

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

1st Fadlan Yusuf Raspati
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia

maharajafadlan@student.telkomuniversity.ac.id

2nd Meta Kallista
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia

metakallista@telkomuniversity.ac.id

3rd Casi Setianingsih
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia

setiacasie@telkomuniversity.ac.id

Abstrak — Sampah berserakan merupakan masalah yang sulit dihilangkan. Masih banyak masyarakat yang sering membuang sampah tidak pada tempatnya. Sampah seperti kantong plastik, botol plastik, styrofoam, dan kaleng merupakan jenis sampah yang sulit untuk dirauikan masih sering dibuang sembarangan oleh masyarakat tanpa memperdulikan dampak yang akan terjadi pada lingkungan sungai. Berdasarkan permasalahan di atas, dibuatlah Pemantau dan Pendeteksi Sampah Plastik Menggunakan Algoritma YOLOv5 (You Only Look Once Version 5), sebuah sistem yang bertujuan untuk mempermudah pengguna untuk memantau dan mendeteksi sampah pada lingkungan sungai. Sistem ini menggunakan algoritma pendeteksi yaitu algoritma YOLOv5 untuk dapat mendeteksi dan mengklasifikasi jenis sampah yang terdapat pada sungai. Dengan bantuan kamera CCTV dan memanfaatkan OpenCV, algoritma YOLO akan dapat bekerja secara bersamaan dengan OpenCV sehingga pengguna dapat memantau lingkungan dan mendapatkan hasil deteksi dari kamera CCTV secara real- time. Pengguna dapat memakai sistem ini dengan mengakses web yang langsung menampilkan output dari kamera CCTV dan pendeteksian yang dihasilkan oleh algoritma YOLOv5.

Kata kunci— YOLOv5, CCTV, Sampah,

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 risikonya 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.

Dilansir dari Liputan6.com, hingga bulan Juli 2021, data Badan Pusat Statistik (BPS) menyebutkan bahwa 46% sungai di Indonesia dalam keadaan tercemar berat, 32% tercemar sedang berat, 14% tercemar sedang dan 8% tercemar ringan. Sungai Citarum menjadi salah satu sungai yang paling kotor sejak sepuluh tahun lalu akibat tercemar banyak limbah hasil pabrik tekstil. Hal ini berdampak pada air sungai

yang berwarna warni dan sudah tidak jernih lagi. Dikutip dari Antaranews.com, Luckmi Purwandari, Direktur Jenderal Pengendalian Pencemaran dan Kerusakan Lingkungan (Ditjen PPKL) Kementerian Lingkungan Hidup dan Kehutanan (KLHK) mengatakan bahwa tercemarnya sungai-sungai Indonesia kebanyakan disebabkan oleh kegiatan pembuangan sampah tidak bertanggung jawab, sehingga biota di aliran sungai kekurangan oksigen dan tidak dapat bertahan hidup.

Dengan dirancangnya sistem Pemantau dan Pendeteksi Sampah Menggunakan Algoritma YOLOv5 yang dibantu dengan menggunakan kamera CCTV ini, diharapkan pengguna dapat memantau area sekitar dari kejauhan menggunakan kamera CCTV dan mengetahui jenis sampah yang terdapat pada area sekitar dari hasil pendeteksian algoritma YOLOv5.

Sistem ini dirancang untuk membantu pengguna agar dapat lebih cepat untuk mengantisipasi terjadinya penumpukan sampah pada sungai dengan memantau lingkungan sekitar sungai menggunakan kamera CCTV dan pendeteksian dari algoritma YOLO.

Dengan dirancangnya sistem ini, diharapkan sistem ini dapat memberikan sebuah ide atau inspirasi bagi pihak-pihak yang berkaitan dengan kebersihan lingkungan, dan juga dapat memberikan kontribusi positif sehingga dapat meningkatkan antisipasi yang lebih awal agar tidak terjadi penumpukan sampah pada sungai yang dapat mengakibatkan bencana banjir.

II. KAJIAN TEORI

A. Object Detection

Object Detection adalah proses menemukan instance objek dari kelas tertentu, seperti wajah, mobil, dan pohon, dalam gambar atau video. Tidak seperti klasifikasi, deteksi objek dapat mendeteksi banyak objek, serta lokasinya di gambar. Detektor objek akan mengembalikan daftar objek yang terdeteksi dengan informasi kelas objek, probabilitas, dan koordinatnya untuk setiap objek. [1]

B. Convolutional Neural Networks (CNN)

Convolutional Neural Networks (CNN) adalah salah satu deep learning yang mampu melakukan pengenalan gambar dan klasifikasi gambar. Metode CNN ini merupakan suatu kelas pada neural network yang biasa digunakan untuk data image. Metode CNN dapat digunakan untuk mendeteksi dan mengenali objek pada sebuah gambar. Metode ini merupakan sebuah metode yang terinspirasi dari cara

mamalia dan manusia untuk memvisualisasikan sebuah citra. Metode CNN memanfaatkan proses konvolusi yang dimana konvolusi ini menggerakkan sebuah kernel ke dalam gambar untuk memfilter gambar menjadi beberapa bagian grid yang memiliki panjang dan tinggi (pixels) yang sama rata. Sehingga komputer mendapatkan informasi representatif baru dari hasil pembagian grid pada bagian gambar tersebut dengan filter yang digunakan. [2]

C. You Only Look Once (YOLO)

You Only Look Once (YOLO) merupakan salah satu model deep learning yang diusulkan pertama kali oleh Joseph Redmond pada tahun 2015. You only look once (YOLO) adalah sebuah algoritma yang dikembangkan untuk mendeteksi sebuah objek secara real-time. Sistem pendeteksian yang dilakukan adalah dengan menggunakan repurpose classifier atau localizer untuk melakukan deteksi. Sebuah model diterapkan pada sebuah citra di beberapa lokasi dan skala. [3]

D. OpenCV

OpenCV (*Open Source Computer Vision Library*) adalah sebuah perangkat lunak yang ditujukan untuk pengolahan citra digital secara real-time yang dibuat oleh Intel, dan sekarang didukung oleh Willow Garage dan Itseez. Program ini terbebas dan ada dalam naungan sumber terbuka dari lisensi BSD. Pustaka ini didedikasikan sebagai besar untuk pengolahan citra digital secara real time.

OpenCV pertama kali diluncurkan secara resmi pada tahun 1999 oleh Inter Research sebagai kelanjutan dari bagian proyek yang menghasilkan aplikasi intensif berbasis CPU, real-time ray tracing dan tembok penampilan 3 Dimensi. Para kontributor utama dalam proyek ini termasuk mereka yang berkecimpung dalam bidang optimasi dari Intel Russia dan juga Tim Pustaka Performasi Intel. [4]

E. Real-time Streaming Protocol (RTSP)

Real-time Streaming Protocol (RTSP) merupakan jaringan komputer yang dirancang untuk sistem komunikasi yang fungsinya untuk mengontrol server media. RTSP biasanya dapat mengontrol sesi media, seperti kamera video, encoder, perekam video digital, dan server streaming. Dengan cara ini, RTSP memungkinkan pengguna untuk mengontrol pemutaran media streaming dengan mengirimkan perintah seperti play, pause, dan rewind. RTSP juga memungkinkan klien untuk menerima data waktu nyata dari server, seperti posisi streaming saat ini atau total durasi.

Selain media streaming, RTSP sering digunakan untuk pengaturan kamera IP. Banyak kamera IP memiliki dukungan bawaan untuk RTSP, sehingga memudahkan untuk mengakses streaming video langsung dari perangkat apa pun yang kompatibel. Sebagai contoh, beberapa sistem keamanan rumah menggunakan RTSP untuk melakukan streaming video langsung dari kamera langsung ke komputer atau perangkat seluler. [5]

F. Dataset

Dataset atau biasa disebut kumpulan data merupakan sekumpulan data yang memiliki atribut tertentu yang menyesuaikan dengan penelitian yang akan dilakukan. Dataset ini berfungsi sebagai data uji pada penelitian agar komputer dapat mengenal dan belajar akan dataset tersebut sehingga komputer dapat melatih dirinya untuk mengetahui

dataset yang digunakan. Dataset bisa didapatkan dengan berbagai cara sesuai dengan kebutuhan penelitian.

G. Roboflow

Roboflow merupakan salah satu platform pada web yang berfungsi untuk membuat dataset pendeteksian. Platform roboflow memiliki beberapa fitur untuk membuat dataset yang akan dijadikan data uji pada komputer, fitur tersebut diantaranya adalah anotasi dataset, partisi dataset, dan augmentasi dataset.

III. METODE PENELITIAN

A. Real-time Streaming Protocol (RTSP)

Pada metode ini, RTSP digunakan untuk memanggil output dari kamera CCTV menggunakan komputer atau laptop. RTSP dapat bekerja dengan memanfaatkan alamat IP dari kamera CCTV yang terhubung dengan *router* internet, sehingga output dari kamera CCTV akan dapat dipanggil menggunakan bantuan OpenCV.

rtsp://UserName:Password@IpAdress:Port/Streaming/Channels/ChannelID

GAMBAR III.1.
Syntax RTSP

Gambar III.1. adalah *syntax* RTSP. *Syntax* diisi berdasarkan informasi yang terdapat pada kamera CCTV yang memiliki alamat IP tersendiri. *Syntax* tersebut dapat memanggil output dari kamera CCTV secara *real-time*.

```
import cv2
# vid = cv2.VideoCapture(0) # For webcam
vid = cv2.VideoCapture("http://wmccpinetop.axiscam.net/mjpg/video.mjpg")
while True:
    _, frame = vid.read()
    print(frame)
    cv2.imshow('Video Live IP cam', frame)
    key = cv2.waitKey(1) & 0xFF
    if key == ord('q'):
        break

vid.release()
cv2.destroyAllWindows()
```

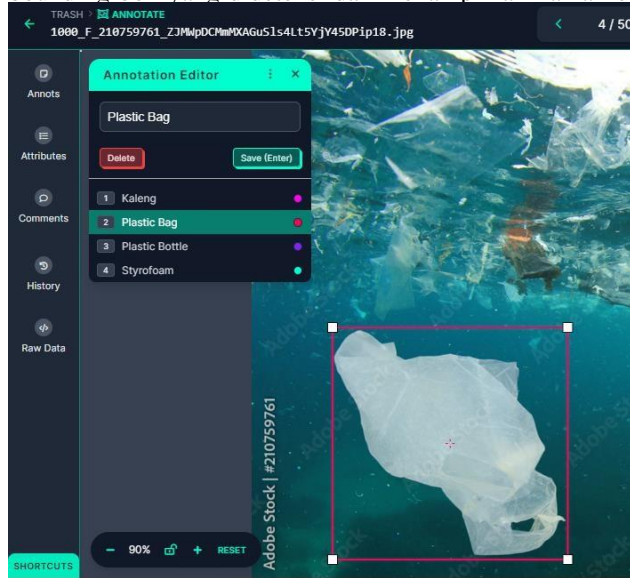
GAMBAR III.2.
Source Code OpenCV Untuk RTSP

Gambar III.2. adalah *source code* untuk menampilkan output dari kamera CCTV secara *real-time* dengan menggunakan bahasa Python. Dengan menggunakan *source code* di atas, output kamera CCTV akan dapat dipanggil menggunakan python tanpa harus membuka web untuk *streaming* video secara *real-time*.

B. Anotasi Dataset

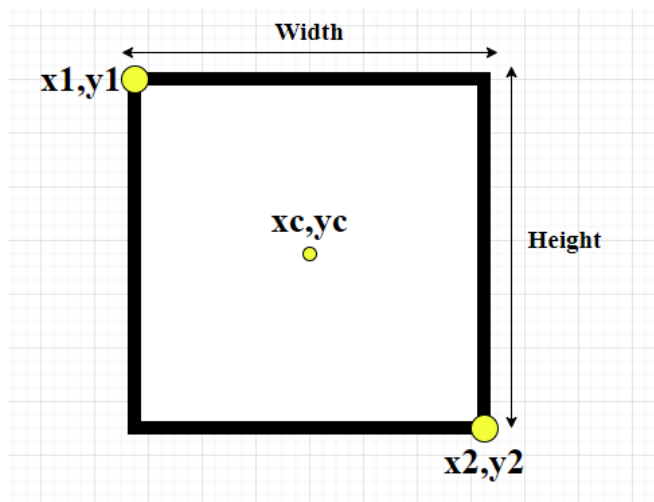
Pengnnotasian dataset dilakukan menggunakan platform Roboflow yang dimana pada platform tersebut memiliki beberapa fitur untuk penganotasian objek pada gambar yang akan dideteksi. Penganotasian dapat dilakukan dengan cara pemberian *bounding box* dan *labeling*. *Bounding box* merupakan persegi panjang yang mengelilingi suatu objek pada gambar. *Bounding box* merupakan kotak pembatas yang bertujuan untuk mengidentifikasi posisi objek yang terdapat pada gambar. Setelah memberi *bounding box* pada objek yang terdapat pada gambar, selanjutnya adalah memberi label. Setiap objek yang telah diberi *bounding box* memiliki sebuah kelas, setiap kelas dikelompokkan agar *bounding box* dapat mengklasifikasi jenis kelas apa yang terdeteksi pada algoritma YOLO tersebut. Kedua metode ini akan bekerja secara bersamaan dengan menampilkan output

bounding box serta menampilkan nilai keyakinan posisi bounding box yang dideteksi dan menampilkan nama kelas



GAMBAR III.3.
Contoh Anotasi Dataset

Pada Gambar III.3. merupakan contoh dari pembuatan dataset dengan menganotasi objek pada gambar. Dataset yang merupakan gambar diunggah ke platform roboflow untuk dianotasi sehingga gambar memiliki atribut yang diantaranya adalah titik sudut kiri atas, titik sudut kanan bawah, lebar, tinggi, kelas, dan nilai keyakinan.



GAMBAR IV.3.
Parameter Bounding Box

Pada Gambar III.3. Sebuah bounding box memiliki beberapa parameter atau atribut yang berfungsi untuk mesin mengidentifikasi dan menentukan kotak pembatas. Parameter tersebut adalah :

2. Kelas : benda yang terdapat pada area bounding box
3. x_1,y_1 : titik sudut kiri atas bounding box
4. x_2,y_2 : titik sudut kanan bawah bounding box
5. x_c,y_c : titik pusat dari bounding box
6. Width : lebar bounding box
7. Height : tinggi bounding box

1. Nilai Kepercayaan : Nilai seberapa besar objek yang terdapat dalam area bounding box sesuai dengan kelas yang telah ditentukan. Misal, kelas mobil, motor, dsb.

Adapun persamaan yang menentukan nilai parameter dari sebuah bounding box :

$$x_c = \frac{x_1 + x_2}{2} \quad (1)$$

Persamaan (1) merupakan persamaan yang menentukan nilai titik koordinat x untuk titik pusat bounding box.

$$y_c = \frac{y_1 + y_2}{2} \quad (2)$$

Persamaan (2) merupakan persamaan yang menentukan nilai titik koordinat y untuk titik pusat bounding box.

$$Width = x_2 - x_1 \quad (3)$$

Persamaan (3) merupakan persamaan yang menentukan nilai lebar dari sebuah bounding box.

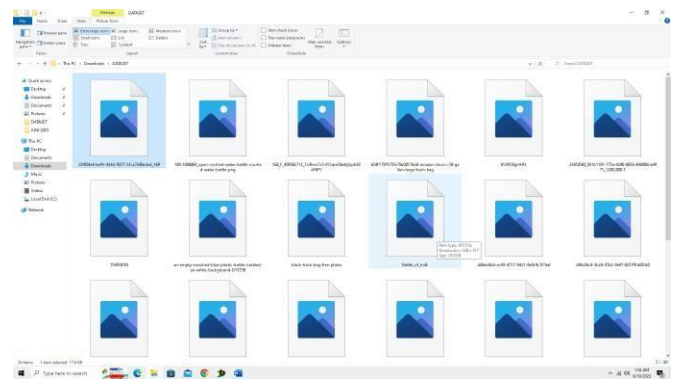
$$Height = y_2 - y_1 \quad (4)$$

Persamaan (4) merupakan persamaan yang menentukan nilai tinggi dari sebuah bounding box.

IV. HASIL DAN PEMBAHASAN

Pada bagian ini, penulis akan menampilkan dan menjelaskan dataset yang telah penulis buat untuk dijadikan data uji pada algoritma YOLO. Berikut beberapa penjelasan mengenai pembuatan dataset algoritma YOLO

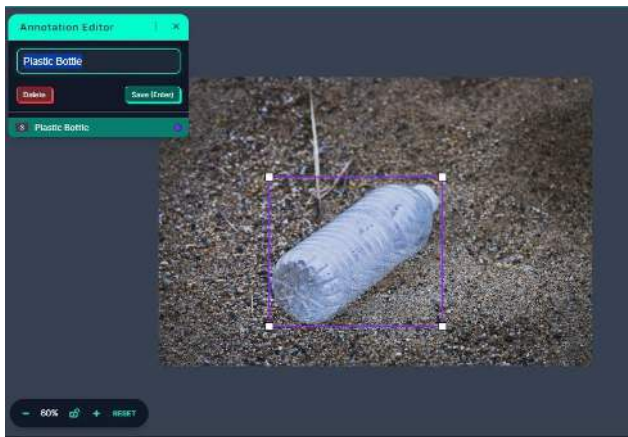
A. Pengumpulan Dataset



GAMBAR IV.1.
Pengumpulan Dataset

Pada Gambar IV.1. merupakan langkah awal untuk mengumpulkan dataset. Dataset yang dikumpulkan adalah berupa format gambar objek yang sesuai dengan penelitian. Penulis mengumpulkan 800 gambar untuk dijadikan dataset yang akan diunggah ke platform Roboflow untuk dilakukan pengantasian gambar sehingga gambar memiliki atribut untuk pembelajaran algoritma YOLO.

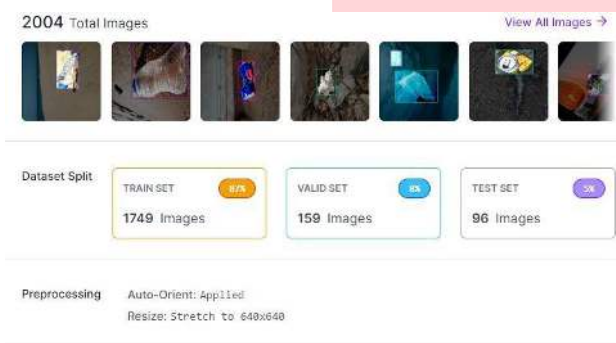
B. Anotasi Dataset



GAMBAR IV.2.
Anotasi Objek

Pada Gambar IV.2. merupakan penganotasian objek yang terdapat pada gambar yang sebelumnya telah diunggah ke platform Roboflow. Penganotasian objek dilakukan dengan cara memberi *bounding box* pada sekeliling objek dan memberikan kelas sesuai dengan objek yang telah diberi *bounding box* tersebut.

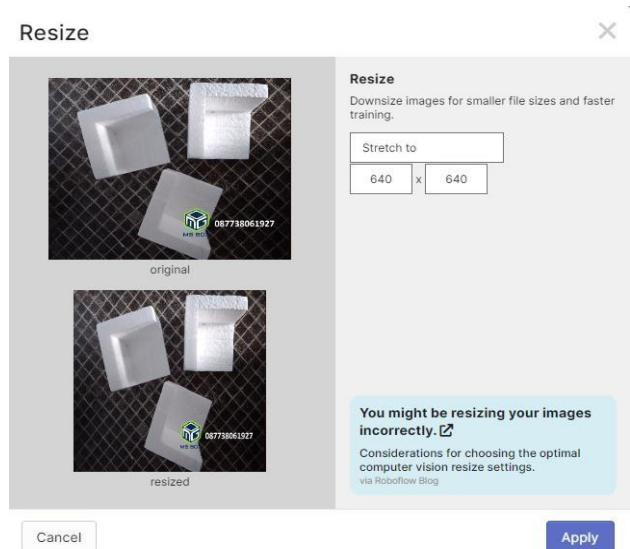
C. Partisi Dataset



GAMBAR IV.3.
Partisi Dataset

Pada Gambar IV.3. merupakan gambar dari partisi dataset. Partisi dataset merupakan pemecahan data pada seluruh dataset menjadi beberapa bagian. Pada algoritma YOLO, seluruh dataset akan dipartisi menjadi 3 bagian. Bagian pertama adalah Data Latih (Train Set), data latih merupakan bagian dataset yang akan menjadi bahan ajar untuk algoritma YOLO agar dapat memprediksi *bounding box* dan mengenali kelas objek yang berada didalam *bounding box* tersebut. Sesuai dengan gambar di atas, data latih cenderung lebih banyak dari bagian lainnya, hal tersebut dikarenakan agar algoritma YOLO dapat lebih banyak belajar memprediksi *bounding box* dan mengenali kelas objek yang terdapat didalam *bounding box* tersebut, sehingga algoritma YOLO dapat memprediksi *bounding box* dan mengenali kelas objek yang terdapat didalam *bounding box* lebih presisi yang akan berpengaruh terhadap akurasi pendeteksian.

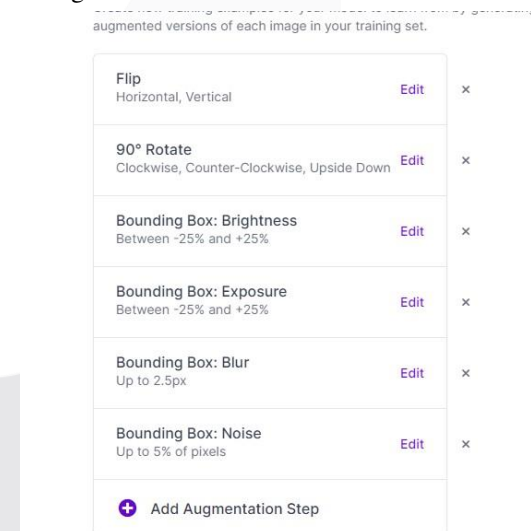
D. Pre-Processing Dataset



GAMBAR IV.4.
Pre-Processing Dataset

Pada Gambar IV.4. adalah bagian *Pre-Processing* Dataset. Pada bagian ini merupakan Langkah untuk menyeimbangkan dataset dengan ketentuan tertentu. *Pre-Processing* yang penulis gunakan adalah merubah ukuran semua dataset menjadi satu ukuran yang sama. Penulis menggunakan ukuran 640x640 untuk semua dataset. Sehingga dataset yang dibuat akan menjadi seimbang atau *balance*. Dengan dataset yang seimbang, hal ini akan mempengaruhi pembelajaran bagi algoritma YOLO, algoritma YOLO akan lebih konsisten untuk mempelajari dataset tersebut karena dataset tersebut telah diseimbangkan dengan sama rata.

E. Augmentasi Dataset



GAMBAR IV.5.
Augmentasi Dataset

Pada Gambar IV.5. merupakan bagian dari Augmentasi Dataset. Pada bagian ini, dataset yang telah diperoleh akan dimodifikasi atau dimanipulasi dengan cara- cara tertentu. Penulis memodifikasi dataset dengan cara mengatur *Flip*, *Rotate*, *Brightness Adjust*, *Exposure Adjust*, *Blur*, dan *Noise* pada gambar dataset. Tujuan memodifikasidataset adalah untuk memperbanyak dataset. Dataset yang telah dimodifikasi akan menjadi model dataset yang

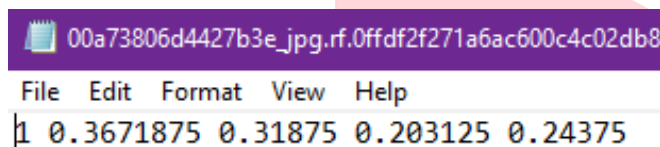
baru dan dapat digunakan untuk melatih algoritma YOLO.

F. Export Dataset

test	4,520,669	4,520,669	File folder	7/16/2023 3:41 ...
train	114,380,423	114,380,423	File folder	7/16/2023 3:41 ...
valid	7,885,054	7,885,054	File folder	7/16/2023 3:41 ...
data.yaml	339	339	YAML File	7/16/2023 3:41 ... 3344C167
README.dataset...	134	134	Text Document	7/16/2023 3:41 ... FA581542
README.robofl...	1,522	1,522	Text Document	7/16/2023 3:41 ... 341EEFC2

GAMBAR IV.6.
File Dataset

Pada Gambar IV.6. merupakan hasil dari semua langkah pembuatan dataset, dilihat pada ketiga file di atas, itu merupakan partisi dataset yang dimana setiap file berisi mengenai gambar objek yang telah diberi *bounding box*, *pre-processing* data, dan augmentasi data. Pada file tersebut juga terdapat berupa file .txt yang isinya merupakan titik koordinat dari bounding box yang terdapat pada dataset, hal ini bertujuan agar algoritma YOLO dapat menentukan posisi dari bounding box pada sebuah gambar.



GAMBAR IV.7.
Titik Koordinat Bounding Box

Pada Gambar IV.7. merupakan titik koordinat dari sebuah bounding box pada gambar. Pada gambar di atas menunjukkan ada 4 titik koordinat dikarenakan sebuah bounding box merupakan bentuk kotak atau persegi panjang yang mengelilingi sekitar objek pada gambar.

V. KESIMPULAN

Semakin berkembangnya teknologi, Pendeteksi Sampah Menggunakan Algoritma YOLO yang dibantu dengan kamera CCTV ini dapat memberikan kemudahan bagi pengguna untuk memantau sampah pada sungai dari kejauhan. Pengguna tidak harus langsung terjun ke area sekitar sungai untuk mengecek kondisi area sekitar sungai, pengguna dapat memantau area sekitar sungai sekaligus mendapatkan informasi sampah melalui deteksi algoritma YOLO dari kejauhan sehingga pengguna dapat mengantisipasi sampah sebelum sampah tersebut menumpuk.

Pembuatan dataset merupakan hal penting untuk pendeteksian ini, pembuatan dataset bertujuan untuk sistem dapat mengenali berbagai objek sampah yang terdapat pada sungai, sehingga sistem apt mempresentasikannya melalui kamera CCTV.

Adapun langkah-langkah untuk pembuatan dataset, diantaranya adalah, pengumpulan dataset, anotasi objek, partisi data, *pre-processing* data, augmentasi data, dan export dataset.

Sistem ini kelak dapat mempermudah petugas kebersihan untuk mengantisipasi terjadinya penumpukan sampah pada sungai dengan cara memantau dan mendapatkan informasi menggunakan kamera CCTV yang dikonfigurasi dengan algoritma YOLO sebagai pendeteksi sampah.

REFERENSI

- [1] D. Schwalm, "Detecting objects on IP camera video with Tensorflow and OpenCV," Analytics Vidhya, 22 March 2020. [Online]. Available: <https://medium.com/analytics-vidhya/detecting-objects-on-ip-camera-video-with-tensorflow-and-opencv-e2c25297a75a>. [Accessed 09 September 2023].
- [2] S. Theophilus, "Roboflow: Converting Annotations for Object Detection," Analytics Vidhya, 03 September 2021. [Online]. Available: <https://medium.com/analytics-vidhya/converting-annotations-for-object-detection-using-roboflow-5d0760bd5871>. [Accessed 09 September 2023].
- [3] V. S. Subramanyam, "Basics of Bounding Boxes," Analytics Vidhya, 16 January 2021. [Online]. Available: <https://medium.com/analytics-vidhya/basics-of-bounding-boxes-94e583b5e16c>. [Accessed 09 September 2023].
- [4] PyShine, "OpenCV and Real time streaming protocol (RTSP)," PyShine, 01 November 2020. [Online]. Available: <https://pyshine.com/Real-time-streaming-protocol-and-opencv-in-Python/>. [Accessed 09 September 2023].
- [5] S. Mulia, "Mengenal OpenCV Dalam Python: Pengertian, Sejarah, Dukungan pada OS, Fitur-fitur," IDMETAFORA, 31 August 2022. [Online]. Available: <https://idmetafora.com/news/read/1177/Mengenal-OpenCV-Dalam-Python-Pengertian-Sejarah-Dukungan-pada-OS-Fitur-fitur.html>. [Accessed 09 September 2023].
- [6] Q. Lina, "Apa itu Convolutional Neural Network?," Medium.com, 02 January 2019. [Online]. Available: <https://medium.com/@16611110/apa-itu-convolutional-neural-network-836f70b193a4>. [Accessed 09 September 2023].
- [7] N. K. Nissa, "Cara Kerja Object Detection dengan YOLO (You Only Look Once)," Pacmann, 23 March 2023. [Online]. Available: <https://pacmann.io/blog/cara-kerja-object-detection-dengan-yolo>. [Accessed 09 September 2023].
- [8] A. Salim, "Object Detection (Case: Plat Detection)," BISA.AI, 01 April 2020. [Online]. Available: <https://medium.com/bisa-ai/object-detection-case-plat-detection-7cb5f53682ae>. [Accessed 09 September 2023].