleh ukuran *bounding box* dengan besar SxSx(B*5+C). Dengan B adalah jumlah banyaknya *bounding box* dalam sebuah input gambar, dan C adalah banyaknya kelas yang dapat dideteksi. Nilai B akan dikalikan dengan angka 5 dikarenakan sebuah *bounding box* memiliki 5 atribut yang perlu disimpan, diantaranya adalah:

1. Koordinat X
2. Koordinat Y
3. Lebar (*width*)

4. Tinggi (*height*)
5. *Confidence score* (Nilai probabilitar sebuah *bounding box* yang mendeteksi sebuah objek)

Semua atribut pada *bounding box* akan melalui proses normalisasi, sehingga akan menghasilkan nilai antara 0 dan 1. Koordinat X dan Y akan dinormalisasi dengan menyesuaikan titik kiri atas dari grid, lebar (*width*) dan tinggi (*height*) juga akan dinormalisasikan dengan menyesuaikan ukuran gambar tersebut. Berikut ilustrasi mengenai proses konvolusi untuk mendapatkan *output bounding box* dan label kelas.

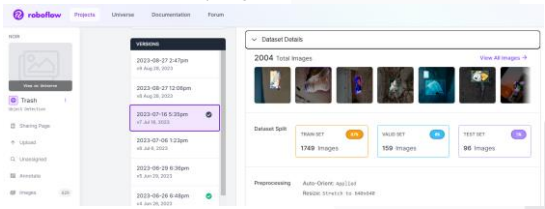


GAMBAR 4
Ilustrasi Pembuatan *Bounding Box*

Pada gambar 4 berikut ini dapat dilihat bahwa proses pertama yang berupa SxS grid pada *input* akan menjalankan 2 proses yaitu *bounding box + confidence* dan *class probability map,* ketika kedua proses itu selesai maka akan menghasilkan hasil gambar terakhir yang akan menampilkan sebuah *bounding box* terhadap objek yang terdeteksi.

F. Roboflow

Roboflow merupakan website yang berfungsi untuk membuat, meminjam, dan membuat sebuah dataset. Pada proyek ini kami membuat dataset sendiri yang memiliki kelas tersendiri.



GAMBAR 4
detail pada daaset

Pada gambar 4 berikut ini merukapan detail dari dataset yang telah kami bikin, dataset ini memiliki total gambar sebanyak 2004 dan telah dibagi menjadi data train sebanyak 1749 gambar, data valid sebanyak 159 gambar, dan data test sebanyak 96 gambar.

Roboflow tersebut akan dipanggil oleh YOLO dengan memasukan *syntax* seperti pafa gamber 5 dibawah ini.

```
!pip install -q roboflow
from roboflow import Roboflow
rf = Roboflow(model_format="yolov5", notebook="roboflow-yolov5")
```

GAMBAR 5
penginstallan roboflow

Pada gambar 5 berikut, penginstalan roboflow akan dilakukan dan YOLO akan dapat menyambungkan antara roboflow dan YOLO itu sendiri. Setelah itu akan dilakukannya pemanggilan dataset yang akan digunakan.

```
from roboflow import Roboflow
rf = Roboflow(api_key="5s6KWdYIUZeql22wjXmP")

from roboflow import Roboflow
rf = Roboflow(api_key="5s6KWdYIUZeql22wjXmP")
project = rf.workspace("noir-1cfxg").project("trash-3i2eu")
dataset = project.version(7).download("yolov5")
```

GAMBAR 6
pemanggilan dataset dari roboflow

Pada gambar 6 berikut ini adalah cara pemanggilan dataset yang berapa pada roboflow agar dapat terpanggil oleh YOLO.

G. Pelatihan Dataset

Pelatihan dataset akan dilakukan oleh YOLO dengan cara yang ada pada gambar 7 berikut

```
python train.py --img 416 --batch 16 --epochs 100 --data {dataset.location}/data.yaml --cfg ./models/custom_yolov5s.yaml --weights '' --name yolov5s_results
```

GAMBAR 7
Training Dataset

Pada gambar 7 berikut ini adalah cara *training* pada sebuah dataset akan dilakukan oleh YOLO. Dengan mengganti seberapa banyaknya img, batch, dan epoch yang ingin diuji dan di-*train* oleh YOLO tersebut.

IV. HASIL DAN PEMBAHASAN

Pada penelitian proyek ini hasil yang akan ditampilkan dan dibahas yaitu berupa hasil output pada YOLO dari dataset yang telah dibuat.

A. Hasil Training Dataset pada YOLOv5

GAMBAR 8
Hasil Pendeteksian Dataset

Pada gambar 8 berikut ini adalah hasil pendeteksian pada dataset yang telah ditraining menggunakan YOLOv5, objek yang dapat terdeteksi pada gambar tersebut akan memiliki *bounding box* yang berisi nama objek yang terbaca dan seberapa besar nilai *confidence* pada objek tersebut. Nilai *confidence* akan berpengaruh terhadap seberapa besar objek tersebut akan tampil berdasarkan apa yang dapat terdeteksi oleh YOLOv5.

B. Implementasi Menggunakan CCTV

GAMBAR 9
Hasil Pengimplementasian YOLOv5

Pada gambar 9 berikut ini, pengujian berhasil dilakukan menggunakan kamera CCTV yang digabungkan oleh YOLOv5 dan

mendapatkan output yang terdeteksi berupa 2 plastik botol dan 1 keresek.

C. Pengujian Pendeteksian Algoritma YOLO Pada Objek Dengan Faktor Jarak dan Intensitas Cahaya

1. Botol Plastik

GAMBAR 10
Botol Plastik Pada Jarak 1M

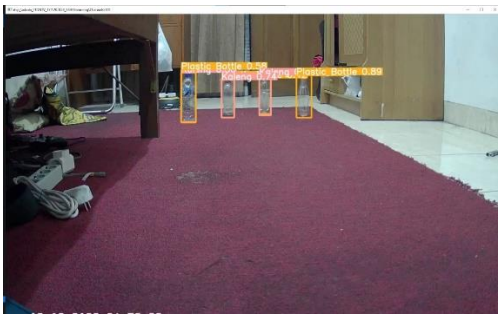
Pada gambar 10 berikut ini, pengujian pendeteksian algoritma YOLO yang dilakukan pada botol plastik dengan jarak 1 meter, intensitas cahaya sebesar 39 lux, dan berada pada ruangan tertutup. Pada hasil pendeteksian dapat terlihat bahwa pendeteksian mengalami *True Positive* dimana YOLO memprediksi label dan mencocokkannya dengan benar sesuai kebenaran dasar.

GAMBAR 11
Botol Plastik Pada Jarak 2M

Pada Gambar 11 berikut ini pengujian pendeteksian algoritma YOLO yang dilakukan pada botol plastik dengan jarak 2 meter, intensitas cahaya sebesar 39 lux, dan berada pada ruangan tertutup. Pada hasil pendeteksian dapat terlihat bahwa pendeteksian mengalami *True Positive* dimana YOLO memprediksi label dan mencocokkannya dengan benar sesuai kebenaran dasar.

GAMBAR 12
Botol Plastik Pada Jarak 3M

Pada Gambar 12 berikut ini pengujian pendeteksian algoritma YOLO yang dilakukan pada botol plastik dengan jarak 3 meter, intensitas cahaya sebesar 39 lux, dan berada pada ruangan tertutup. Pada hasil pendeteksian dapat terlihat bahwa pendeteksian mengalami *True Negative* pada salah satu pendeteksian, yaitu *bounding box* sebelah kanan yang menyatakan bahwa sebuah tas yang terdeteksi menjadi botol plastik.



GAMBAR 13
Botol Plastik Pada Jarak 4M

Pada Gambar 13 berikut ini pengujian pendeteksian algoritma YOLO yang dilakukan pada botol plastik dengan jarak 4 meter, intensitas cahaya sebesar 39 lux, dan berada pada ruangan tertutup. Pada hasil pendeteksian dapat terlihat bahwa pendeteksian mengalami *False Positive* dimana ada 2 pendeteksian yang salah dalam memprediksi 2 objek pada *bounding box,* kelas yang diprediksi merupakan kelas yang salah, yang seharusnya botol plastik tetapi pendeteksian memprediksi botol plastik menjadi sebuah kaleng.

2. Kantong Plastik

GAMBAR 14
Kantong Plastik Pada Jarak 1M

Pada gambar 14 berikut ini pengujian pendeteksian algoritma YOLO yang dilakukan pada kantong plastik dengan jarak 1 meter, intensitas cahaya sebesar 39 lux, dan berada pada ruangan tertutup. Pada hasil pendeteksian dapat terlihat bahwa pendeteksian mengalami *True Positive* dimana YOLO memprediksi label dan mencocokkannya dengan benar sesuai kebenaran dasar.

GAMBAR 15
Kantong Plastik Pada Jarak 2M

Pada gambar 15 berikut ini pengujian pendeteksian algoritma YOLO yang dilakukan pada kantong plastik dengan jarak 2 meter, intensitas cahaya sebesar 39 lux, dan berada pada ruangan tertutup. Pada hasil pendeteksian dapat terlihat bahwa pendeteksian

mengalami *True Positive* dimana YOLO memprediksi label dan mencocokkannya dengan benar sesuai kebenaran dasar.

GAMBAR 16
Kantong Plastik Pada Jarak 3M

Pada gambar 16 berikut ini pengujian pendeteksian algoritma YOLO yang dilakukan pada kantong plastik dengan jarak 3 meter, intensitas cahaya sebesar 39 lux, dan berada pada ruangan tertutup. Pada hasil pendeteksian dapat terlihat bahwa pendeteksian mengalami *True Positive* dimana YOLO memprediksi label dan mencocokkannya dengan benar sesuai kebenaran dasar.

GAMBAR 17
Kantong Plastik Pada Jarak 4M

Pada gambar 17 berikut ini pengujian pendeteksian algoritma YOLO yang dilakukan pada kantong plastik dengan jarak 4 meter, intensitas cahaya sebesar 39 lux, dan berada pada ruangan tertutup. Pada hasil pendeteksian dapat terlihat bahwa pendeteksian mengalami *False Negative* dimana pada gambar 17 terdapat 4 buah kantong plastik tetapi hanya terdapat 3 *bounding box* yang diprediksi.

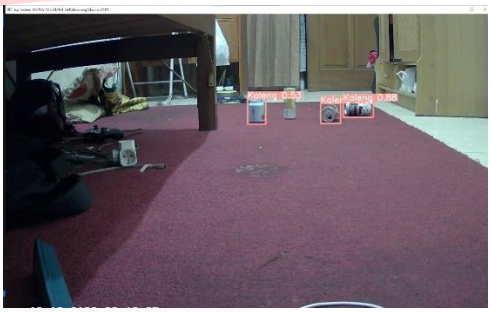
3. Kaleng

GAMBAR 18
Kaleng Pada Jarak 1M

Pada gambar 18 berikut ini pengujian pendeteksian algortima YOLO yang dilakukan pada kaleng dengan jarak 1 meter, intensitas cahaya sebesar 39 lux, dan berada pada ruangan tertutup. Pada hasil pendeteksian dapat terlihat bahwa pendeteksian mengalami *True Positive* dimana YOLO memprediksi label dan mencocokkannya dengan benar sesuai kebenaran dasar.

GAMBAR 19
Kaleng Pada Jarak 2M

Pada gambar 19 berikut ini pengujian pendeteksian algortima YOLO yang dilakukan pada kaleng dengan jarak 2 meter, intensitas cahaya sebesar 39 lux, dan berada pada ruangan tertutup. Pada hasil pendeteksian dapat terlihat bahwa pendeteksian mengalami *False Negative* dimana pada gambar 19 terdapat 5 buah kaleng tetapihanya terdapat 4 *bounding box* yang diprediksi.



GAMBAR 20
Kaleng Pada Jarak 3M

Pada gambar 20 berikut ini pengujian pendeteksian algortima YOLO yang dilakukan pada kaleng dengan jarak 3 meter, intensitas cahaya sebesar 39 lux, dan berada pada ruangan tertutup. Pada hasil pendeteksian dapat terlihat bahwa pendeteksian mengalami *False Negative* dimana pada gambar 20 terdapat 4 buah kaleng tetapihanya terdapat 3 *bounding box* yang diprediksi.

GAMBAR 21
Kaleng Pada Jarak 4M

Pada gambar 21 berikut ini pengujian pendeteksian algortima YOLO yang dilakukan pada kaleng dengan jarak 4 meter, intensitas cahaya sebesar 39 lux, dan berada pada ruangan tertutup. Pada hasil pendeteksian dapat terlihat bahwa pendeteksian mengalami *False Negative* dimana pada gambar 21 terdapat 4 buah kaleng tetapihanya terdapat 3 *bounding box* yang diprediksi.

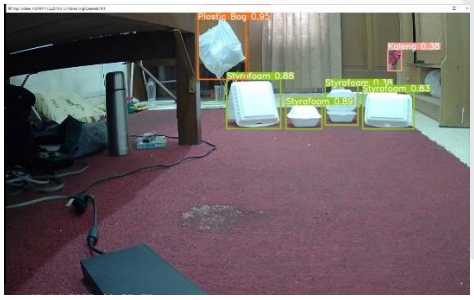
4. Styrofoam

GAMBAR 22
Styrofoam Pada Jarak 1M

Pada gambar 22 berikut ini pengujian pendeteksian algoritma YOLO yang dilakukan pada styrofoam dengan jarak 1 meter, intensitas cahaya sebesar 39 lux, dan berada pada ruangan tertutup. Pada hasil pendeteksian dapat terlihat bahwa pendeteksian mengalami *true positive* bukan hanya pada styrofoam tetapi pada kantong plastik yang dimana YOLO memprediksi label dan mencocokkannya dengan benar sesuai kebenaran dasar.

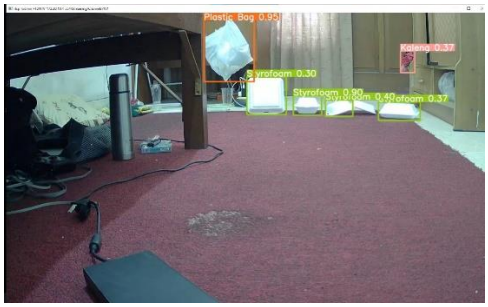
GAMBAR 23
Styrofoam Pada Jarak 2M

Pada gambar 23 berikut ini pengujian pendeteksian algoritma YOLO yang dilakukan pada styrofoam dengan jarak 2 meter, intensitas cahaya sebesar 39 lux, dan berada pada ruangan tertutup. Pada hasil pendeteksian dapat terlihat bahwa pendeteksian mengalami *multiple False Positives* dimana YOLO memprediksi 2 buah objek yang bukan merupakan kantong plastik tetapi memprediksikan kedua objek tersebut sebagai kelas kantong plastik.



GAMBAR 24
Styrofoam Pada Jarak 3M

Pada gambar 24 berikut ini pengujian pendeteksian algoritma YOLO yang dilakukan pada styrofoam dengan jarak 3 meter, intensitas cahaya sebesar 39 lux, dan berada pada ruangan tertutup. Pada hasil pendeteksian dapat terlihat bahwa pendeteksian mengalami *False Positive* dimana YOLO memprediksi sebuah *bounding box* pada objek yang tidak termasuk dalam kelas pendeteksian pada dataset.



GAMBAR 25
Styrofoam Pada Jarak 4M

Pada gambar 25 berikut ini pengujian pendeteksian algoritma YOLO yang dilakukan pada styrofoam dengan jarak 4 meter, intensitas cahaya sebesar 39 lux, dan berada pada ruangan tertutup. Pada hasil pendeteksian dapat terlihat bahwa pendeteksian mengalami *False Positive* dimana YOLO memprediksi sebuah *bounding box* pada objek yang tidak termasuk dalam kelas pendeteksian pada dataset.

V. KESIMPULAN

Pendeteksian sebuah objek dapat dilakukan dengan menggunakan berbagai macam cara, salah satunya adalah dengan menggunakan YOLO. Pendeteksian sampah menggunakan YOLO dengan bantuan CCTV akan mempermudah pengguna dalam mengetahui bagaima keadaan sampah sekitar Sungai dengan jangkau yang panjang dimanapun dan kapanpun.

Pendeteksi sampah tidak akan berjalan jika tidak ada dataset, maka dari itu pembuatan dataset yang bagus sangatlah penting agar YOLO dapat belajar dan memahami setiap objek yang dia pelajari, hasil *output* yang diberikan oleh YOLO akan menghasilkan pendeteksian sesuai dengan dataset yang dipelajari.

REFERENSI

[1] Richard Szeliski “Computer Vision: Algorithms and Applications”, Springer Link, 10 September 2010, [Online], Available: https://link.springer.com/book/10.1007/978-1-84882-935-0#book-header

[2] Lin, C. J., & Jhang, J. Y “Intelligent Traffic-Monitoring System Based on YOLO and Convolutional Fuzzy Neural Networks”, SEMANTIC SCHOLAR, 15 Januari 2022, [Online], Available: https://www.semanticscholar.org/paper/Intelligent-Traffic-Monitoring-System-Based-on-YOLO-Lin-Jhang/262e6de7ef70b8b0be2d11b268cc79075a6c7f13

[3] Zulkhaidi, “Pengenalan Pola Bentuk Wajah dengan OpenCV” , JURTI, Desember 2019, [Online], Available: https://e-journals.unmul.ac.id/index.php/INF/article/view/4033

[4] ARGHADEEP MITRA, “Detection of Waste Materials Using Deep Learning and Image Processing”, Dec7 ,2020, [Online] ,Available: https://scholarworks.calstate.edu/downloads/gx41mn74q

[5] Colin van Lieshout, Kees van Oeveren, Tim van Emmerik, dan Eric Postma, “Automated River Plastic Monitoring Using Deep Learning and Cameras”, ADVANCING EARTH AND SPACE SCIENCE, 28 JUL 2020 , [Online], Available: https://agupubs.onlinelibrary.wiley.com/doi/pdf/10.1029/2019EA000960

[6] Mincheol Kim dkk, “A CNN Image Classification Analysis for 'Clean-Coast Detector' as Tourism Service Distribution”,

Journal of Distribution Science(유통과학연구), 2020.01.30, [Online], Available: https://koreascience.kr/article/JAKO202014862061074.view?orgId=kodisa

[7] Muhammad Faisal dkk, "Faster R-CNN Algorithm for Detection of Plastic Garbage in the Ocean: A Case for Turtle Preservation", Hindawi, 26 May 2022, [Online], Available: https://www.hindawi.com/journals/mpe/2022/3639222/

[8] Mattis Wolf dkk, "Machine learning for aquatic plastic litter detection, classification and quantification (APLASTIC-Q)", IOPscience, 16 November 2020, [Online], Available: https://iopscience.iop.org/article/10.1088/1748-9326/abbd01

[9] Harsh Panwar dkk, "AquaVision: Automating the detection of waste in water bodies using deep transfer learning", Case Studies in Chemical and Environmental Engineering Volume 2, September 2020, [Online], Available: https://www.sciencedirect.com/science/article/pii/S2666016420300244

[10] Chengjuan Ren dkk "Coastal Waste Detection Based on Deep Convolutional Neural Networks", MDPI, 31 October 2021, [Online], Available: https://www.mdpi.com/14248220/21/21/7269/htm

[11] Nur Athirah Zailan dkk, "An automated solid waste detection using the optimized YOLO model for riverine management", frontiers, 12 August 2022, [Online], Available: https://www.frontiersin.org/articles/10.3389/fpubh.2022.907280/full

[12] Oktaviani Ella Karlina dkk, "PENGENALAN OBJEK MAKANAN CEPAT SAJI PADA VIDEO DAN REAL TIME WEBCAM MENGGUNAKAN METODE YOU LOOK ONLY ONCE (YOLO) ", 2020, [Online], Available: https://ejournal.gunadarma.ac.id/index.php/infokom/article/download/2362/186

[13] Matko Glučina dkk, "Detection and Classification of Printed Circuit Boards Using YOLO Algorithm", 29 January 2023, [Online], Available: https://www.mdpi.com/2079-9292/12/3/667