

# Sistem Pendeteksi Sampah Plastik menggunakan Algoritma YOLOv8

Alief Rahman Hakim<sup>1</sup>, Umar Taufiq<sup>1\*</sup>, Dinar Nugroho Pratomo<sup>1</sup>, Firma Syahrian<sup>1</sup>, Candra Febri Nugraha<sup>1</sup>, Ahmad Adhiim Muthahhari<sup>1</sup>

<sup>1</sup>Departemen Teknik Elektro dan Informatika, Sekolah Vokasi, Universitas Gadjah Mada;

aliefrahman02@mail.ugm.ac.id

dinar.nugroho.p@ugm.ac.id

fsyahrian@ugm.ac.id

candra.febri.nugraha@ugm.ac.id

ahmad.adhiim.m@ugm.ac.id

\*Korespondensi: umartaufiq8284@ugm.ac.id;

**Abstract** – Plastic waste is a global issue with an ever-increasing volume. This study utilizes sample data from the Yogyakarta region, where the growing volume of waste remains inadequately managed. Identification and sorting of plastic waste based on resin types are essential to facilitate effective recycling. This research develops a plastic waste sorting detection system using a deep learning approach with the YOLOv8-cls architecture to classify plastics such as High-Density Polyethylene, Polyethylene Terephthalate, Polypropylene, and Polystyrene. Given the dynamic characteristics of plastic waste and the emergence of new forms, the study retrains the baseline YOLOv8n-cls model by comparing replay and fine-tuning methods, exploring freeze layer variations from 0 to 9. Results indicate that the replay method performs best, achieving stable F1-scores on old data in the range of 98–99%, and in some cases, matching or surpassing the baseline model with F1-scores of 99.16%–99.29%, demonstrating the model's reliability in recognizing old objects. Meanwhile, the F1-scores for new data ranged between 85–88%, which is considered high. Analysis and evaluation show that the replay method is more effective and stable in retaining knowledge of old objects while learning new ones. System testing through user acceptance testing achieved a score of 98.4% for the system's ability to sort plastic waste. This detection system is expected to assist the plastic waste sorting industry in handling unmanaged plastic waste quickly and accurately.

**Keywords** – Image Classification, Deep Learning, YOLO, Plastic Waste

**Intisari** – Sampah plastik merupakan permasalahan global dengan volume yang terus meningkat. Penelitian ini menggunakan sampel data dari wilayah Yogyakarta, dengan volume yang terus meningkat namun belum diimbangi penanganan memadai. Identifikasi serta pemilahan sampah plastik berdasarkan gambar pada jenis resin diperlukan untuk memfasilitasi daur ulang efektif. Penelitian ini mengembangkan sistem pendeteksi pemilahan sampah plastik menggunakan pendekatan *deep learning* dengan arsitektur YOLOv8-cls untuk mengklasifikasikan plastik *High Density Polyethylene*, *Polyethylene Terephthalate*, *Polypropylene*, dan *Polystyrene*. Objek sampah plastik memiliki karakteristik yang dinamis dan munculnya bentuk-bentuk baru, penelitian ini melakukan *retraining* pada model *baseline* YOLOv8n-cls dengan membandingkan metode *replay* dan *fine-tuning*, serta mengeksplorasi variasi *freeze* layer dari 0 hingga 9. Hasil menunjukkan metode *replay* menunjukkan kinerja terbaik dengan F1-score pada data lama stabil di kisaran 98–99%, bahkan pada beberapa kasus setara atau melebihi model *baseline* dengan F1-score 99,16%–99,29%, menandakan model tetap andal mengenali objek lama. Sementara itu, F1-Score pada data baru berada di rentang 85–88%, yang masih tergolong tinggi. Analisis dan evaluasi menunjukkan metode *replay* lebih efektif dan stabil dalam mempertahankan pengetahuan objek lama sekaligus mempelajari objek baru. Pengujian sistem menggunakan *user acceptance testing* mendapat nilai 98,4% terhadap kemampuan sistem dalam memilah sampah plastik. Sistem pendeteksian ini diharapkan dapat membantu industri pemilahan sampah plastik dengan cepat dan akurat dalam menangani sampah plastik yang belum tertangani dengan baik.

**Kata kunci** – Klasifikasi Gambar, Pembelajaran Mendalam, YOLO, Sampah Plastik

## I. PENDAHULUAN

Masalah pengelolaan sampah plastik menjadi tantangan signifikan di Provinsi Yogyakarta, dengan peningkatan volume setiap tahunnya yang belum diimbangi oleh penanganan yang efektif. Sampah plastik menyumbang hampir 28% dari total timbunan sampah di daerah ini dan berpotensi menimbulkan dampak lingkungan yang serius oleh media [1], [2]. Salah satu solusi yang menjanjikan adalah implementasi sistem pemilahan sampah plastik otomatis berbasis teknologi *deep learning*, khususnya *Convolutional Neural Network* (CNN), untuk mendeteksi dan mengklasifikasikan jenis resin plastik. Beberapa studi terdahulu telah membuktikan efektivitas pendekatan ini [3] yang mendeteksi plastik *Polyethylene Terephthalate* (PET), *High-Density Polyethylene* (HDPE), *Polypropylene* (PP), dan *Polystyrene* (PS) menggunakan 15 lapisan CNN, serta

penelitian [4] yang memanfaatkan arsitektur YOLO untuk mendeteksi plastik PET dan PET-G pada ban berjalan.

Meskipun menjanjikan, penerapan *machine learning* dalam pendeteksian sampah plastik dihadapkan pada tantangan kualitas dataset yang belum sepenuhnya merepresentasikan kondisi dunia nyata [5]. Dalam pendekatan *supervised learning*, model sangat bergantung pada kelengkapan anotasi. Setiap objek sampah plastik menuntut anotasi yang komprehensif karena memiliki tingkat deformabilitas yang ekstrem dan *intra-class variance* yang sangat tinggi. Sebagai contoh, kelas plastik PP dapat ditemukan dalam wujud gelas utuh, sedotan, hingga masker yang sudah hancur atau terlipat. Jika dataset pelatihan tidak mencakup berbagai variasi bentuk, posisi, dan tingkat kerusakan fisik tersebut, model cenderung mengalami *overfitting* dengan hanya menghafal bentuk ideal. Akibatnya, model gagal mengekstraksi fitur visual pada tumpukan

sampah yang berserakan atau terdistorsi di lapangan. Oleh karena itu, peningkatan kualitas anotasi yang mewakili kondisi dinamis menjadi syarat mutlak agar jaringan saraf dapat memetakan fitur visual secara akurat ke dalam kode resin plastik. Untuk menjawab kebutuhan komputasi visual yang efisien, penelitian ini mengadopsi model YOLOv8-clas. Berbeda dengan generasi pendahulunya yang berfokus utama pada deteksi objek berbasis *bounding box*, YOLOv8 yang dirilis oleh Ultralytics (2023) menyediakan arsitektur khusus klasifikasi gambar (YOLOv8-clas) yang sangat ringan dan presisi. Pemilihan model ini didasari oleh penggunaan *backbone* CSPDarknet53 yang telah dioptimalkan untuk mengekstraksi fitur hierarkis secara lebih baik dibandingkan arsitektur CNN konvensional. Tinjauan literatur oleh Terven dan Cordova-Esparza [6] menegaskan bahwa YOLOv8 menawarkan keseimbangan terbaik antara kecepatan inferensi (*real-time processing*) dan tingkat akurasi tinggi. Keunggulan komputasi yang ringan ini menjadikannya sangat ideal untuk dikembangkan lebih lanjut menggunakan metode *continual learning*.

Namun, tantangan muncul ketika model YOLOv8-clas yang telah dilatih menggunakan dataset standar seperti WaDaBa sering mengalami kesalahan klasifikasi (*misclassified*) saat dihadapkan pada objek sampah plastik berwujud baru. Di sisi lain, melatih ulang model secara keseluruhan dari awal setiap kali ada tambahan data baru sangatlah tidak efisien secara waktu dan komputasi. Oleh karena itu, pendekatan *Continual Learning* (CL) diperlukan, dengan fokus utama mengatasi *catastrophic forgetting*, yaitu hilangnya memori model terhadap pengetahuan lama saat mempelajari pengetahuan baru.

Berdasarkan tinjauan literatur [7], terdapat beberapa pendekatan untuk mengatasi hal ini, seperti metode isolasi parameter dan metode regularisasi (contohnya *Elastic Weight Consolidation/EWC*). Sayangnya, metode regularisasi seringkali gagal mempertahankan performa ketika menghadapi data dengan pergeseran distribusi bentuk yang drastis. Sebagai solusi yang lebih tangguh, penelitian ini mengimplementasikan metode *replay*. Strategi berbasis memori (*memory-based*) ini memungkinkan model untuk belajar dari aliran data baru dengan menyisipkan sebagian sampel lama ke dalam proses pelatihan. Pemilihan ini didukung oleh penelitian [8] dan sangat disarankan oleh pengembang YOLOv8 Jocher [9] karena keterbatasan dan kemudahan penggunaan model.

Secara singkat, penelitian ini bertujuan untuk melakukan pelatihan ulang (*re-training*) pada model YOLOv8-clas menggunakan pendekatan *replay* dalam mendeteksi jenis resin sampah plastik. Kinerja metode *replay* ini akan dikomparasikan secara langsung dengan metode *fine-tuning* (*transfer learning*) konvensional pada dataset baru yang memiliki karakteristik berbeda dari dataset awal. Evaluasi performa dari kedua metode tersebut akan diukur menggunakan metrik *F1-score* dan *forgetting*, dengan tujuan menghasilkan sistem pendeteksi sampah plastik yang lebih

akurat, dinamis, dan memiliki stabilitas pengetahuan yang tinggi.

## II. DASAR TEORI

### A. Machine Learning

*Machine learning* adalah cabang kecerdasan buatan atau *Artificial Intelligence* (AI) yang berfokus pada pengembangan algoritme dan model statistik yang memungkinkan sistem komputer meningkatkan kinerjanya pada tugas tertentu melalui pengalaman, tanpa diprogram secara eksplisit T. M. Mitchell [10]. Pembelajaran ini mencakup berbagai teknik yang memungkinkan komputer belajar dari data dan membuat prediksi atau keputusan berdasarkan data C. M. Bishop [11].

Metode-metode ini secara luas dapat dikategorikan menjadi *Supervised learning*, *unsupervised learning*, dan *reinforcement learning* E. Alpaydin [12]. *Supervised learning* melibatkan pelatihan model pada data berlabel untuk membuat prediksi atau klasifikasi. *Unsupervised learning* bertujuan untuk menemukan pola atau struktur tersembunyi dalam data tak berlabel. *Reinforcement learning* memungkinkan agen mempelajari perilaku optimal melalui interaksi dengan lingkungan [13].

### B. Continual learning

Menurut [14] *Continual Learning* (CL) atau pembelajaran berkelanjutan adalah masalah pembelajaran berurutan di mana sistem diharapkan untuk beradaptasi dengan aliran masukan yang berasal dari keadaan yang dinamis. Situasi yang berubah ini sering dimodelkan sebagai serangkaian tugas yang harus dipelajari secara berurutan tanpa melupakan tugas-tugas yang telah dipelajari sebelumnya. Fokus utama CL dalam kecerdasan buatan adalah mengatasi *catastrophic forgetting*, yaitu kecenderungan model jaringan saraf untuk melupakan informasi lama saat mempelajari informasi baru. Oleh karena itu, pendekatan CL bertujuan untuk memungkinkan sistem belajar, menyimpan, mengingat, dan mentransfer pengetahuan secara berkelanjutan, mirip dengan bagaimana otak manusia bekerja.

*Replay* atau *rehearsal* adalah metode dalam *continual learning* yang digunakan untuk mengatasi *catastrophic forgetting* dengan cara menyimpan sebagian sampel dari tugas-tugas sebelumnya dan menggunakannya kembali dalam pelatihan di masa mendatang. *Replay* berakar dari penelitian awal tentang *continual learning*, di mana metode ini dikembangkan untuk mempertahankan informasi dari tugas sebelumnya dengan menyimpan contoh data tertentu dalam sebuah *memory buffer*. Memori ini kemudian digunakan untuk melatih model bersamaan dengan data baru, sehingga model dapat mengingat kembali informasi lama [14].

### C. Transfer learning

*Transfer learning* adalah teknik dalam pembelajaran mesin di mana model yang telah dilatih pada satu tugas digunakan kembali sebagai titik awal untuk tugas lain yang terkait. Metode ini memanfaatkan pengetahuan yang diperoleh dari dataset sumber untuk meningkatkan kinerja

pada dataset target, terutama ketika data pada tugas baru yang terbatas. *Transfer learning* memungkinkan model untuk lebih cepat beradaptasi dengan tugas baru dan sering kali menghasilkan kinerja yang lebih baik dibandingkan dengan melatih model dari awal. Ada beberapa pendekatan *transfer learning* salah satu yaitu pendekatan *fine-tuning*.

*Fine-tuning* [15] adalah teknik dalam *transfer learning* di mana model yang sudah dilatih pada tugas sebelumnya disesuaikan kembali pada tugas baru dengan melakukan pembaruan parameter pada *layer* tertentu. *Fine-tuning* terjadi setelah *transfer learning*, di mana parameter model yang ditransfer dapat diadaptasi dengan dataset target untuk meningkatkan performa. Sebagai contoh, beberapa layer dari model dapat dibekukan (*frozen*) agar tidak diperbarui, sementara *layer* lain diperbolehkan belajar ulang.

#### D. YOLO

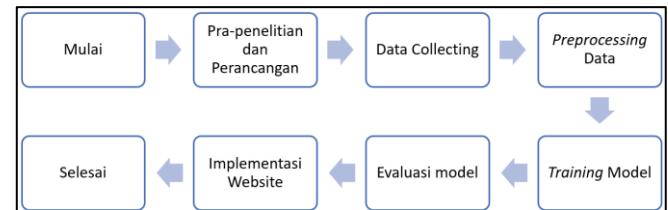
Arsitektur *You Only Look Once* (YOLO) [16] merupakan model *deep learning* berbasis CNN untuk deteksi objek yang dapat mengidentifikasi dan mengklasifikasikan objek dalam satu lintasan gambar dengan cepat dan akurat. Meskipun ada banyak versi dan varian dari arsitektur YOLO, peningkatan dari versi sebelumnya ke versi terbaru sebagian besar terdiri dari kombinasi berbagai jenis arsitektur CNN, optimisasi tahap *post-processing*, dan rangkaian antar lapisan *neural network*. Secara umum, struktur jaringan YOLO kontemporer terdiri dari tiga lapisan utama: *backbone*, *neck*, dan *head*.

*Backbone* berfungsi sebagai ekstraktor fitur dari gambar input. *Backbone* bertanggung jawab untuk mengekstraksi representasi fitur hierarkis yang kaya, yang akan digunakan oleh bagian selanjutnya *Neck* dan *Head* untuk mendeteksi objek. Arsitektur YOLO menggunakan metode *convolution* pada bagian *backbone* nya yaitu pada YOLOv8 menggunakan CSPDarknet53. *Neck* secara umum berfungsi sebagai penghubung *backbone* dan *head* untuk mempercepat deteksi objek. *Head* adalah bagian akhir dari model yang bertanggung jawab untuk melakukan prediksi akhir, yaitu koordinat *bounding box*, skor *confidence*, dan kelas objek.

*Head* Pada arsitektur YOLOv8-klasifikasi (YOLOv8-cl), *head* merupakan komponen yang mengubah representasi fitur yang dihasilkan oleh *backbone* menjadi prediksi label kelas. Tahapan ini diawali dengan proses *pooling* (misalnya *global average pooling*) yang bertujuan mengubah peta fitur berdimensi spasial menjadi sebuah vektor berdimensi tetap. Selanjutnya, vektor ini akan diproyeksikan melalui lapisan *fully connected* (FC) yang jumlah neuronnya sesuai dengan banyaknya kelas yang akan diklasifikasikan. Hasil akhir dari lapisan FC berupa nilai logit atau skor untuk tiap kelas, yang kemudian diproses (misalnya melalui fungsi *softmax*) untuk menghasilkan prediksi akhir. Dengan demikian, *head* pada YOLOv8-cl berperan penting dalam mengekstrak informasi paling relevan dari keluaran *backbone* dan menerjemahkannya menjadi label kelas yang sesuai.

### III. METODOLOGI

Tahapan penelitian ini memiliki serangkaian tahapan. Mulai dari pra-penelitian, pengambilan data pengujian, Pembuatan dataset, pelatihan mode *machine learning*, hingga evaluasi. Serangkaian tahapan akhir tersebut dapat dilihat Gambar 1 pada diagram alur berikut:



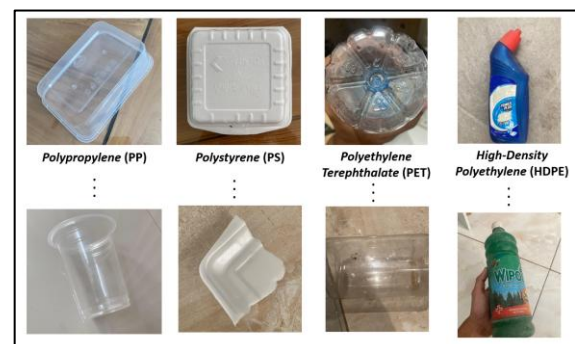
Gambar 1. Alur tahapan penelitian

#### A. Pra-penelitian dan Perancangan

Tahap pra-penelitian dan perancangan mencakup serangkaian proses yang dimulai dari analisis permasalahan, studi literatur, menentukan metode yang sesuai. Hasil analisis permasalahan menunjukkan bahwa model sebelumnya memiliki performa yang buruk dalam memprediksi objek sampah plastik di mana model menggunakan dataset WaDaBa belum cukup untuk mendeteksi sampah baru. Dengan itu penelitian ini ingin menggunakan metode *replay* untuk mengatasi bentuk atau kondisi dinamis pada objek sampah plastik. Menurut selaku pengembang YOLOv8, menyarankan penggunaan pendekatan metode *replay* sebagai solusi untuk mengatasi masalah tersebut. Walaupun begitu saat ini belum ada penelitian resmi yang melakukan pelatihan *replay* menggunakan *pretrain model* YOLO. Berdasarkan permasalahan tersebut, penelitian ini mengusulkan pengembangan sistem pendeteksi sampah plastik yang praktis untuk digunakan oleh pengguna.

#### B. Data Collecting

*Data collecting* merupakan kegiatan pengambilan sampel data pada kondisi dunia nyata. Pada penelitian ini data tersebut berupa objek sampah plastik khususnya yang memiliki label kode resin. Data terbagi menjadi dua yaitu data yang berasal pengumpulan mandiri atau survei pada penelitian ini, dan data yang berasal dari basis data yang bernama WadaBa [3] yang digunakan untuk kebutuhan metode *replay* nantinya. Gambar-gambar sampah plastik baru akan dijadikan data primer, lihat pada Gambar 2.

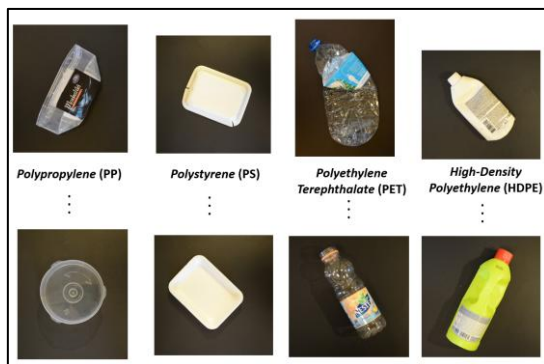


Gambar 2. Data primer

Pada penelitian ini, data sekunder merupakan hasil pembersihan dan penyesuaian dataset WadaBa yang asli (7 kelas) dari proyek sebelum penelitian ini atau dataset yang digunakan model YOLOv8 *baseline* untuk *training*. Di mana kelas pada dataset tersebut adalah PET, HDPE, PP, PS. Adapun distribusi data WadaBa yang digunakan pada model *baseline* akan diuraikan pada Tabel 1 bawah ini. Contoh sampel dataset WadaBa dapat dilihat pada Gambar 3.

Tabel 1. distribusi dataset WadaBa

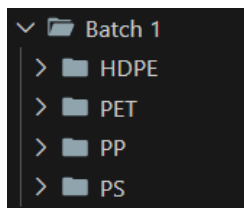
| <b>Class</b> | <b>Training</b> | <b>Validation</b> | <b>Test</b> |
|--------------|-----------------|-------------------|-------------|
| PET          | 462             | 51                | 121         |
| HDPE         | 478             | 53                | 69          |
| PP           | 317             | 35                | 28          |
| PS           | 335             | 37                | 48          |
| Total        | 1,592           | 176               | 266         |



Gambar 3. Data sekunder

Data yang digunakan pada penelitian ini adalah data pengujian (*Test*) dan *data training*. Data testing WadaBa tersebut berfungsi sebagai dataset pengujian pada pelatihan ini. *Data training* WadaBa tersebut dapat digunakan sebagai episodik memori pada *replay buffer* untuk metode *replay*.

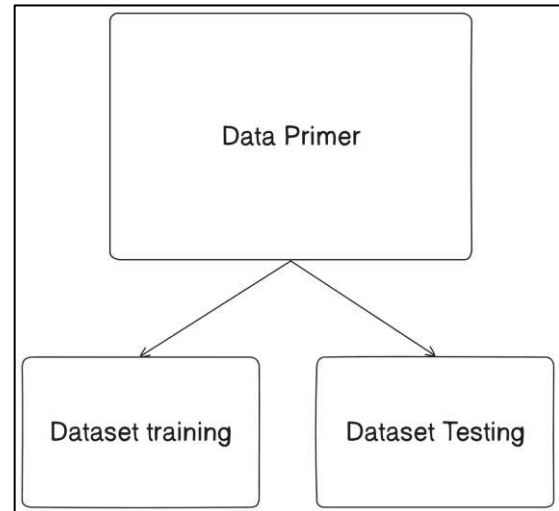
Pada tahap ini data-data hasil survei dan dataset sekunder dikumpulkan pada suatu folder dengan struktur yang dapat dilihat pada Gambar 4.



Gambar 4. Struktur dataset

### C. Preprocessing Data

Pada tahap ini sebelum melakukan pelatihan model, data primer perlu dilakukan pembagian data. Data primer dibagi ke 2 dataset, yaitu dataset *training* dan dataset *testing*. Hal ini dilakukan di awal agar objek plastik dengan bentuk tertentu tersebar merata di dataset testing, dengan begitu model *machine learning* dapat dievaluasi dengan baik. Pembagian dataset tersebut dibagi seperti pada Gambar 5.

Gambar 5. Pembagian dataset *primer*

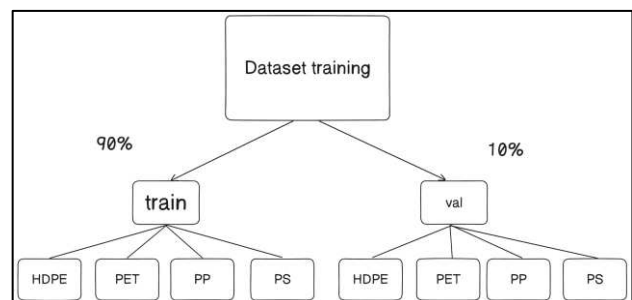
Setelah melakukan pembagian dataset primer, terdapat dataset testing primer. Dan untuk mengetahui performa model pada pengetahuan lama, terdapat dataset testing WadaBa yang digunakan pada model *baseline*. Kedua dataset testing tersebut dapat jelaskan distribusi dataset pada Tabel 2.

Tabel 2. Pembagian dan distribusi dataset primer

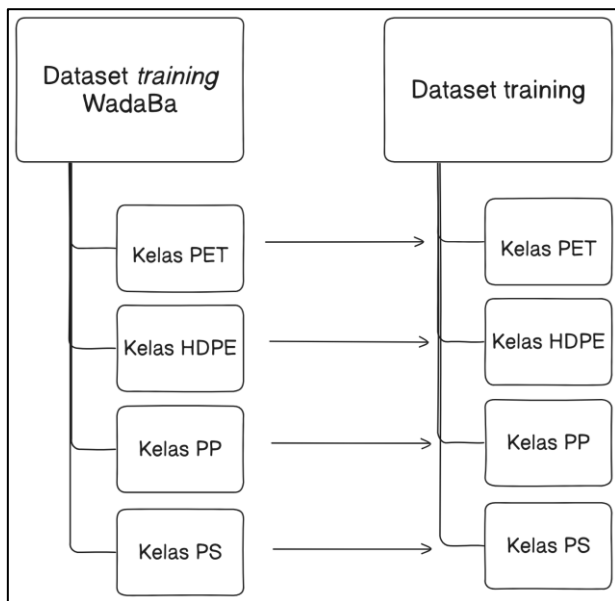
| <b>Class</b> | <b>Training</b> | <b>Test</b> |
|--------------|-----------------|-------------|
| PET          | 153             | 126         |
| HDPE         | 113             | 70          |
| PP           | 119             | 69          |
| PS           | 94              | 34          |
| Total        | 479             | 299         |

Kemudian *data training* dari data primer akan dilakukan augmentasi agar sampel objek lebih banyak. Augmentasi dilakukan dengan pengaturan +kecerahan 70%, -kecerahan 70%, +blur 40% .

Setelah melakukan augmentasi, Kemudian dilakukan *rationing* 90:10 dengan format *training* model YOLO yaitu terdapat 2 folder pada *root folder* yaitu *train* dan *val*. Folder *train* dan *val* memiliki masing-masing kelas PET, HDPE, PP, PS. Dataset ini yang akan digunakan pada metode *fine-tuning*. Gambar 6 berikut adalah Visualisasi dataset yang siap digunakan untuk *training*.

Gambar 6. Dataset *training*

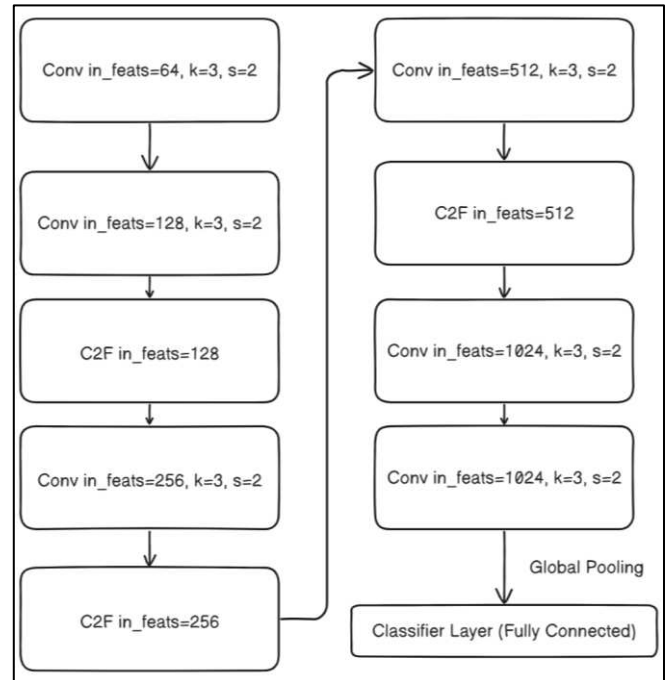
Pada metode *replay* terdapat penambahan data lama (*dataset training* Wadaba) sebagai *replay buffer*. Dengan adanya penambahan *replay buffer* tersebut akan terjadi bias jumlah dataset pada metode *replay*. Di mana tujuan penelitian ini membandingkan metode *replay* dan *fine-tuning*. Pada tahapan pembuatan dataset ini dilakukan penyetaraan jumlah dataset yang sama. Penyetaraan dataset dilakukan setelah melakukan augmentasi pada proses di atas. Dalam melakukan pengambilan *episodic memory* atau *replay buffer* penelitian ini menggunakan metode *random sampling* (penghapusan dan penambahan secara acak). Gambar 7 berikut ini merupakan visualisasi tingkat tinggi (jika dilihat dari persebaran kelas) dataset *training* ditambah dengan *replay buffer*.



Gambar 7. Visualisasi akuisisi sampel dataset Wadaba pada Dataset training

#### D. Arsitektur YOLOv8n-cls

Arsitektur menggunakan model YOLOv8-cls di mana suffix -cls menandakan model YOLO tersebut memiliki tugas klasifikasi. Pada model tugas klasifikasi terdapat 2 bagian pada arsitektur YOLOv8-cls, yaitu *Backbone* dan *Head*. *Backbone* berperan sebagai *feature extractor* pada istilah *deep learning* umum, sedangkan *Head* sebagai lapisan yang terdapat *fully connected*. Pada model tersebut terdapat 9 layer pada *backbone*-nya dan 1 lapisan untuk *head* atau layer klasifikasi. Lapisan-lapisan tersebut divisualisasikan secara umum yang dapat dilihat pada Gambar 8.



Gambar 8. Arsitektur atau struktur layer dari YOLOv8n-cls

Penelitian ini berfokus pada *backbone* di mana sebagai *feature extractor* untuk mengambil informasi dari gambar. Dengan adanya *layer convolution* yang ada penelitian akan melakukan eksperimen *freeze layer convolution* tersebut.

#### E. Pelatihan Model

Pada tahap pelatihan model penelitian ini menetapkan *hyperparameter* umum dan desain eksperimen dari setiap metode pelatihan *replay* dan *fine-tuning*. Tabel 3 berikut ini merupakan pengaturan *hyperparameter* umum.

Tabel 3. Pengaturan *hyperparameter* umum

| <i>Hyperparameter</i> | <i>Value</i>              |
|-----------------------|---------------------------|
| <i>Optimizer</i>      | <i>Default (AdamW)</i>    |
| <i>Epoch</i>          | 20                        |
| <i>Activation</i>     | <i>Relu, SiLu</i>         |
| <i>Loss function</i>  | <i>Cross-Entropy</i>      |
| <i>Learning rate</i>  | <i>Default (Adaptive)</i> |
| <i>Batch size</i>     | <i>Default (16)</i>       |

Pada pelatihan untuk mengomparasi kedua metode terdapat variasi eksperimen dengan melakukan pembekuan lapisan pada model YOLOv8-cls. Daftar varian model nantinya dapat dilihat pada Tabel 4 berikut.

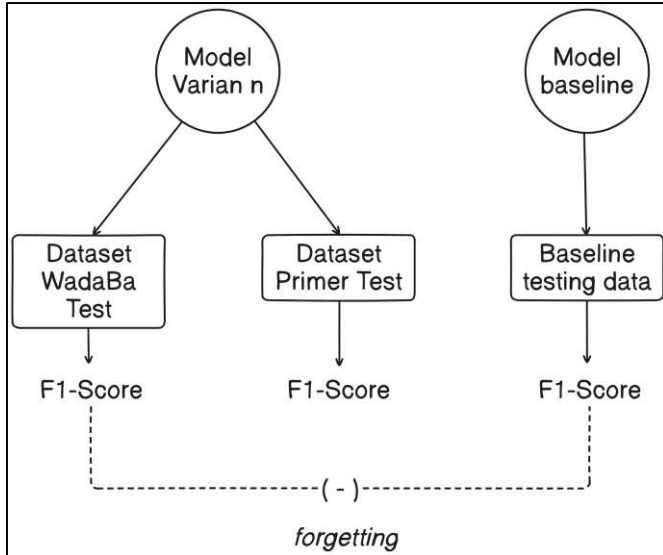
Tabel 4. Daftar varian pada pembekuan lapisan (*freeze layer*)

| Varian | Pembekuan Lapisan |
|--------|-------------------|
| 1      | 0                 |
| 2      | 1                 |
| 3      | 2                 |
| 4      | 3                 |
| 5      | 4                 |
| 6      | 5                 |
| 7      | 6                 |
| 8      | 7                 |
| 9      | 8                 |
| 10     | 9                 |

Pada pelatihan metode *replay* akan menggunakan dataset yang sudah akuisisi oleh *replay buffer* yaitu sebesar 25%, Sedangkan *Fine-tuning* akan menggunakan dataset yang tidak terdapat *episodic memory*.

#### F. Metrik Evaluasi

Penelitian ini menggunakan metrik evaluasi F1-Score, khusus pada penelitian *continual learning* terdapat metrik untuk mengetahui *forgetting* dapat dilihat pada Gambar 9.

Gambar 9. Skema evaluasi metrik *forgetting*

Berikut ini merupakan persamaan matematis pada metrik F1-Score *macro* setiap kelas (*c*) pada persamaan 1, rata-rata F1-score *macro* pada persamaan 2.

$$F1_c = \frac{2 \times Precision_c \times Recall_c}{Precision_c + Recall_c} \quad (1)$$

$$F1_{macro} = \frac{1}{c} \sum_{c=1}^c F1_c \quad (2)$$

#### G. Model Terbaik

Pada penelitian ini untuk menentukan model terbaik dapat dilihat dari keseimbangan antara retensi pengetahuan lama dan akuisisi pengetahuan baru seperti [17]. Dengan begitu model dikatakan mencapai stabilitas dan fleksibilitas dalam tujuan dari *continual learning*.

### IV. HASIL DAN PEMBAHASAN

Pada penelitian ini terdapat beberapa poin hasil dan pembahasan. Mulai dari *preprocessing* pada dataset *training*, *training* model, hasil evaluasi pelatihan, dan penelitian ini juga melakukan implementasi sistem pada aplikasi berbasis web.

#### A. Preprocessing data

Pada dataset *training* dari dataset primer akan dilakukan augmentasi. Setelah melakukan augmentasi adanya penyetaraan total sampel pada dataset untuk melakukan perbandingan metode *fine-tuning* dan *replay*. Berikut ini adalah hasil *rationing* dan distribusi sampel. Pada dataset *training* yang telah dilakukan augmentasi dapat dilihat pada Tabel 5.

Tabel 5. Dataset *finet-tuning*

| Kelas | Training                 | Validation            |
|-------|--------------------------|-----------------------|
| HDPE  | 310 + 104 <i>b</i> = 414 | 33 + 12 <i>b</i> = 45 |
| PET   | 230 + 76 <i>b</i> = 306  | 25 + 8 <i>b</i> = 33  |
| PP    | 242 + 80 <i>b</i> = 322  | 26 + 9 <i>b</i> = 35  |
| PS    | 191 + 63 <i>b</i> = 254  | 21 + 7 <i>b</i> = 28  |

Kemudian untuk dataset *replay*, Tabel 6 berikut ini adalah hasil untuk dataset *replay* yang digunakan untuk pelatihan *replay*. Sampel *replay buffer* (*b*) diambil dari dataset *baseline*.

Tabel 6. Dataset *replay*

| Kelas | Training                 | Validation            |
|-------|--------------------------|-----------------------|
| HDPE  | (117+38 <i>b</i> ) = 414 | (12+5 <i>b</i> ) = 45 |
| PET   | (71+23 <i>b</i> ) = 306  | (7+5 <i>b</i> ) = 33  |
| PP    | (30+10 <i>b</i> ) = 322  | (3+1 <i>b</i> ) = 35  |
| PS    | (44+14 <i>b</i> ) = 254  | (4+2 <i>b</i> ) = 28  |

#### B. Hasil Training

Hasil eksperimen dimulai dari varian 1 dengan *freeze* 0 sampai varian 10 dengan *freeze* 9. Pada penelitian ini terdapat 2 nilai F1-Score, yang pertama adalah F1-Score A adalah nilai pengujian pada dataset WadaBa (pengetahuan lama), F1-Score B adalah nilai pengujian dataset primer (pengetahuan baru). Seluruh hasil eksperimen dapat dilihat pada Tabel 7.

Tabel 7. Hasil seluruh *training* eksperimen

| Varian | Metode      | F1 A   | F1 B   | Forget | Time        |
|--------|-------------|--------|--------|--------|-------------|
| Base   |             | 99.16% | -      | -      | -           |
| 1      | Fine-Tuning | 52.95% | 93.50% | 46.21% | 47.46 menit |
| 2      |             | 48.95% | 95.28% | 50.22% | 52.02 menit |
| 3      |             | 47.42% | 94.20% | 51.74% | 50.94 menit |
| 4      |             | 50.75% | 94.01% | 48.41% | 44.04 menit |
| 5      |             | 46.99% | 93.75% | 52.17% | 43.5 menit  |
| 6      |             | 43.43% | 93.89% | 55.73% | 45.06 menit |
| 7      |             | 47.90% | 94.67% | 51.26% | 44.94 menit |
| 8      |             | 54.35% | 94.01% | 44.81% | 45.84 menit |
| 9      |             | 51.23% | 92.38% | 47.93% | 44.7 menit  |
| 10     |             | 59.08% | 93.01% | 40.08% | 44.16 menit |
| 1      | Replay      | 98.79% | 86.47% | -0.37% | 41.88 menit |
| 2      |             | 98.56% | 86.81% | -0.60% | 42.12 menit |
| 3      |             | 99.16% | 86.40% | 0.00%  | 43.56 menit |
| 4      |             | 98.60% | 87.97% | -0.56% | 43.92 menit |
| 5      |             | 99.16% | 87.60% | 0.00%  | 44.1 menit  |
| 6      |             | 99.29% | 85.88% | 0.13%  | 43.68 menit |
| 7      |             | 98.79% | 87.61% | -0.37% | 43.14 menit |
| 8      |             | 99.16% | 86.21% | 0.00%  | 43.56 menit |
| 9      |             | 99.16% | 85.33% | 0.00%  | 43.08 menit |
| 10     |             | 98.92% | 87.54% | -0.24% | 43.02 menit |

Dari hasil eksperimen dan pengujian tersebut, adapun hasil analisis dan evaluasi yang didapat. Berikut ini adalah aspek yang cukup penting disorot.

#### 1) Evaluasi pada data lama dan *forgetting*

Pada metode *fine-tuning* setelah pelatihan, F1-Score A turun drastis dibanding *Baseline* (99.16%) menjadi kisaran 43%–59%. Terjadi “*catastrophic forgetting*” yang signifikan. Nilai *forgetting* mencapai -40% hingga -56%. Sedangkan pada metode *replay* F1-Score A umumnya berada di kisaran 98.5%–99.3%, bahkan beberapa varian sama atau lebih tinggi dari *Baseline* (99.16%). *Forgetting* mendekati nol (sekitar -

0.6% sampai +0.13%). Sebagian varian benar-benar tidak mengalami penurunan kinerja (*forgetting* = 0%).

#### 2) Evaluasi pada data baru

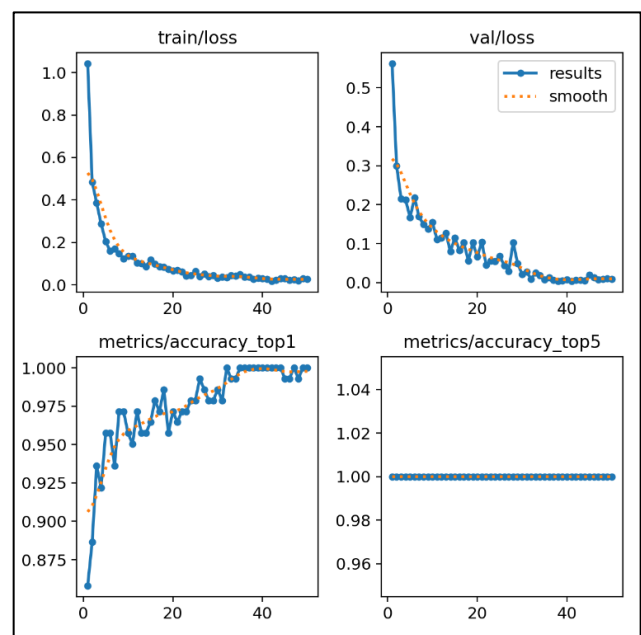
Metode *fine-tuning* mencapai F1-Score B tinggi, berkisar 92%–95%. Artinya, model berhasil beradaptasi dengan baik terhadap data baru meskipun mengorbankan performa pada data lama. Sedangkan pada metode *replay* Rata-rata F1-Score B di rentang 85%–88%. Tidak setinggi *Fine-Tuning*, namun masih cukup kompetitif. *Fine-Tuning* umumnya unggul pada adaptasi data baru, sedangkan *Replay* menyeimbangkan antara data lama dan baru. Ada kompromi antara mempertahankan kinerja lama (A) dan meningkatkan kinerja pada data baru (B).

#### 3) Waktu *training*

Pada metode *Fine-Tuning* Berkisar antara 43.5 menit hingga 52.02 menit (50 epoch). Ada variasi yang cukup lebar; varian dengan *freeze* 1 (varian 2) mencapai waktu terlama (52.02 menit). Metode *replay* sekitar 41.88 menit hingga 44.1 menit (50 epoch). Umumnya sedikit lebih singkat atau setara. *Replay* tidak memakan waktu jauh lebih lama meski perlu sampling data lama. *Fine-Tuning* pada beberapa kasus justru bisa lebih panjang tergantung jumlah layer yang tidak dibekukan.

#### 4) Model terbaik

Metode *replay* sangat baik dalam stabilitas menyeimbangkan performa lama dan baru. Model varian 5 adalah model dengan stabilitas terbaik dengan F1-score B 87.60% yang tergolong tinggi dan sambil menjaga F1 A yang sama dengan model *baseline* (99.16%). Dengan hasil *training* yang memuaskan, hasil *training* varian 5 *replay* dapat dilihat pada Gambar.10.

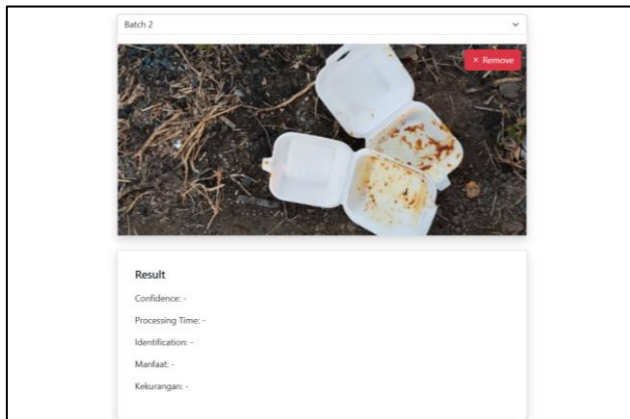
Gambar.10 Hasil *training* varian 5 metode *replay*

### C. Implementasi sistem berbasis web

Aplikasi diimplementasi berbasis web, yang di mana akan digunakan oleh pengguna sebagai sistem *monitoring* pemilahan sampah plastik serta sistem dapat melatih model.

#### 1) Antarmuka Halaman Prediksi

Berikut ini adalah hasil implementasi antarmuka untuk pengguna pada fitur prediksi atau *scan* tipe sampah plastik. Gambar 11 merupakan tangkapan layar peramban pada fitur prediksi yang digunakan pengguna untuk melakukan uji coba pendeteksian objek sampah plastik.



Gambar 11. Antarmuka halaman prediksi

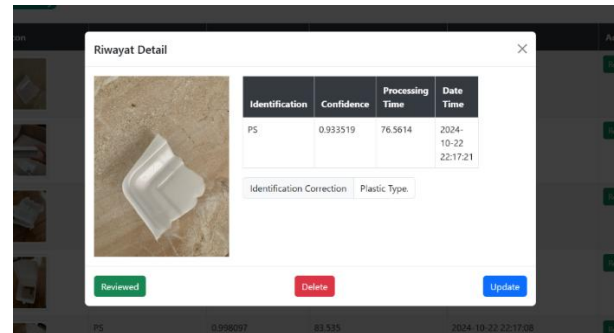
#### 2) Antarmuka Halaman Monitoring

Antarmuka riwayat prediksi berbentuk daftar riwayat prediksi dari yang telah dilakukan oleh pengguna. Antarmuka halaman riwayat prediksi dapat dilihat pada Gambar 12.

The screenshot shows a web interface for monitoring prediction history. It features a table with columns for #, Foto, Identification, Confidence, Processing Time, Date Time, and Action. The table contains 10 rows of data, each representing a prediction record. The interface is designed for easy viewing and management of prediction history.

Gambar 12. Antarmuka halaman riwayat prediksi

Pada aksi *review* akan muncul modal seperti Gambar 13 yang berfungsi sebagai anotasi ulang prediksi yang salah.



Gambar 13. Antarmuka monitoring prediksi

### D. User Acceptance Test

Hasil jawaban responden akan dihitung menggunakan bobot dengan skala *Likert 5* poin untuk menentukan nilai aktual. Jawaban responden dapat dilihat pada Tabel 8 berikut ini.

Tabel 8. Hasil jawaban responden UAT

| No.                                         | Pertanyaan                                                                        | Jawaban |    |   |    |     |
|---------------------------------------------|-----------------------------------------------------------------------------------|---------|----|---|----|-----|
|                                             |                                                                                   | SS      | S  | N | TS | STS |
| 1                                           | Apakah sistem bekerja dengan baik dalam pendeteksian sampah plastik?              | 0       | 5  | 0 | 0  | 0   |
| 2                                           | Apakah sistem mudah untuk digunakan?                                              | 3       | 2  | 0 | 0  | 0   |
| 3                                           | Apakah sistem memiliki respon yang cepat?                                         | 3       | 2  | 0 | 0  | 0   |
| 4                                           | Apakah sistem memiliki tampilan yang baik?                                        | 0       | 4  | 0 | 1  | 0   |
| 5                                           | Apakah sistem memiliki prospek yang baik dalam industri pemilahan sampah plastik? | 4       | 1  | 0 | 0  | 0   |
| Nilai Bobot                                 |                                                                                   | 5       | 4  | 3 | 2  | 1   |
| Jumlah Jawaban                              |                                                                                   | 13      | 14 | 0 | 1  | 0   |
| Nilai Aktual (Jumlah Jawaban × Nilai bobot) |                                                                                   | 65      | 56 | 0 | 2  | 0   |
| Jumlah Skor Aktual                          |                                                                                   | 123     |    |   |    |     |

Setelah mendapat skor aktual, dapat ditentukan pula skor ideal dan skor UAT dengan persamaan 3 di bawah ini:

$$\text{Skor ideal} = 5 \times 5 \times 5 = 125$$

$$\text{Skor UAT \%} = \frac{123}{125} \times 100 = 98.4 \% \quad (3)$$

## V. SIMPULAN

Eksperimen menggunakan variasi *freeze layer* (0–9) pada YOLOv8n-cls untuk membandingkan metode *fine-tuning* dengan *replay*. Hasil menunjukkan bahwa metode *replay* mampu menjaga kinerja Data Lama (A) di kisaran 98–99%, beberapa bahkan setara atau melebihi *baseline* (99,16–99,29%). Meski F1-Score Data Baru (B) sedikit lebih rendah (85–88%) dibanding beberapa varian *fine-tuning*, nilai ini tetap tergolong tinggi. Dengan demikian, metode *replay* efektif mempertahankan kemampuan model terhadap data lama sekaligus beradaptasi pada sampah plastik baru, menjadikannya solusi yang lebih stabil dan berkelanjutan.

Sistem dibangun menggunakan *framework* Flask dan database MySQL, dengan fitur koreksi prediksi secara manual oleh pengguna. Setelah diujikan melalui metode *User Acceptance Testing* (UAT), sistem meraih nilai 98,4%, menandakan bahwa pengguna menerima dan menyetujui fungsionalitasnya dengan sangat baik..

## REFERENSI

- [1] “Portal Berita Pemerintah Kota Yogyakarta - Menengok Pengelolaan Sampah Jadi ‘RDF’ di TPS 3R Nitikan Yogya.” Accessed: Dec. 26, 2024. [Online]. Available: <https://warta.jogjakota.go.id/detail/index/33358>
- [2] admin, “Yogyakarta di Bawah Bayang-Bayang Gunung Sampah,” *perkim.id*. Accessed: Dec. 26, 2024. [Online]. Available: <https://perkim.id/perkotaan/yogyakarta-di-bawah-bayang-bayang-gunungan-sampah/>
- [3] J. Bobulski and J. Piatkowski, “PET waste clasification method and Plastic Waste DataBase WaDaBa”.
- [4] J. Choi, B. Lim, and Y. Yoo, “Advancing Plastic Waste Classification and Recycling Efficiency: Integrating Image Sensors and Deep Learning Algorithms,” *Appl. Sci.*, vol. 13, no. 18, p. 10224, Sep. 2023, doi: 10.3390/app131810224.
- [5] O. Tamin, E. G. Moun, J. Ahmad Dargham, F. Yahya, S. Omatu, and L. Angeline, “Machine Learning for Plastic Waste Detection: State-of-the-art, Challenges, and Solutions,” in *2022 International Conference on Communications, Information, Electronic and Energy Systems (CIEES)*, Veliko Tarnovo, Bulgaria: IEEE, Nov. 2022, pp. 1–6. doi: 10.1109/CIEES55704.2022.9990703.
- [6] J. Terven and D. Cordova-Esparza, “A Comprehensive Review of YOLO Architectures in Computer Vision: From YOLOv1 to YOLOv8 and YOLO-NAS,” 2023, doi: 10.48550/ARXIV.2304.00501.
- [7] M. D. Lange *et al.*, “A continual learning survey: Defying forgetting in classification tasks,” *IEEE Trans. Pattern Anal. Mach. Intell.*, pp. 1–1, 2021, doi: 10.1109/TPAMI.2021.3057446.
- [8] A. Chaudhry *et al.*, “On Tiny Episodic Memories in Continual Learning,” Jun. 04, 2019, *arXiv*: arXiv:1902.10486. Accessed: Sep. 03, 2024. [Online]. Available: <http://arxiv.org/abs/1902.10486>
- [9] G. Jocher, “Is there any solution for Catastrophic Inference with YOLOv8 yet? · Issue #4282 · ultralytics/ultralytics,” GitHub. Accessed: Aug. 13, 2024. [Online]. Available: <https://github.com/ultralytics/ultralytics/issues/4282>
- [10] T. M. Mitchell, *Machine learning*, Nachdr. in McGraw-Hill series in Computer Science. New York: McGraw-Hill, 2013.
- [11] C. M. Bishop, *Pattern Recognition and Machine Learning*, Softcover reprint of the original 1st edition 2006 (corrected at 8th printing 2009). in Information science and statistics. New York, NY: Springer New York, 2016.
- [12] E. Alpaydin, *Introduction to machine learning*, Fourth edition. in Adaptive computation and machine learning series. Cambridge, Massachusetts: The MIT Press, 2020.
- [13] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. in Adaptive computation and machine learning. Cambridge, Mass: The MIT press, 2016.
- [14] B. Wickramasinghe, G. Saha, and K. Roy, “Continual Learning: A Review of Techniques, Challenges, and Future Directions,” *IEEE Trans. Artif. Intell.*, vol. 5, no. 6, pp. 2526–2546, Jun. 2024, doi: 10.1109/TAI.2023.3339091.
- [15] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, “How transferable are features in deep neural networks?,” 2014, doi: 10.48550/ARXIV.1411.1792.
- [16] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You Only Look Once: Unified, Real-Time Object Detection,” May 09, 2016, *arXiv*: arXiv:1506.02640. Accessed: Sep. 15, 2024. [Online]. Available: <http://arxiv.org/abs/1506.02640>
- [17] S.-A. Rebuffi, A. Kolesnikov, G. Sperl, and C. H. Lampert, “iCaRL: Incremental Classifier and Representation Learning,” Apr. 14, 2017, *arXiv*: arXiv:1611.07725. Accessed: Aug. 29, 2024. [Online]. Available: <http://arxiv.org/abs/1611.07725>