Deep learning-based Methods for Detecting and Quantifying floating litter in Riverine Environments

Jia, T.

DOI
10.4233/uuid:46d1a28c-eb01-4f63-aa33-1bd8d866e52e

Publication date
2025

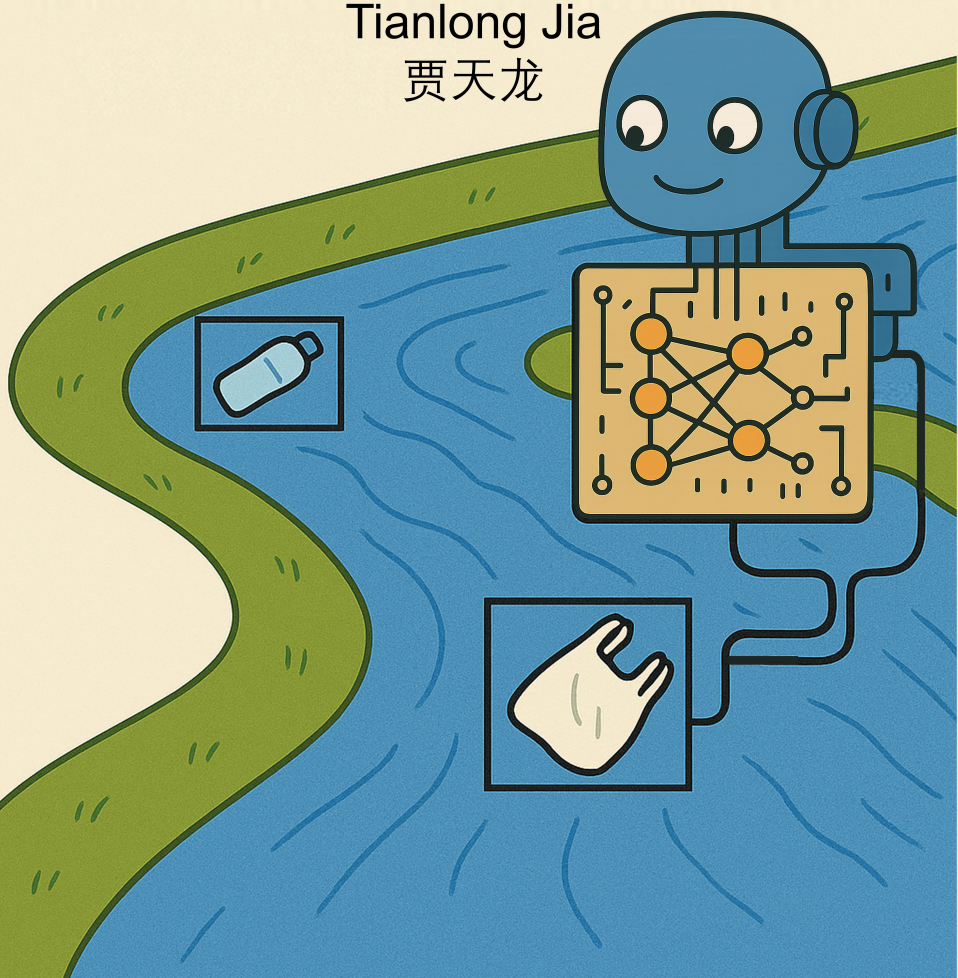
Document Version
Final published version

Citation (APA)
Jia, T. (2025). *Deep learning-based Methods for Detecting and Quantifying floating litter in Riverine Environments.* [Dissertation (TU Delft), Delft University of Technology].
https://doi.org/10.4233/uuid:46d1a28c-eb01-4f63-aa33-1bd8d866e52e

Important note
To cite this publication, please use the final published version (if applicable).
Please check the document version above.

Deep Learning-based Methods for Detecting and Quantifying Floating Litter in Riverine Environments

Tianlong Jia
贾天龙



Deep learning-based Methods for Detecting and Quantifying floating litter in Riverine Environments

DEEP LEARNING-BASED METHODS FOR DETECTING AND QUANTIFYING FLOATING LITTER IN RIVERINE ENVIRONMENTS

Dissertation

for the purpose of obtaining the degree of doctor
at Delft University of Technology
by the authority of the Rector Magnificus, prof. dr. ir. T.H.J.J. van der Hagen,
chair of the Board for Doctorates
to be defended publicly on
Monday 8 September 2025 at 17:30 o'clock

by

Tianlong JIA

Master of Science in Hydraulic Engineering,
Huazhong University of Science and Technology, China
born in Jiamusi, China.

This dissertation has been approved by the promotors.

Composition of the doctoral committee:

Rector Magnificus,	chairperson
Prof. dr. ir. Z. Kapelan,	Delft University of Technology, promotor
Dr. ir. R. Taormina,	Delft University of Technology, copromotor
Ir. R. de Vries,	Noria Sustainable Innovators, external adviser
Independent members:	
Prof. dr. ir. N.C. van de Giesen	Delft University of Technology
Dr. ir. Y. Ding	Delft University of Technology
Dr. ir. T.H.M. van Emmerik	Wageningen University and Research
Prof. dr. ir. G. Fu	University of Exeter
Prof. dr. ir. R. Uijlenhoet (reserve member)	Delft University of Technology

This research was conducted at the AidroLab of Delft University of Technology, in collaboration with Noria Sustainable Innovators and the Directorate-General for Public Works and Water Management of The Netherlands (Rijkswaterstaat). This research was funded by the China Scholarship Council and Rijkswaterstaat, and also partially supported by the Lamminga Fund.

Keywords: Artificial Intelligence, Computer Vision, Environmental Monitoring, Macroplastics, Pollution

Cover by: Tianlong Jia
Email: jiatianlong123456@gmail.com

ISBN (Paperback/softback): 978-94-6384-826-8
ISBN (E-book): 978-94-6518-097-7

An electronic version of this dissertation is available at
http://repository.tudelft.nl/.

This book is dedicated to my family,
for their endless support and patience.

Tianlong Jia

ACKNOWLEDGEMENTS

I still vividly remember the day I first arrived in the Netherlands on December 4th, 2020. It was a cold morning as I traveled from Amsterdam Airport Schiphol to the TU Delft campus. During that ride, I imagined what my PhD journey would look like over the next four years. Thankfully, it turned out to be even more rewarding than I could have imagined. Along this way, I've learned and grown immensely. I'm deeply grateful to the countless people who supported me throughout this journey and made my stories in the Netherlands happen.

First and foremost, I would like to express my heartfelt gratitude to my daily supervisor and co-promotor, Dr. Riccardo Taormina, for enrolling me at AidroLab and providing me with the opportunity to work on this amazing PhD project. I truly appreciate that whenever I make progress in my research and wish to share it, you can always find a meeting time as soon as possible. Your passion and enthusiasm for research have been truly inspiring. You has consistently offered smart ideas and invaluable suggestions, whenever I faced challenges in my research. I am also deeply thankful for your unwavering encouragement throughout my entire research journey.

I would like to sincerely thank my promotor, Prof. Zoran Kapelan, for enrolling me at TU Delft. I still remember our email exchanges and interviews before I was granted the opportunity to pursue this PhD project. Your kind words and patience greatly encouraged me during that time. Throughout my PhD journey, I am deeply grateful for your prompt feedback, insightful advice, and unwavering encouragement whenever I encountered problems. Your careful and detailed revision of my papers, along with your high-level critiques of my research have been invaluable in shaping the quality of my work.

My deep gratitude goes to Ir. Rinze de Vries, my external supervisor, for your significant contributions to my research. I still remember the relaxed atmosphere in Noria's meeting room, where you carefully and patiently guided me through English writing and experiment design. Whenever I found myself lacking certain skills needed for my research, you always came up with solutions to help me develop these skills. I could not have achieved the important milestones without your invaluable guidance in field sampling and devices setup deployment, and insightful suggestions for revising my papers.

I would like to extend a special thank you to Imke Okkerman, Paul Vriend, and Eric Copius Peereboom from Rijkswaterstaat, for your invaluable support and assistance in various aspects of my research.

I want to thank the Noria family Arnoud van der Vaart, Sophie Broere, Fokke Jongerden, Parshva Mehta, Jur van Wijk, and Andre Jehan Vallendar for the great talks and support throughout my research. A special thank you goes to Andre, whose assistance helped me quickly acclimate to life in the Netherlands. I extend my gratitude to Edoardo Antonio Forte, Francesca Anita Lena, and Agatha Zamuner for their valuable assistance in data collection in Vietnam.

During my PhD journey, I had the privilege of closely collaborating with researchers from the hydrology and environmental hydraulics group in Wageningen University and Research. I would like to express my heartfelt gratitude to Tim H.M. van Emmerik, Paolo Tasseron, Tim Janssen, and Louise Schreyers for your assistance.

Thanks to our current and former support colleagues at our department: Betty Rothfuse, Fleur van de Water, Eef Neijenhuis, Louise Holslag, Linda Otten, Maureen Smith, Mariska and Riëlle. Your assistance in promptly organizing various daily matters has greatly saved me time and allowed me to focus more on my research

I am very lucky to have been a part of Riccardo's group, alongside Alexander Garzón Díaz, Roberto Bentivoglio, Tugba Yildizli and Ned. I truly cherish our talks about research, academic conferences, culture and custom, PhD life, travel plans, and, of course, the shared experiences we've had with Riccardo. Thank my colleagues in Sanitary section: Joao Ferreira, Mike Wit, Iosif Kaniadakis, Aashna Mittal, Sander Wingelaar, Yana, Bilal Khan Yusufi, Gladys Wangari Mutahi, Rifki Wahyu Kurnianto, Asif Jan, Simon Kreipl, Job van der Werf, Anurag Bhambhani, and Andrea Deiana. I have truely enjoyed our coffee break and conversations about food, sports, travel plans, and PhD life. I would also like to thank the Chinese PhDs, Postdocs and visiting researchers in our department: Mingliang Chen, Shuo Zhang, Bin Lin, Max, Hongxiao Guo, Lihua Chen, Zhe Deng, Dengxiao Lang, COCO, Yuke Li, Guangzhe Qin, Qin Ou, Yanghui Xu, Sijia Kong, Xingzhou Lyv, Dong Wang, Yi Luo, Guoding Chen, Xiao Feng, Yipeng Wu, and Prof. Zhaoxu Peng. The joyful moments spent with you have made me feel at home and created my amazing memories in the Netherlands.

I am fortunate to have found friends with the same hobby in the Netherlands: Lanny, Tianlei Miao, Happy, and Robert. We have spent countless days in dance studios, sweating together and enjoying breaking dance. Those moments with you have kept me energized and motivated whenever I faced bottlenecks in my research. I want to express my appreciation to my other Chinese friends in the Netherlands: Yu Yao, Yongxia Shi, Xiaohuan Lyv, Yujie Tang, Yifei Li, Desong Du, Yiru Jiao, Yongli Wu, Xiaoyu Liu, Jiao Zhao, Sen Yuan, Chaochong Cai, Shuaiqi Yuan, Liqi Cao, Deqing Mao, Zhenzhen Wu, Dinghao Wu, Yigu Liu, Cheng Chang, Zichao Li, Xinxin Zhang, Yiru Jiao, Di Yan, Chen Wang, Yubao Zhou, Hanting Ye, Minfei Liang, Xinling Yue, Langzi Chang, Hongpeng Zhou, Dawei Fu, Jingyi Liu, Kai Wu, Ziao Wang and Kangmin Mao. Your companionship was the shining light that brightened this journey, especially during the most challenging times. Thank my friends who traveled with me around Europe and America during my PhD journey: Tingting Lilianna, Ziang Zeng, Yi Dai, Yiyun Zhu, Haolong Cheng, Xue Yao, Yuanyuan Tan, Hanbing Wang, Chi Jin, Bo Li, Jun Wen, Yuanli Li, Xinyue Fu, Ruopeng Huang, Xiangyu Shao, Xianjing Liu, Chuang Chen, and Bo Li. Your companionship made these trips truly memorable. I would also like to appreciate my friends outside the Netherlands: Shiliang Duan, Yao Tong, Dexiang Zhang, Chenyang Lai, Yakun Wang, Xuguang Sun, Liyang Gao, JingYi Chen, Shengqi Sun, Zhanxing Xu, Yan Wang, Ao Sun, Yichen Zhou, Yide Wang, Yawen Ou, Xiaoyang Liu, Yun Gao, Jianjiao Feng, and Bonan Wang. Your companionship was the diamond that made this journey brighter during the toughest moments.

I am deeply grateful to my former teachers in China, Prof. Hui Qin and Prof. Hai Sun, who first guided me on the academic path and provided invaluable advice on career

development. Their mentorship has left an indelible mark on my academic journey.

Above all, my deepest gratitude belongs to my family. To my beloved Mom, thank you for always supporting my decisions and granting me the freedom to chase my dreams. To my brothers, sisters, uncles, aunts, and grandparents, your unwavering love and support have been the foundation I've relied upon throughout this journey. To my dear wife Jing, thank you for your love and care. Over these years, we've had countless discussions about research, PhD life, and our future. We have grown into the best versions of ourselves, ready to face the future together. I'm so grateful to have met you in my life.

Tianlong Jia

02-2025

Delft, the Netherlands

CONTENTS

List of Figures

List of Tables

List of Acronyms

This list describes the acronyms used within the body of the thesis:

AE: Artificial Environment
AI: Artificial Intelligence
ANI: Adding New Images
ANI-DA: Adding New Images and performing Data Augmentation
AP: Average Precision
AUV: Autonomous Underwater Vehicle
COCO: Common Objects in Context
CNN: Convolutional Neural Network
CV: Computer Vision
DA: Data Augmentation
DL: Deep Learning
DSGC: Device Setup Generalization Capability
EGC: Environmental Generalization Capability
FN: False Negative
FPS: Frame Per Second
FOV: Field of View
FP: False Positive
FTAL: Fine-Tuning All Layers
FTC: Fine-Tuning the Classifier alone
GGC: Geographical Generalization Capability
GSD: Ground Sampling Distance
IC: Image Classification
IoU: Intersection Over Union
IS: Image Segmentation
ML: Machine Learning
MLOps: Machine Learning Operations
MLP: Multilayer Perceptron
MoCo: Momentum Contrast
mAP: Mean Average Precision
NGC: Non-Aquatic Generalization Capability
NMS: Non-Maximum Suppression
OA: Overall Accuracy
OD: Object Detection
OSPAR: Oslo and Paris Conventions
RoI: Region of Interest
ROV: Remotely Operated Vehicle
SAHI: Slicing Aided Hyper Inference
SimCLR: Simple Framework for Contrastive Learning of Visual Representations

SL: Supervised Learning
SSL: Semi-Supervised Learning
SwAV: Swapping Assignments between Multiple Views of the Same Image
TL: Transfer Learning
TP: True Positive
TUD-GV: TU Delft - Green Village
TUD-HCMC: TU Delft - Ho Chi Minh City
UAV: Unmanned Aerial Vehicle
WUR-HCMC: Wageningen UR - Ho Chi Minh City

SUMMARY

Litter, particularly plastic, accumulating in water bodies is a challenging environmental issue that affects ecosystems, human health and the economy. Rivers are the main pathways of land-based plastic waste to the ocean, but they also act as potential temporary and long-term plastic sinks, where significant amounts of plastic waste accumulate, and even remain trapped for decades. The detection and quantification of floating litter in rivers and urban waterways is thus essential for evaluating pollution levels and informing mitigation actions. However, traditional monitoring methods, such as sampling with nets and booms, are not suitable for large-scale structured monitoring across multiple geographic locations in extensive river systems. Deep Learning (DL) methods have shown great promise in automatic detection and quantification of floating litter from images or videos. Given that this specific field is still in its early stages, this thesis aims to enhance the understanding of DL-based litter detection and quantification in riverine environments, identify key knowledge gaps, and explore methodologies to address these gaps and drive further advancements in this field.

This thesis presents a critical review outlining the state-of-the-art of DL-based detection and quantification of litter in water bodies, highlighting key knowledge gaps in this field, as detailed in Chapter 2. The review results indicate that only a few studies have not achieved satisfactory model generalization performances on new, unseen images under different geographic, environmental, and device setup conditions. Additionally, developing robust models using conventional supervised learning (SL) methods requires a large amount of labeled data for training, that is expensive and laborious. Finally, few studies have used DL methods to measure floating litter fluxes across rivers with broader cross-section, with a limited amount of labeled data. However, a robust detection model with strong generalization capability is particularly crucial for accurately quantifying litter fluxes in large-scale river systems, that is essential for evaluating pollution levels. These gaps are synthesized into the main question of this thesis:

- *How to develop robust DL-based methods for detecting floating litter and quantifying cross-sectional floating litter fluxes in rivers, particularly in contexts with limited labeled data?*

To answer this main research question, three sub-questions are defined based on three knowledge gaps, and then addressed in Chapter 4-6 contained within this thesis:

1. *How to build robust DL models to detect floating litter in rivers, leveraging a relatively large amount of labeled data? (Gap 1, Chapter 4)*

2. *How to build robust DL models to detect floating litter in rivers, leveraging a limited amount of labeled data? (Gap 2, Chapter 5)*

3. *How to develop DL-based methods to quantify cross-sectional floating litter fluxes in rivers, leveraging a limited amount of labeled data? (Gap 3, Chapter 6)*

To answer the above questions, we needed to evaluate multiple DL methods on different datasets for litter detection and quantification. Thus, we created multiple datasets by collecting data from multiple locations in canals and waterways in the Netherlands and Vietnam, as detailed in Chapter 3. The remainder of this thesis mainly focuses on (1) enhancing generalization performances of litter detection models, while reducing the dependence on large-scale labeled dataset, and (2) developing DL-based methods for quantifying cross-sectional floating litter fluxes using these datasets and two existing openly available datasets.

To enhance model generalization performances for litter detection, we first exploited various DL methods including transfer learning (TL) methods and data-centric artificial intelligence (AI) approaches. The tested data-centric AI approaches mainly include data augmentation (DA) and adding new images from new conditions, as detailed in Chapter 4. We evaluated these methodologies using around 4,000 labeled images from a canal in the Netherlands. We found that the most effective TL method for improving detection accuracy involved using a model pre-trained on a large general dataset (i.e., ImageNet) and subsequently fine-tuning the entire network on floating litter images. Among the tested DA techniques, flipping DA techniques improve generalization performances the most, i.e., augmenting the training set with flipped versions of the original images. We also found that trained models generalize well to similar condition (i.e., same camera heights but different viewing angle), but do not generalize well to more complex scenarios (i.e., different camera heights and different viewing angle). Adding a limited number of images from new device setups can significantly improve generalization in such complex scenarios. While these methodologies are effective to achieve robust model performances, developing such models requires a large amount of labeled data, that is labor-intensive and time-consuming to obtain.

To overcome this issue and maintain robust performances, we proposed a semi-supervised learning (SSL) method to detect floating litter, based on a self-supervised learning method, as detailed in Chapter 5. We validated this method on images from canals and waterways in the Netherlands, Indonesia, and Vietnam. When developing litter detection models, we also used the best-performing TL method identified in the previous evaluation. The results show that our method matches or surpasses the supervised learning (SL) benchmark in performance on unseen data collected from the same geographic locations as the training data, and yields more notable improvement when limited labeled data is available for fine-tuning. More importantly, it achieves superior performance on unseen data sourced from different geographic locations as the training data.

Considering the effectiveness of SSL and data-centric AI approaches, we proposed a SSL-based framework to quantify cross-sectional floating litter fluxes in river systems, leveraging a limited amount of labeled data, as detailed in Chapter 6. Additionally, we used the best-performing DA approach identified in the previous evaluation. We developed this framework using images from waterways of the Netherlands, Indonesia and Vietnam, and evaluated flux quantification performances on a Vietnam case study, that was not used for model development. We benchmarked our results against the SL meth-

ods and human visual counting methods. The results indicate that the SSL-based framework substantially underestimates fluxes compared to human measurements. However, the SSL-based framework quantifies litter fluxes nearly twice as high as the baseline SL-based framework, offering estimates that align more closely with human-measured fluxes.

The main contributions of this thesis are: (1) providing insights of the use of TL and data-centric AI approaches to improve generalization capability of models for litter detection, (2) proposing a SSL method for litter detection to improve generalization capability while reducing dependence on large-scale labeled dataset, and (3) proposing a SSL-based framework to quantify cross-sectional floating litter fluxes in river systems, while minimizing the need for extensive labeled data. The methodological innovations and research findings offer valuable insights for both the scientific community and practitioners from industry in mitigating litter pollution in rivers. Further research should focus on: (i) developing a more robust litter detection model, (ii) quantifying floating litter mass fluxes and hotspots in rivers, and (iii) developing DL-based monitoring strategies for riverine litter, as detailed in Chapter 7.

SAMENVATTING

Zwerfafval, met name plastic, dat zich ophoopt in waterlichamen is een complex milieuprobleem dat ecosystemen, de volksgezondheid en de economie beïnvloedt. Rivieren vormen de belangrijkste transportroutes voor landafkomstig plastic afval naar de oceaan, maar fungeren ook als tijdelijke en langdurige opslagplaatsen waar aanzienlijke hoeveelheden plastic zich kunnen ophopen en zelfs tientallen jaren vast kunnen blijven zitten. Het detecteren en kwantificeren van drijvend zwerfafval in rivieren en stedelijke waterwegen is daarom essentieel om vervuilingsniveaus te beoordelen en gerichte maatregelen te nemen. Traditionele monitoringsmethoden, zoals bemonstering met netten en drijvende barrières, zijn echter niet geschikt voor grootschalige, gestructureerde monitoring over meerdere geografische locaties binnen uitgestrekte riviersystemen. Diepgaande leertechnieken (Deep Learning, DL) bieden veelbelovende mogelijkheden voor de automatische detectie en kwantificatie van drijvend zwerfafval op basis van beelden of video's. Aangezien dit onderzoeksveld zich nog in een vroege fase bevindt, richt deze thesis zich op het vergroten van inzicht in DL-gebaseerde detectie en kwantificatie van zwerfafval in rivieromgevingen. Daarnaast worden belangrijke kennishiaten geïdentificeerd en worden methodologieën onderzocht om deze hiaten te dichten en verdere vooruitgang in dit vakgebied te stimuleren.

Deze thesis presenteert een kritische beoordeling die de stand van zaken van DL-gebaseerde detectie en kwantificatie van afval in waterlichamen schetst, met nadruk op belangrijke kennishiaten in dit veld, zoals beschreven in Hoofdstuk 2. De resultaten van de review geven aan dat slechts enkele studies erin zijn geslaagd om bevredigende prestaties in modelgeneralizatie te behalen op nieuwe, niet eerder geziene beelden onder verschillende geografische, milieu- en apparaatomstandigheden. Bovendien vereist het ontwikkelen van robuuste modellen met behulp van conventionele supervised learning (SL)-methoden een grote hoeveelheid gelabelde data voor training, wat kostbaar en arbeidsintensief is. Ten slotte hebben slechts enkele studies DL-methoden gebruikt om drijvend afvalfluxen over rivieren met een bredere dwarsdoorsnede te meten, met een beperkte hoeveelheid gelabelde data. Een robuust detectiemodel met een sterke generalisatiecapaciteit is echter bijzonder cruciaal voor het nauwkeurig kwantificeren van afvalfluxen in grootschalige riviersystemen, wat essentieel is voor het evalueren van vervuilingsniveaus. Deze hiaten worden samengevat in de hoofdvraag van deze thesis:

- *Hoe kunnen robuuste DL-gebaseerde methoden worden ontwikkeld voor het detecteren van drijvend afval en het kwantificeren van dwarsdoorsnede van drijvende afvalfluxen in rivieren, vooral in contexten met beperkte gelabelde gegevens?*

Om deze hoofdonderzoeksvraag te beantwoorden, worden drie deelvragen gedefinieerd op basis van drie kennisleemtes, die vervolgens worden behandeld in Hoofdstuk 4-6 van dit proefschrift:

1. *Hoe bouw je robuuste DL-modellen om drijvend afval in rivieren te detecteren, door gebruik te maken van een relatief grote hoeveelheid gelabelde gegevens? (Kloof 1, Hoofdstuk 4)*

2. *Hoe bouw je robuuste DL-modellen om drijvend afval in rivieren te detecteren, met behulp van een beperkte hoeveelheid gelabelde gegevens? (Kloof 2, Hoofdstuk 5)*

3. *Hoe ontwikkel je DL-gebaseerde methoden om dwarsdoorsnede van drijvend afvalfluxen in rivieren te kwantificeren, door gebruik te maken van een beperkte hoeveelheid gelabelde gegevens? (Kloof 3, Hoofdstuk 6)*

Om de bovenstaande vragen te beantwoorden, moesten we verschillende DL-methoden evalueren op verschillende datasets voor afvaldetectie en kwantificatie. Daarom hebben we meerdere datasets gecreëerd door gegevens te verzamelen op verschillende locaties in kanalen en waterwegen in Nederland en Vietnam, zoals gedetailleerd in Hoofdstuk 3. De rest van deze thesis richt zich voornamelijk op (1) het verbeteren van de generalisatieprestaties van afvaldetectiemodellen, terwijl de afhankelijkheid van grootschalige gelabelde datasets wordt verminderd, en (2) het ontwikkelen van DL-gebaseerde methoden voor het kwantificeren van dwarsdoorsnede-drijvend afvalfluxen met behulp van deze datasets en twee bestaande, openbaar beschikbare datasets.

Om de generalisatieprestaties van het model voor afvaldetectie te verbeteren, hebben we eerst verschillende DL-methoden onderzocht, waaronder transfer learning (TL) en data-centrische kunstmatige intelligentie (AI) benaderingen. De geteste data-centrische AI-benaderingen omvatten voornamelijk data-augmentatie (DA) en het toevoegen van nieuwe afbeeldingen van nieuwe omstandigheden, zoals gedetailleerd in Hoofdstuk 4. We hebben deze methodologieën geëvalueerd met behulp van ongeveer 4.000 gelabelde afbeeldingen van een kanaal in Nederland. We ontdekten dat de meest effectieve TL-methode voor het verbeteren van de detectieprecisie het gebruik van een model dat vooraf is getraind op een groot algemeen dataset (d.w.z. ImageNet) en vervolgens het fijn-afstemmen van het gehele netwerk op afbeeldingen van drijvend afval betrof. Van de geteste DA-technieken verbeterde de flip DA-techniek de generalisatieprestaties het meest, d.w.z. het vergroten van de trainingsset door de originele afbeeldingen te spiegelen. We ontdekten ook dat getrainde modellen goed generaliseren naar vergelijkbare omstandigheden (d.w.z. dezelfde camerahoogtes maar een andere kijkhoek), maar niet goed generaliseren naar complexere scenario's (d.w.z. verschillende camerahoogtes en verschillende kijkhoeken). Het toevoegen van een beperkt aantal afbeeldingen van nieuwe apparaatinstellingen kan de generalisatie in dergelijke complexe scenario's aanzienlijk verbeteren. Hoewel deze methodologieën effectief zijn voor het bereiken van robuuste modelprestaties, vereist het ontwikkelen van dergelijke modellen een grote hoeveelheid gelabelde gegevens, die arbeidsintensief en tijdrovend zijn om te verkrijgen.

Om dit probleem te overwinnen en robuuste prestaties te behouden, hebben we een semi-supervised learning (SSL) methode voorgesteld voor het detecteren van drijvend afval, gebaseerd op een self-supervised learning methode, zoals gedetailleerd in Hoofdstuk 5. We hebben deze methode gevalideerd op afbeeldingen van kanalen en waterwegen in Nederland, Indonesië en Vietnam. Bij het ontwikkelen van afvaldetectiemodellen hebben we ook de best presterende TL-methode gebruikt die werd geïdentificeerd

in de vorige evaluatie. De resultaten tonen aan dat onze methode de prestaties van de supervised learning (SL) benchmark evenaart of overtreft op onzichtbare gegevens die zijn verzameld uit dezelfde geografische locaties als de trainingsgegevens, en een meer merkbare verbetering oplevert wanneer beperkte gelabelde gegevens beschikbaar zijn voor fine-tuning. Belangrijker nog, het behaalt superieure prestaties op onzichtbare gegevens die afkomstig zijn uit verschillende geografische locaties dan de trainingsgegevens.

Gezien de effectiviteit van SSL- en data-centric AI-benaderingen, hebben we een op SSL gebaseerd raamwerk voorgesteld om de dwarsdoorsnede van de drijvende afvalfluxen in rivier-systemen te kwantificeren, door gebruik te maken van een beperkte hoeveelheid gelabelde gegevens, zoals gedetailleerd in Hoofdstuk 6. Daarnaast hebben we de best presterende DA-benadering gebruikt die werd geïdentificeerd in de vorige evaluatie. We hebben dit raamwerk ontwikkeld met afbeeldingen van waterwegen in Nederland, Indonesië en Vietnam, en de prestaties van de flux-kwantificatie geëvalueerd op een Vietnam-case study, die niet werd gebruikt voor de modelontwikkeling. We hebben onze resultaten vergeleken met de SL-methoden en de visuele tellingen door mensen. De resultaten geven aan dat het SSL-gebaseerde raamwerk de fluxen aanzienlijk onderschat in vergelijking met menselijke metingen. Het SSL-gebaseerde raamwerk kwantificeert de afvalfluxen echter bijna twee keer zo hoog als het basis SL-gebaseerde raamwerk, en biedt schattingen die dichter in de buurt komen van door mensen gemeten fluxen.

De belangrijkste bijdragen van dit proefschrift zijn: (1) het bieden van inzichten in het gebruik van TL- en data-centric AI-benaderingen om de generalisatiecapaciteit van modellen voor afvaldetectie te verbeteren, (2) het voorstellen van een SSL-methode voor afvaldetectie om de generalisatiecapaciteit te verbeteren, terwijl de afhankelijkheid van grootschalige gelabelde datasets wordt verminderd, en (3) het voorstellen van een SSL-gebaseerd raamwerk om de dwarsdoorsnede van drijvende afvalfluxen in riviersystemen te kwantificeren, terwijl de behoefte aan uitgebreide gelabelde gegevens wordt geminimaliseerd. De methodologische innovaties en onderzoeksresultaten bieden waardevolle inzichten voor zowel de wetenschappelijke gemeenschap als praktijkmensen uit de industrie in het verminderen van afvalvervuiling in rivieren. Verder onderzoek zou zich moeten richten op: (i) het ontwikkelen van een robuuster afvaldetectiemodel, (ii) het kwantificeren van drijvende afvalmassa-fluxen en hotspots in rivieren, en (iii) het ontwikkelen van DL-gebaseerde monitoringstrategieën voor rivieren-afval, zoals gedetailleerd in Hoofdstuk 7.

1

INTRODUCTION

1

1.1. LITTER POLLUTION IN RIVERS

Plastic pollution in water bodies is a challenging global concern, that negatively affects aquatic ecosystems and human livelihood (Bellou et al., 2021; Borrelle et al., 2020). Kaandorp et al. (2023) estimated an initial amount of floating marine plastics of 3.2 million tonnes in 2020. Rivers are the main pathways of land-based plastic waste to the ocean (Meijer et al., 2021; Schmidt et al., 2017). Lebreton et al. (2017) estimated that the yearly plastic flux transport from rivers to oceans is 1.15 to 2.41 million tonnes, while this estimation comes with large uncertainties (Roebroek et al., 2022). Furthermore, marine plastic litter may wash up on beaches and shores, and substantial amounts of discarded plastic litter has also been detected in lakes (Imhof et al., 2018; van Emmerik & Schwarz, 2020). They become micro- and nanoplastics over the years, associated with severe environmental and health risks (Liu et al., 2021b).

Recent studies indicate that river systems act as plastic reservoirs, where the majority of plastics accumulates, and even retains for decades (van Emmerik et al., 2022b; van Emmerik et al., 2023). Plastic pollution in rivers is a significant concern due to its potential to harm aquatic life and human health, increase flood risk, and break down into microplastics (Al-Zawaidah et al., 2021; van Emmerik & Schwarz, 2020). Recognizing the urgent need to address this issue, 175 countries have agreed to endorse a legally-binding agreement on plastic pollution by 2024 at the UN Environment Assembly (UNEA-5.2). It aims to regulate the full life cycle of plastics, including production, usage, and disposal (Walker, 2022).

Regardless of waste type, detecting and quantifying floating litter (items >5 mm) is key to assess pollution levels in river systems. Such assessment is essential for developing effective pollution reduction measures, such as source reduction and targeted cleaning campaigns (van Emmerik et al., 2019a, 2022c).

1.2. LITTER DETECTION AND QUANTIFICATION METHODS

1.2.1. IN SITU METHODOLOGIES

The in situ methodologies mainly include: (1) physical interception-based sampling, and (2) observation-based sampling (Hurley et al., 2023). Physical interception-based sampling entails the active entrapment and collection of waste from the rivers using tools (e.g., nets and booms), followed by subsequent quantification and categorization of litter. In contrast, observation-based sampling involves the monitoring, quantification, and categorization of visible litter in rivers without physically collecting waste.

PHYSICAL INTERCEPTION-BASED SAMPLING

Nets are commonly used tools for intercepting and collecting floating, submerged, or benthic litter transporting in rivers. They can be installed at fixed points (e.g., bridges, riverbanks, and riverbeds) or towed by boats moving along the rivers (Haberstroh et al., 2021; Munari et al., 2021). They can be placed at specific depths within the water column using buoys or weights, allowing for targeted interception of debris at various levels (Hurley et al., 2023). When the measurement period is finished, the nets are removed from the rivers. Then, the collected litter is analyzed to count, categorize, or weigh the litter items, aiding in the further estimation of fluxes. These nets, typically around 1 m

wide and 0.5 m tall, are easily operated by one or two people. This enables frequent and flexible measurements at various locations across the river width (van Emmerik & Schwarz, 2020). Nets are applied to sample litter in various river systems, such as the Saigon river in Vietnam (van Emmerik et al., 2018a) and the Seine in France (van Emmerik et al., 2019b). However, the effectiveness of these nets is limited to specific flow velocity ranges. High water velocities increase the risk of net damage and the drag force exerted on the nets, making manual deployment and retrieval unsafe. Low water velocities result in insufficient force to keep the nets horizontal, hindering their ability to collect litter.

Booms are floating barriers that act as vertical screens, accumulating floating debris on the river surface, including litter and non-litter materials. While usually employed for clean-up activities, booms can also serve as a sampling method by incorporating procedures to isolate, to count, categorize, or weigh the litter items from the collected debris. Booms can measure floating litter fluxes, covering the entire river width or a partial section. They also can measure near-surface litter fluxes under the floating line by including meshes or screens (Vriend et al., 2020b). This sampling method is most effective in rivers with low flow velocities. At higher velocities, litter items may pass beneath the booms (Roy et al., 2021). Another issue is the potential capture of a large amount of non-litter materials, making measurement complicate. For example, Gasperi et al. (2014) installed booms on the Seine River to collect macroplastics and investigated the amount and composition of plastics. They found that above 90% of the mass of the collected debris was vegetation. This sample composition poses significant challenges in separating the macroplastic component, especially for macroplastic litter items with smaller sizes. It requires additional labor to remove the sorted debris.

OBSERVATION-BASED SAMPLING

Human visual counting methods are the most frequently employed approaches to measure litter fluxes in river systems (Hurley et al., 2023). They involve observers standing at appropriate points (e.g., bridges) and recording the amount of visible litter items over a specified measurement period. Then, the observation results can be used to estimate the fluxes for the entire river at a specific moment in time (van Emmerik & Schwarz, 2020). With the availability of additional litter mass statistics (e.g., the mean mass per litter item), the estimation of litter fluxes can be converted to that of litter mass fluxes (e.g., the mass of litter items across the river width per unit of time) (van Emmerik et al., 2018a). These straightforward methods do not require the specialized equipment, allowing for frequent and inexpensive data collection in various river systems, e.g., the Rhone River in France (Castro-Jiménez et al., 2019) and the Saigon river (van Emmerik et al., 2018a). While visual observation is effective, it is limited to the availability of an appropriate point for observing rivers, such as bridges or other infrastructures that pass over the river. Such locations can be riverbanks for observing narrow rivers (Hurley et al., 2023). Additionally, it is not feasible for monitoring rivers with high litter fluxes, where human counters face challenges in accurately tracking debris over time (van Lieshout et al., 2020). It may be dangerous during extreme events, such as floods (van Emmerik et al., 2023).

Sensor-assisted observation methods involve collecting data with imaging devices, and manually detecting and counting litter items from collected data. The imaging

devices include cameras (van Lieshout et al., 2020), unmanned aerial vehicles (UAVs) (Rocamora et al., 2021; Schreyers et al., 2021) and sonar technologies (Broere et al., 2021). For example, van Lieshout et al. (2020) collected videos using cameras mounted on bridges in Jakarta, Indonesia. Then, they manually counted the floating macroplastic litter items from one-minute video clips and calculated the macroplastic fluxes. Rocamora et al. (2021) collected images by flying a drone at various points along the Segura River in Spain, and then manually counted floating litter items in images. The counting results are subsequently processed to estimate the total volume of floating waste along the river. Broere et al. (2021) used sonar technologies to identify and quantify submerged macroplastic litter items in the Guadalete river, Spain. These imaging devices enable continuous long-term measurements with high consistency, particularly when mounted cameras are used. UAVs are particularly effective in monitoring river sections that are difficult for humans to access. In addition, they are suitable for monitoring rivers with high flow velocities or high litter fluxes by carefully counting items in images or videos (van Lieshout et al., 2020). While these methods eliminate the need for observers to continuous monitor litter in target rivers, detecting and visually counting litter from sensor-collected data remains labor-intensive and time-consuming.

1.2.2. AUTOMATIC DETECTION AND QUANTIFICATION METHODOLOGIES

While the above in situ methodologies are effective, the time-consuming and labor-intensive procedures limit their applicability to long-term structured monitoring systems, including the monitoring of multiple geographic locations with varying environmental conditions in extensive river system (van Emmerik & Schwarz, 2020). Therefore, an automatic and efficient litter detection and quantification approach is needed.

Automated methods based on Computer Vision (CV) have been proposed to automatically detect and quantify litter from images. For example, Kataoka and Nihei (2020) installed a video camera at the Noda Bridge across the Edo River in Japan, and developed an image processing algorithm based on the color difference of the floating litter. This algorithm can detect and quantify the area fluxes of the litter (i.e., the area covered by litter per unit time). Then the area fluxes are further processed to estimate the litter mass flux.

Automated methods based on machine learning (ML) for CV have been applied to efficiently detect litter in water bodies from various types of data, ranging from drone images to satellite imagery. Existing ML applications include models based on Random Forest (Martin et al., 2018), Support Vector Machine (Basu et al., 2021), and Naive Bayes (Biermann et al., 2020). Nevertheless, traditional ML methods usually require time-consuming manual feature engineering and substantial data preprocessing (Bengio et al., 2013). These methods are also known to reach a performance plateau regardless of the amount of data available (Zhu et al., 2016).

Currently, researchers have suggested using Deep Learning (DL)-based CV methods, especially Convolutional Neural Networks (CNNs), for developing efficient alternatives (Garcia-Garin et al., 2021; Jakovljevic et al., 2020; van Lieshout et al., 2020). DL, a subset of ML based on deep biologically inspired artificial neural networks (Granger, 2006), has now superseded traditional ML techniques in many fields of science and technology, including water resources applications (Sit et al., 2020). DL belongs to the family of

representation learning techniques, which replace manual feature engineering via automatic discovery of the representations needed for feature detection from raw data (LeCun et al., 2015). In addition, DL models can lead to increasingly better performances as more data is fed to the models (Wang, Perez, et al., 2017). These characteristics allow DL models based on CNNs to reach state-of-the-art performances in all CV tasks such as image classification (IC), object detection (OD) and image segmentation (IS). A detailed literature review on DL-based litter detection and quantification in rivers is presented in Chapter 2. While preliminary results are promising, this specific field is still in its infancy (van Emmerik & Schwarz, 2020). The researchers must increase their efforts to devise DL-based applications that can help tackle pollution in river systems.

1.3. THESIS RESEARCH FRAMEWORK

1.3.1. KNOWLEDGE GAP STATEMENT

While DL methods offer efficient alternatives for detection and quantification of floating litter in rivers, this specific field is still in its early stages, with many challenges remaining. We conducted a literature review outlining the state-of-the-art of DL-based detection and quantification of litter in rivers, as well as other water bodies. The detailed review and discussion are shown in Chapter 2. Based on findings in Chapter 2, we identified three key knowledge gaps, as follows:

1. **The lack of robust DL models to detect floating litter in rivers.** There is a lack of DL-based detection models with robust out-of-domain generalization performances. This includes models that can detect floating litter in rivers on new, unseen images under the different geographic, environmental, and device setup conditions. Such models are especially crucial for large-scale structured monitoring, enabling the monitoring of multiple geographic locations with varying environmental conditions in extensive river system, without well-labeled and location-specific data for further refinement of DL models.

2. **The requirement of a large amount of labeled data for developing robust models.** All reviewed papers used supervised learning (SL) methods to develop litter detection models. These methods require a large amount of labeled data for training. The labeling work is expensive and laborious. While the community has released a open dataset on floating litter in rivers (van Lieshout et al., 2020), the amount of annotated data available is far below that of comprehensive datasets. While transferring the representations learned from general datasets can reduce the data requirement, these representations are not sufficiently effective to generalize across different locations and environmental conditions. This operation, known as transfer learning, is discussed in more detail in Chapter 2.

3. **The lack of DL methods to quantify cross-sectional floating litter fluxes in rivers, leveraging a limited amount of labeled data.** The current literature mainly focuses on detecting floating litter in rivers, but few studies focus on quantify floating litter fluxes in rivers with wide cross-sections. However, the fluxes quantification is important for assessing pollution levels, thereby facilitating the design of

effective pollution mitigation strategies (van Emmerik et al., 2019a, 2022c). Additionally, existing studies rely on supervised learning models for litter quantification. They require a large amount of labeled data, that is time-consuming and costly to obtain.

While the third research gap highlights the lack of DL methods for quantifying litter fluxes in rivers, it is equally important to quantify litter mass fluxes and litter mass in hotspots (Tasseron et al., 2020; van Emmerik et al., 2022a), as these metrics are also critical for assessing pollution levels in some scenarios. This thesis focuses on DL-based litter flux quantification, due to the time constraints of the doctoral research. Details on litter mass flux and hotspot quantification, along with directions for future research, are presented in Chapter 7.

1.3.2. RESEARCH QUESTIONS

These key gaps are synthesized in the primary objective of this thesis: to develop robust DL-based methods for detecting floating litter and quantifying cross-sectional floating litter fluxes in rivers, particularly in contexts with limited labeled data.

This objective is further synthesized into the main research question:

- *How to develop robust DL-based methods for detecting floating litter and quantifying cross-sectional floating litter fluxes in rivers, particularly in contexts with limited labeled data?*

To answer this main research question, three sub-questions are defined based on the three knowledge gaps in Chapter 2.4, and then addressed in Chapter 4-6 contained within this thesis:

1. *How to build robust DL models to detect floating litter in rivers, leveraging a relatively large amount of labeled data? (Gap 1, Chapter 4)*
2. *How to build robust DL models to detect floating litter in rivers, leveraging a limited amount of labeled data? (Gap 2, Chapter 5)*
3. *How to develop DL-based methods to quantify cross-sectional floating litter fluxes in rivers, leveraging a limited amount of labeled data? (Gap 3, Chapter 6)*

1.3.3. RESEARCH METHODOLOGY AND THESIS OUTLINE

Fig. 1.1 shows the thesis outline, including the mapping of the research sub-questions and chapters.

Chapter 2 presents a critical review outlining the state-of-the-art of DL-based detection and quantification of litter in water bodies, identifying key knowledge gaps. Based on these gaps, we formulated the main research question and sub-questions, that are addressed in Chapter 4-6 of this thesis.

To address these research questions, we need to evaluate multiple DL methods on datasets for litter detection and quantification. However, the available open datasets are limited in size, as highlighted in Chapter 2. Thus, we need to generate multiple datasets with sufficient data from rivers.

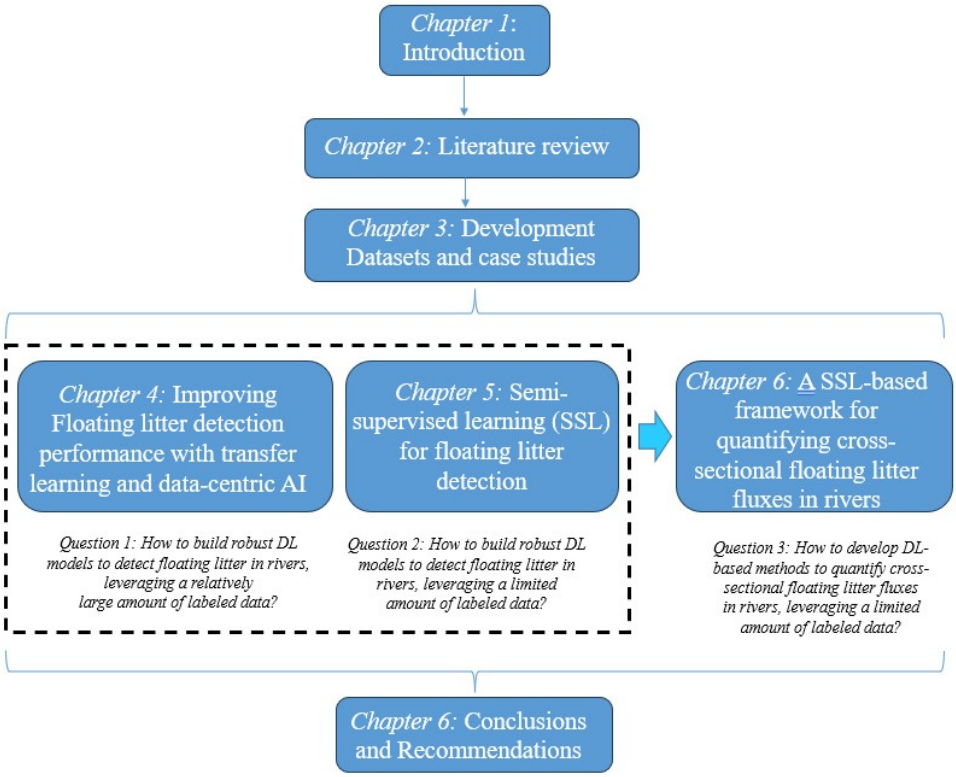


Figure 1.1: Structure of the thesis.

Chapter 3 presents multiple datasets generated by us. We collected data from multiple locations in canals and waterways in the Netherlands and Vietnam. Additionally, this chapter presents two existing openly available datasets used in this thesis.

Chapter 4 presents the studies and findings addressing the first research sub-question. Based on the review findings in Chapter 2, we identified transfer learning and data-centric AI methods as potentially effective approaches for enhancing model generalization capability. Thus, we developed DL models in a supervised manner, and evaluated the benefits of these methods on model out-of-domain generalization capability using a dataset with 4,000 labeled images (see Chapter 3). It is aimed at exploring the potential of various methodologies for building robust DL models to detect floating litter in rivers, leveraging a relatively large amount of labeled data (e.g., 4000 labeled images).

While the results in Chapter 4 show that the aforementioned methods are effective to improve model generalization capability, developing robust supervised models requires a large number of labeled images. Obtaining these labeled images for model development is costly and labor-intensive. Therefore, we need to explore alternative approaches to improve model generalization performance with a limited amount of labeled data in Chapter 5.

Chapter 5 proposes and evaluates a two-stage semi-supervised learning (SSL) method for improving model out-of-domain generalization capability, leveraging a limited number of labeled images (1.8k images with 2.6k annotated litter items), and a large amount of unlabeled images (100k). When developing litter detection models, we used the best-performing transfer learning strategy in Chapter 4. This chapter aims at exploring the potential of SSL methods for building robust DL models to detect floating litter in rivers, leveraging a limited amount of labeled data.

Chapter 6 proposes a SSL-based framework to quantify cross-sectional floating litter fluxes in rivers, leveraging a limited amount of labeled data, considering the effectiveness of SSL methods demonstrated in Chapter 5. When developing models, we also used a data-centric AI method to enhance model performance, as highlighted in Chapter 4. Additionally, we further optimized SSL models to obtain better performances, an aspect not explored in the experiments of Chapter 5. This chapter aims at exploring the potential of SSL-based framework to quantify cross-sectional floating litter fluxes in rivers.

Most importantly, we also addressed the main research question of this thesis in Chapter 6 by integrating the explored methodologies presented in Chapter 4-6, i.e., a SSL-based framework combined with the appropriate transfer learning and data-centric AI approaches for floating litter detection and quantification, leveraging a limited amount of labeled data.

Chapter 7 summaries the conclusions and scientific contribution, and presents the future research outlook.

2

Literature Review

We conducted a systematic review of papers on deep learning(DL)-based detection and quantification of litter in water bodies. The results show that the researchers have employed a variety of DL architectures implementing different CV techniques to detect litter in various aquatic environments. While limited attention has been given to to detecting and quantifying floating litter in rivers, we recommend increasing efforts toward riverine ecosystems, considering their major role in transport and storage of litter. We identified three key knowledge gaps in the study of riverine ecosystems: (i) the lack of robust DL models to detect floating litter in rivers, (ii) the requirement of a large amount of labeled data for developing robust detection model, and (iii) the lack of DL-based quantification of cross-sectional floating litter fluxes in rivers.

This chapter is based on:

Jia, T., Kapelan, Z., de Vries, R., Vriend, P., Peereboom, E. C., Okkerman, I., & Taormina, R. (2023). Deep learning for detecting macroplastic litter in water bodies: A review. Water Research, 231, 119632.

2. LITERATURE REVIEW

2.1. INTRODUCTION

The number of publications on DL-based detection and quantification of litter in rivers remains limited, since research in this field is still in its early stages. Therefore, we examined papers, focusing on various water bodies polluted by litter, such as marine surface, beaches, lakes, and rivers. By reviewing these studies, we could identify common methodologies, techniques, and significant gaps. Moreover, we could better identify the opportunities to address riverine pollution issues by understanding the broader context of litter detection and quantification. Based on the review of these publications, we identified key knowledge gaps in the study of riverine ecosystems.

The chapter is structured as follows. The methodology used to select and analyze the reviewed papers is described in Chapter 2.2. Chapter 2.3 thoroughly reviews the selected papers with critical discussion points. Finally, Chapter 2.4 identifies key knowledge gaps.

2.2. METHODOLOGY

2.2.1. SEARCH METHODOLOGY

In this review, we analyzed 34 peer-reviewed journal papers and conference proceedings published up to 2021, sourced from the "Scopus" and "Web of Science" databases. We employed the following steps to identify these papers. Firstly, we searched for papers published until the end of 2021 by employing three sets of keywords: (1) Deep learning-related keywords included "deep learning", "neural network", "artificial intelligence" and "machine learning"; (2) Litter-related keywords included "plastic", "trash", "litter", "debris" and "garbage"; (3) Water bodies keywords included "marine", "sea", "ocean", "beach", "shore", "river", "channel", "canal", "waterway" and "lake". The literature search identified papers containing combinations of these terms in their titles, keywords, and abstract. After reviewing the abstracts of all papers matching the inclusion criteria, we selected 33 papers, with the first publication dating back to 2016 (Valdenegro-Toro, 2016). Finally, we conducted a snowball search by checking the citations of these publications. The procedure yielded a total of 34 papers, which are listed in Table 2.1 along with the most important details. These papers are ordered by the type of water bodies and then by publication year within type in Table 2.1. If the study features different computer vision (CV) tasks, the review considers each of these tasks separately. When multiple architectures are tested, we report only the architecture achieving the highest performances, which is listed in the "Model architecture" column of Table 2.1.

Table 2.1: Details of reviewed papers

Reference	Water body	Dataset			CV task			TL	DA	GC	Performance evaluation[a]	
		Source	Size (#images)	Split (%)	No. classes	Type	Model architecture				Metric	Performance
Wu et al. (2020)	AE	phone, camera und.[b]	1,400	80/ 0/ 20	3	OD	YOLO v4	✓	✓		mAP	mAP=82.7%
Valdenegro-Toro (2016)	AE	sonar	22,446	70/ 15/ 15	2, 6	IC	CNN		✓		OA, recall, confusion matrix	OA=97.1% (6 object classes)
						OD	CNN with sliding windows				recall	recall=80.8% (binary OD)
Xue et al. (2021b)	marine und.	camera und.[b]	10,000	85/ 0/ 15	7	OD	YOLO v3 with ResNet50 backbone[c]	✓	✓		mAP, AP, F1-score, kappa, confusion matrix	mAP50=53.8%
Bajaj et al. (2021)	marine und.	camera und.[b]	2,900	85/ 0/ 15	3	OD	Inception-ResNetV2	✓				
Tian et al. (2021)	marine und.	camera und.	6,600	94/ 6/ 0	3	OD	Improved YOLO v4[c]				mAP, AP	

Table 2.1: Details of reviewed papers (Continued)

Hegde et al. (2021)	marine und.	camera und.[b], camera[b]	10,000	80/ 20/ 0	4	OD	SSD MobileNet V2	✓	✓	precision, recall, F1-score	
Marin et al. (2021)	marine und.	camera und.[b]	2,395	80/ 20/ 0	6	IC	Inception-ResNetV2[c]	✓	✓	OA, F1-score, kappa, confusion matrix, macro precision, macro recall, macro F1-score, weighted precision, weighted F1-score	OA=91.4%
Politikos et al. (2021)	marine und.	camera und.	635	80/ 15/ 5	11	OD	R-CNN with MobileNetV1 backbone	✓	✓	mAP, AP	mAP50=62%

Table 2.1: Details of reviewed papers (Continued)

Xue et al. (2021a)	marine und.	camera und.[b]	13,914	70/ 15/ 15	7	IC	Shuffle-Xception[c]		✓		OA, average accuracy, precision, recall, F1-score, kappa, confusion matrix	
Deng et al. (2021)	marine und.	camera und.[b]	7,212		22	OD IS	Improved Mask R-CNN[c]		✓		mAP	mAP50=65% mAP50=60.2%
Musić et al. (2020)	marine und.	camera und.[b]	~2,600	60/ 20/ 20	5	IC OD	VGG16[c] YOLO v3	✓	✓		OA	OA=85%
Panwar et al. (2020)	marine und., shores	camera[b]	369	80/ 0/ 20	4	OD	RetinaNet with ResNet-101-FPN backbone	✓		N	mAP, AP	mAP88=81.48%
Fulton et al. (2019)	marine und.	camera und.[b]	6,540	87/ 0/ 13	3	OD	YOLO v2[c]	✓			mAP, AP, Average IoU	mAP=47.9%

Table 2.1: Details of reviewed papers (Continued)

Mifdal et al. (2021)	marine sur.	satellite[b]		2	IS	U-Net		✓		pixel accuracy, F1-score, kappa	pixel accuracy=84.28%
Garcia-Garin et al. (2021)	marine sur.	airborne	796	90/ 0/ 10	2	IC	CNN		✓		OA, precision, recall, F1-score	OA=81%
de Vries et al. (2021)	marine sur.	camera	100,000		2	OD	YOLO v5[c]	✓				
Kylili et al. (2020)	marine sur., shores	camera[b]	1,600	79/ 20/ 1	8	IC	VGG16	✓	✓		OA	OA=90%
Battula et al. (2020)	marine sur.	camera[b]	2,467		2	OD	Resnet-50			N		
Watanabe et al. (2019)	marine sur., shores	camera, phone	189	80/ 0/ 20	4	OD	YOLO v3				mAP	mAP50=77.2%
Kylili et al. (2019)	marine sur.	camera	750	79/ 20/ 1	3	IC	VGG16	✓	✓		OA	OA=86%
Kylili et al. (2021)	shores	camera[b]	2,000	67/ 16/ 17	7	OD	YOLO v5	✓	✓			
						IS	YOLACT++					

Table 2.1: Details of reviewed papers (Continued)

Song et al. (2021)	shores	phone	846	70/ 17/ 13	7	OD	YOLO v5			D	AP, mAP	
Martin et al. (2021)	shores	airborne	750[d]		2, 14	OD	Faster R-CNN			G	precision, recall, F1-score	F1-score=44.2% (binary OD)
Papakonst-antinou et al. (2021)	shores	airborne	22,760[d]	54/ 13/ 33	2	IC	VGG19[c]	✓	✓	G	OA, precision, recall, F1-score	OA=77.6%
Wolf et al. (2020)	shores, rivers	airborne	12,918[d]	80/ 0/ 20	6, 18	IC	CNN	✓	✓		OA, precision, recall, F1-score, confusion matrix	OA=83% (6 class objects), OA=71% (18 class objects)
Gonçalves et al. (2020)	shores	airborne			2	IC	DenseNet			G	precision, recall, F1-score	
Kako et al. (2020)	shores	airborne		64/ 36/ 0	2	IS	MLP			G	pixel accuracy	

Table 2.1: Details of reviewed papers (Continued)

Fallati et al. (2019)	shores	airborne			2	OD	CNN			G, E	precision, recall, F1-score	recall=67%
Thiagarajan and Satheesh Kumar (2019)	shores	camera	135		2	IC	CNN		✓		OA, precision, recall	
						OD	CNN with sliding windows					
Putra and Prabowo (2021)	rivers	phone		90/ 10/ 0	2	OD	YOLO v3 with darknet-53 backbone	✓			AP, mAP	
Lin et al. (2021)	rivers	camera	2,400	91/ 0/ 9	8	OD	FMA-YOLO v5s[c]		✓		AP, mAP	mAP=79.41%
Tharani et al. (2021)	rivers	camera	13,500	93/ 7[e]	3	OD	M2Det(VGG)[c]				AP, mAP	mAP=45.8%
						IS	Improved U-Net[c]	✓				
van Lieshout et al. (2020)	rivers	camera	1,272	85/ 0/ 15	2	OD	Faster R-CNN with Inception V2	✓	✓	G, D	recall	recall=68.7%

Table 2.1: Details of reviewed papers (Continued)

Jakovljevic et al. (2020)	rivers, lakes	airborne 2,608	80/ 20/ 0	IS	ResUNet50[c]	✓	✓		precision, recall, F1-score

Acronyms: Transfer learning (TL), Data augmentation (DA), Generalization capability (GC), Artificial Environment (AE), Image classification (IC), Object detection (OD), Image segmentation (IS), Geographical generalization capability (G), Environmental generalization capability (E), Device setup generalization capability (D), Non-aquatic generalization capability (N), overall accuracy (OA), average precision (AP), mean average precision (mAP), Intersection Over Union (IoU).

[a] The "Metric" column is not populated when the study does not report common CV metrics. The "Performance" column is populated with the test value of the most representative metric.

[b] Part of or all the images in studies are retrieved from public databases or internet.

[c] The study features multiple deep learning architectures for computer vision tasks.

[d] The authors in studies cut the raw images into image tiles. In the "Dataset size" column, we report the total number of image tiles in datasets.

[e] Tharani et al. (2021) split their dataset into 93% for training and validation and 7% for testing.

2.2.2. REVIEW METHODOLOGY

Fig. 2.1 shows the most relevant factors used to classify and analyze the selected 34 papers. The factors are: (i) the water body(ies) polluted by litter (reviewed and discussed in Chapter 2.3.1; (ii) dataset on litter in water including the dataset source(s), dataset label(s), the dataset size and dataset split (Chapter 2.3.2); (iii) details on the CV task(s) performed to detect litter, including the type of CV task(s) and model architecture(s) for each CV task (Chapter 2.3.3); (iv) the machine learning paradigms used for developing model to detect litter, including traditional supervised learning and zero-shot learning methods (Chapter 2.3.4); (v) whether the authors resorted to data augmentation (DA) and transfer learning (TL) techniques to improve model performances (Chapter 2.3.5); (vi) the generalization capability of litter detection models (Chapter 2.3.6); and (vii) details on the metric(s) used for performance evaluation (Chapter 2.3.7).

It is noted that the review and discussion in the following chapters are primarily based on the 34 studies in Table 2.1. Additionally, we searched and reviewed papers published after 2021 using the same methodologies. However, we only reported perspectives and findings from these newer studies, that are different from those of the 34 papers. These findings are presented in Chapter 2.3.4.

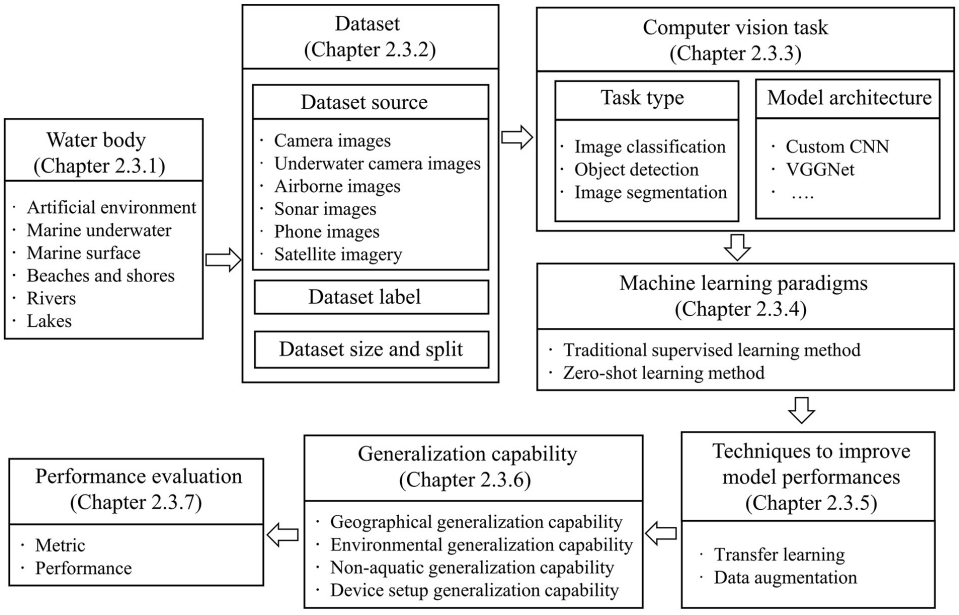


Figure 2.1: Factors reviewed in the reviewed literature.

2.3. REVIEW AND DISCUSSION

2.3.1. WATER BODIES POLLUTED BY LITTER

Fig. 2.2 shows the distribution of water bodies and dataset sources in the reviewed literature. Some papers are counted multiple times since they either consider multiple water

bodies (Jakovljevic et al., 2020; Kylili et al., 2020; Panwar et al., 2020; Watanabe et al., 2019; Wolf et al., 2020) or employ multiple dataset sources (Hegde et al., 2021; Watanabe et al., 2019; Wu et al., 2020). Most studies dealt with litter pollution in real settings, with the exception of two studies that considered a controlled artificial environment (AE). Valdenegro-Toro (2016) and Wu et al. (2020) collected data in a water tank for model training and test, which is time-saving and cost-effective. However, they did not investigate the generalization capability of DL models from studies in AE to field case studies. Field applications are different from studies in AE, because it is generally difficult to replicate the wide variety of litter and environmental conditions (e.g., natural lighting) witnessed in real settings (Valdenegro-Toro, 2016). We encourage further studies to test the performance of DL models trained with AE data when applied to real scenarios to assess the overall suitability of this approach and to allow future benchmarking of different methods.

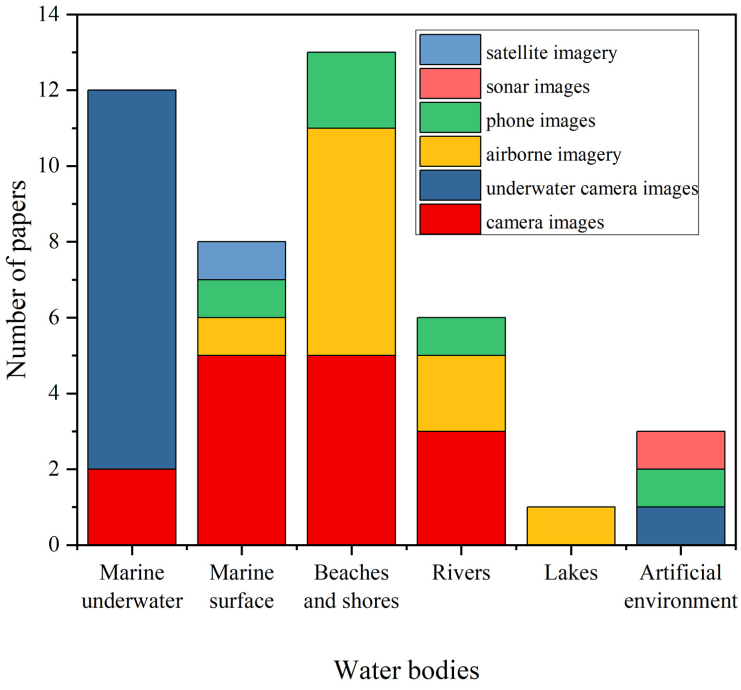


Figure 2.2: Distribution of water bodies and dataset sources. Some papers are counted multiple times since they either consider multiple water bodies or multiple dataset sources.

We categorized the bodies of water examined in the reviewed papers into (i) beaches and shores, (ii) marine underwater, (iii) marine surface, (iv) rivers (including natural rivers, waterways, urban channels and their banks), and (v) lakes. Most of the studies (18 out of 34 papers) focused on litter pollution in marine environments, including marine underwater (11 papers), marine surface (7 papers), and beaches and shores (12 pa-

pers). Fewer studies were concerned with river pollution (6 papers), and only one study dealt with litter in lakes. This is somewhat expected since, contrary to the consolidated body of knowledge for litter pollution in marine environments, the first scientific studies on the quantification of riverine litter date only to the early 2010s (Blettler et al., 2018; van Emmerik & Schwarz, 2020); the first studies concerning lakes are even more recent (Imhof et al., 2018).

2.3.2. DATASETS ON LITTER

EMPLOYED DATASET SOURCES

Researchers collected input data using imaging devices such as standard digital cameras, underwater cameras, cameras mounted on unmanned aerial vehicles (UAVs) or manned aircrafts, cameras mounted on phones, as well as satellite cameras and sonar technologies. As shown in Fig. 2.2, camera images (12 out of 34 papers), underwater camera images (11 papers), and airborne imagery (8 papers) are the three most popular dataset sources, while 4 studies have used phone images and only 1 study resorted to sonar images or satellite imagery.

Due to affordable costs and user-friendliness, digital cameras are popular data gathering devices regardless of the studied body of water (Mustafah et al., 2013). Fixed cameras are installed on bridges to monitor floating litter on the river surface (van Lieshout et al., 2020). One disadvantage is the limited coverage due to their fixed positions and limited viewpoints. Cameras attached to a vessel can survey broader areas (de Vries et al., 2021; Kylili et al., 2019) , although these surveying activities are time-consuming and labor-intensive compared to fixed installations. Five studies used camera images which were partially or completely retrieved from structured databases, such as ImageNet dataset (Deng et al., 2009) or via internet (e.g., Google). Two of these studies (Kylili et al., 2020, 2021) extracted images from the ImageNet dataset. Panwar et al. (2020) retrieved images from the TACO dataset (Proença & Simoes, 2020), and directly utilized the images with annotations from the dataset. Battula et al. (2020) extracted images from a Kaggle dataset[1] and labeled images with bounding boxes to train and test object detection (OD) model. Hegde et al. (2021) retrieved unlabelled images from Google and manually created the annotations to develop and test their models.

Researchers mainly used underwater cameras to collect images under the water surface, e.g., cameras attached to remotely operated vehicles (ROVs) (Wu et al., 2020) or vessels (Politikos et al., 2021). Nine studies used underwater camera images which were partially or completely retrieved from public databases or internet. This drastically reduces the cost of sampling activities in marine underwater environments where sampling requires laborious diving operations, expensive ROVs and/or autonomous underwater vehicles (AUVs) (Valdenegro-Toro, 2016). Five of these studies (Bajaj et al., 2021; Fulton et al., 2019; Marin et al., 2021; Xue et al., 2021a, 2021b) extracted data from the deep-sea debris database[2] provided by Japan Agency for Marine-earth Science and Technology. Four studies (Hegde et al., 2021; Marin et al., 2021; Musić et al., 2020; Wu et al., 2020) retrieved images from internet. While authors of these studies had to manually

[1] https://www.kaggle.com/asdasdasasdas/garbage-classification

[2] Japan Agency for Marine Earth Science and Technology, "Deep-sea Debris Database", available at http://www.godac.jamstec.go.jp/catalog/dsdebris/e/index.html

produce the annotations, one study (Deng et al., 2021) directly utilized the images with annotations from the TrashCan dataset (Hong et al., 2020).

Researchers collected airborne imagery using cameras mounted on UAVs (7 papers) or placed under manned aircrafts (1 paper). UAVs grant versatility since operators can easily customize the flight route and flight height to obtain images at different locations and with different ground sample distances (Fallati et al., 2019). Furthermore, UAVs allows surveying otherwise hard-to-reach locations (Zhang et al., 2017) and eliminate the limitations of fixing sensors on bridges or other infrastructure. However, no-fly zones restrict flying UAVs (e.g., nearby airports). Researchers may also need special training and licenses to operate UAVs, thus increasing the operational costs and, in some cases/countries, making the use of UAVs difficult, if not impossible. As shown in Fig. 2.2, UAVs are particularly suitable for field sampling activities along beaches, since these present fewer flight restrictions and obstacles that can potentially interfere with the UAVs flight (e.g., buildings).

Only four studies used mobile phones to collect images. Phones are easily available for citizens, thus could substantially contribute to citizen science initiatives for data collection. Modern smart phones with high-resolution cameras can obtain high-quality images and thus meet the needs of accurate sampling.

While sonar devices are preferred instruments for target and object recognition in underwater environments, e.g., fish classification and fishery assessment (McCann et al., 2018), only one study in the reviewed literature applied sonar devices to collect underwater images (Valdenegro-Toro, 2016). Although the relatively higher sampling costs hinder the development of autonomous detection and classification directly using sonar images (Qin et al., 2021), sonar devices are promising for underwater litter monitoring as suggested by a recent study (Broere et al., 2021). Sonar sampling can cover a larger area underwater where ROVs or divers cannot safely dive (Neupane & Seok, 2020) because sound waves travel further in water. For these reasons, we encourage further studies to assess their suitability for detecting litter under water surface, especially in real-world settings.

One study (Mifdal et al., 2021) retrieved Sentinel 2 imagery on floating marine litter from the Google Earth Engine dataset catalog. These data are globally available and free of charge, but do not contain specific annotations for litter. After collecting and labeling the satellite imagery, the authors trained DL models to detect floating objects on the sea surface. Compared with other dataset sources, satellite imagery can provide broader geographical coverage that is significant for hotspots monitoring and global environmental monitoring. On the other hand, satellite imagery is not appropriate to detect small and isolated litter floating on the vast sea surface, and cannot be used for observing underwater litter (Watanabe et al., 2019).

DATASET LABELS

Authors do not usually categorize macroplastic litter and other types of litter in their datasets using labels that reflect international guidelines and standards. If we consider the categories defined by the Oslo and Paris Conventions (OSPAR) (Wenneker & Oosterbaan, 2010), we identify 12 categories of macroplastic litter (i.e., bags, bottles, nets, caps/lids, industrial packaging/plastic sheeting, cups, buckets, cutlery/trays/straws, containers, shoes/sandals, rope, and floats/buoys), and 5 categories of other litter (i.e., glass,

paper/cardboard, rubber, metal, and cloth) across all surveyed datasets. When OSPAR defines multiple sub-categories of one plastic product and the specific sub-category is unclear in reviewed papers, we only report its general category. For example, while "bags" is categorized into 6 sub-categories in OSPAR (e.g., small plastic bags and fertilizer/animal feed bags), we only report "bags" in this chapter.

Several studies (13 out of 34 papers) detected plastic in a binary fashion. Among these, only three studies (Garcia-Garin et al., 2021; Kako et al., 2020; van Lieshout et al., 2020) specifically detected the presence of macroplastic litter in images. The remaining studies detected macroplastic litter by including it in a generic "litter" or "trash" or "debris" category. A larger group of studies detected more than 2 classes (22 papers). Among these, one study (Kylili et al., 2019) detected different types of plastic products. Nine studies provided a refined categorization for other types of litter. For example, Panwar et al. (2020) categorized the objects into glass, metal, paper, and plastic. One study (Tharani et al., 2021) detected macroplastic litter by considering three different sizes, included in three generic categories (i.e., small trash, medium trash and large trash). The remaining studies (11 papers) detected different plastic products as well as other object categories. For instance, Watanabe et al. (2019) classified the objects as plastic bottles, plastic bags, drift wood, and other debris. Gathering a balanced dataset with accurate labels becomes challenging as the number of classes increases. We identify 9 studies (Marin et al., 2021; Martin et al., 2021; Musić et al., 2020; Politikos et al., 2021; Tharani et al., 2021; Thiagarajan & Satheesh Kumar, 2019; Tian et al., 2021; Wolf et al., 2020; Xue et al., 2021b) working with unbalanced datasets, featuring classes with very scarce data (e.g., shoes, plastic cups, string and cord). Depending on the sensor used and its resolution, small objects (e.g., straws, toothpicks, and cotton buds) may be far less visible than others (Tharani et al., 2021). To improve detection of rare items or small items, we need to collect more data at higher resolutions (Wolf et al., 2020). Additionally, we can use the Slicing Aided Hyper Inference (SAHI) method (Akyon et al., 2022) to enhance accuracy of small litter detection. The implementation details and performance evaluation of this method are presented in Chapter 6.

DATASET SIZE AND SPLIT

The "Dataset size" column of Table 2.1 reports the size of dataset used in the reviewed papers, not including the data generated via DA. For one study (Wu et al., 2020), only the dataset size including the data generated with DA could be reported. Fig. 2.3 shows the distribution of dataset sizes (the number of images) per dataset source, as reported in 29 of the reviewed papers (see Table 2.1). The dataset size of phone images is small because two other datasets containing phone images and another kind of dataset source (Watanabe et al., 2019; Wu et al., 2020) are featured in the "multiple" category. The size of another dataset containing phone images (Putra & Prabowo, 2021) is unclear, thus it was not reported in Fig. 2.3. One dataset containing 100,000 images (de Vries et al., 2021) is much larger than all the others. These time-lapse images were collected at intervals between 2 s and 10 s during The Ocean Cleanup's North Pacific Mission 3 research expedition. The average dataset length is of around 9000 images. According to Arya et al. (2020), Image classification (IC) generally requires more than 5000 labeled images for each class to train a model with acceptable performances. For binary detection problems, this entails that at least 10,000 images are needed to develop a sufficiently robust detection

model. In the reviewed literature, 11 studies conducted IC tasks (see Table 2.1). Apart from two studies (Gonçalves et al., 2020; Thiagarajan & Satheesh Kumar, 2019), all other 9 studies reported both the specific dataset size and the number of classes in datasets. According to the suggestions of Arya et al. (2020), with the exception of (Papakonstantinou et al., 2021; Valdenegro-Toro, 2016), these studies did not collect sufficient raw data for model training and validation considering the number of classes. Therefore, all studies lacking sufficient data adopted TL and/or DA to improve the performances (see Chapter 2.3.5). Similar considerations may be drawn also for studies presenting OD and image segmentation (IS) applications.

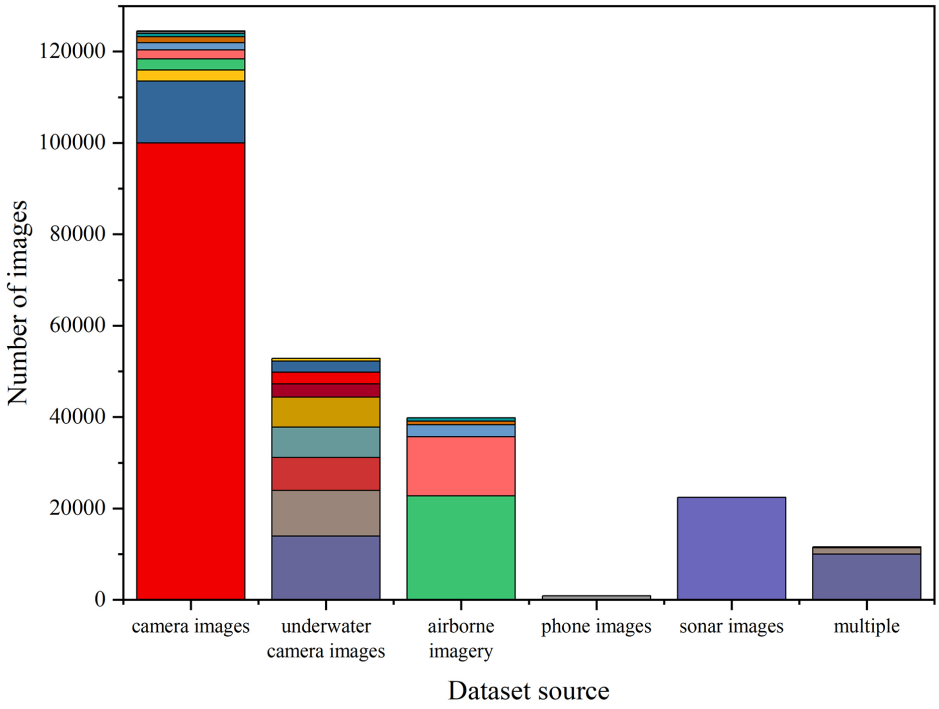


Figure 2.3: Distribution of dataset size (the number of images) per dataset source identified in 29 reviewed papers. Each different block identifies a different dataset. The label "multiple" identifies datasets obtained from multiple dataset sources.

Multiple researchers (e.g., Martin et al. (2021) and van Lieshout et al. (2020)) stressed the importance of a large-scale dataset for DL-based detection of macroplastic litter. Furthermore, three studies (Kylili et al., 2019; Musić et al., 2020; van Lieshout et al., 2020) showed that the increase of training dataset size leads to superior detection performance. For example, van Lieshout et al. (2020) developed a DL model to detect floating macroplastic litter in rivers across Jakarta, Indonesia using a binary classification approach. The precision (i.e., the proportion of objects correctly identified as macroplastic litter with respect to total detections) raised from 49.4% to 59.4% when increasing

the number of labels in the training dataset from about 2000 to 10,000. This study also showed that increasing dataset size further (from 10,000 to 24,000) resulted in smaller improvements (from 59.4%). Indeed, after training with sufficient data to learn basic representations, performance for CV tasks tend to grow logarithmically with dataset size (Sun et al., 2017). Therefore, we suggest gathering and labeling training data with respect to the level of performance required to address the specific challenge.

The "Dataset split" column of Table 2.1 reports the train/validation/test splits of the dataset used. Among 26 papers reporting dataset split, 10 reported the use of both a validation and a test dataset. The validation dataset is commonly used to select the best model by monitoring overfitting during training; on the other hand, the test dataset is employed to assess the generalization capability of the model for "unseen" data. It is not clear whether the remaining studies used part of the training data for validation and model selection, or if they used the test dataset for that purpose. Similarly, some papers report the use of a validation dataset, but not that of a test dataset. In general, we recommend to split the dataset into training (≈80%), validation (≈10%) and test (≈10%) datasets to facilitate robust model selection and unbiased estimation of the generalization error on unseen data.

2.3.3. COMPUTER VISION TASKS FOR LITTER DETECTION

CV TASK TYPES

General CV tasks are image classification, object detection and image segmentation (Chai et al., 2021). Fig. 2.4 shows an example of DL model architecture, and the output of different CV tasks.

IC is the process of classifying the entire image into one category (single-label classification) or multiple categories (multi-label classification) (Wei et al., 2014). The labeling procedure of IC includes annotating a given image with one class label or multiple class labels (see Fig. 2.4, top panel). On the other hand, OD algorithms automatically identify the class and location of different objects in images. The labeling task of OD requires the annotation of objects with class labels and bounding boxes (see Fig. 2.4, middle panel). Consequently, the output of OD models are bounding boxes and class labels for each detected instance. IS divides an image into multiple segments with similar characteristics, enabling a pixel-by-pixel identification of objects of interest. The labeling task requires assigning corresponding labels and pixel-wise masks to target objects (see Fig. 2.4, bottom panel) (Chai et al., 2021). We identify two types of IS among reviewed papers: semantic segmentation (Jakovljevic et al., 2020; Kako et al., 2020; Mifdal et al., 2021; Tharani et al., 2021) and instance segmentation (Deng et al., 2021; Kylili et al., 2021). Semantic segmentation assigns category labels to each pixel in images, while instance segmentation assigns category labels and instance identities to each object pixel (Chai et al., 2021). Thus, semantic segmentation is more suitable to quantify the area occupied by litter, while instance segmentation is more appropriate to discriminate different litter items.

Table 2.1 shows that researchers prefer OD methods to detect litter in aquatic environments (23 out of 34 papers). OD can concurrently identify the type and location of objects in images, thus estimating the number of litter items in an image (van Lieshout et al., 2020). IC (11 papers) is also popular since it is simpler to implement, especially

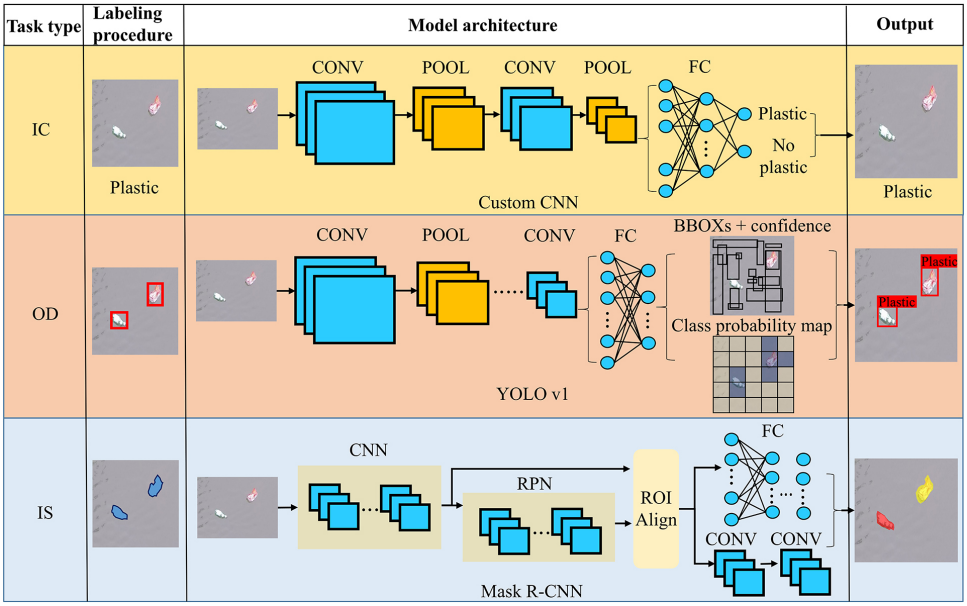


Figure 2.4: Labeling procedure, selected typical model architecture and output of different computer vision tasks. The "IC" row shows an example of binary classification, while the "IS" row shows an example of instance segmentation. Acronyms used: Convolutional layer (CONV), Pooling layer (POOL), Fully connected layer (FC), Bounding boxes (BBOXs), Convolutional neural network (CNN), Region Proposal Network (RPN), Image classification (IC), Object detection (OD), Image segmentation (IS).

by deploying one of the many successful architectures already available from the CV literature. Only 6 studies resorted to IS, arguably because of the substantial amount of time required to properly label the datasets (Jabari et al., 2021). Referring to IC models, all reviewed papers employed single-label algorithms, i.e., binary classifiers and multi-class classifiers. These methods can only process images containing one type of object at a time. On the other hand, multi-label classifiers can identify multiple categories of objects in one image (e.g., macroplastic litter, metal, and rubber) (Chai et al., 2021). Although these classifiers can better capture the diversity of litter in natural environments, no reviewed paper resorted to multi-label IC.

Although most studies (24 out of 34 papers) conducted CV tasks only for detection purposes, 10 studies also attempted the quantification of litter. Of these, 9 papers quantified the number of litter items via OD (de Vries et al., 2021; Martin et al., 2021; Song et al., 2021; van Lieshout et al., 2020), IC (Garcia-Garin et al., 2021; Gonçalves et al., 2020; Papakonstantinou et al., 2021; Wolf et al., 2020) or IS (Kylili et al., 2021). For example, Gonçalves et al. (2020) cut one original image into small portions, and performed IC to classify each portion into "litter" or "no litter". The number of litter items in one image was then calculated by the sum of the number of "litter" portions. However, if there are portions containing more than one item, performing IC tasks will lead to the deviation between the predicted results and the ground truth. Some studies also post-processed the model results to compute spatial litter concentrations (in items/m^2 or items/km^2,

5 papers), fluxes (in items/min/m, 1 paper), mass concentrations (in g/m^2, 1 paper) or mass (in kg, 1 paper). For instance, Martin et al. (2021) computed the concentrations of plastic bottles in beaches by averaging the number of correctly detected bottles over the tested area, and computed its mass concentrations by multiplying concentrations with respect to the median weights of bottles retrieved from Martin et al. (2019). Kylili et al. (2021) computed the total mass of litter in beaches by tallying the known mass of all litter items predicted by the DL model. van Lieshout et al. (2020) computed the macroplastic fluxes floating in rivers by dividing the number of items detected per time unit by the river width. Kako et al. (2020) computed macroplastic volumes via IS. They first detected the edges of macroplastic litter on images, which were then superimposed on a digital surface model containing location and altitude data over a beach. This allowed the litter volume to be computed from the heights and base area surrounded by the edges.

MODEL ARCHITECTURES FOR EACH CV TASK

Most reviewed publications (33 out of 34 papers) used Convolutional Neural Networks (CNNs) based architectures, such as YOLO networks (Xue et al., 2021b) and VGG networks (Kylili et al., 2020). Only one study (Kako et al., 2020) used a more conventional, three-layered Multilayer Perceptron (MLP) neural network. Compared with MLP, CNN can take advantage of the spatial patterns implicit in raw images. Besides, the properties of CNN (i.e., local connections and weight sharing) enable it to learn representations with fewer trainable parameters than MLP (Liang & Hu, 2015).These characteristics have allowed CNN to outperform MLP (Zhang et al., 2018) and to be more widely used for CV. However, none of the reviewed papers featured current state-of-the-art architectures such as Vision Transformers, e.g., Swin Transformer (Liu et al., 2022a) and ConvNeXts (Liu et al., 2022b).

In IC tasks, four studies employed the VGGNet architecture, probably because this architecture was proposed in 2014, and has been applied since then successfully in many fields (Ajit et al., 2020). Custom CNN architectures (4 papers) are also popular, mainly to develop parsimonious models with limited parameters that better match data availability and largely reduce computational efforts. For example, one study (Valdenegro-Toro, 2016) employed a custom 4-layered CNN with 930,000 parameters, much less than the 143.47 million parameters of a deeper VGG model with 19 layers (Martin et al., 2021).

CNN-based OD algorithms are divided into two-stage algorithms and one-stage algorithms. In two-stage algorithms, the first stage generates a set of bounding box proposals that are classified and detected in the second stage (Chai et al., 2021). On the other hand, one-stage algorithms perform classification and bounding box prediction concurrently in a single forward pass of the network. YOLO networks (11 papers) are the most frequently used OD architectures among the reviewed papers. YOLO networks are popular one-stage architectures thanks to their fast processing speed, which can reach the standards required for real-time video processing (Redmon et al., 2016). Although YOLO networks are faster than other architectures, its accuracy may be lower than that of some two-stage OD algorithms such as Faster R-CNN, which has been used in two reviewed papers.

U-Net (3 papers) is the most frequently employed IS architecture (Huang et al., 2020). One study (Jakovljevic et al., 2020) employed ResUNet50, which is based on a hybrid

between the popular ResNet (He et al., 2016) and U-Net architectures. Building blocks of ResNet pre-trained on the ImageNet dataset are added to the U-Net.

Two studies deployed DL models, Resnet-50 neural network (Battula et al., 2020) and SSD MobileNet V2 (Hegde et al., 2021) to perform OD in edge computing devices, e.g., processing boards connected to fixed cameras or installed in ROVs. For example, Hegde et al. (2021) stored a trained detection model in a Raspberry Pi board. The device used the model to detect litter from the surrounding environment as sampled by the attached underwater camera.

Model complexity plays an important role for DL models that will eventually run in real-time or on edge computing devices. Researchers should thus further investigate the suitability of small architectures with good classification performances, such as MobileNetV2 (≈2.4 million parameters) (Dong et al., 2020), and SqueezeNetV1 (≈1.2 million parameters) (Gholami et al., 2018). These “light” architectures can be easily transferred to edge devices and play a significant role in tackling litter pollution in water bodies. Pruning algorithms can successfully reduce model complexity and enable edge computing. For instance, Tian et al. (2021) proposed a pruned YOLO v4 capable of accurate OD for underwater camera images with only 7% of the original parameters.

2.3.4. MACHINE LEARNING PARADIGMS

All reviewed studies in Table 2.1 resorted to traditional supervised learning approaches for developing the detection models. In supervised learning, the model is trained to perform its task from examples of paired input/output data, where the output data is carefully labeled, or annotated, by humans. The typical labeling procedure used for each CV task is shown in Fig. 2.4). However, such models can not recognize objects that unseen for models during training. This limitation poses a challenge for developing models with high generalization capability used for structural monitoring of floating litter in rivers, as further discussed in Chapter 2.3.6.

In 2024, Nguyen and Dang (2024) proposed a zero-shot segmentation framework to detect seafloor litter, leveraging the zero-shot learning method, a transformative machine learning paradigm that allows models to identify categories they have never encountered during model training (Sun et al., 2021). It offers a promising solution to address the challenges of collecting sufficient labeled data for the wide variety of litter categories, as discussed in Chapter 2.3.2. More importantly, this approach demonstrates significant potential for developing a robust model that generalizes well to new litter category in a zero-shot (or few-shot) manner, with minimal prior data on new litter category. It is noted that the importance on zero-shot generalization capability highlighted by this publication, supports our argument in Chapter 2.4.1.

This zero-shot segmentation framework mainly include two modules: (1) an interpretable Contrastive Language–Image Pre-training (iCLIP) model for point prompt generation (Li et al., 2022), and (2) Segment-Anything Model (SAM) for zero-shot segmentation (Kirillov et al., 2023). Further details on these modules can be found in Nguyen and Dang (2024). Most interestingly, the iCLIP model is trained on image samples paired with text supervision (e.g., "a photo of a glass"), rather than on images with mask annotations indicting the shape and location of objects of interest (see Fig. 2.4). Then, the SAM predicts segmentation masks for each object category, based on the point prompt

generated by the iCLIP model. This allows the framework to identify and segment new litter categories (e.g., plastic bottle) simply by providing text prompts (e.g., "a photo of a plastic bottle"), without retraining models on new images with mask annotations, that are expensive to obtain.

2.3.5. TECHNIQUES TO IMPROVE DL MODEL PERFORMANCES

TRANSFER LEARNING

Transfer learning (TL) involves the transfer of prior knowledge from a related task to a new task (Pan & Yang, 2009). The usual TL approach involves (1) pre-training a base network on a base dataset and task (e.g., image classification on ImageNet), and (2) transferring the learned feature knowledge to a target network to be fine-tuned on a target dataset and task. In the base task, the first few layers of the base network extract generic low-level features (e.g., edges, lines, and corners), that generalizes to many datasets and tasks. The remaining layers extract more high-level, complex and abstract feature knowledge (e.g., object boundaries and contours), that specializes to a target dataset and task (Yosinski et al., 2014). This operation improves learning of the target task by: (1) providing a better starting point for training and preventing the model from falling into local minima (Fulton et al., 2019); (2) limiting the number of parameters to be optimized to a subset of the layers of the network; and (3) reducing data-labeling efforts by reducing the amount of training data needed to reach satisfactory performances on the new task.

Most of the reviewed papers (19 out of 34 papers) adopted TL, regardless of the CV task performed. For example, Musić et al. (2020) performed IC task to detect five categories of litter, i.e., plastic, glass, metal, paper and cardboard. They pre-trained the VGG16 on ImageNet dataset and fine-tuned the final layers of the VGG16 on a new dataset containing images from these five categories of litter. Although the objects in ImageNet are quite different from the detected litter, the pre-trained model improves detection because it recognizes generic features (e.g., edges, and basic shapes) in its early layers.

Researchers usually used models pre-trained on the ImageNet dataset or the Common Objects in Context (COCO) dataset (Lin et al., 2014). ImageNet is preferred for IC (5 papers), but also applied for OD (2 papers) and IS (2 papers). The COCO dataset is a common choice for OD (6 papers). The CIFAR-10 dataset (Recht et al., 2018) and the PASCAL VOC dataset (Everingham et al., 2010) have also been used.

While several authors resorted to TL to develop their models, with the exception of (Marin et al., 2021), no studies have thoroughly assessed its benefits with respect to training from scratch or fine-tuning the entire architecture (not just the classifier). Such investigations can be justified by the reported good performances of small architectures such as MobileNetV2 and SqueezeNetV1. Furthermore, the representation learned on large open source datasets may not always reflect typical features of images with litter (e.g., variety of litter, presence of water in the background).

DATA AUGMENTATION

Data augmentation (DA) reduces model overfitting by increasing the amount of available training data via augmentation or transformation of the images in the original training dataset (Shorten & Khoshgoftaar, 2019). This technique can also improve the performances of models when dealing with imbalanced datasets by creating more samples of

underrepresented classes.

DA usually involves automatic procedures performing geometrical transformations as well as color space transformations on available images. The DA methods used in reviewed papers include flipping (11 papers), rotation (10 papers), zooming in/out (4 papers), shifting (4 papers), noise addition (3 papers), cropping (2 papers), shearing (2 papers), copy-paste augmentation (2 papers), changes in brightness (1 paper), and mosaic data augmentation (1 paper). Fig. 4.1 shows an example of several DA techniques. Copy-paste augmentation is an advanced DA technique, whose purpose is to copy objects from a source image and paste them to a target image (Ghiasi et al., 2021). For example, Lin et al. (2021) employed such technique to superimpose labeled target objects cropped from real-world images against realistic backgrounds. Mosaic data augmentation combines 4 cropped images to create a synthetic image, that is often used for data augmentation in OD tasks (Lin et al., 2021).

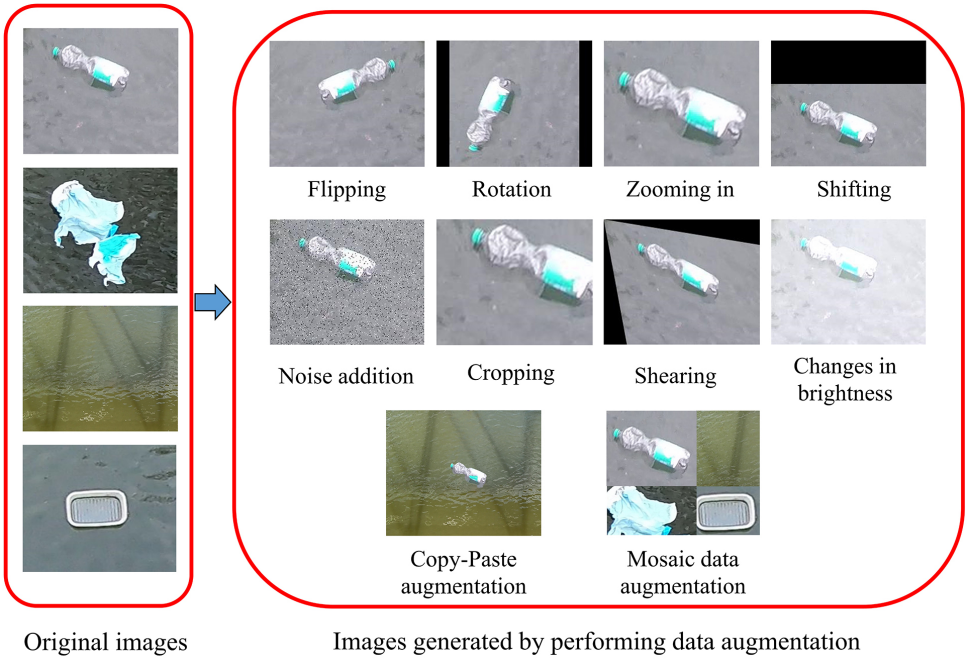


Figure 2.5: Examples of data augmentation techniques used in reviewed papers to improve model performances. Left: original images; Right: images generated by performing geometrical transformations (e.g., flipping, rotation, zooming in, shifting, cropping, and shearing), and other basic (e.g., changes in brightness, and noise addition), or advanced transformations (e.g., copy-paste augmentation, and mosaic data augmentation).

Flipping is the most popular choice as it preserves the original features of litter in the images and maintains fidelity with respect to the original label. On the other hand, the addition of noise or changes in brightness may alter the original images too much, thus degrading model performances. Rotation, zooming in, shifting, cropping, shearing, and mosaic data augmentation may instead lead to the omission of some of the originals objects of interest in the new images, forcing relabeling and partially nullifying the benefits

of DA (Shorten & Khoshgoftaar, 2019).

While most studies (20 out of 34 papers) applied DA (see Table 2.1), only three studies have thoroughly evaluated the benefits of DA with respect to training the same architecture on the original dataset. van Lieshout et al. (2020) showed that model precision marginally raised from 59.4% to 63.4% when using flipping data augmentation methods. Lin et al. (2021) also showed the model performances increased slightly when employing mosaic data augmentation. Musić et al. (2020) used copy-paste augmentation by superimposing computer-generated litter on realistic backgrounds. However, adding these images to the training dataset resulted in poorer prediction performances on the real-world dataset. Thus, researchers should discuss the benefits of different DA method for litter detection models in more depth.

2.3.6. GENERALIZATION CAPABILITY

DL models for CV exploit spatial inductive biases and shared weights to recognize features and objects regardless of their position in the image (Battaglia et al., 2018). While this favors generalization to unseen data, good detection performances at a single location or for similar environmental conditions do not guarantee that the model can be successfully applied or "transferred" to other situations and case studies. Achieving satisfactory out-of-domain generalization capability is a prerequisite for deploying large scale monitoring strategies based on DL, especially with respect to transferability across different bodies of water, locations, and device setups.

We identify four different forms of out-of-domain generalization capability in the reviewed papers: (1) geographical generalization capability, (2) environmental generalization capability, (3) non-aquatic generalization capability, and (4) device setup generalization capability. Geographical generalization capability represents the generalization capability of the model at different locations under roughly the same environmental conditions (such as weather, presence of waves, wind conditions, and terrain shading). Environmental generalization capability refers model testing in different environmental conditions. Non-aquatic generalization capability involves models trained with data from non-aquatic environments and tested on aquatic environments (or vice versa). Lastly, device setup generalization capability represents the generalization capability for different device setups, such as the flight altitude of UAVs, or the setting angle between a fixed camera and the water surface.

Despite the importance of generalization, only few studies (9 out of 34 papers) directly addressed these aspects, with two studies (Fallati et al., 2019; van Lieshout et al., 2020) considering two different forms of generalization capability (see Table 2.1). The majority of these 9 studies are with respect to geographical generalization capability (6 papers). Five papers (Fallati et al., 2019; Kako et al., 2020; Martin et al., 2021; Papakonstantinou et al., 2021; van Lieshout et al., 2020) studied geographical generalization capability by training and testing on different case studies, respectively. For example, Papakonstantinou et al. (2021) trained DL models on UAV images captured from certain beaches, and tested it on UAV images collected from different beaches. Compared with geographical generalization capability, there are less studies concerning non-aquatic generalization capability (2 papers), device setup generalization capability (2 papers), and environmental generalization capability (1 paper). For instance, Panwar et al. (2020)

trained a model on images of macroplastic litter gathered across streets and forests, and tested it on images with macroplastic litter under the sea surface and on beaches. Song et al. (2021) assessed device setup generalization capability by using a phone mounted on a tripod to collect training and test data from different heights at one beach. Fallati et al. (2019) evaluated environmental generalization capability by collecting training and test data at different time of the day. The authors also used a UAV to collect training and test data at different beaches to assess the geographical generalization capability of the model.

Among the 9 papers addressing generalization capability, 4 papers (Battula et al., 2020; Kako et al., 2020; Panwar et al., 2020; Song et al., 2021) did not discuss the performances of DL models trained and tested in different conditions. Only 1 paper (Papakonstantinou et al., 2021) reported promising geographical generalization capability, with a precision metric of 83%. The models in the remaining studies did not show satisfactory generalization performances when tested for different geographical, environmental, or device setup conditions with reported precision between 20% and 63.8%. For example, van Lieshout et al. (2020) showed that the performances of a trained model working reasonably well for one location deteriorated quickly for an unseen location, with a decrease in precision from 68.7% to 54%. These new images featured substantially more organic material (e.g., leaves and branches) than those used for training. The presence of organic material, unaccounted for during training, thus hindered robust detection of floating litter. The authors also showed that the generalization performances increased when including images from different locations in the training dataset. In general, we believe the community should increase efforts to develop DL models with robust generalization that can operate well across different conditions.

2.3.7. PERFORMANCE EVALUATION

The "Metric" column of Table 2.1 reports the performance metrics used by the authors when these reflect common options used for CV (Padilla et al., 2020; Wambugu et al., 2021) and are unambiguous.

For IC tasks, the majority of studies used the overall accuracy (OA) metric (9 out of 11 papers) to evaluate performances over all classes. Precision (6 papers), recall (7 papers), and F1-score (6 papers) were the most popular choices to evaluate performances for each class. These metrics should be preferred for imbalanced datasets since OA misrepresents the minority classes. For example, Wolf et al. (2020) worked on an imbalanced dataset including 18 categories of objects. Although good average performance were reported for all classes (OA=71%), minority classes such as carton (25 images in total) were poorly detected (F1-score=0.46).

For binary OD, common metrics include recall (4 out of 8 papers), precision (2 papers), and F1-score (2 papers). For multi-class OD, the majority of studies employed average precision (AP, 8 out of 17 papers) and mean average precision (mAP, 11 papers) to assess performances for each class object and over all classes, respectively. The value of these metrics depends largely on the selected threshold for determining the Intersection Over Union (IoU), a number that quantifies the degree of overlap between the predicted and ground-truth bounding boxes. With some exceptions (Deng et al., 2021; Panwar et al., 2020; Politikos et al., 2021; Putra & Prabowo, 2021; Song et al., 2021; Watanabe et al.,

2019; Xue et al., 2021b), these important thresholds are rarely reported in reviewed papers. Based on common benchmarks (e.g., COCO and PASCAL VOC), we recommend using a threshold IoU=0.5 when estimating fluxes (e.g., number of items across the river width per unit of time), while higher thresholds (e.g., up to 0.95) should be used to quantify mass concentrations (e.g., hotspot areas).

For binary semantic segmentation tasks, two studies (Kako et al., 2020; Mifdal et al., 2021) used pixel accuracy metrics to assess performances on detecting macroplastic litter. For multi-class semantic segmentation tasks, one paper (Jakovljevic et al., 2020) used precision, recall, and F1-score metrics to evaluate performances for each class. No papers reported results in terms of IoU or mean IoU, which are the preferred metrics for semantic segmentation as they account for unbalanced datasets. For multi-class instance segmentation, one paper (Deng et al., 2021) employed mAP to evaluate performances over all classes.

The "Performance" column of Table 2.1 reports the test value of the most representative metric across all classes. However, since the proposed methodologies have been tested on different macroplastic datasets in disparate experimental settings, a direct comparison is unfeasible. More interestingly, some papers report encouraging evidence on the effectiveness of DL methods with respect to accurate, but time-consuming, sampling methods. For instance, de Vries et al. (2021) found a satisfactory correlation (R2=0.7) between DL-detected spatial concentrations of macroplastics on the sea surface and manta-trawling ground truth observations. Song et al. (2021) reported a small error (<5%) between the number of litter items on a beach yielded by actual counting and those detected by Yolo v5. Kako et al. (2020) reported similar figures (<5%) for the volumetric difference of beached plastic debris between surveys and MLP-based IS. These results suggest that using DL for automatic detection and quantification of litter is a valid alternative to traditional sampling methodologies.

2.4. SUMMARY OF KEY KNOWLEDGE GAPS

Our review shows that the majority of reviewed papers focus on detecting litter in marine environments, while less attention is devoted to detecting freshwater litter. Recent research indicated that most debris leaking into the environment does not reach the oceans, but instead accumulates in river systems (Tramoy et al., 2020; van Emmerik et al., 2022b; Weideman et al., 2020), resulting in damaged ecosystems (Blettler et al., 2018). Monitoring the source, transport, and sink points of riverine litter is thus essential to quantify global litter pollution transport and effectively reduce pollution (van Emmerik & Schwarz, 2020). Therefore, we advocate for greater efforts on applying DL to tackle riverine litter pollution problems in the future.

Based on the findings reported in Chapter 2.3 and the significance of monitoring litter in rivers, we identified three major knowledge gaps:

1. *The lack of robust DL models to detect floating litter in rivers*

2. *The requirement of a large amount of labeled data for developing robust detection model*

3. *The lack of DL-based quantification of cross-sectional floating litter fluxes in rivers, leveraging a limited amount of labeled data*

2.4.1. ROBUST DL MODEL TO DETECT FLOATING LITTER IN RIVERS

Despite some promising initial efforts, research on the generalization capability of DL models for detecting floating litter is insufficient. Most studies have evaluated in-domain generalization performances, while only a few studies have assessed out-of-domain generalization performances. In-domain generalization refers to the model performance on new, unseen images under the same geographic, environmental, and device setup conditions, while out-of-domain generalization refers to unseen images from different conditions, as well as different case studies (e.g., waterway networks within a country). Besides, review results show that the models proposed so far do not retain satisfactory out-of-domain generalization performances under different geographical, environmental, or device setup conditions.

We argued that DL-based detection models with robust zero-shot out-of-domain generalization capability should be developed. This capability enables DL models to detect previously unseen objects from different geographic, environmental, and device setup conditions, without requiring training data of these unseen objects. Such models are especially crucial for large-scale structured monitoring, enabling the monitoring of multiple geographic locations with varying environmental conditions in extensive river system, without well-labeled and location-specific data for further refinement of DL models.

2.4.2. REQUIREMENT OF LARGE AMOUNT OF LABELED DATA FOR MODEL DEVELOPMENT

All reviewed papers employed supervised learning methods to develop DL models. These methods require large quantities of annotated training data for supervised learning to achieve robust performance. As reported in Chapter 2.3.2, the average dataset size across 34 reviewed papers is around 9,000 labeled images. The manual labeling work is costly, time-consuming and relies on domain-specific knowledge on floating litter detection. The community has released a open dataset (van Lieshout et al., 2020), with 1,272 images and 14,968 annotated floating macroplastic litter items in rivers. However, the amount of annotated data available is far below that of comprehensive datasets, e.g., ImageNet with over 14 million images and almost 20,000 categories. This may hinder achieving broad model generalization and effective transferability, which underpins robust and versatile computer vision systems for structural monitoring of floating litter.

To partially overcome this limitation, researchers often used transfer learning approaches (see Chapter 2.3.5), that transfer knowledge from a related task to a new task. While transfer learning is a powerful technique, its effectiveness declines when the base and target tasks become less similar (Yosinski et al., 2014). To develop DL models for floating litter detection, reviewed studies pre-trained models on comprehensive datasets, such as ImageNet. However, the high-level features in these datasets have limited relevance with respect to floating litter imagery. This may hinder performances and generalization capability.

2.4.3. DL-BASED QUANTIFICATION OF CROSS-SECTIONAL FLOATING LITTER FLUXES, LEVERAGING A LIMITED AMOUNT OF LABELED DATA

The current literature mainly focuses on detecting litter in rivers, and only few studies link DL-based detection to the quantification of litter, mainly with respect to the number of litter items. However, stakeholders require this information to design cleaning campaigns, and mitigate the impact of pollution on the environment and human health (Tasseron et al., 2020; van Emmerik et al., 2018b).

Only one study (van Lieshout et al., 2020) quantifies the floating litter fluxes in rivers. They used a supervised learning model with a single bridge-mounted camera, to estimate plastic fluxes in a narrow waterway in Jakarta, Indonesia. The development of their model requires a large amount of labeled data, that is time-consuming and costly to obtain. Additionally, van Calcar and van Emmerik (2019) reported that the horizontal distribution of floating litter fluxes along some wider rivers is highly uneven, based on observations from 24 locations in rivers across seven countries in Europe and Asia, e.g., the Saigon River (300 m wide), in Vietnam. Relying on observation from a single or low number of locations to estimate litter fluxes across a wide river may lead to significant under- or overestimation (van Emmerik et al., 2019a). Therefore, more efforts should go into developing better DL-methods for quantifying cross-sectional floating litter fluxes in wide rivers, leveraging a limited amount of labeled data.

3

DEVELOPMENT DATASETS AND CASE STUDIES

This chapter is based on:

Jia, T., Vallendar, A. J., de Vries, R., Kapelan, Z., & Taormina, R. (2023). Advancing deep learning-based detection of floating litter using a novel open dataset. Frontiers in Water, 5, 1298465.

Jia, T., de Vries, R., Kapelan, Z., van Emmerik, T. H. M., & Taormina, R. (2024). Detecting floating litter in freshwater bodies with semi-supervised deep learning. Water Research, 266, 122405.

Jia, T., Taormina, R., de Vries, R., Kapelan, Z., van Emmerik, T. H. M., Vriend, P. & Okkerman, I. (2025). A Semi-supervised Learning-Based Framework For Quantifying Litter Fluxes in River Systems (Submitted)

3.1. INTRODUCTION

To answer the research questions in this thesis, we needed to evaluate multiple deep learning methods on different datasets for litter detection and quantification. However, the community has released open datasets with limited data, as highlighted in Chapter 2. To overcome this limitation, we generated multiple datasets by collecting data from multiple locations in canals and waterways in the Netherlands and Vietnam. This chapter presents these datasets, along with two existing openly available datasets used in this thesis for model development and evaluation.

3.2. DATASETS AND CASE STUDIES

We collected data from five locations (1) The TU Delft - Green Village (TUD-GV), the Netherlands, (2) Oostpoort, the Netherlands, (3) Amsterdam, the Netherlands, (4) Groningen, the Netherlands, and (5) the TU Delft - Ho Chi Minh City (TUD-HCMC), Vietnam. We also present two open datasets in this chapter: (1) Jakarta, Indonesia (van Lieshout et al., 2020), and (2) Wageningen UR - Ho Chi Minh City (WUR-HCMC) (van Emmerik et al., 2024). The details of these datasets are shown in Table 3.1.

Table 3.1: Details on 7 datasets used in this thesis

Name	Collection location	Collection device	Image resolution (pixel×pixel)	Device height (m)	No. images [1]
TUD-GV	Delft, the Netherlands	GoPro Hero 4, GoPro MAX 360, Huawei P30 Pro	1920×1080	2.7	7,965
Oostpoort	Delft, the Netherlands	GoCam3, GoPro MAX 360	3840×2160, 1920×1440	5	562
Amsterdam	Amsterdam, the Netherlands	GoPro Hero 10	5568×4176	1-2	92
Groningen	Groningen, the Netherlands	Obscape HQ	2592×1944	4	63
TUD-HCMC	Ho Chi Minh City, Vietnam	Pentax K-serie	6016×4000		508
Jakarta	Jakarta, Indonesia	Dahua Easy4ip	2560×1440, 1920×1080	4.5	526
WUR-HCMC	Ho Chi Minh City, Vietnam	GoPro Hero 11, DJI Phantom 4 Pro	5568×4872, 5464×3070	7.4-18.6 (cameras) 11-14 (drones)	935

[1] In these columns, we only reported the maximum number of images used for model development in this thesis.

3.2.1. THE TU DELFT - GREEN VILLAGE DATASET

We created the TUD-GV dataset from experiments conducted during 10 days in February and April 2021 in a small drainage canal at The Green Village —a field lab facility in the TU Delft Campus, the Netherlands. Fig. 3.1 shows the monitoring setup. We captured data using two action cameras (GoPro HERO4 and GoPro MAX 360) and a phone (Huawei P30 Pro) mounted on four different locations on a bridge. All devices recorded videos with a resolution of 1080p, a linear field of view, and a FPS (frame per second) of 24 (for the action cameras) or 30 (for the phone). We opted for data collection in a semi-controlled environment as it is time-saving and cost-effective.

First, we collected the litter objects from canals in Alkmaar (the Netherlands) with

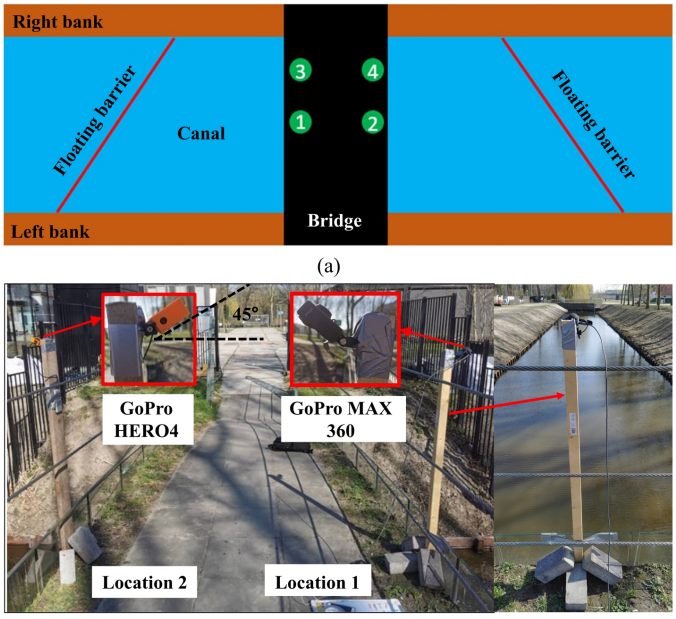


Figure 3.1: Monitoring setup at The Green Village: (a) view from the top with the four different filming locations (1-4) on the bridge; (b) details of some camera installation on Location 1 and Location 2.

the help of volunteers, as well as from household waste from nearby neighborhoods. In total, we gathered 626 items, including plastic bottles, plastic bags, miscellaneous plastic objects, as well as metal tins, paper and cardboard items. Examples of litter objects can be found in Fig. 3.2. Then, we placed the collected litter on the water surface of the canal at The Green Village and captured images as the floating litter moved on the water surface due to wind. Finally, we used floating barriers (see Fig. 3.1 (a)) to intercept floating litter after data collection to prevent water pollution.

Table 3.2 shows the details of the TUD-GV dataset, including device specifications, weather condition, litter class and the number of images. We recorded a total of 165 videos, from which we selected 9473 images (703 phone images and 8770 camera images) to create the TUD-GV dataset. These images contain canal and household floating litter under two different weather conditions (sunny and cloudy), taken from two device heights above the water surface (2.7m and 4.0m) and two viewing angles (0 and 45 degrees).

Fig. 3.3 provides examples of images from different device settings (device height and viewing angle). The collected images reflect all possible combinations of device used, device settings, type of litter, and environmental conditions. The set of household litter from the 2.7 m/45° setup is comprised solely of cloudy weather images, while some images from the 4 m/45° and 4 m/0° setups contain sun glints, as shown in Fig. 3.3 (c) and (d). Images from the 4 m/0° setup were cropped to exclude the bridge, as shown in Fig. 3.3 (c).

Figure 3.2: Monitoring setup at The Green Village: (a) view from the top with the four different filming locations (1-4) on the bridge; (b) details of some camera installation on Location 1 and Location 2.

Table 3.2: TUD-GV dataset details

Device	Device degree (°)	Device height (m)	Weather conditions	Litter class	No. images per class				No. images
					No litter	Little litter	Moderate litter	Lots of litter	
GoPro HERO4, GoPro MAX 360, Huawei P30 Pro	0	2.7	Sunny, Cloudy	Canal litter, Household waste	1151	1429	1971	1305	5856
	0	4			555	331	350	124	1360
	45	2.7			399	293	348	166	1206
	45	4			302	246	298	205	1051

Inspired by the categorization scheme of CrowdWater (van Emmerik et al., 2020), we manually labeled the images in the TUD-GV dataset into four classes: *no litter* (0 items), *little litter* (1-2 items), *moderate litter* (3-5 items), and *lots of litter* (6-10 items) according to the number of litter items in images (see Fig. 3.3). The images and labels are available for download from Zenodo at https://doi.org/10.5281/zenodo.7636124.

Additionally, we annotated litter items in 1,501 images using bounding boxes to indicate their locations. These images and bounding box annotations are available for download from Zenodo at https://doi.org/10.5281/zenodo.13730228.

3.2.2. THE OOSTPOORT DATASET

We generated the Oostpoort dataset from experiments conducted during 26 days from February to March 2022, in a canal at Oostpoort, Delft, the Netherlands. We collected data employing two action cameras (GoCam3 and GoPro MAX 360). Fig. 3.4 shows monitoring setups including cameras that are mounted outside the windows of a tower at Oostpoort, at a height of about 5m above the water surface. We recorded video sequences with a time-lapse recording (1 image/30 sec), and a FPS (frame per second) of 17.98. We generated the Oostpoort dataset by saving images from these videos. The

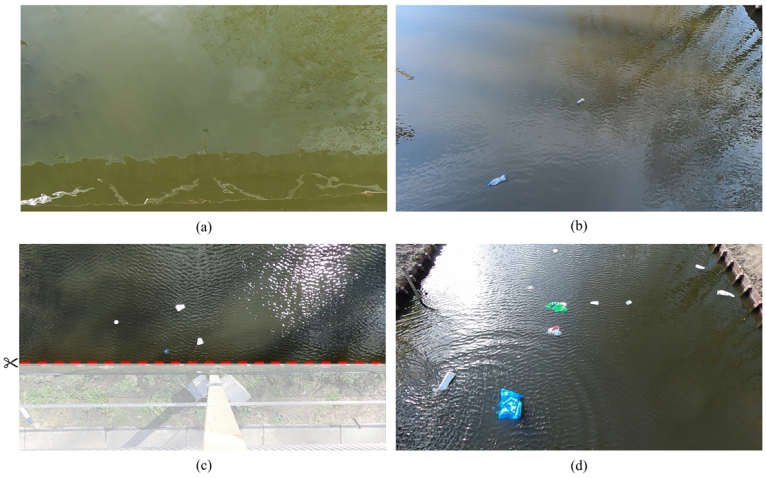


Figure 3.3: Examples of images from the TUD-GV dataset captured from four device setups including (a) 2.7m/0°, (b) 2.7m/45°, (c) 4m/0°, and (d) 4m/45°. The captions for the four images are (a) no litter, (b) little litter, (c) moderate litter, and (d) lots of litter, respectively. The image (c) was cropped to omit the bridge.

Oostpoort images and bounding box annotations are available for download from Zenodo at https://doi.org/10.5281/zenodo.13730298.

Figure 3.4: Monitoring setups at the Oostpoort.

Examples of images can be found in Fig. 3.5. Some images in this dataset contain fauna and various extents of organic material (e.g., leaves and branches), that increases the complexity of the environment owing to their diverse range of color patterns, shapes and sizes. Organic material and floating litter clutter together in garbage patches in some images, making litter harder to detect (van Lieshout et al., 2020).

Figure 3.5: Examples of Oostpoort images.

3.2.3. THE AMSTERDAM DATASET

We created the Amsterdam dataset from one experiment conducted on 1st March 2023, in canals and ponds at Amsterdam, the Netherlands. We recorded images using an action camera (GoPro Hero 10). All images used in this study are captured by the device positioned at a distance of maximum 2 m from the water surface. Examples of these images can be found in Fig. 3.6. The Amsterdam images and bounding box annotations are available for download from Zenodo at https://doi.org/10.5281/zenodo.13730370.

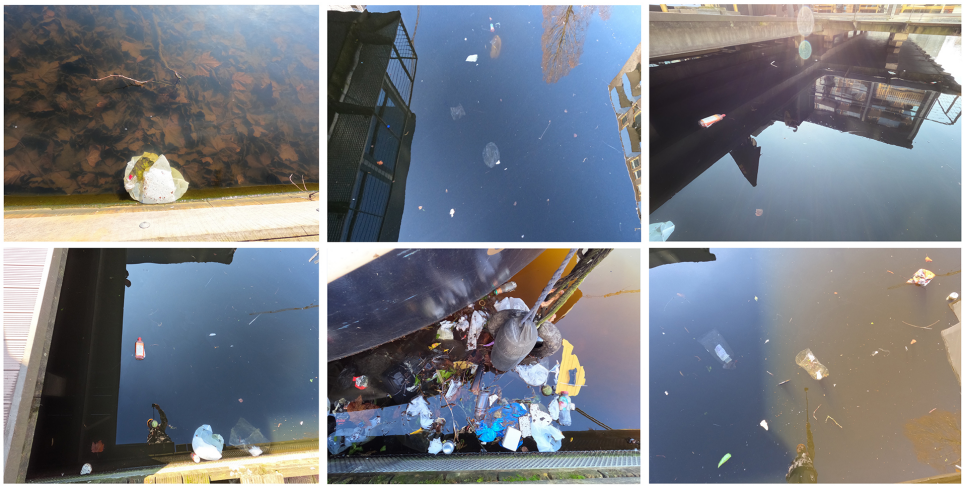


Figure 3.6: Examples of Amsterdam images.

3.2.4. THE GRONINGEN DATASET

We conducted several experiments in a canal in Groningen, the Netherlands, in 2023. We captured data employing a security cameras (Obscape HQ time-lapse). Fig. 3.7 shows

monitoring setups including cameras mounted on a bridge at a height of 4m. We recorded images with a time-lapse recording (1 image/6 sec). Examples of images are shown in Fig. 3.8. The Groningen images and bounding box annotations are available for download from Zenodo at https://doi.org/10.5281/zenodo.13730384.

Figure 3.7: Monitoring setups at Groningen.

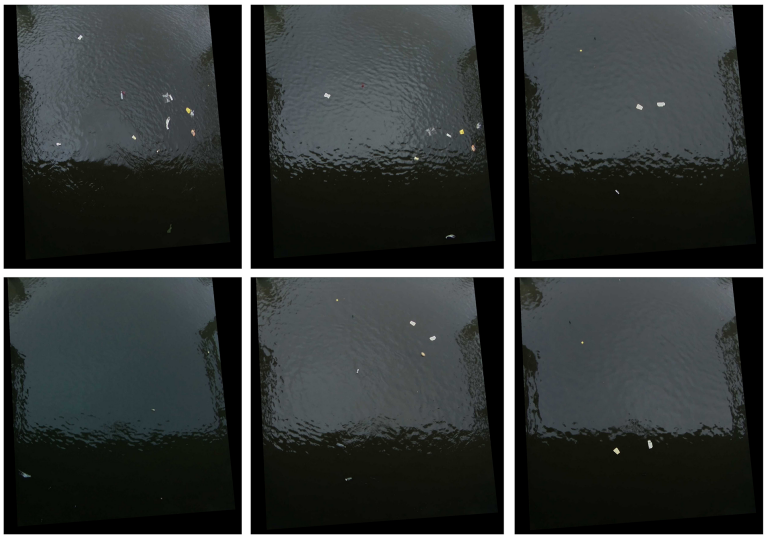


Figure 3.8: Examples of Groningen images. The images used in the experiments are cropped to omit the structure.

3.2.5. THE TU DELFT - HO CHI MINH CITY DATASET

We conducted measurements at two bridges across the Saigon River in Ho Chi Minh City, Vietnam: (1) Binh Loi and (2) Thu Thiem over two days during the wet season in September 2023. Fig. 3.9 shows the location of these bridges in the Saigon River, and our

sampling points on each bridge. The Binh Loi bridge is located in the central part of the city, while the Thu Thiem bridge is situated at the downstream end.

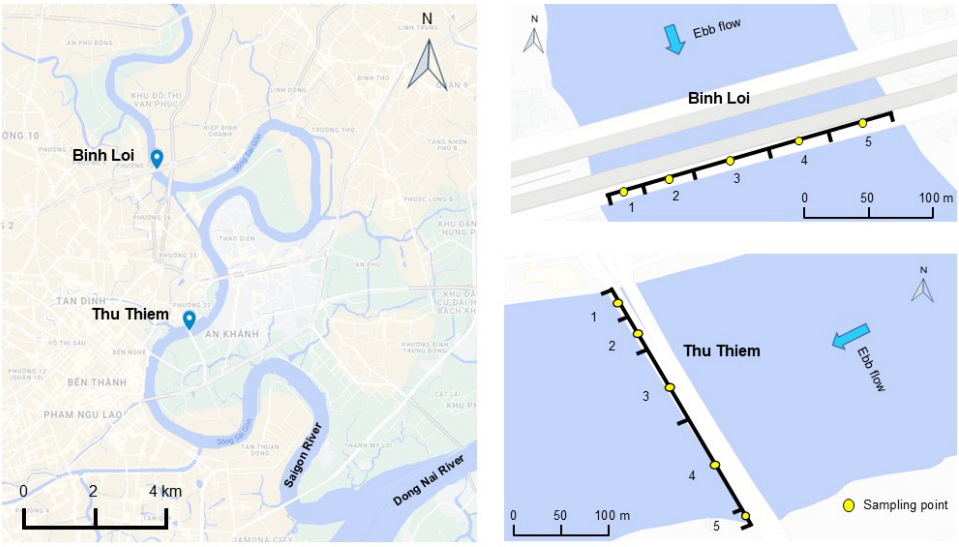


Figure 3.9: The location of Binh Loi and Thu Thiem bridges in the Saigon River (left) and sampling points for each bridge (right).

Table 3.3 shows the details of the measurements. We divided each bridge into five transects, and monitored floating litter at the center of each transect. The length of these transects was carefully selected to ensure that the bridge piers were not visible within the camera's field of view during sampling. All measurements were performed in the southernmost side of bridges during the ebb tide.

Table 3.3: Details of the measurements at Thu Thiem and Binh Loi bridges on the Saigon River

Bridge	River width (m)	Date	Sampling point	Transect width (m)	Observation area width (m)	No. measurement rounds	Sampling duration per point (s)	Time-lapse interval (s)	No. images	No. litter items	No. annotated litter items
Thu Thiem	285	09/09	1	35	7	4	130	10	199	51	64
			2	58							
			3	70							
			4	60							
			5	62							
Binh Loi	228	12/09	1	28	7	6	130	10	309	108	114
			2	33							
			3	69							
			4	85							
			5	12							

On each measurement day, we conducted 4 or 6 rounds of measurements. During each round, we captured images (6016×4000 pixels) sequentially from the sampling point 1 to 5, using a handheld camera (Pentax K-series) over a period $\Delta t_{i,m}$ of 120 or 130 seconds. The camera was oriented nearly vertically with respect to the water surface, with a time-lapse recording (1 image/10 seconds). The observation area width for each sampling point is 7 m, and the ground sampling distance (GSD) of each image is 0.12

cm/pixel. Due to instability in the operation of the handheld camera, we only selected the images without heavy blur for measuring litter fluxes, as reported in Table 3.3. Finally, we built the TUD-HCMC dataset, including the $\text{Test}_{\text{Thu Thiem}}$ and $\text{Test}_{\text{Binh Loi}}$ subsets with 199 and 309 images collected from the Thu Thiem and Binh Loi bridge, respectively. We annotated litter items in images using bounding boxes to indicate their locations. Since some items appear in multiple consecutive images, the number of annotated litter items exceeds the actual number of litter items in the rivers (see Table 3.3). Examples of images are shown in Fig. 3.10.

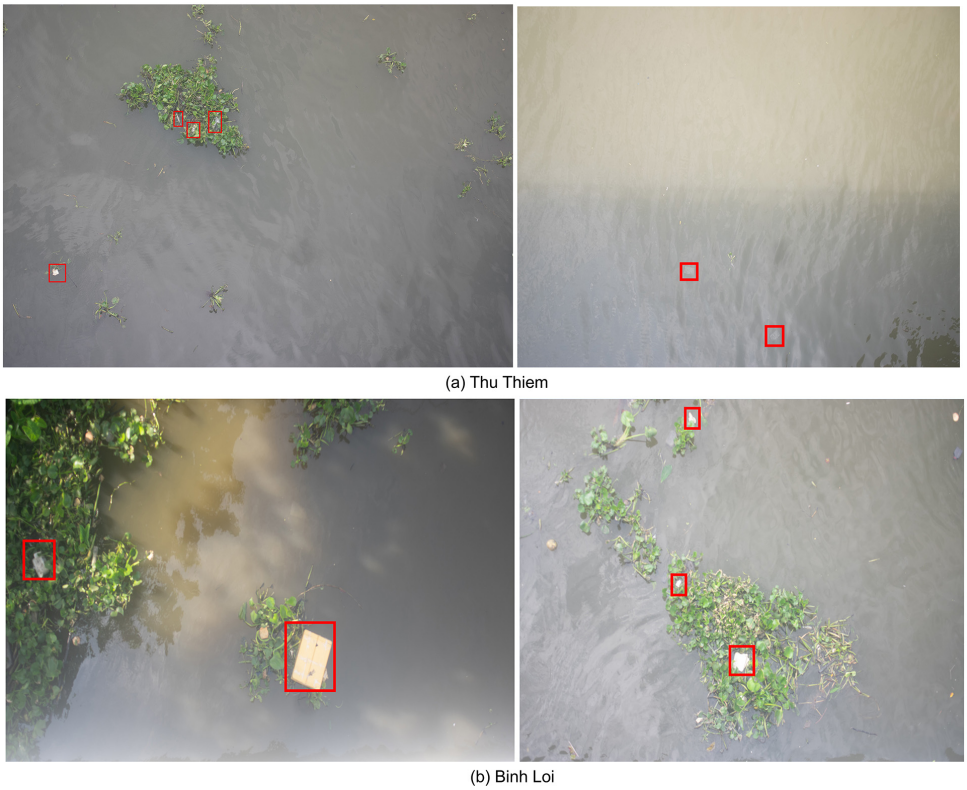


Figure 3.10: Examples of images from TUD-HCMC dataset, including (a) Thu Thiem and (b) Binh Loi images. Ground-truth litter is shown in red bounding boxes.

3.2.6. THE JAKARTA DATASET

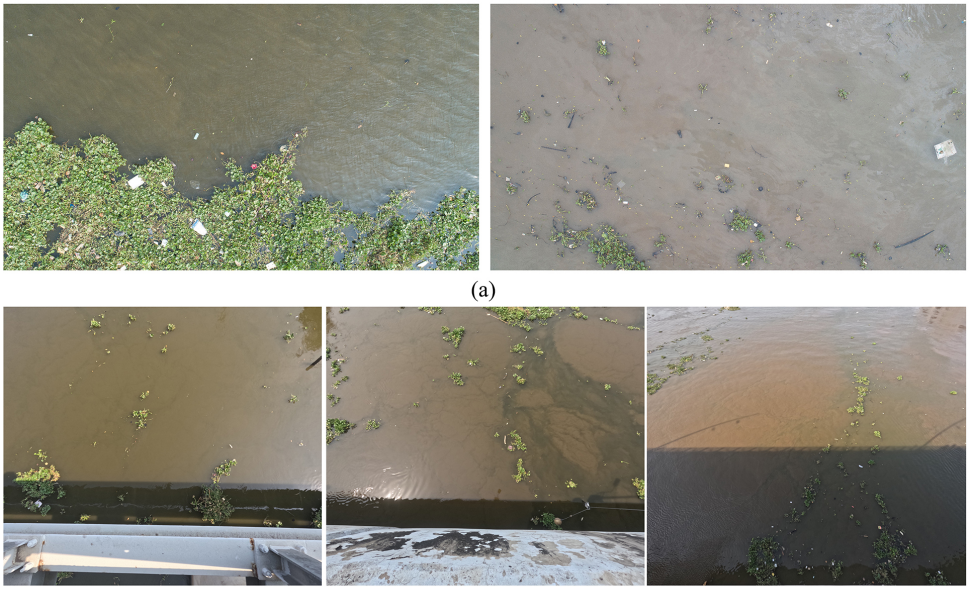
The Jakarta Dataset is an object detection dataset with 526 images and 11064 annotated floating macroplastic litter items. van Lieshout et al. (2020) collected these images using a camera mounted mounted on bridges at five different waterways in Jakarta, Indonesia, from 30 April to 12 May 2018. These images were taken from the view angle of 6 degrees, under various levels of organic material on river surface (i.e., no organic debris, some organic debris, and many organic debris). Most images (1,108) have relatively still water

surfaces, but the remaining images (164) have waves. Examples of images are shown in Fig. 3.11.

Figure 3.11: Examples of images from Jakarta dataset.

3.2.7. THE WAGENINGEN UR - HO CHI MINH CITY DATASET

The Wageningen UR - Ho Chi Minh City (WUR-HCMC) dataset was created by van Emmerik et al. (2024) from WUR. It includes 15,495 images collected from experiments conducted during 8 weeks from February to April 2023, at five locations of the Saigon river at Ho Chi Minh City, Vietnam. They captured images using drones (DJI Phantom 4 Pro) at the Thanh Ho and Quy Kien locations, as well as bridge-mounted cameras (Gopro Hero 11) at the Phu Long, Binh Loi and Thu Thiem bridges. They flew drones across the river width at the altitude ranging from 11 to 14m above the river surface. They installed bridge-mounted cameras with a time-lapse recording (31 image/10 sec) at the height ranging from 7.4 to 18.6m above the river surface. Examples of images are shown in Fig. 3.12.



(a)

(b)

Figure 3.12: Examples of WUR-HCMC images collected by (a) drones and (b) cameras.

4

IMPROVING FLOATING LITTER DETECTION PERFORMANCE WITH TRANSFER LEARNING AND DATA-CENTRIC AI

To improve floating litter detection performances, we first tried transfer learning (TL) methods and data-centric artificial intelligence (AI) approaches, based on the findings in literature review. We evaluated three data-centric artificial AI approaches: (1) data augmentation (DA), (2) adding new images to original training data (ANI), and (3) adding new images and performing DA (ANI-DA). We developed models with five deep learning architectures for a multi-class image classification using about 4000 labeled images from the TU Delft-Green Village dataset. We evaluated the benefits of TL and DA on models' in-domain generalization capability. Additionally, we assessed the benefits of three data-centric AI approaches on models' out-of-domain generalization capability, to unseen litter items and new device settings, such as increasing the cameras' height and tilting them to 45°. The results show that fine-tuning all layers is more effective than the common approach of fine-tuning the classifier alone. Among the tested DA techniques, simple image flipping boosts models' in-domain generalization capability the most, while other methods have little impact on the performance. Models retain good out-of-domain generalization capability which drops significantly only for the most complex scenario tested, but the overall accuracy raises significantly to around 75% when adding a limited amount of images to training data, combined with flipping augmentation (i.e., ANI-DA methods).

This chapter is based on:

Jia, T., Vallendar, A. J., de Vries, R., Kapelan, Z., & Taormina, R. (2023). Advancing deep learning-based detection of floating litter using a novel open dataset. Frontiers in Water, 5, 1298465.

4.1. INTRODUCTION

While some studies have applied deep learning (DL) methods to detect floating litter in rivers with promising results, there is a lack of DL-based detection models with robust generalization performances, as discussed in Chapter 2. This includes models that can detect floating litter for different geographical, environmental or device setup conditions as well as models that can generalize across different case studies (e.g., waterway networks within a country). One major challenge in developing such robust models is the need of large annotated datasets to train and validate robust DL models. These datasets should include images collected from in-situ experiments, with different devices and instrumental settings across various sampling locations under different environmental conditions. Acquiring a sufficiently large dataset can be time-consuming, tedious, and costly.

To partially address this challenge, we found that researchers often utilize techniques such as transfer learning (TL), as discussed in Chapter 2. Moreover, they usually used data-centric artificial intelligence (AI) approaches to improve model performance, such as data augmentation (DA) (see Chapter 2). We do not present the details of TL and DA in this chapter. Readers are referred to Chapter 2.3.5 for more details.

Data-centric AI aims to improve model performances by training on cleaner and more informative datasets (Motamedi et al., 2021). Several studies have shown the benefits of employing these approaches for a wide variety of computer vision-related industrial applications (Im et al., 2021; Tang et al., 2020; Zhou et al., 2019). These approaches usually entail improving the quality of existing data by resorting to pre-processing techniques, systematic labeling, and expert knowledge. For instance, sun glints on the surface of rivers can lead to the misclassification of floating objects (Jakovljevic et al., 2020). Some pre-processing techniques can dilute the effects of these unwanted reflections and boost model performances, as shown already for applications in defect detection and eye tracking (Im et al., 2021; Singvi et al., 2012).

We identified two data-centric AI methods used to improve model detection performance: (1) DA, and (2) Adding New Images (ANI) to the training dataset, from the literature review in Chapter 2. ANI methods involve adding new images collected from new geographical or environmental or device setup conditions to the original training dataset, which improves model generalization in these new conditions. Only one study (van Lieshout et al., 2020) in the literature has evaluated the benefits of ANI methods (see Chapter 2.3.6). van Lieshout et al. (2020) found that a model that performed well for one location in Jakarta, Indonesia, did not generalize well to a different location of the same city, resulting in a drop in precision from 68.7% to 54.0%. This degradation in performance was attributed to the presence of a large amount of organic material (e.g., leaves and branches) in the new images, which was not accounted for during training. They also demonstrated that the generalization performances to different locations improved when the ANI methods were used, by including images from different locations in the training dataset.

TL and data-centric AI approaches are particularly important to develop models with good out-of-domain generalization capability, which is essential for deploying large scale monitoring campaigns. Therefore, we selected these methodologies to improve model generalization capability to detect floating litter in rivers, leveraging a relatively large

amount of labeled data. However, only a few studies (Maharjan et al., 2022; van Lieshout et al., 2020) evaluated their benefits compared to alternatives (e.g., training models from scratch or with non-augmented datasets). Additionally, no study exists on DL-based litter detection reporting a rigorous comparison of TL strategies, DA techniques, ANI methods, and their effects on generalization.

To fill this gap, this chapter evaluates the benefits of these approaches in enhancing model generalization capability to detect floating litter in rivers using the TU Delft-Green Village (TUD-GV) dataset (see Chapter 3). The findings presented in this chapter contribute to answering the first research sub-question of this thesis:

- *How to build robust DL models to detect floating litter in rivers, leveraging a relatively large amount of labeled data?*

The remainder of the chapter is structured as follows. Chapter 4.2 presents the methodology used in this study, including the DL architectures, TL methods, DA techniques, and three data-centric AI approaches to improve generalization capability. Chapter 4.3 describes three sets of experiments, including the datasets used, the experimental setup, and performance evaluation. In Chapter 4.4, we presented and discussed the experimental results. Finally, we summarized the conclusions in Chapter 4.5.

4.2. METHODOLOGY

4.2.1. DEEP LEARNING ARCHITECTURES

We framed the problem of floating litter detection as a multi-class image classification task. We employed five major CNN architectures that have demonstrated good performance on ImageNet classification: ResNet50 (25.6M parameters) (He et al., 2016), InceptionV3 (23.9M) (Szegedy et al., 2016), DenseNet121 (8.1M) (Huang et al., 2017), MobileNetV2 (3.5M) (Sandler et al., 2018), and SqueezeNet (1.2M) (Iandola et al., 2016). The reader is referred to the literature for more details on the employed architectures.

A typical CNN for image classification consists of several convolutional blocks and a classifier. The convolutional blocks are made up of convolutional and pooling layers, which are used to extract features from images. The classifier typically consists of fully connected dense layers that are used to classify images based on the features extracted by the convolutional base (Subramanian et al., 2022). For the purpose of this study, we replaced the original classifier in each CNN architecture with a global average pooling layer followed by a dense layer with a softmax activation function for multi-class classification (i.e., 4 classes). Global average pooling summarizes the feature maps produced by the convolutional base to reduce overfitting and computational costs.

4.2.2. TRANSFER LEARNING

We evaluated the benefits of the most common TL strategies (Guo et al., 2020): (1) fine-tuning the classifier alone (FTC), and (2) fine-tuning all layers (FTAL). We evaluated the effect of transferring features learned on the ImageNet IC task to floating litter detection, a common approach in the field (see Chapter 2). The ImageNet dataset is a widely used benchmark dataset for IC tasks, with more than 20,000 categories (e.g., balloon and strawberry) and over 14 million images. In the FTC strategy, we first loaded the model

pre-trained on ImageNet, then replaced and fine-tuned the classifier on the TUD-GV dataset while freezing the convolutional base (i.e., weights remain fixed during training). In the FTAL strategy, we fine-tuned all layers of the model on the TUD-GV dataset after loading the ImageNet weights as the starting point and replacing the classifier. We compared the effectiveness of FTC and FTAL against the performances obtained by training the models from scratch, that is with random weight initialization.

4.2.3. DATA-CENTRIC AI APPROACHES

We applied three data-centric AI methods to improve the models' generalization capability, including (1) DA; (2) ANI; and (3) using the best DA method after adding new images to the training dataset (ANI-DA).

We evaluated the benefits of four different DA techniques separately, including (1) flipping, (2) brightening, (3) darkening, and (4) adding random salt-and-pepper noise; we also tested (5) mixing all the four aforementioned techniques, an approached hereafter identified as MIX DA. Fig. 4.1 shows examples of each DA technique. Flipping has been shown to be effective on benchmark datasets, such as ImageNet and CIFAR-10 (Recht et al., 2018). Since lighting biases often hinder image classification and object detection (Shorten & Khoshgoftaar, 2019), we also assessed the effect of variations in brightness on model performances. Furthermore, adding noise to images can help CNN models discover more robust features in images (Shorten & Khoshgoftaar, 2019). Techniques such as cropping, rotation, or zooming were not assessed because they may cause the omission of original objects of interest in the new images, leading to undesirable label transformations (Shorten & Khoshgoftaar, 2019).

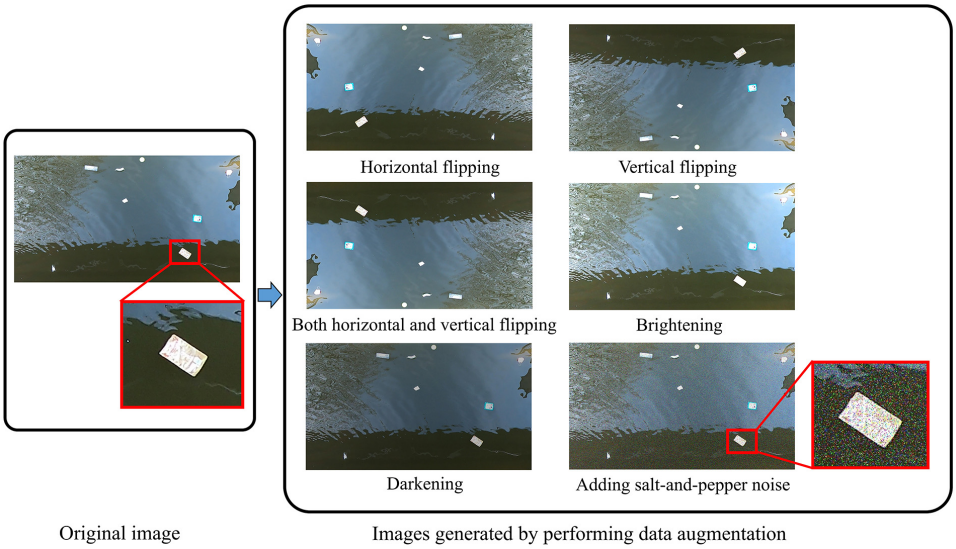


Figure 4.1: Examples of data augmentation techniques used. Left: an original image; Right: images generated by performing horizontal flipping (top row, left), vertical flipping (top row, right), combined horizontal and vertical flipping (middle row, left), brightening (middle row, right), darkening (bottom row, left), and adding salt-and-pepper noise (bottom row, right).

We adopted three types of flipping methods: horizontal flipping (i.e., reversing pixels of an image in the horizontal direction), vertical flipping (i.e., reversing pixels in the vertical direction), and combined horizontal and vertical flipping (i.e., reversing pixels in the horizontal direction and then reversing those in the vertical direction). Each type of flipping was performed to generate one new image from one original image. For brightness augmentation, we used the function provided in the Python Imaging Library (Hadi et al., 2016) by changing the brightness parameter. We generated three new images with different brightness levels by using three random brightness parameters (range [1.1, 1.4]). A brightness parameter value of “0" creates an image with a black color, while a value of “1" returns the original image. Values above “1" create brighter images. Similarly, we employed three random brightness parameters (range [0.6, 0.9]) for darkness augmentation. To add random salt-and-pepper noise, we used the function provided in the Scikit-image library (van der Walt et al., 2014) by changing noise ratio values. We created three new images with different levels of noise by using three random noise ratio values (range [0.01, 0.15]). The noise ratio is the proportion of salt-and-pepper noise in the range [0, 1]. A higher noise ratio value means that there is more salt noise than pepper noise (Azzeh et al., 2018). Each DA method mentioned above was applied to generate three new images for each original training image. MIX DA includes all images generated by the other four DA methods, resulting in a total of 12 new images for each original training image.

4.3. EXPERIMENTS

We conducted three experiments using the TUD-GV dataset. Fig. 4.2 shows the flowchart of them. Experiment 1 aims to compare the in-domain generalization capability of the five DL architectures on the same types of litter items (i.e., canal litter) and device settings (i.e., cameras' height is 2.7m and cameras' angle is 0°), with and without TL. Experiment 2 aims to assess the improvement in in-domain generalization of the five different DA approaches on the two best performing models from Experiment 1. Experiment 3 aims to evaluate and improve the out-of-domain generalization capability of the best models trained on images from 2.7m/0°, for unseen litter (i.e., household waste) and different device setups (camera height/angle: 2.7m/45°, 4m/0°, and 4m/45°).

4.3.1. EXPERIMENT 1: TRANSFER LEARNING IN IN-DOMAIN GENERALIZATION

With the first experiment we compared the detection performances of the five chosen DL architectures (i.e., ResNet, InceptionV3, DenseNet121, MobileNetV2, and SqueezeNet) and we assessed the benefits of the FTC and FTAL strategies described in Chapter 4.2.2. We used a shuffled subset of 4005 images with canal litter for model development. Following the split ratio recommended in Chapter 2, we subdivided into training, validation, and test datasets following the 80/10/10 split detailed in Table 4.1. Ratios between the different classes is kept constant across the different datasets. All images have been recorded from the action cameras with the 2.7m/0° setup.

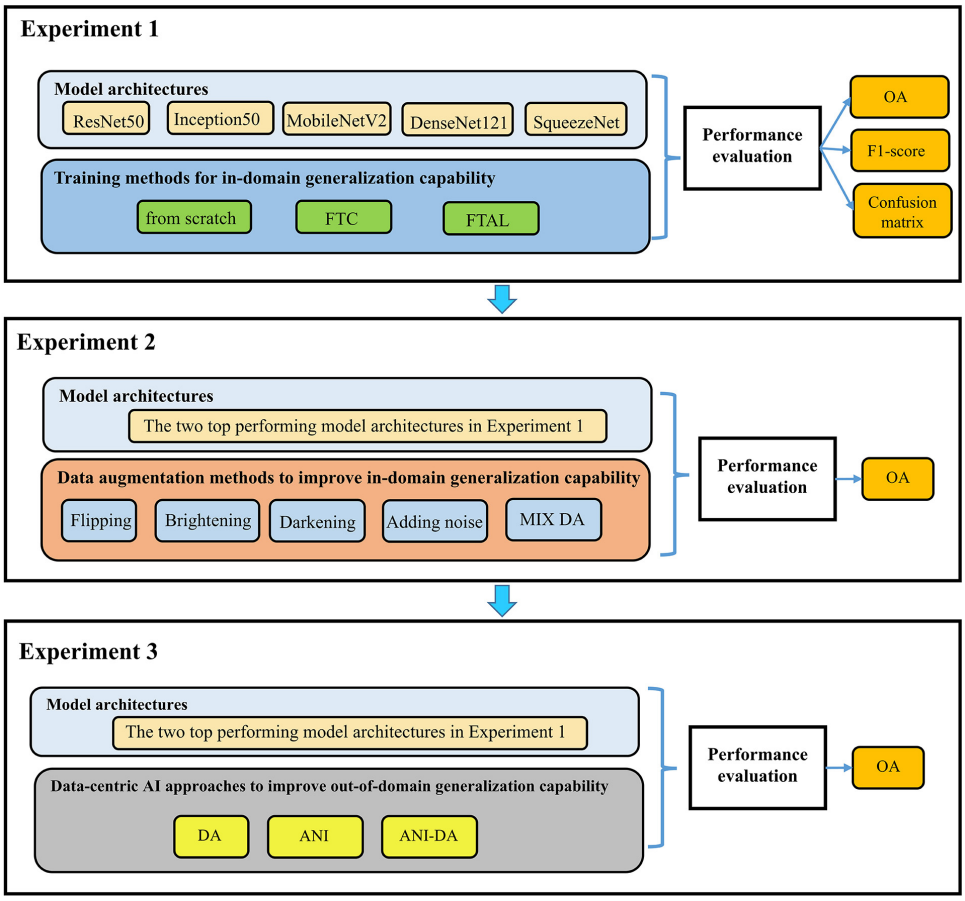


Figure 4.2: The flowchart of three experiments. Acronyms used: Fine-tuning the classifier alone (FTC), Fine-tuning all layers (FTAL), Overall accuracy (OA), Mixing all the four aforementioned techniques (MIX DA), Data augmentation (DA), Adding new images to original training dataset (ANI), Adding new images and performing DA (ANI-DA).

Table 4.1: Datasets for Experiment 1

Dataset name	Device setup (device height/angle)	Litter source	No. images	No. images per class			
				No litter	Little litter	Moderate litter	Lots of litter
Train	2.7m/0°	canal litter	3203	508	752	1088	855
Validation	2.7m/0°	canal litter	399	63	94	136	106
Test	2.7m/0°	canal litter	403	64	95	136	108

4.3.2. EXPERIMENT 2: DATA AUGMENTATION TECHNIQUES IN IN-DOMAIN GENERALIZATION

We applied the five different DA techniques in Chapter 4.2.3 to the two top performing baseline models emerging from Experiment 1. These were retrained on the augmented

datasets yielded by applying each DA techniques, resulting in 12812 or 41639 training images (MIX DA). For a fair comparison against the baselines, we used the same Validation and Test datasets of Experiment 1 (see Table 4.1).

4.3.3. EXPERIMENT 3: DATA-CENTRIC AI APPROACHES IN OUT-OF-DOMAIN GENERALIZATION

To assess the out-of-domain generalization capability to unseen litter items and different device setups, we evaluated the two selected baseline models on the four test datasets reported in Table 4.2. These datasets include camera images of household waste (different from the canal litter present in the original training dataset), filmed with 2.7m/0°, 2.7m/45°, 4m/0°, and 4m/45° device setups, respectively. We performed a misclassification analysis for the best performing baseline model to better understand which features in the test datasets posed challenges to generalization. Next, we evaluated the effects of the methods presented in Chapter 4.2.3 to improve the out-of-domain generalization capability. We implemented the ANI and ANI-DA methods by retraining the two baseline models on the $\text{Train}_{\text{ANI}}$ and $\text{Train}_{\text{ANI-DA}}$ datasets of Table 4.2, respectively. We created the $\text{Train}_{\text{ANI}}$ dataset by adding 1523 images (4726 total) of canal litter to the Train dataset of Experiment 1, from the three missing device setups (2.7m/45°, 4.0m/0°, and 4.0m/45°). These 4726 images still featured canal litter, but were captured from three missing setups to better represent the out-of-domain distributions. The $\text{Train}_{\text{ANI-DA}}$ dataset was created by performing DA on $\text{Train}_{\text{ANI}}$, resulting in a total of 18904 training images. We validated the models for both ANI and ANI-DA cases on the $\text{Validation}_{\text{ANI}}$ dataset, obtained by adding 188 images of canal litter to the Validation dataset of Experiment 1. We compared these models against the baselines of Experiment 1 and the best performing models with DA of Experiment 2.

Table 4.2: Datasets for Experiment 3

Dataset name	Device setup (device height/angle)	Litter source	No. images	No. images per class			
				No litter	Little litter	Moderate litter	Lots of litter
$\text{Train}_{\text{ANI}}$	All[1]	canal litter	4726	1099	1106	1500	1021
$\text{Train}_{\text{ANI-DA}}$	All[1]	canal litter	18904	4396	4424	6000	4084
$\text{Validation}_{\text{ANI}}$	All[1]	canal litter	587	136	138	187	126
$\text{Test}_{\text{2.7m/0°}}$	2.7m/0°	household waste	574	145	126	207	96
$\text{Test}_{\text{2.7m/45°}}$	2.7m/45°	household waste	689	242	193	173	81
$\text{Test}_{\text{4.0m/0°}}$	4.0m/0°	household waste	610	213	163	165	69
$\text{Test}_{\text{4.0m/45°}}$	4.0m/45°	household waste	376	61	71	121	123

[1] "All" device setups includes 2.7m/0°, 2.7m/45°, 4.0m/0°, and 4.0m/45°.

4.3.4. TRAINING SETUP AND PROCEDURE

We resized the RGB images from their original size of 1980 x 1080 x 3 to 224 x 224 x 3 pixels to match the input dimensions of the original pre-trained models. Similarly, we rescaled the input values from a range of 0 to 255 per pixel to a range of 0 to 1. After preliminary trials, we trained all models using a batch size of 16 for 100 epochs. To prevent overfitting, we selected the model parameters from the epoch with the highest validation accuracy. In Experiment 1, we compared five different learning rates (0.1, 0.01, 0.001, 0.0001, and 0.00001) for each model architecture, and only used the best learning rate

in Experiments 2 and 3. We introduced class weights to the cross-entropy loss function used during training to address the slightly imbalanced datasets we created (Wolf et al., 2020). The weight of each class was calculated as the ratio of the total number of images to the number of images in that particular class.

To minimize the effect of randomization, we repeated the training 10 times for each model in all experiments. All the results reported in Chapter 4.4 are mean values calculated from these runs, unless we discuss the outcomes of misclassification analysis, which we conducted on the best performing models out of the 10 runs.

We implemented the DL architectures using the *Python* programming language (version 3.8.5) and the *Keras* DL framework (version 2.6.0). We used the implementations and pre-trained weights from *tf.keras.applications* for all architectures, except for SqueezeNet, retrieved from Malli (2019). Model development was performed on a local NVIDIA GeForce RTX 3090 GPU (24GB).

4.3.5. PERFORMANCE EVALUATION

To evaluate model performances of floating litter detection, we used four metrics commonly employed in multi-class IC tasks: overall accuracy (OA), precision, recall, and F1-score (see Chapter 2). We used OA to summarize model performance across all classes. This metric measures the percentage of correctly identified images out of the total images in the dataset. It is calculated as follows:

$$OA = \frac{\sum_{i=1}^{K} C_{i,i}}{N} \tag{4.1}$$

where N is the total number of images; K represents the number of classes; and $C_{i,i}$ denotes the number of images that are actually in class i and identified as such.

We used precision, recall and F1-score to assess the performances for each class. Precision for class i is written as follows:

$$Precision_i = \frac{TP_i}{TP_i + FP_i} \tag{4.2}$$

where TP_i (True Positive) represents the number of correctly classified images of class i; and FP_i (False Positive) represents the number of images misclassified as class i.

Recall for class i is expressed as follows:

$$Recall_i = \frac{TP_i}{TP_i + FN_i} \tag{4.3}$$

where FN_i (False Negative) represents the number of images that are actually in class i but classified as other classes. Precision reflects how accurate is the model in identifying relevant samples. It identifies the percentage of correctly identified positive samples over the total identified positive samples. On the other hand, recall represents the model's ability to identify all relevant samples. It is the percentage of correctly identified positive samples over the total positive samples. F1-score combines the two metrics by computing their harmonic mean. It is expressed as follows:

$$F1\text{-}score_i = \frac{2 * Precision_i * Recall_i}{Recall_i + Precision_i} \tag{4.4}$$

4.4. RESULTS AND DISCUSSION

4.4.1. EXPERIMENT 1: TRANSFER LEARNING IN IN-DOMAIN GENERALIZATION

Table 4.3 reports the average training time and OA on the Test dataset for the five architectures trained from scratch or fine-tuned after TL. In this table, we only reported the learning rate that yields the best average OA on the validation set for each architecture. The full evaluation of the five architectures with all tested learning rates can be found in Table 8.1 in Appendix 8. The FTAL method consistently outperforms the other methods in in-domain generalization, regardless of the architecture. When using the FTAL method, we obtained OA ranging from 85.0% to 87.6% on the Test set. Training models from scratch performs slightly worse than the FTAL method, with OA ranging between 77.8% and 83.5%. The FTC method performs the worst, with OAs varying between 62.3% and 73.3% depending on the architecture. For example, switching from FTC to FTAL with ResNet50 yields a significant improvement of +22.7% in OA. Although less performing, the FTC method consistently takes the least training time, costing between 2 to 9 seconds for each training epoch. That is approximately 2 to 5 times faster than using the FTAL method or training the models from scratch. This is expected since training or fine-tuning the entire network takes significantly more time than fine-tuning the classifier alone.

Table 4.3: Learning rate, training time, and overall accuracy of all architectures for Experiment 1

Model	Scheme	Learning rate	Training time per epoch (s)	Overall accuracy (%)
ResNet50	from scratch	0.001	22	83.3
	FTC	0.01	8	62.3
	FTAL	0.001	13	**85.0**
InceptionV3	from scratch	0.001	21	83.0
	FTC	0.0001	7	66.5
	FTAL	0.001	20	**85.7**
DenseNet121	from scratch	0.0001	28	83.5
	FTC	0.001	9	73.3
	FTAL	0.0001	18	**87.6**
MobileNetV2	from scratch	0.01	19	81.7
	FTC	0.0001	4	72.7
	FTAL	0.001	19	**86.2**
SqueezeNet	from scratch	0.00001	5	77.8
	FTC	0.0001	2	65.8
	FTAL	0.0001	4	**87.6**

These results suggest that, while the features learned from ImageNet may not fully transfer to the task of classifying floating litter, initializing model parameters with pretrained weights on the ImageNet dataset provides a better starting point for the models than random initialization. Thus, the FTAL method may enable models to achieve better performance faster. This aligns with the findings of other studies demonstrating a

decrease in the transferability of learned features when the base task (e.g., classification on ImageNet) differs significantly from the target task (Yosinski et al., 2014).

Our findings are similar to those reported by Marin et al. (2021) for a study on CNN architectures detecting underwater litter. The authors classified images into six classes: glass, metal, plastic, rubber, other trash, and no trash. Even for this case, the FTAL strategy proved more successful than resorting to FTC, with best performance on the test dataset of OA=91.4% compared to 83.0%.

We found that DenseNet121 outperforms the other architectures, regardless of the the training procedure adopted, with a maximum OA of 87.6%. The superior performances of DenseNet121 may stem from the dense connectivity patterns in its architecture, which favors feature propagation and reuse across layers, while reducing the total number of trainable weights (Huang et al., 2017). Despite having only 1.2M parameters, SqueezeNet also achieves the highest OA of 87.6%. Due to its size, SqueezeNet is the fastest to train, however its detection performance depends significantly on the training procedure adopted, with a difference of +21.8% between FTC and FTAL. SqueezeNet requires less trainable parameters to achieve high accuracy due to its innovative architecture that makes use of 1x1 filters (9X fewer parameters than common 3x3 filters) and “fire modules" (Iandola et al., 2016). These results might have practical implications for distributed monitoring of litter on edge computing devices (e.g., Raspberry Pi or other single-board computers connected to a camera), where litter recognition is performed locally using with limited resources (Liu et al., 2021a).

Table 4.4 presents the F1-score per class for the five architectures using the FTAL method. Precision and recall can be found in Table 8.2 in Appendix 8. All models perform similarly across different classes, showing best performances for “no litter" or “lots of litter" with F1-scores of up to 0.98 and 0.89, respectively. The models show good but lower accuracy for the other two classes, with F1-scores ranging from 0.79 to 0.86. The features for these two intermediate classes may not be highly distinctive, leading to a higher probability of misclassification. For example, Table 4.5 shows the confusion matrix for DenseNet121 using FTAL. We observed a relatively high number of errors for images belonging to the “moderate litter" class, are sometimes confused with “little litter" (14 case) or “lots of litter" (5 case), resulting in the lowest F1-scores for this class across all architectures, ranging from 0.79 to 0.83.

Table 4.4: F1-score per class of all architectures trained using the FTAL strategy for Experiment 1

Model	F1-score			
	No litter	Little litter	Moderate litter	Lots of litter
ResNet50	0.98	0.83	0.79	0.87
InceptionV3	0.98	0.84	0.80	0.87
DenseNet121	0.97	0.86	0.83	0.89
MobileNetV2	0.97	0.86	0.81	0.86
SqueezeNet	0.98	0.88	0.83	0.87

Table 4.5: Confusion matrix of the best performing DenseNet121 trained with the FTAL strategy for Experiment 1

True label	Predicted label			
	No litter	Little litter	Moderate litter	Lots of litter
No litter	**63**	1	0	0
Little litter	3	**84**	8	0
Moderate litter	0	14	**117**	5
Lots of litter	0	0	12	**96**

4.4.2. EXPERIMENT 2: DATA AUGMENTATION TECHNIQUES IN IN-DOMAIN GENERALIZATION

Fig. 4.3 compares the average in-domain generalization performances of the best configurations of SqueezeNet and DenseNet121 from Experiment 1 against that obtained by retraining these baselines using the different DA techniques in Chapter 4.2.3. The baseline performance is indicated by a horizontal dashed line at OA=87.6% since the performances on the Test dataset is the same for both models. The results show that the flipping technique is the most effective in improving model performances, with a significant improvement in OA (+2.0% for SqueezeNet and +4.1% for DenseNet121) compared to the baseline models. This confirms that flipping augmentation is recommended as it does not distort the features in the images with respect to the original label. The other techniques show a slight increase or decrease in OA (from -0.1% to +0.8%), possibly due to the excessive transformation of the original images (Shorten & Khoshgoftaar, 2019). Although using brightening and darkening techniques should increase model robustness to different lighting conditions, these techniques may not be as effective in this particular case since the original images in the TUD-GV dataset were taken in both sunny and cloudy weather. The MIX DA strategy results in a good increase in OA (+1.9% for SqueezeNet and +3.4% for DenseNet121), however, these gains are lower than those achieved by flipping alone. Additionally, the training times for MIX DA are approximately three times longer (see Table 8.3 in Appendix 8).

DenseNet121 outperforms SqueezeNet when using flipping or MIX DA techniques, with an increase in OA of +2.1% and +1.4%, respectively. The OA of DenseNet121 is also higher when using the other DA techniques, although the difference is not as significant. It is generally accepted that a more complex model, such as DenseNet121, can benefit more when trained on a sufficiently large dataset, as it has more capacity to learn and capture patterns in the data. In comparison, a lightweight model like SqueezeNet may not be able to fully take advantage of additional training data generated through DA (Zhu et al., 2016). Therefore, it may be necessary to increase model complexity in order to fully leverage the benefits of additional training data. However, the training times for DenseNet121 are five to six times longer than for SqueezeNet (see Table 8.3 in Appendix 8). This trade-off should be considered when choosing a model for a particular specific litter detection task.

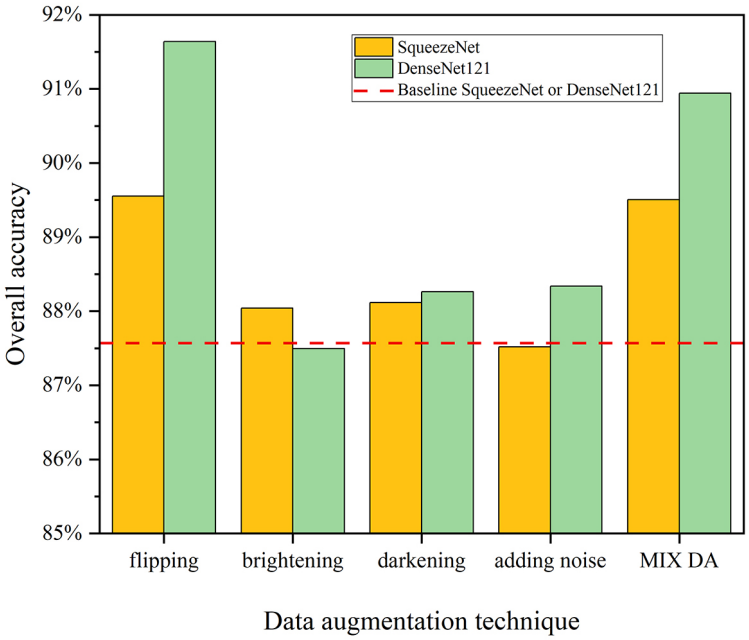


Figure 4.3: In-domain performances of SqueezeNet and DenseNet121 using different DA techniques for Experiment 2. The horizontal dashed line represents the OA of the baseline models, trained without DA. DA techniques include (1) flipping, (2) brightening, (3) darkening, (4) adding noise, and (5) mixing the four above-mentioned techniques (MIX DA).

4.4.3. EXPERIMENT 3: DATA-CENTRIC AI APPROACHES IN OUT-OF-DOMAIN GENERALIZATION

Fig. 4.4 compares the out-of-domain generalization performances of the baseline models against that of the models modified using three data-centric AI approaches described in Chapter 4.2.3. We implemented DA and ANI-DA by applying flipping augmentation alone, due to its demonstrated effectiveness in Experiment 2. The results show that both SqueezeNet and DenseNet121 trained on data with canal litter captured with the 2.7m/0° setup (i.e., Train dataset in Table 4.1) can generalize well to household waste litter under the same device setup (i.e., $\text{Test}_{2.7\text{m}/0°}$ of Table 4.2), achieving OA of 84.4% and 85.3%, respectively. Although the generalization capability in this case is already satisfactory, it can be further improved. Specifically, DenseNet121 models trained with DA and ANI-DA show significant increases is OA of +5.4% and +6.2%, respectively, while ANI alone does not provide a similar boost. Lesser improvements are also measured for SqueezeNet. Although ANI-DA performs the best, it requires the time-consuming and costly collection of new data. Therefore, simple flipping augmentation may be the most cost-effective method to improve the generalization capability under the same device setup.

The SqueezeNet and DenseNet121 baselines exhibit good performances on $\text{Test}_{2.7\text{m}/45°}$, with OA of 90.7% and 83.8%, respectively. Overall, results are similar or better than for

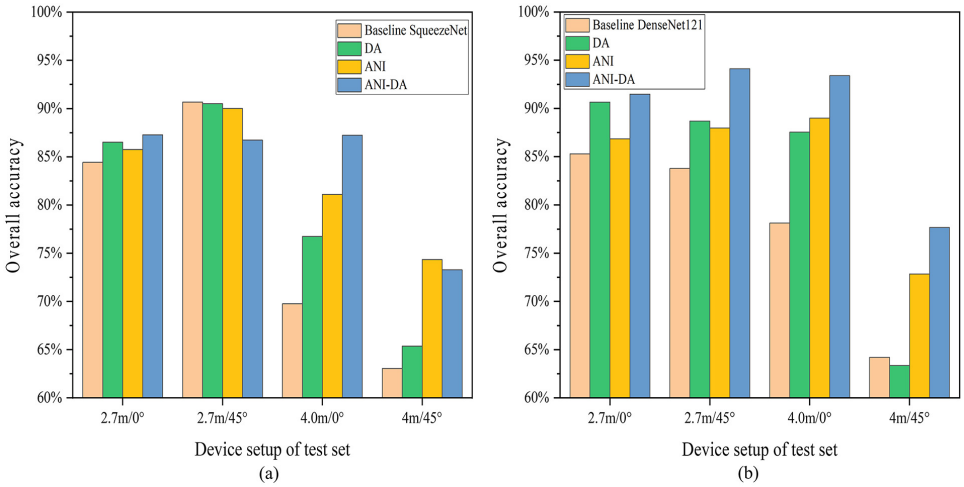


Figure 4.4: Out-of-domain generalization performances of SqueezeNet (a) and DenseNet121 (b) on Experiment 3, featuring baseline models and models leveraging techniques for improved generalization. Comparison performed on datasets of four device setups (2.7m/0°, 2.7m/45°, 4m/0°, and 4m/0°) with household litter. Acronyms used: Data augmentation (DA), Adding new images to original training dataset (ANI), Adding new images and performing DA (ANI-DA).

the simpler $\text{Test}_{2.7\text{m}/0^\circ}$ because $\text{Test}_{2.7\text{m}/45^\circ}$ consists exclusively of images taken in cloudy weather. Sunny weather images are harder to classify due to the presence of sun glints (Jakovljevic et al., 2020). The three approaches significantly improve the generalization capability of DenseNet121, with OA=94.1% for ANI-DA. On the other hand, we could not improve the performances of SqueezeNet further, suggesting that this small architecture cannot incorporate larger amount of data effectively. Nonetheless, SqueezeNet still retains good generalization capability with OAs above 86.7%.

The generalization of the baseline models drops significantly for the more complex device setups, i.e., $\text{Test}_{4\text{m}/0^\circ}$ and $\text{Test}_{4\text{m}/45^\circ}$. SqueezeNet achieves an OA of 69.8% and 63.1% on these test datasets, respectively; while DenseNet121 obtains an OA of 78.1% and 64.2%. To gain insight into the factors contributing to these poor performances, we conducted a qualitative inspection of 192 and 147 images misclassified by the best baseline SqueezeNet model. Fig. 4.5 shows common errors, including (a) identifying sun glints as extra litter (126 cases in $\text{Test}_{4\text{m}/0^\circ}$), (b) undetected items of small size (41 and 79 cases in $\text{Test}_{4\text{m}/0^\circ}$ and $\text{Test}_{4\text{m}/45^\circ}$, respectively), and (c) unseen objects during training (e.g., a PVC pipe and a wood stick, 40 cases in $\text{Test}_{4\text{m}/45^\circ}$). DL models are known to suffer from sun glints, changes in the scale and in the distribution of items (Jakovljevic et al., 2020; Singh & Davis, 2018; van Lieshout et al., 2020).

The ANI method outperforms simple flipping augmentation on these harder datasets, with improvements of around +11% for both architecture in each setup. While flipping grants significant increases of up to +9.4% in $\text{Test}_{4\text{m}/0^\circ}$, it fails to support generalization for the more complex $\text{Test}_{4\text{m}/45^\circ}$. This suggests that simple DA fails to boost generalization when the out-of-domain distribution is significantly different from the training one

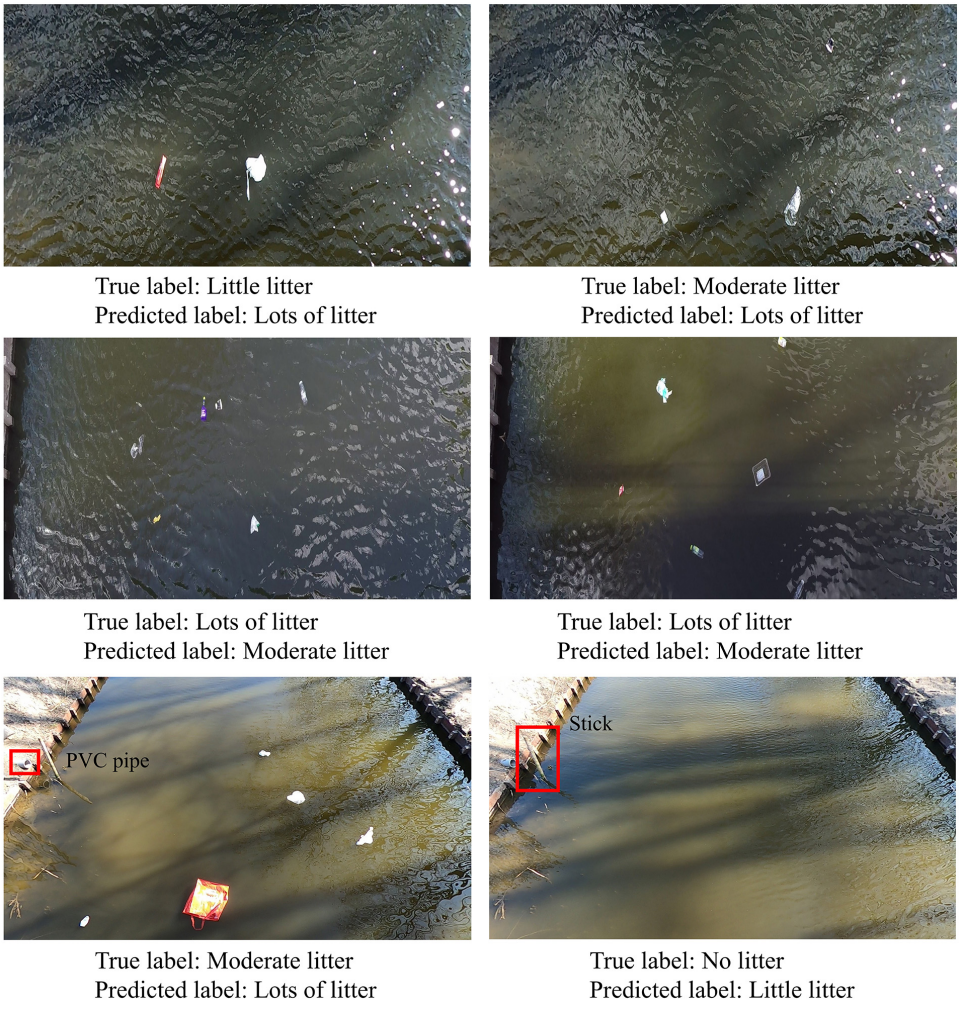


Figure 4.5: Common misclassified examples in the $\text{Test}_{4m/0°}$ and the $\text{Test}_{4m/45°}$ datasets for the best baseline SqueezeNet model. Common misclassification include identifying sun glints as litter, failure to detect small-sized litter, and detection of background objects or external items.

(e.g., different items, camera heights, and viewing angle). In these cases, collecting new data from the new setup is necessary to achieve satisfactory performances. Performing DA after gathering new images can result in further improvements, as demonstrated in $\text{Test}_{4m/0°}$ for both SqueezeNet and DenseNet121 (i.e., OA of 87.2% and 93.4%, respectively) and in $\text{Test}_{4m/45°}$ for DenseNet121 (OA=77.7%).

4.4.4. LIMITATIONS

We acknowledge some limitations in the TUD-GV dataset used in this study and approach, that necessitate further developments for real-world applications. First, although the TUD-GV dataset features items collected from canals, the level of litter degradation does not fully represent the situation encountered in many real contexts. Second, the current research does not account for the interference of vegetation and natural debris, that are intrinsically present in real-world scenarios. Similarly, images gathered from our semi-controlled experiments in a stagnant canal—although representative of urban areas— do not account for the complexity of dynamic environments such as rivers and coastal areas where litter interaction with flow, waves, and other factors is commonplace. Third, the TUD-GV dataset does not include images collected during nighttime, thus it can not be used for developing models to detect and quantify the floating litter items during nighttime. Fourth, this study does not focus on maximizing model performance by pre-processing the raw input images before DA. Tiling images into smaller patches (e.g., 224*224) will likely boost performances by retaining the original image quality (Wolf et al., 2020), although this would require relabeling all tiles. Lastly, real-world applications demand more sophisticated computer vision tasks than the image classification performed here. Object detection and image segmentation methods are preferred approaches to identify, quantify and track floating litter in water bodies from images or videos.

4.5. CONCLUSIONS

In this chapter, we carried out a thorough evaluation of different transfer learning (TL) methods and three different data-centric AI approaches to improve in-domain and out-of-domain generalization performances for floating litter detection, using the TU Delft-Green Village dataset. Three data-centric AI approaches include: (1) data augmentation (DA), (2) adding new images to original training data (ANI), and (3) adding new images and performing DA (ANI-DA). The main findings of this study are as follows:

1. We obtained the best in-domain generalization performances by loading models pre-trained on ImageNet, replacing the classifier, and fine-tuning the entire network on floating litter images. The benefits of this TL approach in terms of detection performance outweigh the shorter training times required by fine-tuning the classifier alone. Transferring the convolutional base from ImageNet seems a better approach than training the models from scratch, at least for our experiments.

2. We recommend flipping DA to improve model in-domain generalization performances at relatively low cost, since the additional images are easy to generate, and maintain high fidelity to the original labels while providing extra training information. On the other hand, brightening, darkening, and adding noise do not show a significant improvement in detecting floating litter.

3. The trained models generalize well to similar conditions, such as detecting unseen litter items from images captured at the same height, but with different viewing angles (i.e., 45°). Flipping DA may boost out-of-domain generalization performances in these circumstances, but it is insufficient when transferring to more

complex scenarios (e.g., different camera heights and different viewing angle). We demonstrated that adding a limited amount of images from these new settings to the original training dataset can substantially improve generalization in these cases.

5

SEMI-SUPERVISED LEARNING FOR FLOATING LITTER DETECTION

While previous analysis show that the transfer learning and data-centric artificial intelligence approaches are effective to improve model generalization capability, developing robust models in a supervised manner requires a large number of labeled images. Obtaining these labeled images for model development is costly and labor-intensive. To address this issue, we proposed a two-stage semi-supervised learning method to detect floating litter based on the Swapping Assignments between multiple Views of the same image (SwAV). SwAV is a self-supervised learning approach that learns the underlying feature representation from unlabeled data. In the first stage, we used SwAV to pre-train a ResNet50 backbone architecture on about 100k unlabeled images. In the second stage, we added new layers to the pre-trained ResNet50 to create a Faster R-CNN architecture, and fine-tuned it with a limited number of labeled images (up to ≈1.8k images with 2.6k annotated litter items). We developed and validated our semi-supervised floating litter detection methodology for images collected in canals and waterways of Delft (the Netherlands) and Jakarta (Indonesia). We tested for out-of-domain generalization performances in a zero-shot fashion using additional data from Ho Chi Minh City (Vietnam), Amsterdam and Groningen (the Netherlands). We benchmarked our results against the same Faster R-CNN architecture trained via supervised learning alone by fine-tuning ImageNet pre-trained weights. The findings indicate that the semi-supervised learning method matches or surpasses the supervised learning benchmark (e.g., average precision and F1-score) when tested on new images from the same training locations. We measured better performances when little data (up to ≈200 images with about 300 annotated litter items) is available for fine-tuning and with respect to reducing false positive predictions. More importantly, the proposed approach demonstrates clear superiority for generalization on the unseen locations, with improvements

This chapter is based on:

Jia, T., de Vries, R., Kapelan, Z., van Emmerik, T. H. M., & Taormina, R. (2024). Detecting floating litter in freshwater bodies with semi-supervised deep learning. Water Research, 266, 122405.

in average precision of up to 12.7%. We attribute this superior performance to the more effective high-level feature extraction from SwAV pre-training from relevant unlabeled images.

5.1. INTRODUCTION

The results in Chapter 4 show that the transfer learning and data-centric artificial intelligence approaches are effective to improve model generalization capability. However, our findings show that even using the best-performing transfer learning method and developing supervised models (SL) on a relatively large amount of labeled data (4000 images), their generalization capability is still limited. The main reason is that the effectiveness of transfer learning declines when the base and target tasks become less similar (Yosinski et al., 2014). The literature review in Chapter 2 indicates that previous studies pre-trained models on one of the following comprehensive datasets: (i) ImageNet, (ii) COCO (Lin et al., 2014), (iii) CIFAR-10 (Recht et al., 2018), and (iv) Pascal VOC (Everingham et al., 2010). However, the high-level features in these datasets have limited relevance with respect to floating litter imagery. This may hinder performances and generalization capability.

Moreover, obtaining these and additional labeled images for model development and refinement is costly and labor-intensive, and relies on domain-specific knowledge on floating litter detection (Guo et al., 2021). Although previous studies have not reported the time required for manually generating labels for a litter detection dataset, studies from other fields indicate significant time involved. For example, annotating 1000 instances across 91 common categories (e.g., car) using pixel-level segmentation masks in the COCO dataset requires more than 22 worker hours (Lin et al., 2014).

To overcome the limitations associated with SL, the deep learning research community is increasingly investigating self- and semi-supervised learning methods due to their data efficiency and generalization capability (Liu et al., 2021c). Self-supervised learning operates by using the unlabeled input data to automatically generate its own labels, learning the underlying representations from the data itself without explicit guidance (Misra & Maaten, 2020). The mainstream self-supervised learning approaches include two categories: generative and discriminative. The generative self-supervised approach learns feature representations by performing pixel-level reconstruction, but requires extensive computational resources (Goodfellow et al., 2014; Kingma & Welling, 2013). Discriminative self-supervised learning approaches learns feature representations using objective functions which are similar to those used in supervised learning method. This is achieved by employing pretext tasks, where both inputs and labels are derived from an unlabeled dataset (Chen et al., 2020).

Typical pretext tasks include relative position prediction (Doersch et al., 2015), Jigsaw puzzle (Noroozi & Favaro, 2016), and rotation prediction (Gidaris et al., 2018). However, these heuristic tasks might limit the generality of the learned representations (Chen et al., 2020). More recently, discriminative approaches based on contrastive learning have gained momentum (Jaiswal et al., 2020). Contrastive self-supervision obtains representations by distinguishing between positive pairs (similar instances) and negative pairs (dissimilar instances) (Jaiswal et al., 2020). For example, the Simple framework for Contrastive Learning of visual Representations (SimCLR) generates two different views from each input image by performing data augmentation (Chen et al., 2020). The positive pairs include two augmented views from the same image, while the negative pairs are formed by sampling two augmented views from different images. Other successful alternatives include Swapping Assignments between multiple Views of the same image (SwAV) (Caron et al., 2020), and Momentum Contrast (MoCo) (He et al., 2020).

Semi-supervised learning (SSL) enhances self-supervised pre-trained models regardless of the method used. SSL leverages a small amount of labeled data to address specific downstream tasks such as image classification, object detection, and image segmentation (Reddy et al., 2018). This operation can also be regarded as a form of transfer learning, where knowledge is transferred from pretext tasks using unlabeled data. Recent studies have shown that SSL methods outperform traditional supervised learning approaches for applications on large-scale datasets (e.g., ImageNet), as well as domain-specific applications, including agriculture (Güldenring & Nalpantidis, 2021) and medical imaging (Miller et al., 2022). While SSL approaches are promising, they have not been applied to detect floating litter.

In this chapter, we proposed a two-stage semi-supervised learning method based on the SwAV approach for detecting floating litter in rivers. In the first stage, we use SwAV to pre-train a ResNet backbone architecture on about 100k unlabeled images with floating litter. First, SwAV uses data augmentation methods to create multiple augmented views from the input unlabeled image. Then, SwAV enables models to learn data representations using a "swapped" prediction mechanism, leveraging the inherent similarities between the views. In the second stage, we create a Faster R-CNN architecture for object detection by adding new deep learning layers to the backbone, and fine-tune them using only a limited number of labeled images (≈1.8k) with 2.6k annotated litter items. This process facilitates the transfer of knowledge learning from SwAV pre-training is transferred to the Faster R-CNN for litter detection task. Based on the analysis of the effectiveness of transfer learning and data augmentation presented in Chapter 4, we believe that the proposed SSL method, integrates these methods, can enhance model generalization capability for litter detection.

In this chapter, we developed and validated the methodology for images collected in canals and waterways of the Netherlands, Indonesia, and Vietnam. These images were sourced from in six datasets, where four were generated by us, while the other two were obtained from publications (see Chapter 2.3.2). Furthermore, we assessed the transferability of low-level (e.g., edges) and high-level (e.g., entire objects and shapes) representations learned via SwAV pre-training. The findings presented in this chapter contribute to answering the second research sub-question of this thesis:

- *How to build robust deep learning models to detect floating litter in rivers, leveraging a limited amount of labeled data?*

5.2. METHODOLOGY

5.2.1. OVERVIEW OF THE SEMI-SUPERVISED LEARNING APPROACH

We propose a two-stage semi-supervised learning method for detecting floating litter based on Swapping Assignments between multiple Views of the same image (SwAV). The approach includes a self-supervised learning stage and supervised learning stage. Fig. 5.1 shows the schematic illustration of the proposed SSL method. In the first stage, we used SwAV to pre-train a ResNet50 network (He et al., 2016) with a large quantity of unlabeled data. We used the ResNet50 as the backbone of our methods since most of the studies in contrastive learning successfully employ variants of this architecture (Jaiswal et al., 2020). To obtain the final model, we first created a Faster R-CNN architec-

ture for object detection (Ren et al., 2015) by adding extra deep learning layers after the pre-trained ResNet50. Then, we fine-tuned the resulting deep learning model using a limited amount of labeled data to perform the specific litter detection downstream task. We describe SwAV and Faster R-CNN in Section 5.2.2 and Section 5.2.3, respectively. Section 5.2.4 presents details on the implementation of the self-supervised pre-training methods, while the supervised stage is illustrated in Section 5.2.5.

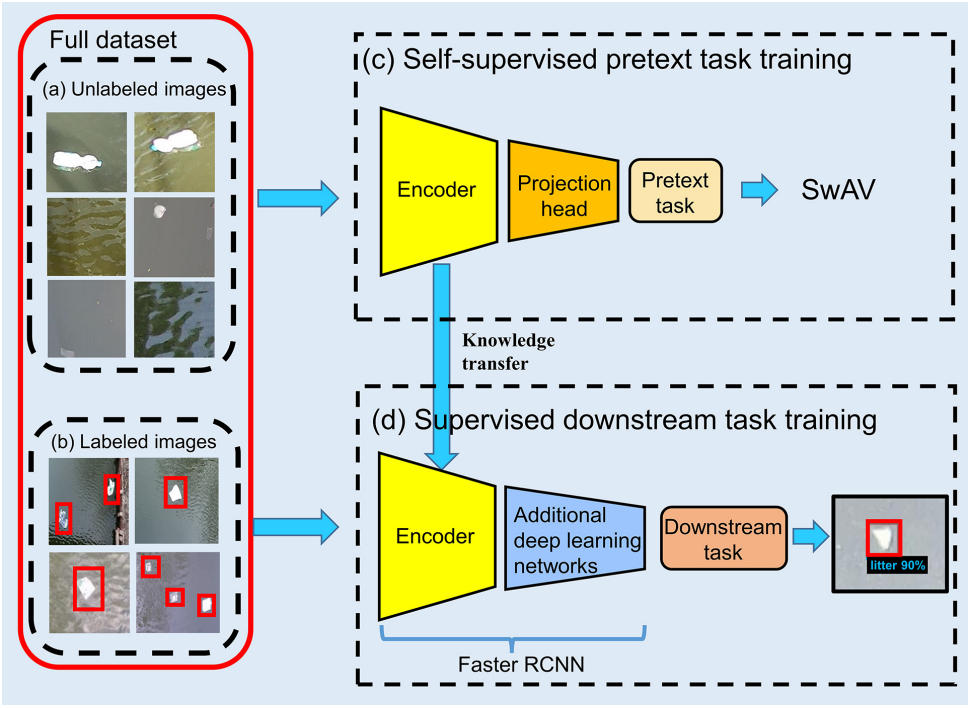


Figure 5.1: The schematic illustration of the proposed two-stage semi-supervised learning method. In the self-supervised learning stage (c), we used SwAV to pre-train a ResNet50 encoder network combined with a projection head, using a large number of unlabeled images (a); Then, we added additional deep learning network to ResNet50 backbone to create a Faster R-CNN architecture. In the supervised learning stage (d), we fine-tuned the Faster R-CNN to learn a specific litter detection downstream task in a supervised manner, using a limited amount of labeled data (b).

5.2.2. SWAPPING ASSIGNMENTS BETWEEN MULTIPLE VIEWS OF THE SAME IMAGE (SWAV)

SwAV is a cluster-based self-supervised contrastive learning method (Caron et al., 2020). Models learn the underlying representations from the data by performing a clustering assignment prediction between various augmentations (or "views") of the same input image. Fig. 5.2 shows the schematic illustration of SwAV. The process begins with data augmentation (e.g., multi-crop and flipping) to generate multiple views of the input image X. In Fig. 5.2, we only show the multi-crop augmentation method, that crops an image randomly into two global views with standard resolution crops (e.g., 224×224 pixels)

and several local views with smaller resolution crops (e.g., 96×96 pixels). For simplicity, we only present two views (x_1, x_2). These views are processed by the same encoder network f_θ (e.g., ResNet50) followed by a projection head (e.g., 2-layer multilayer perceptron) to generate two corresponding feature vectors (z_1, z_2). To perform the online clustering assignment, SwAV uses the Sinkhorn−Knopp algorithm (Cuturi, 2013) to map the feature vectors to a set of prototypes C comprising K prototype vectors. Each prototype represents a cluster in the feature space. This operation results in the generation of the codes Q_1 and Q_2. The uniqueness of SwAV lies in its "swapped" prediction mechanism. Here, the code Q_2, derived from the view x_2, is predicted using the characteristics of the view x_1 and vice versa. This prediction method leverages the inherent similarities between the views, as they originate from the same image. Consequently, SwAV refines its learning of data attributes by forecasting the code of one image view based on the features of its counterpart. This is achieved by minimizing the loss of the following function:

$$L(z_1, z_2) = l(z_1, Q_2) + l(z_2, Q_1) \tag{5.1}$$

5

where $l(z, Q)$ measures the fit between the feature z and the code Q. It can be computed as follows:

$$l(z_1, Q_2) = -\sum_k Q_2^{(k)} \log p_1^{(k)} \tag{5.2}$$

$$l(z_2, Q_1) = -\sum_k Q_1^{(k)} \log p_2^{(k)} \tag{5.3}$$

$$p_1^{(k)} = \frac{\exp\left(\frac{1}{\tau}\mathbf{z}_1^\top \mathbf{c}_k\right)}{\sum_{k'} \exp\left(\frac{1}{\tau}\mathbf{z}_1^\top \mathbf{c}_{k'}\right)} \tag{5.4}$$

$$p_2^{(k)} = \frac{\exp\left(\frac{1}{\tau}\mathbf{z}_2^\top \mathbf{c}_k\right)}{\sum_{k'} \exp\left(\frac{1}{\tau}\mathbf{z}_2^\top \mathbf{c}_{k'}\right)} \tag{5.5}$$

where C_k is the k-th prototype vector in C, and τ denotes the temperature parameter that controls the sharpness of the probability distribution (Caron et al., 2020).

Two major core components of SwAV are clustering assignment and multi-crop augmentation strategy. SwAV's clustering assignment avoids the direct comparison of negative and positive pairs in contrastive learning. That reduces the computational overhead and potential noise introduced by large sets of negative samples, leading to more efficient and robust model training compared to other contrastive learning methods. The multi-crop augmentation strategy improves performance of self-supervised methods with only a small increase in the memory and computational cost. These allow SwAV outperform other recent and successful contrastive learning methods (e.g., SimCLR and MoCo) on the ImageNet classification benchmark (Caron et al., 2020).

5.2.3. FASTER R-CNN FOR LITTER DETECTION

Fig. 5.3 shows the detailed architecture of the Faster R-CNN with a ResNet backbone. The Faster R-CNN is a two-stage detection network, including four modules: (1) feature

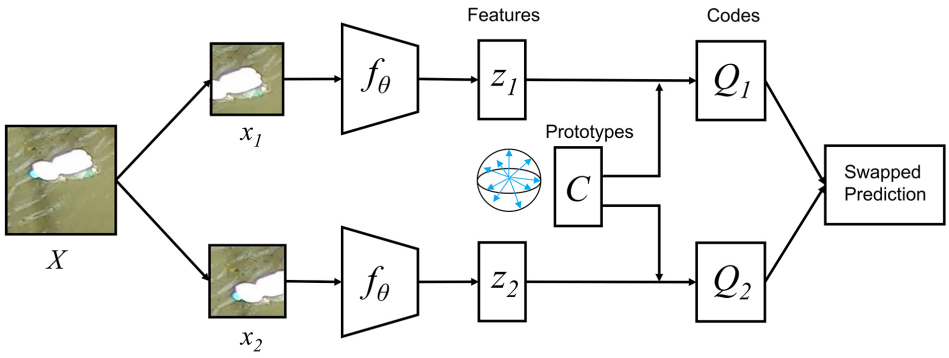


Figure 5.2: The schematic illustration of SwAV adapted from (Caron et al., 2020). First, each image X is augmented into two different views (x_1, x_2), that are processed by the encoder f_θ to obtain two feature vectors (z_1, z_2). Then, the codes of these two features (Q_1, Q_2) are computed by mapping them to prototypes C. Finally, SwAV learns data representations by solving a "swapped" prediction problem, where the code Q_2 is predicted using the view x_1 and vice versa.

5

extraction; (2) object proposal generation; (3) Region of Interest (RoI) pooling; and (4) classification with a confidence level and location prediction (Li et al., 2019). Confidence refers to the probability assigned by the Faster R-CNN when classifying each bounding box. In the first stage of the Faster R-CNN, the backbone extracts relevant feature maps from the input data. Then, the region proposal network, a fully convolutional network, generates region proposals from the shared feature maps. These region proposals together with the feature maps are fed into the RoI pooling layer, performs the pooling operation to integrate feature maps of region proposals with different scales into fixed size feature maps. In the second stage, the extra-network predicts the category with a confidence level and the precise location of objects from each region proposal in the fixed size feature maps.

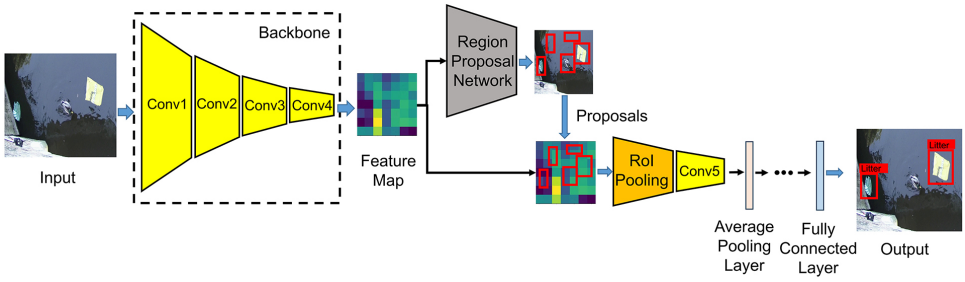


Figure 5.3: The schematic illustration of the Faster R-CNN with ResNet backbone. The basic ResNet (yellow blocks) mainly includes two parts: (1) convolutional blocks Conv1 to Conv4, and (2) Conv5. In the first stage of the Faster R-CNN, the backbone first extracts feature maps from the input data. Then, the Region Proposal Network produces region proposals from these feature maps. Furthermore, the feature maps and region proposals are fed into the RoI Pooling layer, that converts the feature maps of proposals into fixed size feature maps for the final classification and location prediction in the second stage.

The ResNet mainly includes two parts: (i) convolutional blocks Conv1 to Conv4, and (ii) Conv5 (He et al., 2016). Both parts are pre-trained by SwAV in the self-supervised learning stage. Then, the Faster R-CNN is constructed by using Conv1 to Conv4 as the backbone and adding Conv5 after the RoI pooling layer.

5.2.4. SWAV PRE-TRAINING

To evaluate the benefits of self-supervised pre-training, we used two pre-training methods for all experiments: (1) SwAV-FTAL, and (2) SwAV-Scratch, as inspired by two transfer learning strategies in Chapter 4. The SwAV-FTAL method first initializes the ResNet backbone with ImageNet weights, and then uses SwAV to fine-tune all the layers (FTAL) of the backbone on the unlabeled images. ImageNet weights used in this study were created by training the ResNet50 on 1.2 million images (1,000 categories) from the full ImageNet dataset. We selected ImageNet weights since transferring features learned from the ImageNet image classification task to other domain tasks is a widely used approach to detect floating litter, as highlighted in the literature review in Chapter 2. The SwAV-Scratch method uses SwAV to pre-train the ResNet50 from scratch. It involves initializing the ResNet50 backbone with random weights, and then using SwAV to pre-train all the layers of the backbone on the unlabeled images.

5.2.5. FINE-TUNING FOR LITTER DETECTION

To perform the litter detection downstream task, we fine-tuned Faster R-CNN architectures built on the pre-trained ResNet50 backbone. We compared two different approaches for fine-tuning, that entail freezing either 4 convolutional blocks (F4, from Conv1 to Conv4 in Fig. 5.3) or 2 (F2, Conv1 and Conv2) of the ResNet backbone, respectively. During fine-tuning, only the unfrozen layers of the Faster R-CNN are updated. The F2 method is a common method used to transfer low-level feature knowledge learned from pre-training to the downstream task. In contrast, the F4 method transfers both low-level and high-level feature knowledge. In situations where only a small dataset is available for model fine-tuning, maintaining relevant high-level features becomes crucial as it drastically reduces the number of weights to fine-tune. By examining the F4 modality, we aim to evaluate whether the high-level features learned via SwAV pre-training enhances the model's generalization capabilities in data scarce conditions. This investigation can help understand whether this approach can lead to the development of foundational models for litter quantification across multiple locations (Oquab et al., 2023).

5.3. EXPERIMENTS

We conducted multiple experiments to investigate the potential of SSL for floating litter detection using images from: (1) The TU Delft - Green Village (TUD-GV), the Netherlands, (2) Oostpoort, the Netherlands, and (3) Jakarta, Indonesia (see Chapter 3). Moreover, we tested the generalization capability of our proposed method using images captured in three other locations: (1) Amsterdam and (2) Groningen, the Netherlands, and (3) The Wageningen UR - Ho Chi Minh City (WUR-HCMC), Vietnam (see Chapter 3).

We evaluated both *in-domain* as well as *out-of-domain* generalization capability. In-

domain generalization refers to the model performance on new, unseen images from the same geographic locations, while out-of-domain generalization refers to unseen images from other geographic locations. We compared the results with those obtained from a supervised learning benchmark, providing a robust reference point. Additionally, we investigated how the litter detection performance varies with the availability of labeled data for fine-tuning. This aspect is crucial for assessing the models' practical applicability in scenarios with limited annotated resources. Complementing this analysis, we evaluated the relevance of low-level and high-level representations learned from SwAV pre-training with respect to generalization. This examination can share further insights on the suitability of SSL for developing large-scale monitoring networks for quantifying floating litter across multiple locations.

5.3.1. DATA SELECTION

Table 5.1: Data used in this chapter, sourced from TU Delft-Green Village (TUD-GV), Oostpoort, and Jakarta dataset.

	Subset			Total
	TUD-GV	Oostpoort	Jakarta	
Total images	1,501	562	526	2,589
Total image tiles	44,188	71,445	16,762	132,395
No. image tiles with litter annotated	1,969	401	1,399	3,769
No. annotated litter items	2,542	457	2,531	5,530

We created the Delft-Jakarta dataset by selecting random images from the TUD-GV, Oostpoort, and Jakarta locations, as reported in Table 5.1. These images were sliced into tiles with a standard size of 224×224 pixels, to match the input dimensions of ResNet50 (Pham et al., 2021). Example image tiles are shown in Fig 9.1 in Appendix 9. We used the Delft-Jakarta dataset to train and validate the models, and to test their in-domain generalization performance. In total, we extracted a total of 132,395 image tiles from the Delft-Jakarta datasets. These were used to randomly create the non-overlapping subsets for self-supervised pre-training (116,286 tiles), supervised fine-tuning (1,756 tiles), validation (164 tiles), and testing (14,189 tiles), detailed in Table 5.2. Almost 90% of the tiles were used for self-supervised pre-training with SwAV ($\text{Train}_{\text{self}}$). These tiles have no labels. We used a maximum of 1,756 image tiles for supervised fine-tuning ($\text{Train}_{100\%}$), containing a total of 2,628 annotated litter items. The annotations are bounding boxes representing the location of floating litter items, without further categorization. To better assess model performance with respect to the availability of labels, we created six smaller fine-tuning datasets by reducing the number of tiles and annotations down to 5% ($\text{Train}_{80\%}$ to $\text{Train}_{5\%}$). We used a maximum of 164 image tiles and 282 annotations for model validation ($\text{Validation}_{100\%}$), maintaining a 9-to-1 ratio with respect to the data available for fine-tuning. For consistency, we created six smaller validation datasets ($\text{Validation}_{80\%}$ to $\text{Validation}_{5\%}$). We created a Test dataset by including 1,849 tiles with 2,620 annotations. To better evaluate the models performance with respect to false pos-

itives, we included 12,340 image tiles with no floating litter.

Table 5.2: The Delft-Jakarta subsets used in the experiments.

Learning method	Training dataset			Validation dataset			Test dataset			No. tiles without litter
	Name	No. annotated litter items	No. tiles	Name	No. annotated litter items	No. tiles	Name	No. annotated litter items	No. tiles	
Self-supervised	$Train_{self}$	0	116,286							
Semi-supervised and supervised	$Train_{100\%}$	2,628	1,756	$Validation_{100\%}$	282	164	Test	2,620	1,849	12,340
	$Train_{80\%}$	2,076	1,389	$Validation_{80\%}$	224	117				
	$Train_{60\%}$	1,594	1,059	$Validation_{60\%}$	171	100				
	$Train_{40\%}$	1,013	702	$Validation_{40\%}$	115	70				
	$Train_{20\%}$	527	368	$Validation_{20\%}$	62	55				
	$Train_{10\%}$	282	180	$Validation_{10\%}$	27	22				
	$Train_{5\%}$	124	84	$Validation_{5\%}$	13	9				

To evaluate out-of-domain generalization, we sliced randomly selected images from the Amsterdam, Groningen and WUR-HCMC datasets, as detailed in Table 5.3. The tiles in these subsets contain both images with annotated litter and without litter. Example image tiles are shown in Fig 9.2 in Appendix 9.

5

Table 5.3: The Amsterdam, Groningen and WUR-HCMC datasets used to evaluate out-of-domain generalization.

	Subset			Total
	Amsterdam	Groningen	WUR-HCMC	
Total images	9	63	27	99
Total image tiles	3,623	5,544	13,032	22,199
No. image tiles with litter annotated	152	439	766	1,357
No. annotated litter items	204	525	1,091	1,820
No. image tiles without litter	3,471	5,105	12,266	20,842

5.3.2. DEVELOPED MODELS AND EXPERIMENTS

For brevity, we indicated models built via pre-training with the SwAV-FTAL method and fine-tuning with the F2 method, as SwAV-FTAL-F2 across all experiments. Other models are named in the same way, e.g., SwAV-FTAL-F4, SwAV-Scratch-F2, and SwAV-Scratch-F4. We compared the effectiveness of SSL against baseline supervised learning models which are developed without the SwAV pre-training step. These models are Faster R-CNNs fine-tuned on labeled data, built on ResNet50 backbones initialized with ImageNet weights (see Fig. 5.1 (b) and (d)). For consistency, we used two types of baseline models: (1) Baseline-F2, and (2) Baseline-F4, that uses the F2 and F4 methods for fine-tuning, respectively.

We developed all models by using the Delft-Jakarta subsets in Table 5.2. Specifically, we built the SSL models by first pre-training a ResNet50 encoder with a projection head of 2-layer multilayer perceptron on the $Train_{self}$ subset. We then fine-tuned the Faster R-CNN derived from the ResNet50 backbone on all the seven available subsets for supervised learning, i.e., $Train_{100\%}$ to $Train_{5\%}$. We performed model validation on

the respective Validation subsets. The Baseline supervised learning models are developed in the same fashion, but without SwAV pre-training. The Delft-Jakarta Test subset is used for evaluating the in-domain generalization. On the other hand, we evaluated out-of-domain generalization using the image tiles from Amsterdam, Groningen and WUR-HCMC detailed in Table 5.3. For out-of-domain generalization, we tested only the models fine-tuned using the maximum amount of the Delft-Jakarta labeled data, i.e., $\text{Train}_{100\%}$. We used the SwAV-FTAL-F2, SwAV-Scratch-F2 and Baseline-F2 methods to evaluate the quality of transferred low-level representations. Similarly, we investigated the relevance of high-level representations by implementing the SwAV-FTAL-F4, SwAV-Scratch-F4 and Baseline-F4 methods.

5.3.3. PERFORMANCE ASSESSMENT

To assess model performance of floating litter detection, we used two commonly employed metrics in the literature review in Chapter 2: i) AP50, representing the Average Precision (AP) with an Intersection over Union (IoU) threshold of 50% and ii) F1-score computed using the same threshold. The IoU measures the ratio of the overlap area of prediction and ground truth to their union area, which is described as follows (Chen et al., 2024):

$$IoU = \frac{area(bbox_{pred} \cap bbox_{gt})}{area(bbox_{pred} \cup bbox_{gt})} \tag{5.6}$$

where $bbox_{pred}$ and $bbox_{gt}$ are the predicted bounding box and the ground-truth bounding box, respectively. The larger the IoU, the greater the overlap of these two bounding boxes (Xue et al., 2021b). After setting an IoU threshold, we can compute the elements of the confusion matrix for the object detection task. For each ground-truth box, we have a True Positive (TP) if there is at least one overlapping predicted box with IoU equal or above the threshold. Predicted boxes overlapping the ground-truth with IoU less than the threshold are marked as False Positives (FP). If more bounding boxes sufficiently overlap with the ground truth, we mark as TP only the one with the highest confidence (Dollár & Lin, 2014). The others are marked as FP. FPs also include incorrect detection of nonexistent objects. False Negatives (FN) are the undetected ground-truth bounding boxes.

The AP is the average precision of the models for a given IoU threshold. It is computed as the area under the precision-recall curve (Padilla et al., 2020). The precision p and recall r are expressed as follows:

$$p = \frac{TP}{TP + FP} \tag{5.7}$$

$$r = \frac{TP}{TP + FN} \tag{5.8}$$

Precision measures the accuracy of the positive predictions, denoted by the ratio of correctly identified positive cases (TP) to the total number of cases identified as positive (TP + FP). On the other hand, recall is the ratio of correctly identified positive cases (TP) to the actual total positive cases (TP + FN). It assesses the model's ability to detect

all relevant instances. For object detection, the precision-recall curve is computed by i) sorting all detections in descending order based on their confidence level, ii) accumulating all TPs and FPs, iii) and computing p and r for each cumulative detection (Dollár & Lin, 2014; Padilla et al., 2020). In the computation of r for the accumulated detections, the denominator term is constant and equal to the total amount of ground-truth boxes. After creating the precision-recall curve, we can calculate AP by integrating the area under it:

$$AP = \int_0^1 p(r)dr \tag{5.9}$$

AP is an average measure that can sometimes obscure model weaknesses, e.g., a model might achieve good AP through a few highly accurate detections but perform poorly on others. The computation method for the precision-recall curve can also introduce challenges since the precision at each recall level can be subject to fluctuations due to the model's varying confidence levels across different detections (Padilla et al., 2020). The F1-score may provide a more balanced metric of precision and recall at the same IoU threshold. The F1-score is computed as the harmonic mean of p and r, is calculated as follows:

$$F1 - score = \frac{2 * p * r}{p + r} \tag{5.10}$$

The F1-score captures a model's accuracy in detecting objects (recall) while minimizing incorrect detections (precision), making it crucial for contexts where false positives and false negatives have significant implications. Thus, combining AP50 and F1-score allows for a more thorough assessment of both localization accuracy and overall detection efficacy.

5.3.4. BOUNDING BOX REFINEMENT WITH NON-MAXIMUM SUPPRESSION

Before making the final predictions and computing performances, we refined the output bounding boxes via Non-Maximum Suppression (NMS) (Hosang et al., 2017). NMS is a post-processing technique often applied after object detection to eliminate redundant bounding boxes, and ensure that each detected object is represented by the single most probable box. It compares the overlap of boxes using IoU and suppresses all boxes except the one with the highest confidence score when the overlap exceeds a specific threshold. For all our experiments and developed models, we set the IoU NMS threshold equal to 0.5 for consistency. This is a common value that balances the need to reduce box overlap against the risk of missing closely spaced objects.

5.3.5. TRAINING SETUP AND PROCEDURE

We implemented all experiments with the *Python* programming language (version 3.8.16), using the *PyTorch* deep learning framework (version 1.8.1), in combination with the *VISSL* (Goyal et al., 2021) and the *Detectron2* (Wu et al., 2019) libraries. We trained and tested all deep learning models on a NVIDIA Tesla V100S PCIe GPU (32 GB) ((DHPC), 2022).

We used default *VISSL* hyperparameters for SwAV pre-training, including a cluster with 3000 prototype vectors. We pre-trained for 100 epochs, using the SGD optimizer with cosine annealing learning rate scheduling (Loshchilov & Hutter, 2016), with the initial rate of 0.075 and the minimum value of 7.5×10^{-5}. We applied four default *VISSL* data augmentation methods: (1) multi-crop with 8 views (2×[224×224]+6×[96×96]), (2) horizontal flipping, (3) color distortion, and (4) Gaussian blur.

In the supervised learning stage, we fine-tuned the Faster R-CNN with default *Detectron2* hyperparameters, including an SGD optimizer with a fixed learning rate of 0.02, a weight decay of 0.0001 and a momentum of 0.9. Before being fed into Faster R-CNN, the input images were resized while preserving the aspect ratio. The shortest side of each image was randomly scaled to one value among {640, 672, 704, 736, 768, 800} pixels, while ensuing that the longest side did not exceed 1333 pixels. We fine-tuned all Faster R-CNN models for 100 epochs, and selected the model yielding the highest validation accuracy for further model evaluation. Before being fed into Faster R-CNN for inference, input images were resized while preserving the aspect ratio. The shortest edge of image was resized to a length between 800 and 1333 pixels, ensuring the longest edge does not exceed 1333 pixels.

We implemented the Baseline methods using the same fine-tuning hyperparameters. We trained all models for 100 epochs, saving the learned parameters yielding the highest validation accuracy.

5.4. RESULTS AND DISCUSSION

5.4.1. IN-DOMAIN DETECTION PERFORMANCES FOR VARYING DATA AVAILABILITY

Fig. 6.3 compares the AP50 detection performance on Delft-Jakarta Test subset for the SwAV-FTAL-F2, SwAV-Scratch-F2 and Baseline-F2 methods. The three methods perform similarly when relatively more data is available for fine-tuning (i.e., $\text{Train}_{60\%}$ to $\text{Train}_{100\%}$ subsets), with an AP50 ranging from 62.8% to 65.8%. When less labeled data is available (i.e., $\text{Train}_{5\%}$ to $\text{Train}_{40\%}$ subsets), the SwAV-FTAL-F2 method performs best in most cases, obtaining an AP50 ranging from 44.3% to 60.4%. This yields a slight improvement in AP50 of up to 2.3%, compared to the baseline method (AP50=44.4%~59.3%). The SwAV-Scratch-F2 method performs worst (AP50=37.3%~57.4%), yielding a slight decrease in AP50 varying from 5% to 7.1%, compared to the baseline method in half of these cases. Fig. 6.3 also indicates a general upward trend in performance with increasing amount of labeled data, regardless of the approach used. The observed performance plateau could be attributed not only to the limited size of our labeled dataset, but also to the lack of hyper-parameter tuning and the fact that only a single training run was conducted, due to computational limitations (SwAV pre-training time: 12 min/epoch). The stochastic nature of neural network training means that multiple runs yields different results, possibly influencing the observed performance ceiling (Punjani & Fleet, 2021).

At first glance, these results suggest that transferring low-level representations learned by SwAV on unlabeled, but relevant data, does not yield substantial improvements with respect to simple transfer from ImageNet. In particular, learning from scratch via SwAV hinders performance when little data is available for fine-tuning, although the situa-

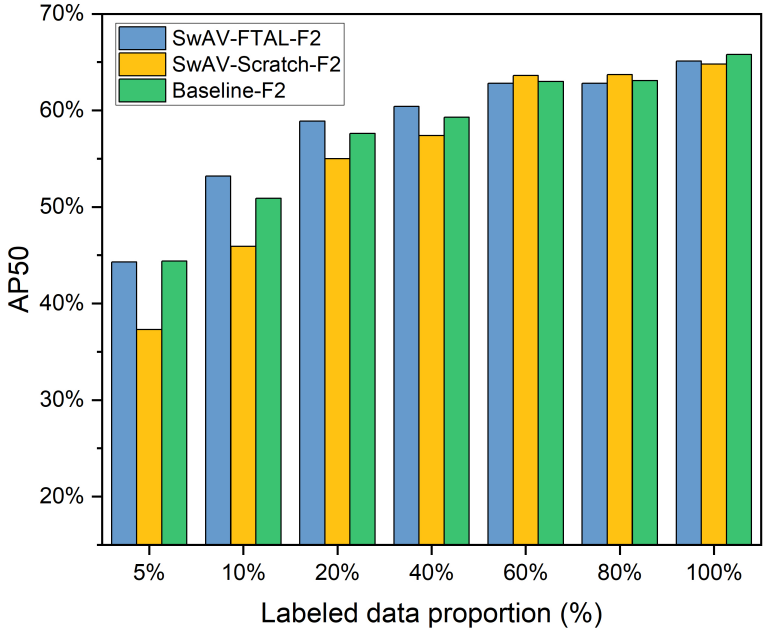


Figure 5.4: AP50 detection performance of the SwAV-FTAL-F2, SwAV-Scratch-F2 and Baseline-F2 methods on the Test subset with different proportion of labeled data for fine-tuning.

tion rapidly improves when more labels are available. However, one must consider that the ImageNet dataset (1.2 million images) contains over 10 times more images than the $\text{Train}_{\text{self}}$ subset used for SwAV pre-training. The availability of large amounts of data enables ResNet50 to learn robust low-level features that are used by the deeper layers fine-tuned for the downstream litter detection task with Faster R-CNN. Furthermore, the ImageNet pre-trained weights are the product of extensive optimization on substantial computational resources, which contrasts sharply with our constrained SwAV pre-training that involved limited runs and no hyper-parameter tuning. Despite these limitations, we achieved comparable results, showcasing the potential effectiveness of our methodology. Better performances can be obtained by scaling the datasets and the computational efforts. Literature reports strong increases in SSL performances with larger SwAV pre-training datasets, e.g., from 1.2 million to 14 million to 1 billion (Goyal et al., 2022).

Regardless of the above limitations in our SwAV implementation, the SSL methods outperform the baseline when considering other metrics. Table 5.4 reports the Test dataset confusion matrix, precision, recall and F1-score for the three methods fine-tuned on $\text{Train}_{100\%}$. The Baseline-F2, yields overall marginally better recall (0.74 vs 0.71), but substantially lower precision (0.48 vs 0.57) than the SSL methods. This results in a lower

F1-score (0.58 vs 0.63) due to a much higher number of FPs. Similar worse performances are found for images without litter, where the number of FPs of the baseline is around double that of the SSL methods.

Table 5.4: Confusion matrix, Precision, Recall and F1-score on the Delft-Jakarta Test subset for models fine-tuned on the $\text{Train}_{100\%}$ dataset. False positives are also reported for 12,340 additional images without litter.

Method	Test dataset						Images without litter
	TP	FN	FP	Precision	Recall	F1-score	FP
SwAV-FTAL-F2	1,850	770	1,391	**0.57**	0.71	**0.63**	3,666
SwAV-Scratch-F2	1,832	788	**1,359**	**0.57**	0.70	**0.63**	**3,594**
Baseline-F2	**1,926**	**694**	2,093	0.48	**0.74**	0.58	7,453
SwAV-FTAL-F4	**1,775**	**845**	2,024	0.47	**0.68**	0.55	6,788
SwAV-Scratch-F4	1,680	940	**1,373**	**0.55**	0.64	**0.59**	**3,192**
Baseline-F4	1,590	1,030	2,296	0.41	0.61	0.49	9,167

The benefits of SwAV pre-training clearly emerge when preserving the high-level feature representations, as reported in Fig. 5.5. The results show that both the SwAV-FTAL-F4 and SwAV-Scratch-F4 methods significantly outperform the Baseline-F4 benchmark, regardless of the amount of labeled data available for fine-tuning. The SwAV-FTAL-F4 method performs best in most cases, achieving an AP50 ranging from 39.9% to 60.5%. The SwAV-Scratch-F4 method performs worse when very limited labeled data is available, but then achieves comparable or higher scores, with the highest reported score of 60.9% for $\text{Train}_{80\%}$. The baseline method obtains AP50 varying between 19.3% and 51.1%. These values are particularly low when little data is available for fine-tuning (i.e., Train_5 and $\text{Train}_{10\%}$ subsets), where SwAV-FTAL-F4 and SwAV-Scratch-F4 yield improvements in AP50 of up to 20%. The SSL approaches only requires 20% of the labeled data (527 annotated litter items) to achieve similar or better performance (AP50=53.3%) than what obtained by the baseline method with 100% of labeled data (2,628 annotated litter items, AP50=51.1%). Similar to the plateau discussed in Fig. 6.3, the drop in performance when moving from $\text{Train}_{80\%}$ to $\text{Train}_{100\%}$ can be linked to the limited overall size of our labeled dataset, the randomness of single runs, and lack of hyper-parameterization. For example, Bolton et al. (2023) reported a similar phenomenon caused by the lack of hyper-parameterization. They trained DL models to identify aircraft engine types with a learning rate of 0.01, but the performance drops as the size of training data. However, when setting the learning rate to 0.001, they found the performance improvement with the increase of training dataset size.

The better performance of the SSL methods are further detailed in Table 5.4 for the three models fine-tuned on $\text{Train}_{100\%}$. The Baseline-F4 performs the worst in all metrics, with a substantial decrease in TP, followed by a detrimental increase in both FN and FP. Interestingly, the SwAV-Scratch-F4 method retains the highest F1-score (0.59), due to a substantially lower number of FP. The lower precision of Baseline-F4 suggests that the high-level features learned from ImageNet are not sufficiently relevant to the specific

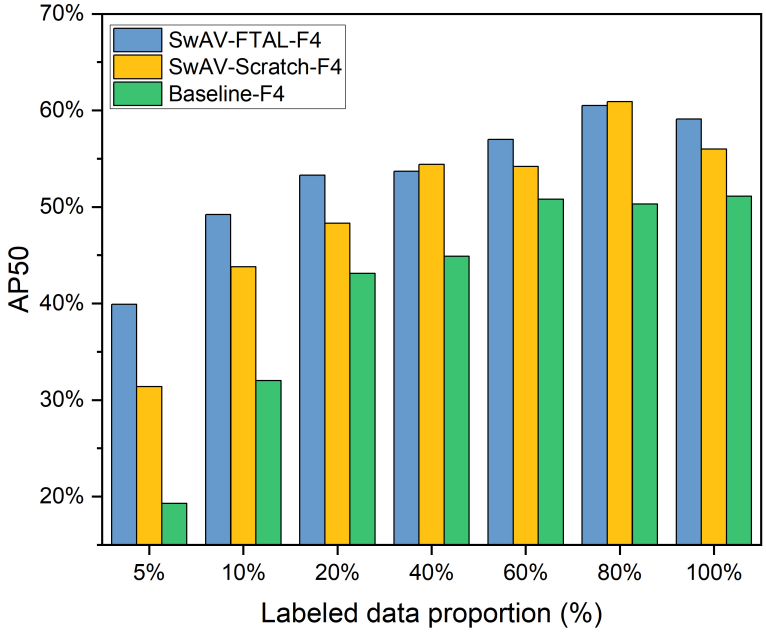


Figure 5.5: AP50 detection performance of the SwAV-FTAL-F4, SwAV-Scratch-F4 and Baseline-F4 methods on the Test subset with different proportion of labeled data for fine-tuning.

nuances of the litter detection task. Visual inspection of the predicted bounding boxes highlights that Baseline-F4 wrongly identifies waves, organic material, and the reflection of structures on banks and bridge as litter, as shown for example in Fig. 5.6. SwAV pre-training helps the models distinguish between the features of litter and non-litter items, as well as background characteristics. ImageNet initialization may partially hinder this process if insufficient data is available for fine-tuning, as hinted by the lower precision of SwAV-FTAL-F4 with respect to SwAV-FTAL-F2 and SwAV-Scratch-F4. Nonetheless, initializing SwAV with ImageNet weights seems useful when labeled data is particularly scarce (e.g., $\text{Train}_{5\%}$ to $\text{Train}_{20\%}$ subsets).

5.4.2. OUT-OF-DOMAIN GENERALIZATION CAPABILITY

The results illustrated in Chapter 5.4.1 suggest that when sufficient fine-tuning data is available, the SSL approach does not offer significant in-domain generalization advantages with respect to simple transfer of ImageNet pre-trained models. This can change by overcoming the discussed constraints on the small datasets used for SwAV pre-training and the limited computational resources. Despite these limitations, the scenario shifts favorably towards SSL when considering out-of-domain generalization, as done for zero-shot floating litter detection to the unseen locations in Amsterdam, Groningnen and

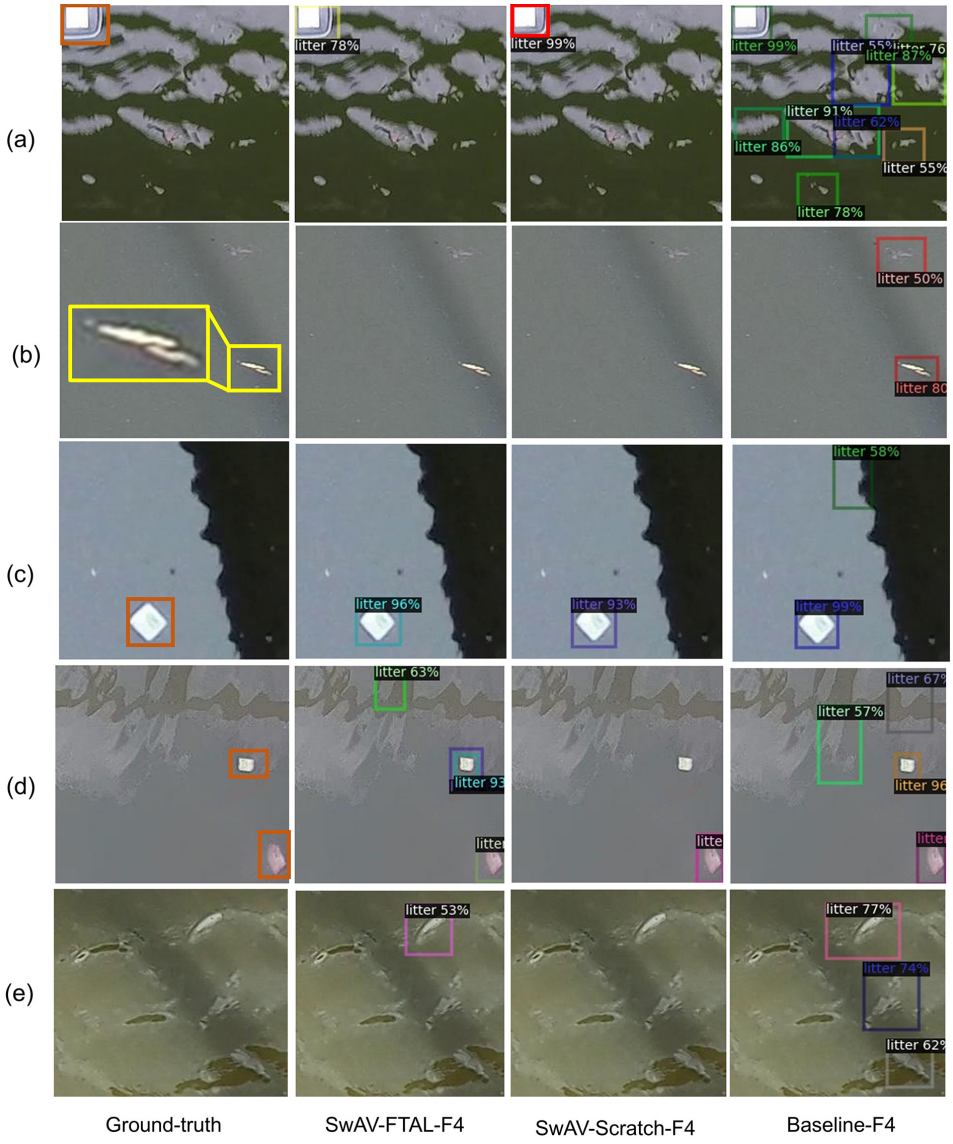


Figure 5.6: Example of predicted bounding boxes for the Faster R-CNN on the Delft-Jakarta Test subset and images without litter using (1) SwAV-FTAL-F4, (2) SwAV-Scratch-F4, and (3) Baseline-F4 methods. The models were fine-tuned on the $\text{Train}_{100\%}$ subset. Common misdetections of Baseline-F4 include the identification of waves ((a) and (e)), organic materials (b), and reflection of structures on banks (c) and bridge (d) as litter. Ground-truth litter is shown in red bounding boxes in the top row.

WUR-HCMC. As shown in Fig. 5.7 for all models fine-tuned on $\text{Train}_{100\%}$, SwAV pre-trained methods consistently match or surpass baseline performances. For example, in the Amsterdam dataset, both SwAV-FTAL-F4 and Baseline-F2 achieved a AP50 of around

45%. In Groningen, SwAV-FTAL-F4 outperforms the best baseline model by 12.7%, reaching an AP of 49.5%. In WUR-HCMC, SwAV-FTAL-F2 exceeds the baseline by over 7.5% with a 20.6% AP50. Further analysis on the confusion matrices and related metrics in Table 9.1-9.3 in Appendix 9 reinforces SwAV's advantage in out-of-domain scenarios. Except for Baseline-F2 in Groningen, which exhibits high precision and fewer FPs due to subpar sensitivity, the SSL models lead in all other metrics for all case studies.

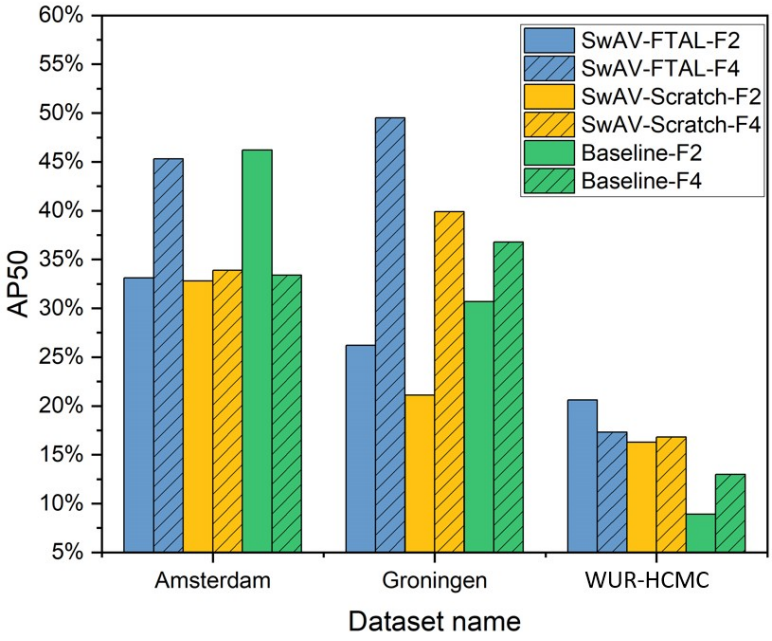


Figure 5.7: Zero-shot generalization capability of the models fine-tuned on $\text{Train}_{100\%}$ for the three unseen locations: Amsterdam, Groningen, and WUR-HCMC.

The better performances of SSL are reflected also in the visual inspection of the detections, done for SwAV-FTAL-F4 and Baseline-F4 on some example images of the three unseen case studies in Fig. 5.8. The baseline method displays fewer correct detection and increased misdetections, especially with respect to organic material, waves and other disturbances or reflective elements on the water surface. These findings collectively suggest that SwAV pre-training notably aids in adapting to new environments, particularly when retaining high-level features. The F4 SSL models are the best overall performers, despite we did not employ the best models emerging from the Delft-Jakarta Test dataset for the evaluation of out-of-domain generalization (i.e., those fine-tuned on $\text{Train}_{80\%}$). Expectedly, performance dips in more challenging conditions, such as in Ho Chi Minh City. Here, factors like lower resolution at the ground due to higher sensor elevation and the introduction of drone imagery, which were not part of the training dataset, further differentiate this dataset from the Delft-Jakarta dataset used for model development.

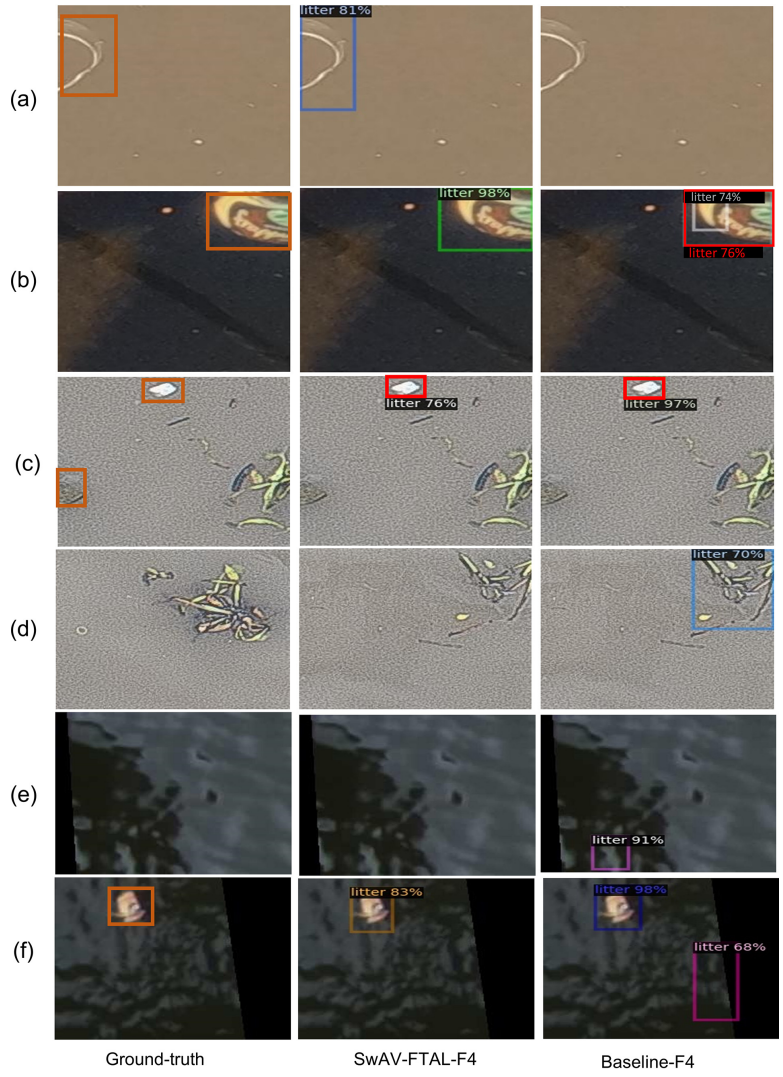


Figure 5.8: Detection results of the Faster R-CNN with ResNet50 backbone on Amsterdam, Groningen, and WUR-HCMC subsets using SwAV-FTAL-F4 and Baseline-F4 methods. The models were fine-tuned on the $\text{Train}_{100\%}$ subset. Both methods can detect litter items in (b), (c) and (f), and only the SwAV-FTAL-F4 method can detect the litter item in (a). Common misdetection of the Baseline-F4 method includes identifying organic materials (d) and wave ((e) and (f)) as litter. Ground-truth litter is shown in red bounding boxes in the top row.

5.5. CONCLUSIONS

The previous studies show that the transfer learning and data-centric artificial intelligence approaches are effective to enhance generalization capability of the supervised learning (SL) model trained on a relatively large amount of labeled data (4000 images). However, model's generalization capability is still limited. Moreover, they require ex-

tensive labeled data, a time-consuming and expensive process. Therefore, we need explore alternative approaches to improve model generalization performance with a limited amount of labeled data.

To overcome this challenge, we proposed a semi-supervised learning (SSL) approach based on SwAV, a self-supervised method that pre-trains deep learning models by discerning data patterns without requiring annotated images. To demonstrate the suitability of this new approach, we carried out experiments on camera images from the Delft (the Netherlands) and Jakarta (Indonesia) using a Faster R-CNN with a ResNet50 backbone. We compared the performance of standard transfer learning from ImageNet against the use of SwAV pre-training on around 100k unlabeled images. All models were fine-tuned using a maximum of around 1.8k images from the same locations. Our results show that the SSL approach performs at par or better than the supervised learning benchmark in average precision and F1-score, when tested on unseen images gathered from the same locations of the training dataset. The improvements are more noticeable when less data (up to ≈200 images with around 300 annotated litter items) is available for fine-tuning and with respect to the prediction of false positives. More importantly, testing for zero-shot generalization capability on unseen locations in Ho Chi Minh City (Vietnam), Amsterdam and Groningen (Netherlands) shows the clear superiority of SSL. This is mainly due to the extraction of better high-level representations via SwAV pre-training on relevant unlabeled images. Better performances are reported when initializing the SSL models with ImageNet weights.

6

A Semi-supervised Learning-Based Framework For Quantifying Cross-sectional Floating Litter Fluxes in Rivers

Based on the previous analysis of the effectiveness of semi-supervised learning (SSL), and data-centric artificial intelligence (AI) approaches, we propose a SSL-based framework for quantifying cross-sectional floating litter fluxes in river systems, with the limited availability of labeled data. When developing models, we used a data-centric AI method (i.e., flipping data augmentation) to enhance model performance. This framework includes four steps: (a) collecting camera images of river surfaces from multiple locations along the target river cross-section, (b) developing a robust litter detection model using SSL methods, (c) applying the developed model to detect litter items in images, and (d) post-processing the detection results to quantify cross-sectional floating litter fluxes. In step (c), we introduced a Slicing Aided Hyper Inference (SAHI) method to enhance accuracy of small litter detection. We optimized SSL models developed in our previous study by increasing pre-training epochs and pre-training dataset sizes, using images from waterways of the Netherlands, Indonesia and Vietnam, that were used for model pre-training and fine-tuning. Additionally, we assessed the zero-shot out-of-domain detection performance of SSL models and litter flux quantification performance of the proposed framework on a Vietnam case study, that was not used for model pre-

This chapter is based on:

Jia, T., Taormina, R., de Vries, R., Kapelan, Z., van Emmerik, T. H. M., Vriend, P. & Okkerman, I. (2025). A Semi-supervised Learning-Based Framework For Quantifying Litter Fluxes in River Systems (Submitted)

training or fine-tuning. We benchmarked our results against the SL methods and human visual counting methods. The results show that SSL models benefit from longer pre-training time and larger pre-training dataset size. The SSL models outperforms baseline SL models in zero-shot out-of-domain generalization in the case study, consistent with our previous findings. Furthermore, the SAHI method correctly identifies 45 additional small litter items (with areas below 1,000 cm^2), compared to the results obtained without the SAHI method, leading to improvement in the F1-score of up to 0.19. The flux measurement results indicate that the SSL-based framework substantially underestimates fluxes by a factor of 3-4 compared to human measurements, since the SSL model usually fails to detect transparent litter items and items entrapped in water hyacinths. However, the SSL-based framework quantifies litter fluxes nearly twice as high as the baseline SL-based framework, offering estimates that align more closely with human-measured fluxes.

6.1. INTRODUCTION

Literature review in Chapter 2 highlights a key knowledge gap in deep learning (DL)-based detection and quantification of litter in rivers: the lack of DL-based quantification of cross-sectional floating litter fluxes in rivers. Our findings in Chapter 5 and Chapter 6 indicate the effectiveness of data-centric artificial intelligence (AI) approaches and semi-supervised learning (SSL) in improving model generalization capability. Especially SSL methods reduces the reliance on large amounts of labeled data. Therefore, to fill the above gap, we proposed a SSL-based framework for measuring cross-sectional floating litter fluxes in river systems, with the limited availability of labeled data for model development. When developing models, we used an effective data-centric AI method (i.e., flipping data augmentation) to enhance model performances, as demonstrated in Chapter 4. Additionally, we further optimized SSL models developed in Chapter 5 to obtain better performances by increasing pre-training epochs and pre-training dataset size. This optimization was not done in Chapter 5. Literature reports the significant impact of these two factors on SSL performance on ImageNet classification tasks (Caron et al., 2020; Goyal et al., 2022).

Litter detection results in Chapter 4 and Chapter 5 show that detecting small litter or litter located far away from the imaging devices still remains a significant challenge. These litter items are represented by a limited number of pixels in images, resulting in insufficient details, that hinders their accurate detection with common object detection models (e.g., Faster R-CNN and YOLO). Specifically, the input images are usually resized to a smaller size (e.g., 640×640 pixel for YOLO network) by DL models before model training and inference, which causes small items to appear even smaller, further complicating detection (van Emmerik et al., 2024). Thus, in this chapter, we introduced a Slicing Aided Hyper Inference (SAHI) method to enhance the detection of small litter, by slicing input images into small tiles and resizing them to a larger dimension.

We developed and validated the SSL-based framework and SAHI method using images collected from canals and waterways in the Netherlands, Indonesia, and Vietnam. The findings presented in this chapter contribute to answering the third research sub-question of this thesis:

- *How to develop DL-based methods to quantify cross-sectional floating litter fluxes in rivers?*

6.2. METHODOLOGY

6.2.1. OVERVIEW OF THE SEMI-SUPERVISED LEARNING-BASED FRAMEWORK FOR QUANTIFYING LITTER FLUXES

Fig. 6.1 shows the proposed SSL-based framework for quantifying cross-sectional floating litter fluxes in flowing rivers. This framework includes four steps: (a) collecting data from locations of target rivers with digital cameras; (b) developing a DL model for litter detection using SSL methods; (c) applying the DL model to detect and count litter items in each collected image; and (d) post-processing the detection results to quantify litter fluxes. In step (b), we can develop models using existing openly available plastic datasets (e.g., see Chapter 3), parts of data from target rivers, or a combination of both.

We described the details on data collection in Chapter 6.2.2. The methodology for model development is presented in Chapter 6.2.3. Chapter 6.2.4 gives details on the litter flux estimation.

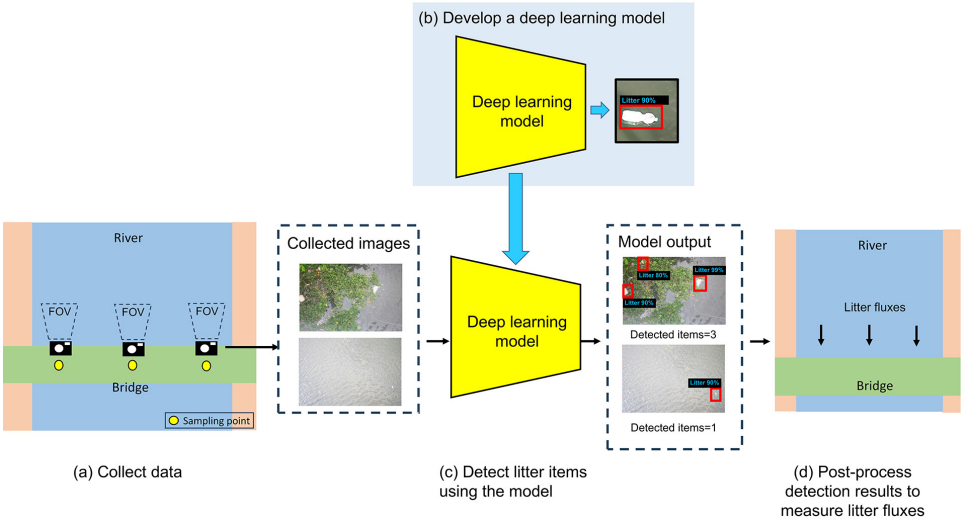


Figure 6.1: The schematic illustration of deep learning-based framework for quantifying cross-sectional floating litter fluxes. First, we used digital cameras to collect images at multiple sampling points on a bridge over the river surface (a). These images capture all floating litter items in camera's field of view (FOV). Second, we developed a deep learning model for litter detection using a semi-supervised learning method (b). Third, we used the developed model to detect litter from the collected images, providing the number of items detected in each image (c). Lastly, we post-processed the detection results to measure cross-sectional floating litter fluxes (d).

6.2.2. DATA COLLECTION FROM TARGET RIVERS

In this framework, we used digital cameras to capture images at multiple sampling points on an infrastructure (e.g., a bridge) of target rivers over the river surface (see Fig. 6.1 (a)), due to their affordable costs and user-friendliness compared to other devices (e.g., drones), as highlighted in Chapter 2. Thus, we selected such devices to enhance the practical applicability of the proposed framework. The cameras can either be: (1) fixed on the bridge at each sampling point for continuous monitoring, or (2) handheld for a pre-defined period to survey multiple sampling points, both with a time-lapse recording. Among these, fixed cameras are more suitable for long-term structured monitoring, as they can be deployed to automatically capture images at pre-defined time periods and frequencies over extended periods of time, while requiring little equipment maintenance (e.g., camera power supply). The time-lapse interval (seconds per frame) is determined based on the actual river plastic flow rate, ensuring that all floating litter items within the observation area are captured in images. The width of the observation area, that is smaller than the full river width, depends on the camera's field of view (FOV) and the height of the bridge above the water surface. Each sampling point can be measured

multiple times during a pre-defined period $\Delta t_{i,m}$ [h] on one measurement day (van Emmerik et al., 2022a). For example, van Emmerik et al. (2024) mounted a single camera at 5 sampling points on three bridges along the Saigon River. They captured 31 images at 10-second intervals, up to 8 times for each sampling point.

6.2.3. METHODOLOGY FOR LITTER DETECTION MODEL DEVELOPMENT

Given that the effectiveness of flipping data augmentation method and SSL methods shown in Chapter 4 and Chapter 5, we adopted the same SSL approach from Chapter 5 to develop a robust model for litter detection. Additionally, we applied flipping data augmentation method to enhance model performance by increasing the number of labeled images for model fine-tuning.

We applied a SAHI method (Akyon et al., 2022) to enhance the model's generalization to small litter in target rivers, as explained below. In this study, we did not develop DL models capable of automatically identifying and counting the same litter item appearing in multiple consecutive images as a single instance. Thus, we manually reviewed the detected litter items and corrected the counts before estimating floating litter fluxes.

Fig. 6.2 shows the schematic illustration of the SAHI method for detecting floating litter. First, the SAHI method slices the original input image into smaller overlapping tiles with a width of W_s and height of H_s (e.g., 400×400 pixels) with an overlap ratio. For simplicity, Fig. 6.2 (a) shows slicing process with the overlap ratio of 0. Then, each sliced tile is resized into a larger dimension with a weight of W_r and height of H_r. Each resized tile is fed into the Faster R-CNN. Finally, the predictions in tiles (i.e., the yellow bounding boxes in Fig. 6.2 (c)) are mapped back to the original input image dimensions. The SAHI method employs Non-Maximum Suppression (NMS) to refine duplicate predictions for the same object in overlapping regions of adjacent tiles (Hosang et al., 2017). The NMS measures the overlap between the predicted bounding boxes in overlapping regions using Intersection over Union (IoU), and filters out redundant boxes with higher IoU overlap than a predefined IoU NMS threshold, retaining the boxes with confidence score higher than a certain confidence threshold (Akyon et al., 2022).

6.2.4. LITTER FLUX ESTIMATION

We post-processed the detection results from the DL model to quantify cross-sectional floating litter fluxes. First, we calculated the mean litter fluxes f_i [items/h] for sampling point i, using the following equation (Schreyers et al., 2023):

$$f_i = \frac{1}{M_i} \sum_{m=1}^{M_i} \frac{N_{i,m}}{\Delta t_{i,m}} \tag{6.1}$$

where $N_{i,m}$ [items] is the total number of litter items detected by the model in the images collected at sampling point i during the m-th measurement within the time period $\Delta t_{i,m}$ [h]. M_i denotes the total number of sampling events at sampling point i.

Then, we calculated the total cross-sectional floating litter fluxes F [items/h] using the following equation, as derived from van Emmerik et al. (2022a):

$$F = \frac{1}{S} \sum_{i=1}^{S} \frac{f_i}{w_i} \cdot W \tag{6.2}$$

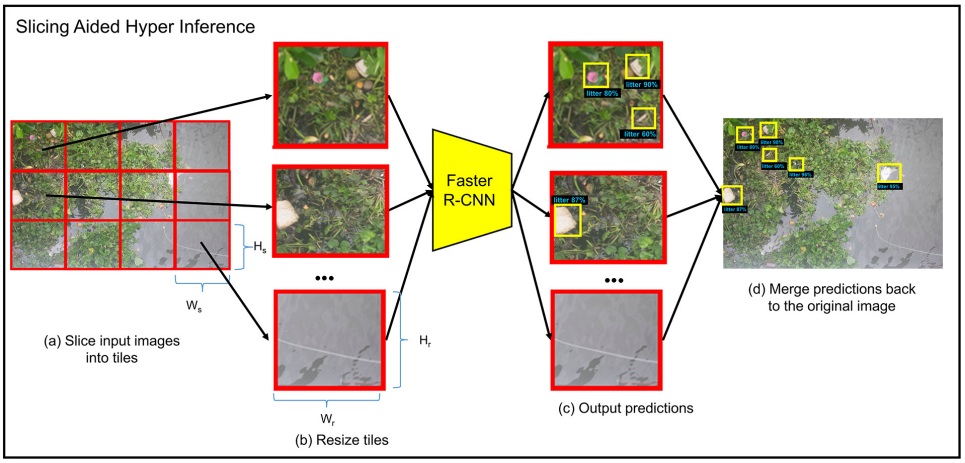


Figure 6.2: The schematic illustration of Slicing Aided Hyper Inference (SAHI) for detecting floating litter. First, the SAHI method divides the input images into smaller (overlapping) tiles (a), and resizes them into a larger scale (b). Then, we used the Faster R-CNN to detect litter in each resized tile (c). Finally, these detections (yellow bounding boxes) are merged back to the original input image (d).

where S is the total number of sampling points in a bridge. w_i [m] is the width of the observation area at sampling point i. W [m] is the total river width.

6.3. EXPERIMENTS

We conducted three experiments in this chapter. We trained and validated SSL models using data from six locations: (1) The TU Delft - Green Village (TUD-GV), the Netherlands, (2) Oostpoort, the Netherlands, (3) Amsterdam, the Netherlands, (4) Groningen, the Netherlands, (5) Jakarta, Indonesia, and (6) Wageningen UR - Ho Chi Minh City (WUR-HCMC). We evaluated litter detection and quantification performances in a case study, TU Delft - Ho Chi Minh City (TUD-HCMC). The details of these datasets are shown in Chapter 3.

In Experiment 1, we optimized the SSL models developed in Chapter 5 with longer pre-training time, and larger pre-training dataset size. We evaluated models' improvement in in-domain litter detection performances (see Chapter 6.3.2). In Experiment 2, we evaluated SSL models' zero-shot out-of-domain litter detection performances with and without SAHI method (see Chapter 6.3.3). Moreover, we compared the litter detection results against those obtained from a SL benchmark in these two experiments. In Experiment 3, we evaluated the capability of the proposed SSL-based framework for floating litter flux quantification (see Chapter 6.3.4). We compared the results of the SSL-based framework with those of a SL-based framework and a conventional human visual counting method.

In Experiment 1 and 2, we evaluated models' in-domain and zero-shot out-of-domain litter detection performance, respectively. In-domain generalization performance indicates the model performance on new, unseen data collected from the same geographic locations as the training data. In contrast, out-of-domain generalization performance

indicates the performance on unseen data sourced from different geographic locations. Zero-shot out-of-domain generalization refers to the capability of DL models to detect previously unseen objects from different geographic locations, without requiring training data of these unseen objects. This capability is especially crucial for large-scale structured monitoring, enabling the monitoring of multiple geographic locations with varying environmental conditions in extensive river system, without well-labeled and location-specific data for further refinement of DL models.

To evaluate model performance for litter detection, we used the same metrics as in Chapter 5: (1) Average Precision (AP50), (2) F1-score, (3) precision, and (4) recall.

6.3.1. DATA SELECTION

For developing models, we randomly selected images from the TUD-GV, Oostpoort, Amsterdam, Groningen, Jakarta and WUR-HCMC dataset, as detailed in the "Total images" column of Table 6.1. We aimed to evaluate models' out-of-domain generalization performance to a new TUD-HCMC case study in Experiment 2 and 3. Thus, we only selected 935 images collected at the Quy Kien and Thanh Ho location from the WUR-HCMC dataset for model development. This selection ensures that images from the Binh Loi and Thu Thiem location in the TUD-HCMC dataset remain unseen during model pre-training and fine-tuning. We sliced the selected images into tiles and achieved a total of 501,983 image tiles with a standard size of 224×224 pixels, matching the input size required for ResNet50.

Table 6.1: Details on the images for model development

Image source	Total images	Total image tiles	No. image tiles without labels	No. image tiles with litter annotated	No. annotated litter items
TUD-GV	3,777	91,565	90,112	1,453	1,719
Oostpoort	562	78,043	77,710	333	342
Amsterdam	92	36,864	36,712	152	204
Groningen	63	5,350	5,193	157	167
Jakarta	526	16,789	16,433	356	501
WUR-HCMC	935	273,372	273,317	55	63
Total	5,955	501,983	499,477	2,506	2,996

In Experiment 1, we trained and validated models, and evaluated their in-domain generalization capability using these 501,983 image tiles. We randomly sampled tiles to create the non-overlapping subsets, including (1) 499,477 tiles (99.5%) for SwAV pre-training, (2) 1,128 tiles for supervised fine-tuning, (3) 125 tiles for model validation, and (4) 1,253 tiles for model testing, as outlined in Table 6.2. We used a maximum of 499,477 tiles without annotations for SwAV pre-training (Train_{500k}). To further investigate model performance regarding to the availability of unlabeled data for SwAV pre-training, we generated five additional smaller pre-training subsets by gradually reducing the number of tiles down to 25k (Train_{300k} to Train_{25k}). We used up to 1,128 image tiles for fine-tuning SSL and baseline SL models in a supervised manner ($\text{Train}_{100\%}$). These tiles contain 1,349 litter items annotated by bounding boxes indicating their locations, without

further classification. To evaluate model performance with respect to the availability of labeled data, we generated two smaller fine-tuning subsets by decreasing the number of labeled tiles down to 20% ($Train_{60\%}$ and $Train_{20\%}$). We used a maximum of 125 tiles containing 158 annotations to validate models ($Validation_{100\%}$), with a 9:1 ratio relative to the tiles for fine-tuning ($Train_{100\%}$). For consistency, we generated two smaller validation subsets ($Validation_{60\%}$ and $Validation_{20\%}$). We generated the Test subset with 1,253 tiles and 1,489 annotations for testing models' in-domain generalization performance.

Table 6.2: The subsets for model development in Experiment 1

Learning method	Training dataset			Validation dataset			Test dataset		
	Name	No. annotated litter items	No. tiles	Name	No. annotated litter items	No. tiles	Name	No. annotated litter items	No. tiles
Self-supervised	$Train_{500k}$	0	499,477						
	$Train_{300k}$	0	299,679						
	$Train_{200k}$	0	203,454						
	$Train_{100k}$	0	99,887						
	$Train_{50k}$	0	49,941						
	$Train_{25k}$	0	24,966						
Semi-supervised and supervised	$Train_{100\%}$	1,349	1,128	$Validation_{100\%}$	158	125			
	$Train_{60\%}$	800	677	$Validation_{60\%}$	91	75	Test	1,489	1,253
	$Train_{20\%}$	276	226	$Validation_{20\%}$	33	25			

6

6.3.2. EXPERIMENT 1: IN-DOMAIN DETECTION PERFORMANCE

With the first experiment, we assessed the benefits of (1) varying pre-training epochs, and (2) varying pre-training dataset sizes on the in-domain generalization performance of SSL models. This examination is essential for assessing the effectiveness of representations learned from different scales of pre-training dataset for generalization. It also offers insights into the effectiveness of SSL methods in scenarios with limited labeled samples, but with abundant unlabeled images and sufficient computational resources for extensive hyperparameter tuning.

For developing SSL models, we first initialized the ResNet50 backbone with ImageNet weights, and then using SwAV to pre-train all the layers of the ResNet50 network with a projection head of 2-layer multilayer perceptron on all six pre-training subsets, (i.e., $Train_{500k}$ to $Train_{25k}$ subset) in the self-supervised learning stage. Due to the limited computational resources, we performed SwAV pre-training for 100, 200, and 300 epochs (Caron et al., 2020; Chen et al., 2020). In the supervised fine-tuning phase, we fine-tuned the Faster R-CNN architecture derived from the SSL backbone on $Train_{100\%}$ to $Train_{20\%}$ subset. Before fine-tuning, we performed horizontal flipping data augmentation technique to the fine-tuning subset, generating one new image for each original image (see Chapter 4). During fine-tuning, we froze the first four convolutional blocks of the ResNet50 backbone network. It allows the Faster R-CNN to retain relevant low-level features (e.g., edges and texture) in the first two blocks, as well as high-level features (e.g., object shapes) in the last tow blocks, learned from SwAV pre-training. Most important, these high-level features significantly improve the model's in-domain and out-of-domain generalization performance in data scarce conditions, as highlighted in Chapter 5. Model validation was conducted on the respective Validation subsets, i.e.,

$\text{Validation}_{100\%}$ to $\text{Validation}_{20\%}$ subset. We selected the SSL model that achieved the highest validation accuracy across the three different pre-training epoch settings. Then, we evaluated its in-domain performance on the Test subset.

We compared the effectiveness of SSL models with baseline SL models, developed with the supervised fine-tuning phase (see Fig. 5.1 (d) in Chapter 5), but without the SwAV pre-training phase (see Fig. 5.1 (c) in Chapter 5). These SL models are Faster R-CNNs supervised fine-tuned on images with annotated litter, with ResNet50 backbones initialized using ImageNet weights. During fine-tuning, the first four convolutional blocks of the ResNet50 backbone network were frozen. They were fine-tuned, validated, and tested on the same subsets used for SSL model development.

6.3.3. EXPERIMENT 2: ZERO-SHOT OUT-OF-DOMAIN DETECTION PERFORMANCE

To evaluate the zero-shot out-of-domain generalization performance for litter detection, we tested the best-performing SSL and SL model developed in Experiment 1, on the $\text{Test}_{\text{Thu Thiem}}$ and $\text{Test}_{\text{Binh Loi}}$ subsets, as outlined in Table 3.3 in Chapter 3. We did not re-train these models on any data from the Thu Thiem and Binh Loi location.

EVALUATION OF THE SAHI METHODS

We compared performance of the SSL model using the SAHI method and that without SAHI during model inference on $\text{Test}_{\text{Thu Thiem}}$ and $\text{Test}_{\text{Binh Loi}}$ subsets. Inspired by Akyon et al. (2022) and Gia et al. (2024), we tested four configurations of width W_s and height H_s for the selected SSL model: (1) 400×400, (2) 640×640, (3) 1280×1280, and (4) 1920×1920 pixels. The configuration yielding the best detection performance for each subset was selected for subsequent steps of this experiment.

When applied to detect litter in the TUD-HCMC case study, models may produce a high number of misdetections, due to the limited data available for SwAV pre-training and supervised fine-tuning, as shown in Chapter 5. To reduce these misdetections, we refined the output bounding boxes by setting a high confidence threshold value before making the final predictions and computing performance metrics. This threshold defines the minimum confidence level required for a detected object to be considered as a valid detection. Increasing this threshold excludes low-confidence predictions, but may also result in missing some true positives with confidence scores below the threshold. Thus, we compared the SSL model's performance using three confidence threshold values (0.5, 0.7 and 0.9) with the best W_s and H_s settings. The confidence threshold value yielding the best performance for each subset was chosen for following steps of this experiment.

It is noted that selecting optimal hyperparameters based on test performance is not a standard practice in machine learning. However, the aim of this experiment was to evaluate the benefit of the SAHI method, while utilizing as much data from TUD-HCMC case study for testing as possible.

EVALUATION OF THE SSL AND SL METHODS

To minimize the influence of randomization, we repeated the fine-tuning process for a total of 10 times for both SSL and SL models. Then, we evaluated the detection performance of all models, using the SAHI method with W_s, H_s and confidence threshold

settings that yielded the best performance in the previous evaluation, ensuring that the pre-processed input images by SAHI were the same before being fed to both SSL and SL models.

6.3.4. EXPERIMENT 3: LITTER FLUX MEASUREMENT

To evaluate the zero-shot out-of-domain flux quantification capability of the proposed SSL-based framework, we used the best-performing SSL models for the Thu Thiem and Binh Loi locations from the 10 runs conducted in Experiment 2. We estimated floating litter fluxes, using the approach introduced in Chapter 6.2.4. Additionally, we evaluated flux quantification capability of the SL-based framework similarly, but replacing the SSL model with the best-performing SL model from 10 runs, as illustrated in Fig. 6.1 (a) and (c). Furthermore, we compared these results against those obtained using the conventional human counting method, where litter items were manually observed and counted directly from the images. We used the Pearson correlation coefficient (r) (Cohen et al., 2009) to assess the linear correlation between fluxes measured by DL-based frameworks (i.e., the SSL- and SL-based framework) and human counting methods across 10 sampling points in the case study. This coefficient ranges from -1 to 1. A higher positive value indicates a stronger positive correlation between two variables. The reader is referred to the work of Cohen et al. (2009) for more details on this coefficient. Litter items appearing in multiple consecutive images were counted only once across all methods and frameworks.

6.3.5. TRAINING SETUP AND PROCEDURE

We implemented model training and evaluation using *Python 3.8.16* and *PyTorch 1.8.1*, with the *VISSL* (Goyal et al., 2021), *Detectron2* (Wu et al., 2019) and SAHI (Akyon et al., 2021) libraries. In the self-supervised learning stage, we performed four data augmentation strategies: (1) multi-crop with 6 views (2×[160×160]+4×[96×96]), (2) horizontal flipping, (3) color distortion, and (4) Gaussian blur. Other settings for SwAW pre-training and fine-tuning are same with those in Chapter 5.

For SAHI, we used the default *SAHI* hyperparameters, including an overlap ratio of 0.2 and an IoU NMS threshold value of 0.5 for refining predictions in overlapping region of adjacent tiles. During inference, we set W_r and H_r, following the default Faster R-CNN setting in *Detectron2* framework. The shortest edge of sliced tile was resized to a length between 800 and 1333 pixels, while the longest edge was scaled by preserving the aspect ratio, ensuring it did not exceed 1333 pixels.

6.4. RESULTS AND DISCUSSION

6.4.1. EXPERIMENT 1: IN-DOMAIN DETECTION PERFORMANCE

SSL MODEL PERFORMANCE FOR VARYING PRE-TRAINING EPOCHS

Table 6.3 presents the AP50 detection performance of the SSL methods on the Validation$_{100\%}$ subset, evaluated with varying pre-training epochs and pre-training dataset sizes. Results for the Validation$_{60\%}$ and Validation$_{20\%}$ subsets are shown in Table 10.1 in Appendix 10. We observed that increasing the pre-training epochs from 100 to 200 usually leads to an improvement in model performance, as indicated by AP50 improvements rang-

ing from 0.2% to 4.2%, while an additional 100 pre-training epochs requires substantial computational resources (e.g., 278 hours per 100 epochs on the Train_{500k} subset). This finding is similar to that reported by Caron et al. (2020). The authors pre-trained the ResNet50 using SwAV for 100, 200, 400, and 800 epochs on 1.28 million unlabeled images from the ImageNet dataset. Their results demonstrate a 3.2% improvement in top-1 accuracy on the ImageNet classification task as pre-training epochs increase from 100 to 800. Furthermore, we found that this improvement is more noticeable, when a large amount of data is available for pre-training. For example, the SSL models pre-trained on Train_{200k} and Train_{500k} achieve an AP50 improvement ranging from 3.4% to 4.2% by increasing epochs, while the SSL models pre-trained on Train_{50k} and Train_{100k} only obtain a AP50 improvement ranging from 0.2% to 0.4%. We attribute this superior performance to the more robust feature representations learned from SwAV pre-training from a larger amount of data for longer training time, which enhance the performance of Faster R-CNN for the downstream litter detection task.

Table 6.3: Pre-training time and validation accuracy (AP50) on the $\text{Validation}_{100\%}$ subset of all SSL models for Experiment 1. The bold entities are the best results for models pre-trained on each pre-training dataset

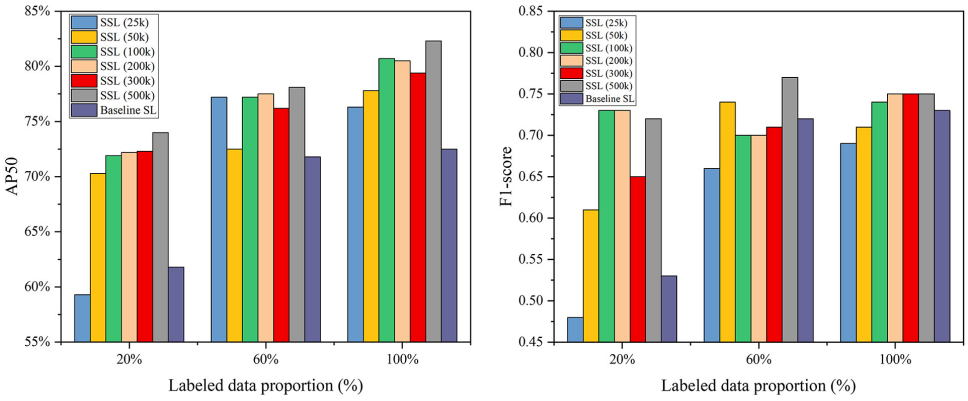
Pre-training dataset	No.pre-training epochs	Pre-training time (h/100 epochs)	AP50
Train_{25k}	100	17	**80.2%**
	200		78.8%
	300		77.1%
Train_{50k}	100	33	80.2%
	200		**80.4%**
	300		79.4%
Train_{100k}	100	56	81.8%
	200		**82.2%**
	300		81.9%
Train_{200k}	100	117	78.0%
	200		**82.2%**
	300		80.7%
Train_{300k}	100	168	82.4%
	200		**82.9%**
	300		81.0%
Train_{500k}	100	278	80.2%
	200		**83.6%**
	300		80.5%

Table 6.3 also demonstrates a decline in AP50 ranging from 0.3% to 3.1%, when epochs

increase from 200 to 300. It could be attributed to the limited size of pre-training dataset (500k images). Caron et al. (2020) reported improved performance with longer pre-training time, but used a significantly larger dataset (1.28 million). Another reason is the single pre-training run conducted, due to computational limitations. The inherent stochasticity of neural network training leads to variations in results across multiple runs, potentially affecting the observed performance (Punjani & Fleet, 2021).

PERFORMANCE FOR VARYING PRE-TRAINING DATASET SIZES

The benefit of larger pre-training dataset on model performance is more noticeable from the results shown in Fig. 6.3, that shows in-domain generalization performance of the SSL and baseline SL methods on the Test subset, with varying proportion of labeled data for fine-tuning. It reveals a general upward trend in AP50 and F1-score for SSL models, as the pre-training dataset size increases, irrespective of the amount of labeled data available for fine-tuning. The performance improvement is particularly noticeable when scaling the pre-training dataset from a small size (<100k) to a larger size, with AP50 increasing by 5.6% to 14.7% and F1-score improving by 0.06 to 0.25. For instance, when models are fine-tuned on the $\text{Train}_{100\%}$ subset, the AP50 improves from 76.3% to 82.3%, and the F1-score increases from 0.69 to 0.75, as the pre-training dataset size increases from 25k to 500k. These findings underscore the advantages of large-scale datasets, enabling models to learn more effective low-level and high-level representations. This improvement is especially significant in scenarios with limited labeled data for fine-tuning (i.e., $\text{Train}_{20\%}$), where AP50 increases by 14.7% and F1-score improves by 0.25.



(a) AP50 detection performance

(b) F1-score detection performance

Figure 6.3: AP50 (a) and F1-score (b) detection performance of the SSL and baseline SL methods on the Test subset with different proportion of labeled data for fine-tuning. The six SSL models were pre-trained on Train_{25k}, Train_{50k}, Train_{100k}, Train_{200k}, Train_{300k}, and Train_{500k} subset, respectively.

We observed a performance plateau when increasing the pre-training dataset size from 100k to 500k. It could be attributed not only to the limited size of the pre-training dataset (500k) and the limited pre-training epochs, but also to the constraints of conducting only a single training run, imposed by computational resource limitations. Literature demonstrates notable performance improvements for SSL models with larger

SwAV pre-training datasets, scaling from 1.28 million to over 1 billion images (Goyal et al., 2022). Thus, we believe that better performance could be achieved by scaling the unlabeled dataset size to over 1 million and conducting a large number of training runs.

Fig. 6.3 also demonstrate that the most SSL models significantly outperform the baseline SL benchmarks, irrespective of the amount of data used for SwAV pre-training or fine-tuning. The SSL method performs best in most cases, obtaining AP50 values ranging from 59.3% to 82.3%, and F1-scores from 0.48 to 0.77. In comparison, the baseline SL method achieves AP50 values varying from 61.8% to 72.5%, and F1-scores from 0.53 to 0.73. These values are particularly low when fine-tuning data is limited (i.e., $\text{Train}_{20\%}$ subsets), where SSL models yield improvements of up to 12% in AP50 and 0.20 in F1-score. The SSL model requires only 20% of the labeled images (226 images with 276 annotated litter items) combined with 500k unlabeled images to achieve comparable or superior performance (AP50=74.0%, and F1-score=0.72) to that of the baseline SL method, which relies on 100% of labeled images (1,128 images with 1,349 annotated litter items, AP50=72.5%, and F1-score=0.73). These findings highlight the benefits of transferring low-level and high-level representations learned by SwAV from unlabeled yet domain-relevant data, leading to notable improvements compared to simple transfer from ImageNet. While the features extracted from ImageNet are general, they are not sufficiently relevant to the specific litter detection task.

6.4.2. EXPERIMENT 2: ZERO-SHOT OUT-OF-DOMAIN DETECTION PERFORMANCE

PERFORMANCE FOR SAHI METHODS

Table 6.4 and 6.5 present the performance of SSL models with or without SAHI methods on the $\text{Test}_{\text{Thu Thiem}}$ and $\text{Test}_{\text{Binh Loi}}$ subset, respectively. The SSL model achieving the highest AP50 on the Test subset in Experiment 1, was selected for these evaluations (i.e., model pre-trained on the Train_{500k} subset and fine-tuned on the $\text{Train}_{100\%}$ subset). The results demonstrate that SSL models using SAHI methods significantly outperform those without SAHI in all metrics for the Thu Thiem location, and in recall and F1-score for the Binh Loi location under the same confidence threshold settings (0.5). Especially, the SAHI method achieves an improvement in F1-score of up to 0.19, compared to models without SAHI across two locations.

Table 6.4 and 6.5 also present the performance of SSL models with SAHI, along with the best W_s and H_s settings under varying confidence thresholds. Increasing confidence threshold from 0.5 to 0.9 usually yields a slight decline in TP and a significant reduction in FP, since a large number of low-confidence FPs are filtered out. This adjustment leads to a slight decrease in recall, but a notable improvement in precision and F1-score. For example, the model with SAHI (W_s, H_s = 1280 pixel) achieves a substantial increase in precision of 0.13 and F1-score of 0.07, with a minor decrease in recall of 0.04 for the Thu Thiem location, when the confidence threshold is raised from 0.5 to 0.9.

For the Thu Thiem location, the model without SAHI fails to detect any litter items correctly (TP = 0) and produces 6 FPs, resulting in precision, recall, and F1-score values of 0. In contrast, the SAHI method under the same confidence threshold settings (0.5) correctly detects 9~27 litter items (TP) depending on the W_s and H_s settings, achieving higher precision (0.02~0.22), recall (0.14~0.42), and F1-score (0.05~0.24), while it gen-

Table 6.4: Confusion matrix, Precision, Recall and F1-score on the $\text{Test}_{\text{Thu Thiem}}$ subset for SSL models, evaluated with varying inference hyperparameters (i.e., W_s, H_s and confidence threshold score). The model was fine-tuned on the $\text{Train}_{100\%}$ subset. The bold entity is the best F1-score

$W_s \times H_s$ (pixel × pixel)	Confidence threshold	TP	FP	FN	Precision	Recall	F1-score
No SAHI	0.5	0	6	64	0.00	0.00	0.00
400×400	0.5	21	838	43	0.02	0.33	0.05
640×640	0.5	27	539	37	0.05	0.42	0.09
1280×1280	0.5	22	95	42	0.19	0.34	0.24
	0.7	21	74	43	0.22	0.33	0.26
	0.9	19	40	45	0.32	0.30	**0.31**
1920×1920	0.5	9	32	55	0.22	0.14	0.17

Table 6.5: Confusion matrix, Precision, Recall and F1-score on the $\text{Test}_{\text{Binh Loi}}$ subset for SSL models, evaluated with varying inference hyperparameters (i.e., W_s, H_s and confidence threshold score). The model was fine-tuned on the $\text{Train}_{100\%}$ subset. The bold entity is the best F1-score

$W_s \times H_s$ (pixel × pixel)	Confidence threshold	TP	FP	FN	Precision	Recall	F1-score
No SAHI	0.5	7	22	107	0.24	0.06	0.10
400×400	0.5	49	3438	65	0.01	0.43	0.03
640×640	0.5	68	2749	46	0.02	0.60	0.05
1280×1280	0.5	69	625	45	0.10	0.61	0.17
1920×1920	0.5	39	229	75	0.15	0.34	0.20
	0.7	36	153	78	0.19	0.32	0.24
	0.9	30	70	84	0.30	0.26	**0.28**

erates a significant number of false positives (FP = 32~838). For the Binh Loi location, the model without SAHI correctly detects only a few litter items (TP = 7) and generates few FPs (22), resulting in a precision of 0.24, but with very low recall (0.06) and F1-score (0.10). In contrast, the SAHI method detects a significantly higher number of litter items (TP = 39~69), but also generates a large number of FPs (229~3438). This leads to lower precision (0.01~0.15), but higher recall (0.34~0.61) and F1-score (0.03~0.20), compared to those obtained by the model without SAHI.

Fig. 6.4 shows the area of all litter items correctly detected by SSL models with or without SAHI method in the $\text{Test}_{\text{Thu Thiem}}$ (W_s, H_s = 1280 pixel) and $\text{Test}_{\text{Binh Loi}}$ (W_s, H_s = 1920 pixel) subset. The area of each litter item is approximately calculated by multiplying its ground-truth bounding box area (pixel^2) by the square of the GSD of images

($cm^2/pixel^2$). The results show that the model without the SAHI method only correctly detect 7 "big" litter items with area above 1,000 cm^2, while fails to detect all "small" litter items with area below 1,000 cm^2. In contrast, the model with the SAHI method not only correctly identifies these 7 "big" litter items, but also detects 9 additional "big" litter items, and 45 additional "small" litter items.

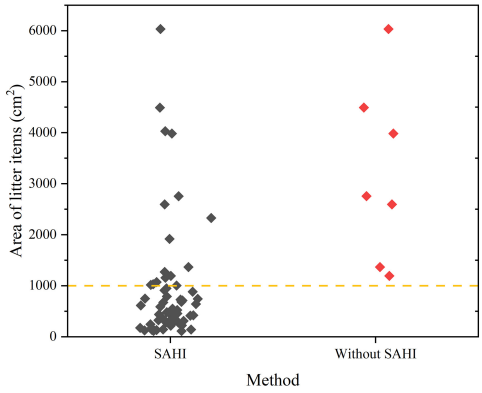


Figure 6.4: The areas of litter items correctly detected by SSL models with or without SAHI method in the $\text{Test}_{\text{Thu Thiem}}$ (W_s, H_s = 1280 pixel) and $\text{Test}_{\text{Binh Loi}}$ (W_s, H_s = 1920 pixel) subset. The confidence threshold is 0.5.

Visual inspection of the predicted bounding boxes highlights the effectiveness of the SAHI method in handling diverse object sizes, as shown for examples in Fig. 6.5 and 6.6. The accurate detection of "small" litter by SAHI methods can be primarily attributed to its slicing and resizing process, which enlarges these objects, thereby providing sufficient details for the model to recognize them effectively.

The aim of this work was not to precisely measure the actual size of litter items detected by models with or without the SAHI method, but to show clear evidence that the SAHI method can correctly detect a large number of litter items with area smaller than a specific threshold, that models without the SAHI method fail to identify. To the best of our knowledge, no study has reported the precise value of specific threshold, as it depends on the GSD, that is determined by sensor elevation and properties (Andriolo et al., 2023). For example, a CNN model without the SAHI method may correctly detect a plastic bottle in images captured by sensors at a low elevation, where the GSD is low and the bottle appears relatively large (i.e., represented by many pixels). However, this model may fail to detect the same bottle in images taken by the same sensors at a higher elevation, where the GSD is higher, making the bottle appears relatively small (i.e., represented by fewer pixels). This issue arises from the lack of scale invariance in CNNs, that refers to a model's ability to maintain consistent outputs regardless of object scale (Singh & Davis, 2018).

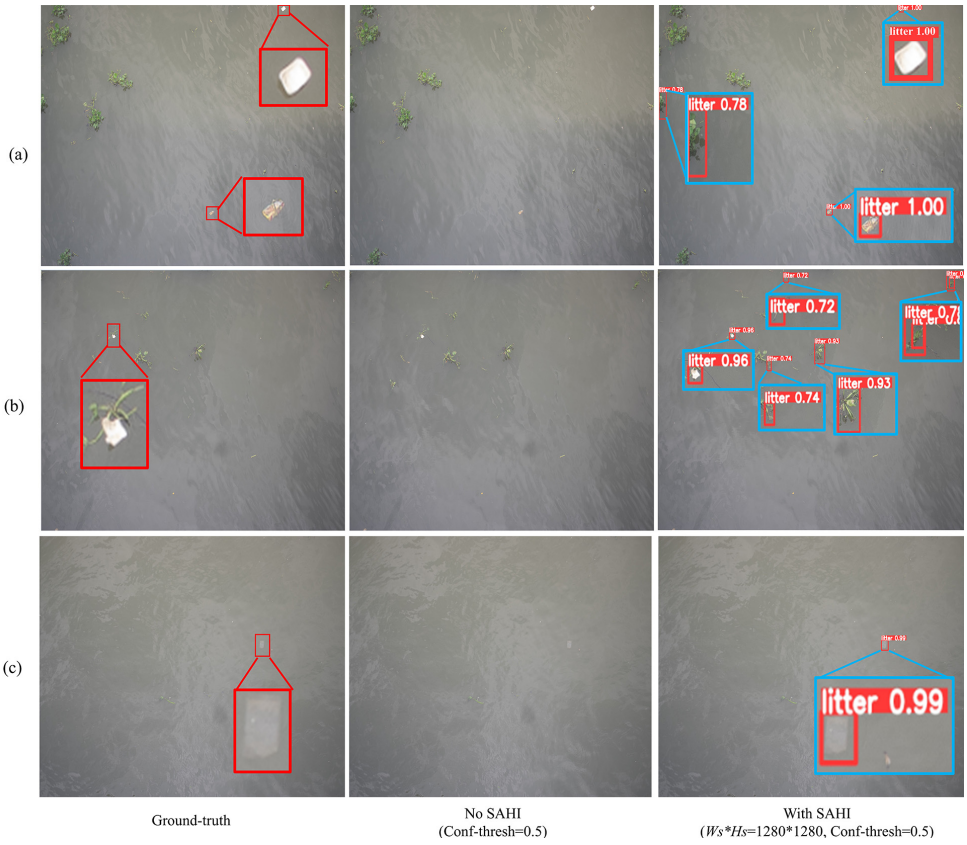


Figure 6.5: Examples of predicted bounding boxes for models with and without the SAHI method on the $\text{Test}_{\text{Thu Thiem}}$ subset. The Faster R-CNN model was fine-tuned on the $\text{Train}_{100\%}$ subset. During inference, we set W_S and H_S to 1280, with a confidence threshold score (Conf-thresh) of 0.5. Without the SAHI method, the model usually fails to detect all "small" litter items with area below 1,000 cm^2 in (a)-(c). With SAHI, the model correctly detects some small, including two in (a), one in (b), and one in (c).

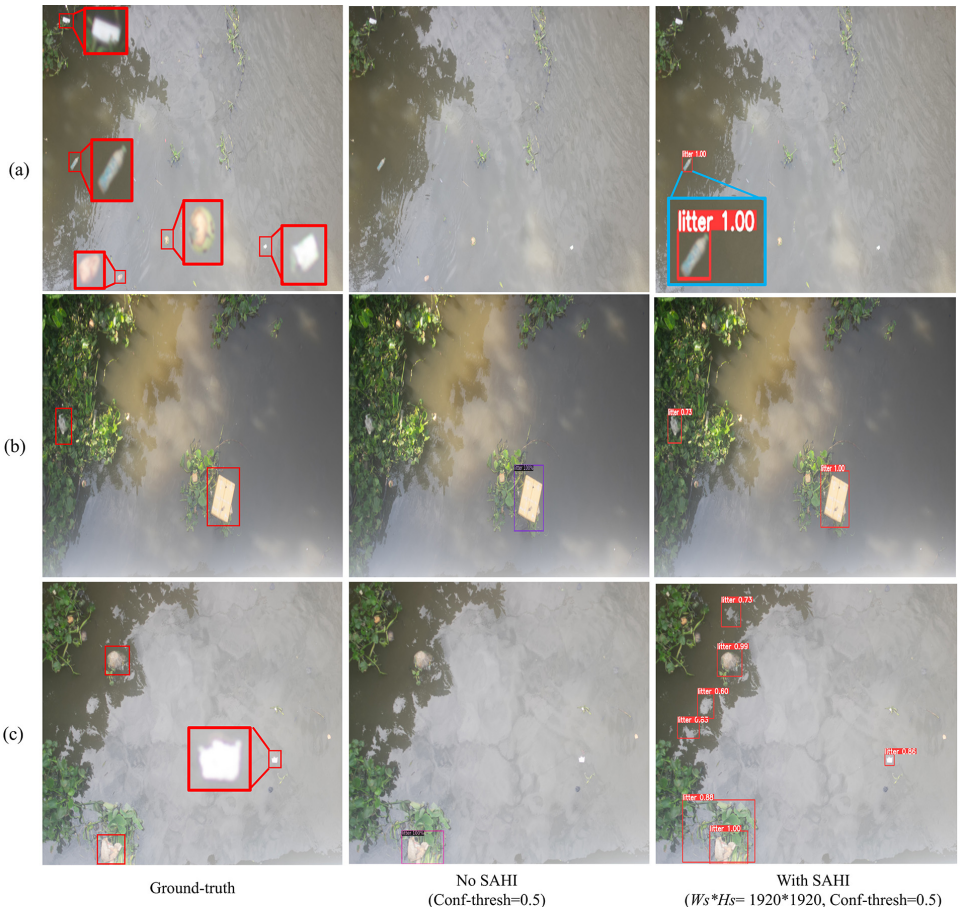
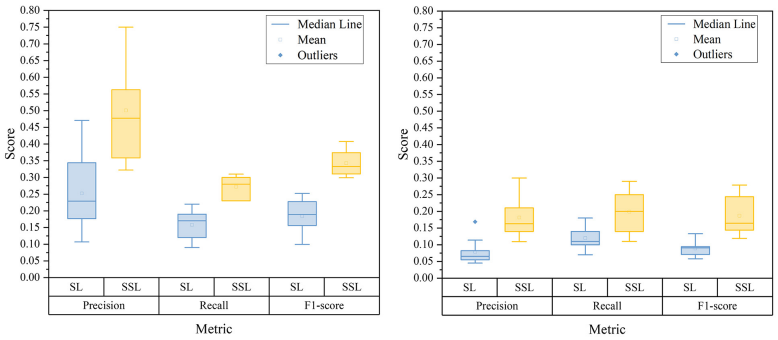


Figure 6.6: Examples of predicted bounding boxes for models with and without the SAHI method on the $\text{Test}_{\text{Binh Loi}}$ subset. The Faster R-CNN model was fine-tuned on the $\text{Train}_{100\%}$ subset. During inference, we set W_s and H_s to 1920, with a confidence threshold score (Conf-thresh) of 0.5. Without the SAHI method, the model correctly detects two "big" items with area above 1,000 cm^2 in (b) and (c), but fails to detect "small" items with area below 1,000 cm^2. With SAHI, the model correctly detects some "small" items, as well as two "big" items.

Performance for SSL and SL methods

Fig. 6.7 illustrates the zero-shot generalization performance of SSL and baseline SL models with SAHI on the unseen Thu Thiem (W_s, H_s = 1280 pixel) and Binh Loi (W_s, H_s = 1920 pixel) location. The confidence threshold is 0.9. These models were fine-tuned on the $\text{Train}_{100\%}$ subset across 10 runs. The results demonstrate that SSL methods significantly outperform the baseline SL methods across all metrics for both locations. For the Thu Thiem location, the SSL method achieves substantial improvements of 0.25 in median precision, 0.11 in median recall and 0.14 in median F1-score, compared to the baseline SL method. Similarly, for Binh Loi location, the SSL method show enhancement of 0.09 in median precision, 0.09 in median recall, and 0.07 in median F1-score. These superior performance are further reflected by visual inspection of the predicted bounding boxes, as depicted in Fig. 10.3 and Fig. 10.4 in Appendix 10. The baseline SL method yields few correct detections and a high misdetection probability, particularly with respect to water hyacinth, reflective elements on the river surface, and other disturbances. These findings indicate that the feature representations learned through SwAV pre-training significantly enhance the model's out-of-domain generalization capability to new environments, consistent with our findings in Chapter 5.



(a) Detection performance at Thu Thiem

(b) Detection performance at Binh Loi

Figure 6.7: Zero-shot generalization performance on precision, recall, and F1-score metrics of SSL and baseline SL methods for the two unseen locations: Thu Thiem and Binh Loi bridge. The models were fine-tuned on the $\text{Train}_{100\%}$ subset.

While the SSL methods achieve much better performance than the baseline SL methods, all metric values remain low for practical application purposes in the TUD-HCMC case study. Specifically, the SSL method obtains a median F1-score of 0.33 and 0.16 for the Thu Thiem and Binh Loi location, respectively. After conducting a quantitative analysis of litter items in TUD-HCMC images, we found that approximately 40% of all litter items are transparent or are entrapped in water hyacinths, as shown in Fig. 6.8. The developed model usually fails to detect transparent litter, due to the insufficient differentiation between the features of transparent litter and water surface. This challenge is further exacerbated under poor lighting conditions. Additionally, the water hyacinths cover large areas of the litter, resulting in insufficient visible details of litter for accurate detection. These occlusions can also distort the litter's shape and texture, making recognition even more difficult. For example, in the Thu Thiem, the SSL model fails to detect

the majority of transparent litter items (14 out of 17 cases) and all entrapped items (5 cases). Similarly, in Binh Loi, all transparent litter items (8 cases) and most entrapped items (26 out of 33 cases) remain undetected. Thus, we explained the low recall and in turn the low F1-score, by the failure to detect these two types of litter. An additional contributing factor may be dataset imbalance. While the pre-training dataset maintains a relatively balanced distribution of samples between rivers in the Netherlands (42%) and those in Vietnam (55%), the fine-tuning dataset shows a strong imbalance (81% vs. 2%, see Table 6.1), which limits the model's generalization capability to rivers in Vietnam.

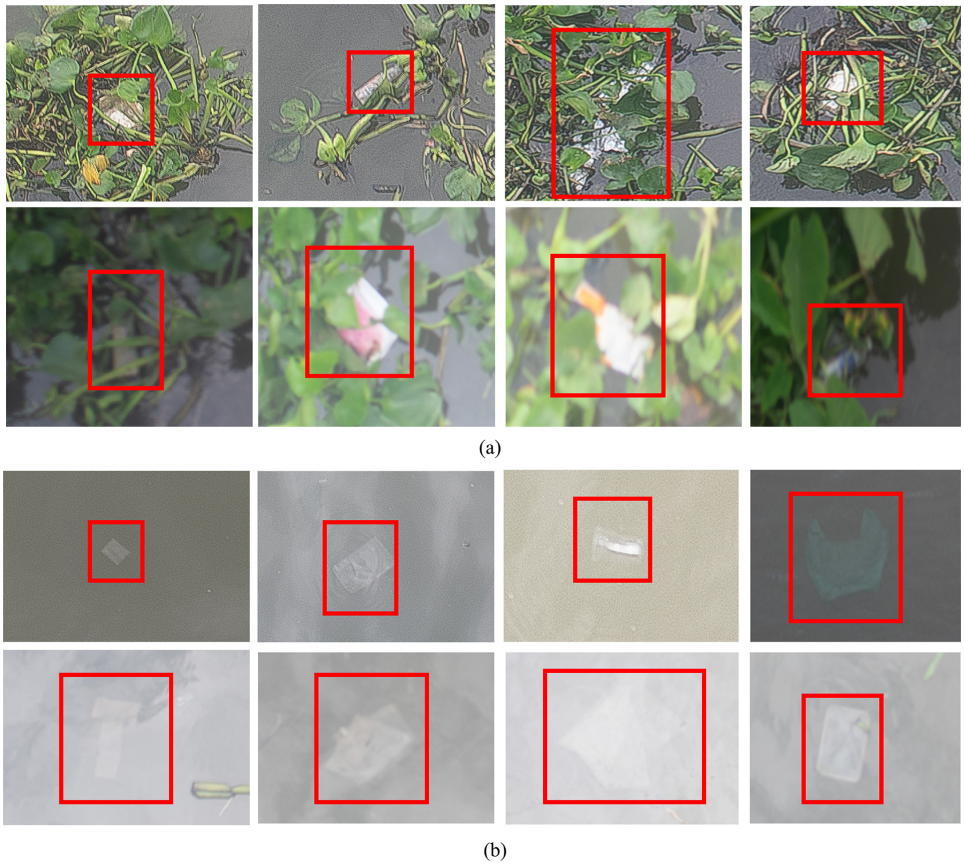
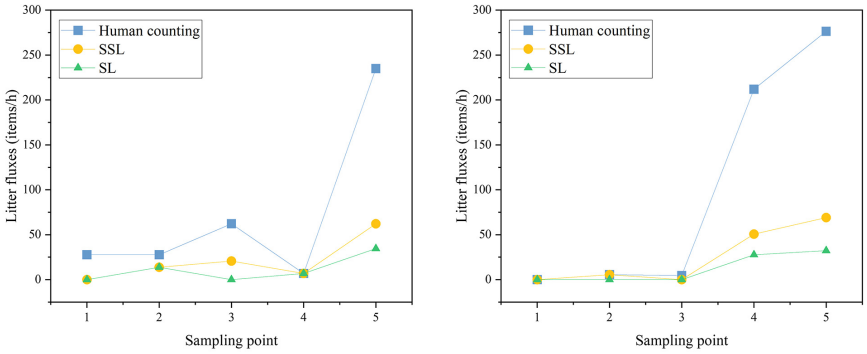


Figure 6.8: Examples of litter items undetected by the Faster R-CNN on the $\text{Test}_{\text{Thu Thiem}}$ and $\text{Test}_{\text{Binh Loi}}$ subset using the SSL method. The models were fine-tuned on the $\text{Train}_{100\%}$ subset. These include litter items entrapped in water hyacinths (a) and transparent items (b). Ground-truth litter is shown in red bounding boxes.

6.4.3. EXPERIMENT 3: LITTER FLUX MEASUREMENT

Fig. 6.9 shows the horizontal distribution of cross-sectional floating litter fluxes, measured by multiple frameworks and methods. We measured fluxes by including the cor-

rectly detected litter items by models (i.e., TPs). For this measurement evaluation, we selected the best-performing SSL and baseline SL models (achieving the highest F1-score) across the 10 runs from Experiment 2. The results indicate that the SSL-based and baseline SL-based frameworks yield identical flux measurements for most low-fluxes region (human-measured fluxes < 50 items/h), such as 14 items/h for sampling point 2 at Thu Thiem and 0 items/h for sampling point 1 at Binh Loi (see Fig. 3.9 in Chapter 3). However, for the high-flux region (human-measured fluxes > 50 items/h), the SSL-based framework significantly outperforms the SL-based framework by consistently measuring higher fluxes, aligning more closely with those measured by humans. For example, at sampling point 5, fluxes measured by the SSL-based framework is 27 items/h and 37 items/h higher than that by the SL-based framework for the Thu Thiem and Binh Loi bridges, respectively. This is mainly attributed to the higher recall achieved by the SSL models compared to the baseline SL models, as described in Chapter 6.4.2.



(a) Litter fluxes for each sampling point at Thu Thiem

(b) Litter fluxes for each sampling point at Binh Loi

Figure 6.9: Horizontal distribution of cross-sectional floating litter fluxes, measured by the SSL-based and baseline SL-based framework, and human counting method. We measured the mean litter fluxes by including the correctly detected litter items by models (i.e., true positives). The SSL and baseline SL models are best-performing models in 10 runs from Experiment 2.

Fig. 6.9 also demonstrates that the concentration of litter items is highest near the eastern riverbanks (i.e., sampling point 5), accounting for approximately 70% at Thu Thiem and 60% at Binh Loi. This spatial distribution can be mainly explained by the flow direction and river morphology. Floating litter fluxes are likely highest in the outer curves of the river (van Emmerik et al., 2018a), as observed at sampling point 5 for the Binh Loi and Thu Thiem bridges during ebb tides. van Emmerik et al. (2018a) also reported a similar spatial distribution of litter fluxes based on measurement taken at 12 sampling points on the Thu Thiem bridge.

Fig. 6.10 presents the linear fit of fluxes measured by the SSL-based and SL-based framework against those measured via human counting. The results indicate a strong positive correlation between fluxes measured by human and that by DL-based frameworks, regardless of whether the SSL or baseline models are used. However, the SSL-based framework demonstrates a stronger correlation with human counting (the Pearson correlation coefficient r=0.99), compared to the SL-based framework (r=0.93).

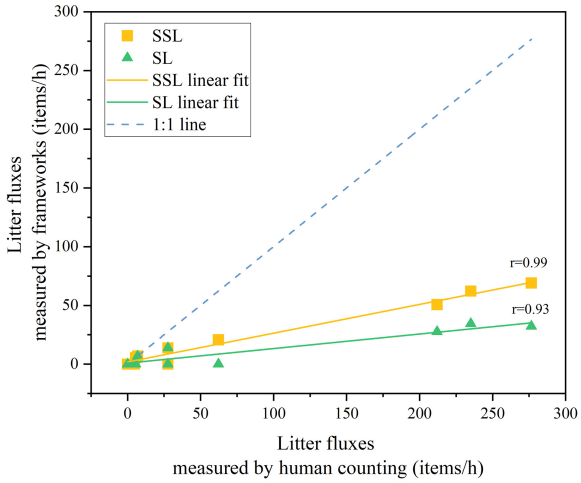


Figure 6.10: Comparison of the mean litter fluxes of 10 sampling points with linear fit analysis, including the Pearson correlation coefficient (r): SSL-based framework, baseline SL-based framework, and human counting method.

Nevertheless, both DL-based frameworks significantly underestimates fluxes in high-flux regions, compared to human measurements. Interestingly, van Lieshout et al. (2020) reported contrasting findings, where DL models estimate relatively higher fluxes for video clips with high litter fluxes, compared to human measurements. The high fluxes, reaching up to 35 items/(min·m) in some video clips, poses a significant challenge for human counters to accurately identify and count each transported litter item. Thus, this discrepancy can be attributed to the limitation on how many objects per minute human observers can realistically count. However, human observers in this study did not face such challenge, since we counted litter items directly from images rather than videos, ensuring reliable human measurements. The lower fluxes measured by DL-based frameworks in this study is explained by the low detection accuracy of DL models, as discussed in Chapter 6.4.2.

Fig. 6.11 shows the total cross-sectional floating litter fluxes at the Thu Thiem and Binh Loi bridges. Both DL-based frameworks substantially underestimate the fluxes, compared to human counting. Specifically, the fluxes measured by the SSL-based framework is approximately 3 times lower than human measurements at the Thu Thiem bridge, and 4 times lower at the Binh Loi bridge. Despite this underestimation, the fluxes measured by the SSL-based framework (858 items/h at Thu Thiem and 826 items/h at Binh Loi) are nearly double those of the baseline SL-based framework (464 items/h and 413 items/h, respectively). This improvement also highlights the superior capability of the SSL-based framework for flux measurement, compare to the SL-based framework.

Our human-measured cross-sectional fluxes at the Thu Thiem bridge align with the findings of van Emmerik et al. (2019a). They reported fluxes measured by human visual counting methods at 12 sampling points on this bridge in 2018, including data from 5 days in September. Using the approach presented in Chapter 6.2.4, we estimated cross-

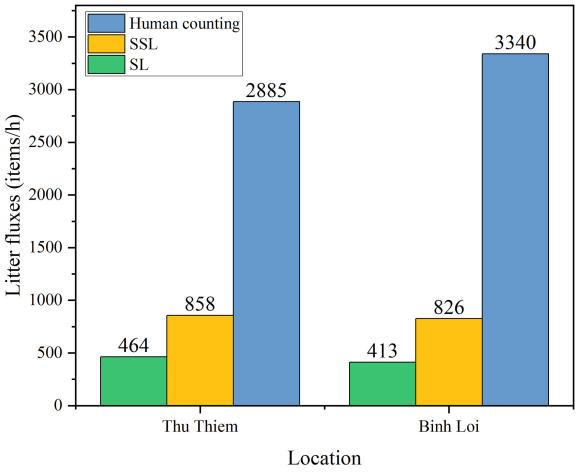


Figure 6.11: The cross-sectional floating litter fluxes at the Thu Thiem and Binh Loi bridge, measured by the SSL-based and baseline SL-based framework, and human counting method.

sectional fluxes, and found that fluxes during ebb tides for these 5 days in September ranged from 1,409 to 31,195 items/h. Our measured fluxes (2,885 items/h) in this study falls within this range.

6.4.4. LIMITATIONS

The aim of this study is not to deploy and optimize an automated system for floating litter flux measurement, but rather to demonstrate that self-supervision can serve as an effective approach toward developing such a system. The performance improvements achieved by SSL methods over SL methods further strengthen the findings of our study in Chapter 5. While their zero-shot detection accuracy remains lower than the expensive and time-consuming human counting method in the TUD-HCMC case study, their performance holds significant potential for improvement through cost-effective approaches.

We proposed several recommendations to further enhance model performance. First, increasing the amount of unlabeled data and extending the training time are beneficial, as shown in Chapter 6.4.2. Second, collecting a limited number of labeled images from target rivers for fine-tuning SSL models and optimizing hyperparameters in SAHI could further enhance model accuracy (van Lieshout et al., 2020). This is particularly effective when using images containing specific litter types from target rivers, e.g., transparent litter and litter entrapped in water hyacinths in the TUD-HCMC case study. Due to the highly limited number of images we collected in the TUD-HCMC case study, we did not perform standard hyperparameter optimization in this study. Third, we suggest applying data augmentation methods to increase the number of labeled instances of specific litter types. For example, water hyacinths of varying sizes can be extracted from source images, and pasted within the bounding boxes of litter items in target images, provided that the hyacinth area is smaller than the corresponding bounding box. This operation

simulates scenarios where litter is partially occluded by vegetation, thereby enhancing the model's robustness to such challenging scenarios. Lastly, the framework could also benefit from replacing the ResNet50 backbone with state-of-the-art architectures, such as transformers that have demonstrated their effectiveness in foundational models like GPT, DINOv2, and Prithvi (Dosovitskiy et al., 2020).

A major shortcoming of developed models in this study is their inability to automatically identify and count the same litter item appearing in multiple consecutive images as a single item, resulting in overestimated fluxes. While we manually corrected the number of litter items detected by models to avoid flux overestimation, more automated methods are needed to address this issue, such as employing DeepSORT for tracking the detected objects across consecutive images (Wojke et al., 2017).

6.5. CONCLUSIONS

Based on the previous analysis of the effectiveness of semi-supervised learning (SSL), and data-centric artificial intelligence (AI) approaches, we proposed a SSL-based framework to measure cross-sectional floating litter fluxes in river systems, with the limited availability of labeled data. We incorporated a data-centric AI method (i.e., flipping data augmentation) to enhance model performance. Additionally, we used a Slicing Aided Hyper Inference (SAHI) method to enhance accuracy of small litter detection. To demonstrate the effectiveness of the proposed framework, we conducted multiple experiments on camera and drone images from canals and waterways of the Netherlands, Indonesia, and Vietnam. We benchmarked our measurement results against the baseline SL and human counting methods. Our main findings are as follows:

(1) The SSL models benefit from longer pre-training time and larger pre-training dataset size. Especially, when a large amount of data (200k images) is available, increasing pre-training epochs from 100 to 200 achieves an improvement in average precision (AP50) of 4.2%. Moreover, scaling the pre-training dataset size from 20k to 500k yields an improvement in AP50 of 14.7% and F1-score of 0.24, with a limited amount of labeled data for fine-tuning (226 images with 276 annotated litter items).

(2) The SSL methods significantly outperform the baseline SL methods in in-domain and out-of-domain detection performance. Compared to baseline SL benchmarks, SSL methods achieve an in-domain AP50 increase of 12% and F1-score increase of 0.2, and a zero-shot out-of-domain median F1-score increase of up to 0.14. It can be primarily attributed to the extraction of more informative and robust feature representations through self-supervised pre-training on relevant unlabeled images.

(3) The SAHI method enables the SSL models' ability to accurately detect 45 additional "small" litter items (area < 1,000 cm^2) in the Vietnam case study, compared to the results obtained from the same SSL models without SAHI. This improvement also lead to an increase in F1-score by 0.34 and 0.19 for two locations in the case study, respectively.

(4) The cross-sectional floating litter fluxes measured by the SSL-based framework are nearly double those of the baseline SL-based framework, demonstrating closer alignment with human-measured fluxes in the Vietnam case study. While the SSL-based framework exhibits a strong positive correlation with human-measured fluxes across 10 sampling points (the Pearson correlation coefficient r=0.99), it significantly underestimates fluxes by a factor of 3 to 4, compared to human measurements. One of the main

reasons is the challenge of correctly detecting transparent litter and litter entrapped in water hyacinth, which together account for around 40% of the litter items in the Vietnam case study.

While we tested this new framework with cameras only for river surfaces, it can also be used with drones. It can be even extended to measure litter fluxes under river surface, provided that images are captured using similar devices (e.g., underwater cameras) or sonar technologies. Additionally, combining SSL methods with images collected from manned aircraft can enhance the detection of macroplastic hotspots on a larger scale, e.g., marine surfaces (Garcia-Garin et al., 2021).

7

CONCLUSIONS AND RECOMMENDATIONS

7.1. THESIS SUMMARY

This thesis aims to enhance the understanding of the current state of deep learning (DL)-based detection and quantification of floating litter in rivers, identify key knowledge gaps, and exploit methodologies to address these gaps. Through a comprehensive literature review in this field, we identified three key knowledge gaps, which were synthesized into the main research question of this thesis:

- *How to develop robust DL-based methods for detecting floating litter and quantifying cross-sectional floating litter fluxes in rivers, particularly in contexts with limited labeled data?*

To answer this main research question, three sub-questions are defined based on three knowledge gaps, and then addressed in Chapter 4-6 contained within this thesis:

1. *How to build robust DL models to detect floating litter in rivers, leveraging a relatively large amount of labeled data?* (Gap 1, Chapter 4)
2. *How to build robust DL models to detect floating litter in rivers, leveraging a limited amount of labeled data?* (Gap 2, Chapter 5)
3. *How to develop DL methods to quantify cross-sectional floating litter fluxes in rivers, leveraging a limited amount of labeled data? (Gap 3, Chapter 6)*

To address research questions, we needed to assess multiple DL methods on different datasets for litter detection and quantification. Thus, we generated multiple datasets by collecting data from multiple locations in canals and waterways in the Netherlands and Vietnam (Chapter 3), and proposed and evaluated multiple methodologies for litter detection and quantification (Chapter 4-6) using these datasets and two existing openly available datasets.

First, we evaluated multiple transfer learning and data-centric Artificial Intelligence (AI) approaches to enhance model performance, leveraging a relatively large amount of labeled data (Chapter 4). However, obtaining these labeled images for model development is costly and labor-intensive. To overcome this challenge, we proposed and evaluated a semi-supervised learning (SSL) method to improve model performance, leveraging a limited number of labeled images (Chapter 5). When developing SSL models, we also used the best-performing transfer learning method, as identified in the previous analysis in Chapter 4. Moreover, we proposed and assessed a SSL-based framework for quantifying cross-sectional floating litter fluxes in rivers, integrating the above explored methodologies (Chapter 6). Finally, we addressed the main research question of this thesis by integrating the explored methodologies presented in Chapter 4-6, i.e., a SSL-based framework combined with the appropriate transfer learning and data-centric AI approaches for floating litter detection and quantification, leveraging a limited amount of labeled data.

7.2. THESIS FINDINGS

7.2.1. THESIS CONCLUSIONS

We present the main conclusions for each research question.

- *How to build robust DL models to detect floating litter in rivers, leveraging a relatively large amount of labeled data?*

  We found that selecting effective transfer learning and data-centric AI approaches can benefit in building robust SL models to detect floating litter in rivers, with a relatively large amount of labeled data. Transferring the convolutional base from ImageNet and fine-tuning the entire network on floating litter images is the best TL method to improve in-domain generalization performance, compared to the other two tested training methods. Additionally, data augmentation techniques such as flipping augmentation can effectively enhance in-domain generalization performance at a low cost. In contrast, augmentations such as brightening, darkening, and adding noise do not lead to significant improvements. Moreover, adding a limited number of images from new device settings to the original training dataset can significantly enhance out-of-domain generalization performance in complex scenarios, involving both different camera heights and different viewing angles.

- *How to build robust DL models to detect floating litter in rivers, leveraging a limited amount of labeled data?*

  We proposed a SSL method for floating litter detection to improve model generalization capability, leveraging a limited amount of labeled data. We benchmarked our results against the same model architecture trained via SL alone. The results show that our method performs on par or better than the SL method in terms of in-domain generalization performances (i.e., at the same locations). Moreover, the enhancements are particularly noticeable when only a limited amount of labeled data available for fine-tuning. More importantly, this method outperforms the SL method in out-of-domain generalization performances (i.e., across unseen locations).

- *How to develop DL-based methods to quantify cross-sectional floating litter fluxes in rivers, leveraging a limited amount of labeled data?*

  We proposed a SSL-based framework to estimate cross-sectional floating litter fluxes in rivers, with the limited availability of labeled data for model development. We incorporated a data-centric AI method (i.e., flipping data augmentation) to further improve model performance. Moreover, we used a Slicing Aided Hyper Inference (SAHI) method to improve the accuracy of small litter detection. Our measurement results were benchmarked against baseline SL models and human visual counting methods. The results show that the SAHI enables model to correctly detect much more small litter items. The SSL-based framework demonstrates a stronger correlation with human counting, compared to the SL-based framework. Additionally, the SSL-based framework estimates fluxes nearly twice as high as the baseline SL-based framework, offering estimates that align more closely with human-measured fluxes.

7.2.2. THESIS SCIENTIFIC CONTRIBUTION

The scientific contributions of this thesis are summarized as follows:

1. Proposing a novel DL-based framework for quantifying cross-sectional floating litter fluxes in rivers, with limited labeled data.
2. Proposing a novel method to enhance litter detection with limited labeled data.
3. Providing new insights to enhance floating litter detection performance with transfer learning and data-centric AI.
4. Generating multiple new datasets for developing DL models for floating litter detection and quantification.

The datasets generated by this thesis are accessible via Zenodo, as detailed in Chapter 3. The source code corresponding to the models and methodologies presented in Chapters 4, 5, and 6 is available at the following repositories:
(1) Chapter 4: https://github.com/TianlongJia/deep_plastic
(2) Chapter 5: https://github.com/TianlongJia/deep_plastic_SSL
(3) Chapter 6: https://github.com/TianlongJia/deep_plastic_Flux_SSL
Instructions for using the code are also provided in each of the repositories.

7.2.3. IMPLICATIONS FOR ENVIRONMENTALLY SUSTAINABLE DEVELOPMENT

The proposed methods and frameworks can be directly applied to develop litter monitoring methods in river networks worldwide. These monitoring methods can provide litter quantification results (e.g., litter fluxes) and pollution level assessments to stakeholders aiming to reduce environmental pollution, including practitioners and researchers. These assessments could support practitioners in devising mitigation strategies and targeted cleanups. Moreover, researchers can leverage the quantification results to investigate litter sources, transport pathways, distributions, retention dynamics, and long-term trends. These results can also facilitate the study of correlations between plastic transport and influencing factors, such as discharge, tidal dynamics, and rainfall (van Emmerik et al., 2019a, 2022a). These studies could contribute to the development of effective mitigation strategies. For example, van Emmerik et al. (2024) conducted continuous sampling over two months along the Saigon river, Vietnam, collecting around 16,000 images. Manually detecting and counting litter items in such a large number of images is impractical. Thus, they used the YOLOv8 deep learning model to detect and count three objects: (1) litter free from water hyacinth, (2) litter entrapped in water hyacinth, and (3) water hyacinth. Their results illustrate litter distribution at five locations along the river and reveal interactions between litter and water hyacinth, with over 73% of all floating plastics found entrapped by water hyacinths. Based on these findings, they suggested that the current removal practices of water hyacinths could be optimized to also remove plastic litter.

7.3. RECOMMENDATIONS FOR ENGINEERING PRACTICE

- **Developing models for floating litter detection with SSL methods**

When only a limited amount of labeled data from target rivers is available, we suggest practitioners using our proposed SSL method to develop models for litter detection. These models can be pre-trained on a large amount of unlabeled images (e.g., datasets generated in this thesis) using a self-supervising method (e.g., SwAV). This approach overcomes the challenge of requiring extensive labeled datasets for developing robust models. Additionally, it allows models to achieve higher generalization capability by using more unlabeled data, that is more easily accessible and less costly to collect, compared to labeled data. It can significantly reduce the cost and time required for data collection and annotation. This presents a more cost-effective option for developing robust models for large-scale structured monitoring in river systems.

- **Enhancing floating litter detection through additional efforts**

  In addition to applying SSL methods when only a limited amount of labeled data is available, we also suggest performing data augmentation techniques (e.g., flipping) on the labeled data to further enhance SSL model performance. This approach leverages the existing labeled data to generate additional training samples, thereby improving model performance without the need to collect new data from the target rivers. In scenarios where numerous small litter items are present in target rivers and cameras are positioned high above the water surface, resulting in low-resolution images, we suggest using the SAHI method to improve the detection of small litter items. If the above methods still do not achieve satisfactory performance in target rivers with different geographic, environmental, and device setup conditions, we finally suggest adding a small amount of labeled data collected from these new conditions to further improve generalization.

- **Enhancing cross-sectional floating litter flux quantification in rivers**

  We suggest practitioners considering our proposed SSL-based framework to quantify litter fluxes as an alternative to human visual counting method, particularly in scenarios requiring long-term and frequent monitoring where resources for human counting are limited, or where conditions become dangerous, such as during extreme flood events. This framework can provide more accurate flux estimations compared to conventional SL-based frameworks. Furthermore, this framework is particularly effective, when the availability of labeled training data is limited. By reducing the reliance on extensive labeled datasets, it significantly lowers the cost and time required for data collection and annotation.

7.4. RECOMMENDATIONS FOR FUTURE WORK

To address a pollution problem of global scale, we recommend researchers focus further efforts on:

1. Developing a more robust litter detection model.
2. Quantifying floating litter mass fluxes and hotspots in rivers.

3. Developing DL-based monitoring strategies for riverine litter.

7.4.1. DEVELOPMENT OF A ROBUST FLOATING LITTER DETECTION MODEL

Following the transfer learning methods, data-centric AI approaches (see Chapter 4), and the proposed SSL methods (see Chapter 5), we can build a robust DL model to detect floating litter in rivers. To further develop more robust models, we recommend to increase efforts to (1) develop foundational models for floating litter detection, and (2) generate a labeled dataset using data-centric AI approaches, as suggested in the following paragraphs.

TOWARDS FOUNDATIONAL MODELS FOR LITTER DETECTION

Foundational models are a recent transformative paradigm in DL. By leveraging vast amounts of data via self-supervision (e.g., SwAV methods presented in Chapter 5), these models achieve remarkable general understanding and adaptability, which allows them to reach unprecedented performances when fine-tuned for specialized tasks. This paradigm shift is exemplified by the OpenAI GPT series, a family of self-supervised foundational models that, in their latest iterations, launched the current AI revolution by enabling the development of ChatGPT via specialization (Achiam et al., 2023; Brown et al., 2020). More importantly, our preliminary explorations in Chapter 5 shows the significant benefits of self-supervised learning methods on models' generalization. Thus, we believe foundational models tailored for floating litter detection could significantly enhance large-scale monitoring, and mitigate this environmental issue, whether from camera imagery or satellites.

To address a pollution problem of global scale, we must significantly expand our dataset to include millions of images from diverse geographical locations. Based on the experimental findings in Chapter 6, we argue that scaling this approach is necessary to yield more robust models (Goyal et al., 2022). Gathering vast quantities of diversified data is a necessary step, but not sufficient. Additional efforts must be directed towards implementation strategies, hyper-parameter optimization, and the selection of suitable DL architectures. For instance, considering the efficacy of transformers in state-of-the-art foundational models like GPT, DINOv2, and Prithvi, adopting similar architectures could be beneficial (Dosovitskiy et al., 2020).

GENERATING A LABELED DATASET USING DATA-CENTRIC AI APPROACHES

After developing the above foundational models, we can obtain robust floating litter detection models by fine-tuning these foundational models using a labeled dataset (see Chapter 5). To build such labeled dataset, we recommend using three data-centric AI approaches: (1) collecting additional data from in-situ experiments, (2) enforcing consistency in the labeling procedure, and (3) using advanced data augmentation techniques.

The data-centric approach favors the collection of additional data from in-situ experiments whenever possible. In particular, we suggest gathering more training images at various sampling locations under different environmental conditions, and extending the collection to different devices and instrumental settings for extensive monitoring applications (e.g., river networks, nation-wide initiatives). We also recommend collecting more data about rare items, such as those entrapped in water hyacinths (see Chapter 6).

Enforcing consistency in the labeling procedure can results in similar improvements (Jain et al., 2021). Manual labeling may introduce significant human error and bias in data, which in turn may severely undermine model performances. Clear guidelines (with illustrative examples) and cross-checking between multiple labelers strengthen the consistency of labeled data towards achieving reliable performances (Lavitas et al., 2021).

We should resort to advanced data augmentation techniques to increase the number of the images collected in the field, such as copy-paste augmentation described in Chapter 2.3.5. Contrary to some traditional techniques (e.g., cropping), this augmentation procedure does not change the original features of target litter or omits objects in the newly generated images. Studies form other fields suggest substantial benefits for different CV tasks (Dwibedi et al., 2017; Ghiasi et al., 2021).

7.4.2. QUANTIFICATION OF FLOATING LITTER MASS FLUXES AND HOTSPOTS IN RIVERS

Stakeholders need detailed information on floating litter mass fluxes or mass of litter in hotspots to design more effective cleaning campaigns, and mitigate the impact of pollution on the environment and human health (Tasseron et al., 2020; van Emmerik et al., 2018b). Future studies should focus on the development of new methods for quantifying litter mass fluxes and hotspots in rivers.

LITTER MASS FLUXES

Litter mass fluxes can be expressed as the mass of litter items across the river width per unit of time (van Emmerik et al., 2018b). To quantify it, we recommend using the proposed SSL-based framework to measure cross-sectional floating litter fluxes in target rivers, as highlighted in Chapter 6. Then, a limited number of experiments must be conducted in target rivers by (i) sampling litter using nets, (ii) counting the number of samples, and (iii) weighing them. The sampled average densities can be computed by dividing the weight of litter by the number of items in experiments. Finally, mass fluxes is measured by multiplying fluxes with respect to sampled average densities (van Emmerik et al., 2018b).

MASS OF LITTER IN HOTSPOTS

Litter hotspots are locations where a large amount of litter accumulates on the water surface due to favorable morphological and environmental conditions (Moy et al., 2018). We suggest collecting hotspot images at various locations along target rivers using UAVs, that can provide an overview of pollution with low human labor costs and high-resolution images. We suggest developing DL models for semantic segmentation tasks on these images using the SSL method. Such models can precisely quantify the area of target objects, as shown in Chapter 2.3.3. Next, limited experiments are needed to (i) collect hotspot images and measure the true area of them, (ii) sample litter in these hotspots (e.g., using nets), and (iii) weigh litter. The spatial average densities of hotspots can be computed by dividing the weight of the litter by the true area occupied by them. After accounting for image resolution (e.g., pixel to area ratio), the mass of litter in hotspots can be obtained by multiplying the pixels identified as hotspots by the image segmentation algorithm by the spatial average densities.

7.4.3. DL-BASED MONITORING OF RIVERINE LITTER

Monitoring riverine litter sources, transport, distribution, sinks, and trends is crucial for decision-makers to devise mitigation strategies and targeted cleanups (van Emmerik et al., 2022c; Vriend et al., 2020a). Effective monitoring methods enhance these efforts. To enable automated, long-term monitoring, we recommend integrating machine learning operations (MLOps) (Ruf et al., 2021) into robust DL-based litter quantification in river systems. MLOps is a continuous integration/continuous deployment process implemented for machine learning-based solution. It enables long-term utilization and refinement of machine learning-based solutions by automating all key phases such as data management, model deployment, and model validation. The integration of MLOps could results in the following steps leading to DL-based structural monitoring of litter. The first step could include the selection of DL tasks (e.g., image classification, object detection, and image segmentation), DL architectures (e.g., Faster R-CNN and U-Net) and monitoring devices (e.g., cameras and drones) for the specific problem. In the second step, the MLOps infrastructure could be initially deployed on selected pilot projects to develop baseline DL models for litter quantification and validate their performance. This will require systematic data gathering and ground truth measurements (e.g., visual inspection of recordings, and comparison against visual counting). After establishing a satisfactory baseline, long-term monitoring on the selected pilot studies can start.

Concurrently, the infrastructure can be strategically extended, employing the baseline DL models for monitoring at new locations. Following the data-centric AI approaches (see Chapter 4 and 7.4.1), we can add novel, accurately labeled images at these new (or existing) locations to improve the generalization capability of the baseline models. This could also include adding litter categories of interests underrepresented in the training dataset or performing tailored ground truth validation for more accurate quantification. As witnessed in other fields of application (Ruf et al., 2021), I believe several iterations of the proposed MLOps approach may lead to robust and automated structural monitoring of litter.

8

APPENDIX TO CHAPTER 4

8.1. MODEL PERFORMANCES AND TRAINING TIME IN EXPERIMENT 1

Table 8.1: Training time and overall accuracy of five architectures employing fine-tuning strategies or trained from scratch in Experiment 1

Model	Learning rate	From scratch		FTC		FTAL	
		Training time per epoch (s)	Overall accuracy (%)	Training time per epoch (s)	Overall accuracy (%)	Training time per epoch (s)	Overall accuracy (%)
ResNet50	0.1	13	76.0	8	62.0	22	68.7
	0.01	20	80.5	8	62.3	19	81.6
	0.001	22	83.3	8	56.0	13	85.0
	0.0001	18	80.3	8	43.8	22	84.4
	0.00001	22	65.0	8	36.5	22	72.0
InceptionV3	0.1	21	77.8	7	66.3	19	77.8
	0.01	21	80.6	7	65.7	20	80.5
	0.001	21	83.0	7	64.9	20	85.7
	0.0001	19	80.3	7	66.5	15	85.5
	0.00001	21	69.5	8	59.2	20	67.8
DenseNet121	0.1	26	75.3	9	69.3	29	80.2
	0.01	29	81.0	9	70.7	28	82.0
	0.001	29	83.4	9	73.3	29	87.2
	0.0001	28	83.5	9	71.9	18	87.6
	0.00001	29	74.8	9	61.2	29	71.9
MobileNetV2	0.1	19	33.8	4	70.3	18	34.7
	0.01	19	81.7	4	70.4	19	81.5
	0.001	19	81.1	4	72.0	18	86.2
	0.0001	19	67.5	4	72.7	19	77.3
	0.00001	19	42.7	4	65.6	19	69.7
SqueezeNet	0.1	5	33.7	2	61.0	4	33.7
	0.01	5	33.7	2	63.5	5	33.7
	0.001	5	33.7	2	64.7	5	33.7
	0.0001	5	54.5	2	65.8	5	87.6
	0.00001	5	77.8	2	57.2	4	84.2

Table 8.2: Precision, recall and F1-score per class for five architectures using the FTAL method

Model	Metric	Class			
		No plastic	Little plastic	Moderate plastic	Lots of plastic
ResNet50	precision	0.98	0.81	0.78	0.91
	recall	0.97	0.85	0.80	0.84
	F1-score	0.98	0.83	0.79	0.87
InceptionV3	precision	0.97	0.84	0.80	0.89
	recall	0.98	0.84	0.81	0.85
	F1-score	0.98	0.84	0.80	0.87
DenseNet121	precision	0.97	0.85	0.82	0.92
	recall	0.98	0.87	0.84	0.86
	F1-score	0.97	0.86	0.83	0.89
MobileNetV2	precision	0.99	0.86	0.80	0.87
	recall	0.96	0.85	0.82	0.86
	F1-score	0.97	0.86	0.81	0.86
SqueezeNet	precision	0.98	0.86	0.80	0.94
	recall	0.97	0.89	0.86	0.82
	F1-score	0.98	0.88	0.83	0.87

8.2. MODEL PERFORMANCES AND TRAINING TIME IN EXPERIMENT 2 AND 3

Table 8.3: Training time, performances of data augmentation techniques (Experiment 2), and the evaluation of generalization capability (Experiment 3) of SqueezeNet and DenseNet

Architecure	Method	Training time per epoch (s)	Overall accuracy on test dataset (%)				
			Test	$\text{Test}_{2.7m/0°}$	$\text{Test}_{2.7m/45°}$	$\text{Test}_{4m/0°}$	$\text{Test}_{4m/45°}$
SqueezeNet	flipping	19	89.6	86.5	90.5	76.8	65.4
	brightening	18	88.0				
	darkening	18	88.1				
	adding noise	20	87.5				
	Mix DA[1]	52	89.5				
	ANI[2]	5		85.7	90.0	81.1	74.3
	ANI-DA[3]	27		87.3	86.7	87.2	73.3
DenseNet121	flipping	110	91.6	90.7	88.7	87.6	63.4
	brightening	113	87.5				
	darkening	103	88.3				
	adding noise	99	88.3				
	Mix DA	307	90.9				
	ANI	41		86.8	88.0	89.0	72.8
	ANI-DA	164		91.5	94.1	93.4	77.7

[1] "MIX DA": Mixing four data augmentation techniques, including flipping, brightening, darkening, and adding noise.
[2] "ANI": Adding new images.
[3] "ANI-DA": Adding new images and performing data augmentation technique on the entire dataset.

9

Appendix to Chapter 5

9.1. IMAGE EXAMPLES

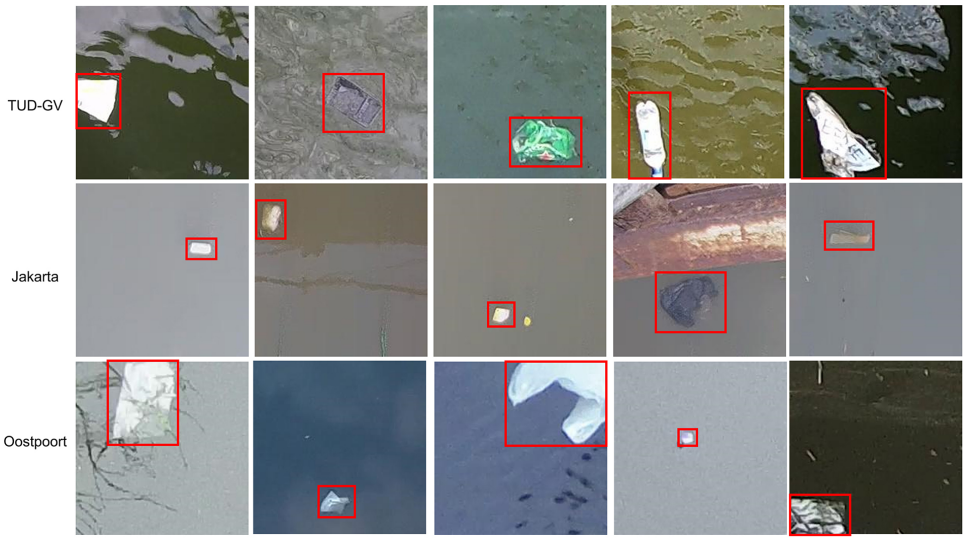


Figure 9.1: Examples of images tiles (224×224 pixels) from TUD-GV, Jakarta and Oostpoort dataset.

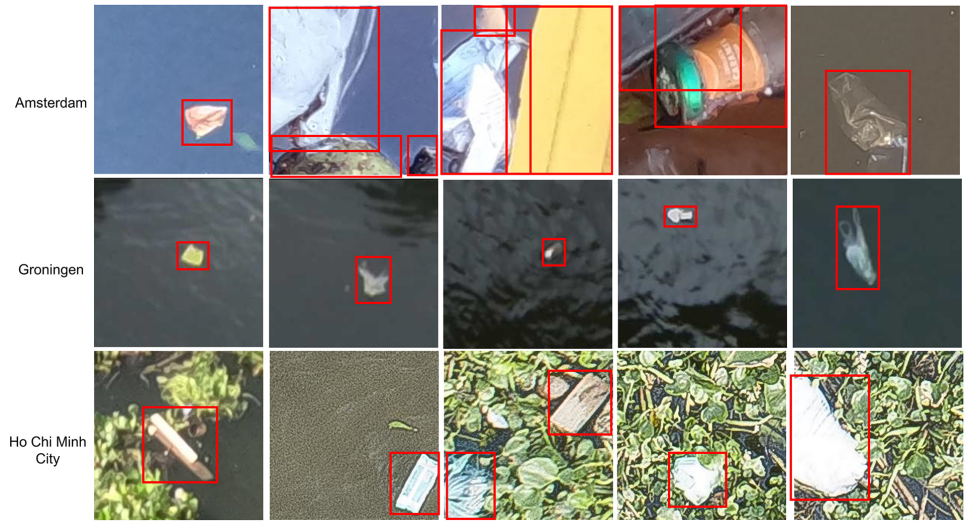


Figure 9.2: Examples of images tiles (224×224 pixels) from Amsterdam, Groningen and WUR-HCMC dataset.

9.2. CONFUSION MATRICES AND PERFORMANCE METRICS FOR OUT-OF-DOMAIN GENERALIZATION

Table 9.1: Model performances of the Faster R-CNN with ResNet50 backbone using various methods on Amsterdam images. The model was fine-tuned on the Train100% dataset

Method	Images with litter annotated						Images without litter
	TP	FN	FP	Precision	Recall	F1-score	FP
SwAV-FTAL-F2	81	123	**90**	**0.47**	0.40	0.43	**1,530**
SwAV-Scratch-F2	83	121	109	0.43	0.41	0.42	1,839
Baseline-F2	114	90	160	0.42	0.56	**0.48**	2,617
SwAV-FTAL-F4	**138**	**66**	241	0.36	**0.68**	0.47	2,249
SwAV-Scratch-F4	104	100	129	0.45	0.51	**0.48**	1,695
Baseline-F4	95	109	180	0.35	0.47	0.40	2,115

Table 9.2: Model performances of the Faster R-CNN with ResNet50 backbone using various methods on Groningen images. The model was fine-tuned on the Train100% dataset

Method	Images with litter annotated						Images without litter
	TP	FN	FP	Precision	Recall	F1-score	FP
SwAV-FTAL-F2	143	382	53	0.73	0.27	0.40	430
SwAV-Scratch-F2	117	408	56	0.68	0.22	0.34	401
Baseline-F2	165	360	**28**	**0.85**	0.31	0.46	**67**
SwAV-FTAL-F4	**283**	**242**	137	0.67	**0.54**	**0.60**	151
SwAV-Scratch-F4	227	298	99	0.70	0.43	0.53	219
Baseline-F4	208	317	167	0.55	0.40	0.46	468

Table 9.3: Model performances of the Faster R-CNN with ResNet50 backbone using various methods on Ho Chi Minh City images. The model was fine-tuned on the Train100% dataset

Method	Images with litter annotated						Images without litter
	TP	FN	FP	Precision	Recall	F1-score	FP
SwAV-FTAL-F2	**340**	**751**	1,128	0.23	**0.31**	0.27	5,889
SwAV-Scratch-F2	268	823	613	0.30	0.25	0.27	5,291
Baseline-F2	254	837	1,436	0.15	0.23	0.18	7,326
SwAV-FTAL-F4	310	781	954	0.25	0.28	0.26	4,009
SwAV-Scratch-F4	272	819	**434**	**0.39**	0.25	**0.30**	**2,946**
Baseline-F4	236	855	929	0.20	0.22	0.21	4,300

10

APPENDIX TO CHAPTER 6

10.1. VALIDATION ACCURACY OF ALL SSL MODELS FOR EXPERIMENT 1

Table 10.1: Validation accuracy (AP50) on the $\text{Validation}_{60\%}$ and $\text{Validation}_{20\%}$ subsets of all SSL models for Experiment 1. The bold entities are the best results for models pre-trained on each pre-training dataset

Pre-training dataset	No.pre-training epochs	AP50 ($\text{Validation}_{60\%}$)	AP50 ($\text{Validation}_{20\%}$)
Train_{25k}	100	77.8%	71.4%
	200	**80.9%**	**78.0%**
	300	76.5%	70.2%
Train_{50k}	100	**80.5%**	68.4%
	200	78.4%	**81.1%**
	300	80.2%	68.8%
Train_{100k}	100	80.5%	77.4%
	200	**83.4%**	**78.1%**
	300	82.1%	76.9%
Train_{200k}	100	**80.1%**	77.2%
	200	79.2%	**78.9%**
	300	78.4%	73.6%
Train_{300k}	100	83.2%	**78.6%**
	200	80.8%	78.5%
	300	**83.5%**	77.0%
Train_{500k}	100	77.8%	**83.9%**
	200	**83.5%**	80.2%
	300	81.7%	82.4%

10.2. EXAMPLE OF PREDICTED BOUNDING BOXES FOR TUD-HCMC CASE STUDY IN EXPERIMENT 2

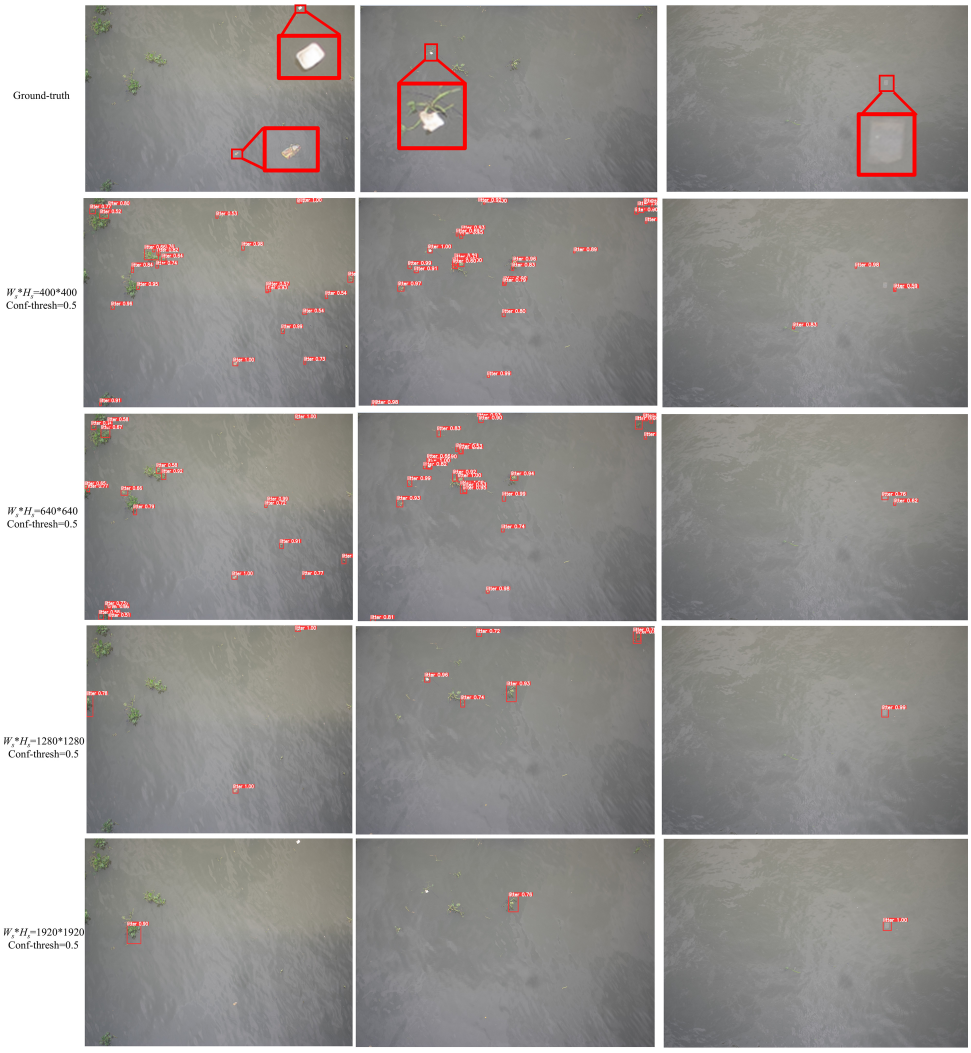


Figure 10.1: Examples of predicted bounding boxes for the Faster R-CNN model with the SAHI method on the $\text{Test}_{\text{Thu Thiem}}$ subset. We used the SSL method to develop the Faster R-CNN model, that was fine-tuned on the $\text{Train}_{100\%}$ subset. During inference, we used various W_s and H_s hyperparameters, with a confidence threshold score of 0.5. Acronyms used: Confidence threshold (Conf-thresh).

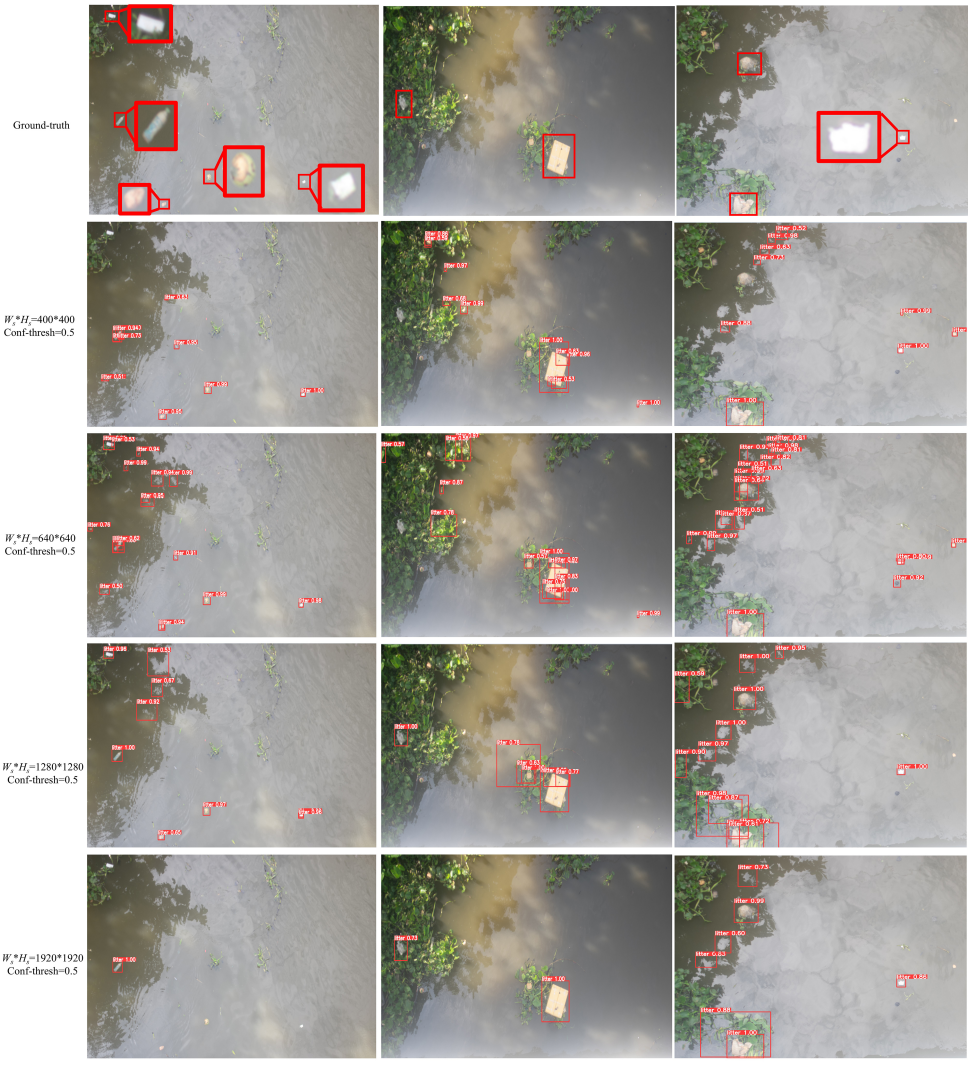


Figure 10.2: Examples of predicted bounding boxes for the Faster R-CNN model with the SAHI method on the $\text{Test}_{\text{Binh Loi}}$ subset. We used the SSL method to develop the Faster R-CNN model, that was fine-tuned on the $\text{Train}_{100\%}$ subset. During inference, we used various W_s and H_s hyperparameters, with a confidence threshold score of 0.5. Acronyms used: Confidence threshold (Conf-thresh).

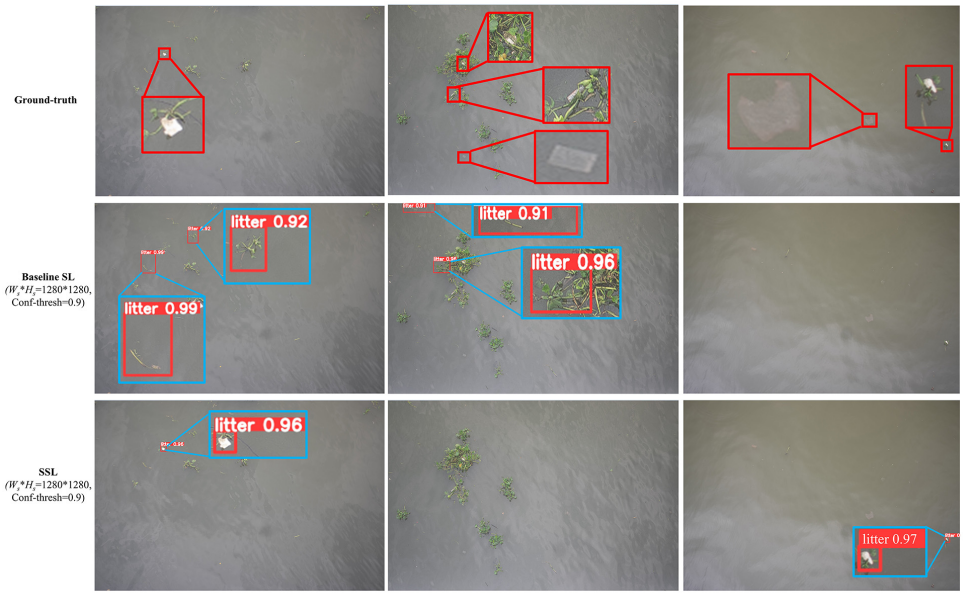


Figure 10.3: Example of predicted bounding boxes for the Faster R-CNN on the $\text{Test}_{\text{Thu Thiem}}$ subset and using the SSL and baseline SL methods. The models were fine-tuned on the $\text{Train}_{100\%}$ subset. The baseline method wrongly detects water hyacinth as litter in (a) and (b), and fails to correctly detect all litter items in (a), (b) and (c). The SSL method correctly detects two litter items in (a) and (c), while fails to detect two submerged items in (b) and (c), and two items entrapped in water hyacinth in (b). Ground-truth litter is shown in red bounding boxes in the top row. Acronyms used: Confidence threshold (Conf-thresh).

Figure 10.4: Example of predicted bounding boxes for the Faster R-CNN on the $\text{Test}_{\text{Binh Loi}}$ subset and using the SSL and baseline SL methods. The models were fine-tuned on the $\text{Train}_{100\%}$ subset. The baseline method wrongly detects the reflection of trees (a and c) and water hyacinth (b) as litter, and fails to correctly detect all litter items in (a), (b) and (c). The SSL method correctly detects four litter items, while fails to detect two items in (a) and (b). Ground-truth litter is shown in red bounding boxes in the top row. Acronyms used: Confidence threshold (Conf-thresh).

BIBLIOGRAPHY

Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F. L., Almeida, D., Altenschmidt, J., Altman, S., Anadkat, S., et al. (2023). Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Ajit, A., Acharya, K., & Samanta, A. (2020). A review of convolutional neural networks. *2020 International Conference on Emerging Trends in Information Technology and Engineering (ic-ETITE)*, 1–5. https://doi.org/10.1109/ic-ETITE47903.2020.049

Akyon, F. C., Altinuc, S. O., & Temizel, A. (2022). Slicing aided hyper inference and fine-tuning for small object detection. *2022 IEEE International Conference on Image Processing (ICIP)*, 966–970.

Akyon, F. C., Cengiz, C., Altinuc, S. O., Cavusoglu, D., Sahin, K., & Eryuksel, O. (2021, November). *SAHI: A lightweight vision library for performing large scale object detection and instance segmentation*. Zenodo. https://doi.org/10.5281/zenodo.5718950

Al-Zawaidah, H., Ravazzolo, D., & Friedrich, H. (2021). Macroplastics in rivers: Present knowledge, issues and challenges. *Environmental Science: Processes & Impacts*, *23*(4), 535–552.

Andriolo, U., Topouzelis, K., van Emmerik, T. H., Papakonstantinou, A., Monteiro, J. G., Isobe, A., Hidaka, M., Kako, S., Kataoka, T., & Gonçalves, G. (2023). Drones for litter monitoring on coasts and rivers: Suitable flight altitude and image resolution. *Marine pollution bulletin*, *195*, 115521.

Arya, D., Maeda, H., Ghosh, S. K., Toshniwal, D., Mraz, A., Kashiyama, T., & Sekimoto, Y. (2020). Transfer learning-based road damage detection for multiple countries. *arXiv preprint arXiv:2008.13101*.

Azzeh, J., Zahran, B., & Alqadi, Z. (2018). Salt and pepper noise: Effects and removal. *JOIV: International Journal on Informatics Visualization*, *2*(4), 252–256.

Bajaj, R., Garg, S., Kulkarni, N., & Raut, R. (2021). Sea debris detection using deep learning: Diving deep into the sea. *2021 IEEE 4th International Conference on Computing, Power and Communication Technologies (GUCON)*, 1–6.

Basu, B., Sannigrahi, S., Sarkar Basu, A., & Pilla, F. (2021). Development of novel classification algorithms for detection of floating plastic debris in coastal waterbodies using multispectral sentinel-2 remote sensing imagery. *Remote Sensing*, *13*(8), 1598.

Battaglia, P. W., Hamrick, J. B., Bapst, V., Sanchez-Gonzalez, A., Zambaldi, V., Malinowski, M., Tacchetti, A., Raposo, D., Santoro, A., Faulkner, R., et al. (2018). Relational inductive biases, deep learning, and graph networks. *arXiv preprint arXiv:1806.01261*.

Battula, S., Kumar, M., Panda, S. K., Rao, U. M., Laveti, G., & Mouli, B. (2020). Online ocean monitoring using edge iot. *Global Oceans 2020: Singapore–US Gulf Coast*, 1–7.

Bellou, N., Gambardella, C., Karantzalos, K., Monteiro, J. G., Canning-Clode, J., Kemna, S., Arrieta-Giron, C. A., & Lemmen, C. (2021). Global assessment of innovative solutions to tackle marine litter. *Nature Sustainability*, *4*(6), 516–524.

Bengio, Y., Courville, A., & Vincent, P. (2013). Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, *35*(8), 1798–1828.

Biermann, L., Clewley, D., Martinez-Vicente, V., & Topouzelis, K. (2020). Finding plastic patches in coastal waters using optical satellite data. *Scientific reports*, *10*(1), 5364.

Blettler, M. C., Abrial, E., Khan, F. R., Sivri, N., & Espinola, L. A. (2018). Freshwater plastic pollution: Recognizing research biases and identifying knowledge gaps. *Water research*, *143*, 416–424.

Bolton, S., Dill, R., Grimaila, M. R., & Hodson, D. (2023). Ads-b classification using multivariate long short-term memory–fully convolutional networks and data reduction techniques. *The Journal of Supercomputing*, *79*(2), 2281–2307.

Borrelle, S. B., Ringma, J., Law, K. L., Monnahan, C. C., Lebreton, L., McGivern, A., Murphy, E., Jambeck, J., Leonard, G. H., Hilleary, M. A., et al. (2020). Predicted growth in plastic waste exceeds efforts to mitigate plastic pollution. *Science*, *369*(6510), 1515–1518.

Broere, S., van Emmerik, T., González-Fernández, D., Luxemburg, W., de Schipper, M., Cózar, A., & van de Giesen, N. (2021). Towards underwater macroplastic monitoring using echo sounding. *Frontiers in earth science*, *9*, 628704.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. (2020). Language models are few-shot learners. *Advances in neural information processing systems*, *33*, 1877–1901.

Caron, M., Misra, I., Mairal, J., Goyal, P., Bojanowski, P., & Joulin, A. (2020). Unsupervised learning of visual features by contrasting cluster assignments. *Advances in neural information processing systems*, *33*, 9912–9924.

Castro-Jiménez, J., González-Fernández, D., Fornier, M., Schmidt, N., & Sempéré, R. (2019). Macro-litter in surface waters from the rhone river: Plastic pollution and loading to the nw mediterranean sea. *Marine Pollution Bulletin*, *146*, 60–66.

Chai, J., Zeng, H., Li, A., & Ngai, E. W. (2021). Deep learning in computer vision: A critical review of emerging techniques and application scenarios. *Machine Learning with Applications*, *6*, 100134.

Chen, G., Zhang, K., Wang, S., & Jia, T. (2024). Phyl v1. 0: A parallel, flexible, and advanced software for hydrological and slope stability modeling at a regional scale. *Environmental Modelling & Software*, *172*, 105882.

Chen, T., Kornblith, S., Norouzi, M., & Hinton, G. (2020). A simple framework for contrastive learning of visual representations. *International conference on machine learning*, 1597–1607.

Cohen, I., Huang, Y., Chen, J., Benesty, J., Benesty, J., Chen, J., Huang, Y., & Cohen, I. (2009). Pearson correlation coefficient. *Noise reduction in speech processing*, 1–4.

Cuturi, M. (2013). Sinkhorn distances: Lightspeed computation of optimal transport. *Advances in neural information processing systems*, *26*.

de Vries, R., Egger, M., Mani, T., & Lebreton, L. (2021). Quantifying floating plastic debris at sea using vessel-based optical data and artificial intelligence. *Remote Sensing*, *13*(17), 3401.

Deng, H., Ergu, D., Liu, F., Ma, B., & Cai, Y. (2021). An embeddable algorithm for automatic garbage detection based on complex marine environment. *Sensors*, *21*(19), 6391.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., & Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. *2009 IEEE conference on computer vision and pattern recognition*, 248–255.

(DHPC), D. H. P. C. C. (2022). DelftBlue Supercomputer (Phase 1).

Doersch, C., Gupta, A., & Efros, A. A. (2015). Unsupervised visual representation learning by context prediction. *Proceedings of the IEEE international conference on computer vision*, 1422–1430.

Dollár, P., & Lin, T.-Y. (2014). Detectron2.

Dong, K., Zhou, C., Ruan, Y., & Li, Y. (2020). Mobilenetv2 model for image classification. *2020 2nd International Conference on Information Technology and Computer Application (ITCA)*, 476–480.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al. (2020). An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

Dwibedi, D., Misra, I., & Hebert, M. (2017). Cut, paste and learn: Surprisingly easy synthesis for instance detection. *Proceedings of the IEEE international conference on computer vision*, 1301–1310.

Everingham, M., Van Gool, L., Williams, C. K., Winn, J., & Zisserman, A. (2010). The pascal visual object classes (voc) challenge. *International journal of computer vision*, *88*, 303–338.

Fallati, L., Polidori, A., Salvatore, C., Saponari, L., Savini, A., & Galli, P. (2019). Anthropogenic marine debris assessment with unmanned aerial vehicle imagery and deep learning: A case study along the beaches of the republic of maldives. *Science of The Total Environment*, *693*, 133581.

Fulton, M., Hong, J., Islam, M. J., & Sattar, J. (2019). Robotic detection of marine litter using deep visual detection models. *2019 international conference on robotics and automation (ICRA)*, 5752–5758.

Garcia-Garin, O., Monleón-Getino, T., López-Brosa, P., Borrell, A., Aguilar, A., Borja-Robalino, R., Cardona, L., & Vighi, M. (2021). Automatic detection and quantification of floating marine macro-litter in aerial images: Introducing a novel deep learning approach connected to a web application in r. *Environmental Pollution*, *273*, 116490.

Gasperi, J., Dris, R., Bonin, T., Rocher, V., & Tassin, B. (2014). Assessment of floating plastic debris in surface water along the seine river. *Environmental pollution*, *195*, 163–166.

Ghiasi, G., Cui, Y., Srinivas, A., Qian, R., Lin, T.-Y., Cubuk, E. D., Le, Q. V., & Zoph, B. (2021). Simple copy-paste is a strong data augmentation method for instance segmentation. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2918–2928.

Gholami, A., Kwon, K., Wu, B., Tai, Z., Yue, X., Jin, P., Zhao, S., & Keutzer, K. (2018). Squeezenext: Hardware-aware neural network design. *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 1638–1647.

Gia, B. T., Khanh, T. B. C., Trong, H. H., Doan, T. T., Do, T., Le, D.-D., & Ngo, T. D. (2024). Enhancing road object detection in fisheye cameras: An effective framework integrating sahi and hybrid inference. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 7227–7235.

Gidaris, S., Singh, P., & Komodakis, N. (2018). Unsupervised representation learning by predicting image rotations. *arXiv preprint arXiv:1803.07728*.

Gonçalves, G., Andriolo, U., Pinto, L., & Duarte, D. (2020). Mapping marine litter with unmanned aerial systems: A showcase comparison among manual image screening and machine learning techniques. *Marine pollution bulletin*, *155*, 111158.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. (2014). Generative adversarial nets. *Advances in neural information processing systems*, *27*.

Goyal, P., Duval, Q., Reizenstein, J., Leavitt, M., Xu, M., Lefaudeux, B., Singh, M., Reis, V., Caron, M., Bojanowski, P., Joulin, A., & Misra, I. (2021). Vissl.

Goyal, P., Duval, Q., Seessel, I., Caron, M., Misra, I., Sagun, L., Joulin, A., & Bojanowski, P. (2022). Vision models are more robust and fair when pretrained on uncurated images without supervision. *arXiv preprint arXiv:2202.08360*.

Granger, R. (2006). Engines of the brain: The computational instruction set of human cognition. *AI Magazine*, *27*(2), 15–15.

Güldenring, R., & Nalpantidis, L. (2021). Self-supervised contrastive learning on agricultural images. *Computers and Electronics in Agriculture*, *191*, 106510.

Guo, J., Wang, Q., & Li, Y. (2021). Semi-supervised learning based on convolutional neural network and uncertainty filter for façade defects classification. *Computer-Aided Civil and Infrastructure Engineering*, *36*(3), 302–317.

Guo, Y., Li, Y., Wang, L., & Rosing, T. (2020). Adafilter: Adaptive filter fine-tuning for deep transfer learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, *34*(04), 4060–4066.

Haberstroh, C. J., Arias, M. E., Yin, Z., & Wang, M. C. (2021). Effects of hydrodynamics on the cross-sectional distribution and transport of plastic in an urban coastal river. *Water Environment Research*, *93*(2), 186–200.

Hadi, Z., Sulaiman, N., Halin, I. A., & Yunus, N. A. M. (2016). Implementation of image enhancement techniques based on intel edison platform. *2016 1st International Conference on Information Technology, Information Systems and Electrical Engineering (ICITISEE)*, 17–20.

He, K., Fan, H., Wu, Y., Xie, S., & Girshick, R. (2020). Momentum contrast for unsupervised visual representation learning. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 9729–9738.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

Hegde, R., Patel, S., Naik, R. G., Nayak, S. N., Shivaprakasha, K., & Bhandarkar, R. (2021). Underwater marine life and plastic waste detection using deep learning and

raspberry pi. *Advances in VLSI, Signal Processing, Power Electronics, IoT, Communication and Embedded Systems: Select Proceedings of VSPICE 2020*, 263–272.

Hong, J., Fulton, M. S., & Sattar, J. (2020). Trashcan 1.0 an instance-segmentation labeled dataset of trash observations.

Hosang, J., Benenson, R., & Schiele, B. (2017). Learning non-maximum suppression. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 4507–4515.

Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. Q. (2017). Densely connected convolutional networks. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 4700–4708.

Huang, H., Lin, L., Tong, R., Hu, H., Zhang, Q., Iwamoto, Y., Han, X., Chen, Y.-W., & Wu, J. (2020). Unet 3+: A full-scale connected unet for medical image segmentation. *ICASSP 2020-2020 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, 1055–1059.

Hurley, R., Braaten, H. F. V., Nizzetto, L., Steindal, E. H., Lin, Y., Clayer, F., van Emmerik, T., Buenaventura, N. T., Eidsvoll, D. P., Økelsrud, A., et al. (2023). Measuring riverine macroplastic: Methods, harmonisation, and quality control. *Water Research*, 119902.

Iandola, F. N., Han, S., Moskewicz, M. W., Ashraf, K., Dally, W. J., & Keutzer, K. (2016). Squeezenet: Alexnet-level accuracy with 50x fewer parameters and < 0.5 mb model size. *arXiv preprint arXiv:1602.07360*.

Im, D., Lee, S., Lee, H., Yoon, B., So, F., & Jeong, J. (2021). A data-centric approach to design and analysis of a surface-inspection system based on deep learning in the plastic injection molding industry. *Processes*, *9*(11), 1895.

Imhof, H. K., Wiesheu, A. C., Anger, P. M., Niessner, R., Ivleva, N. P., & Laforsch, C. (2018). Variation in plastic abundance at different lake beach zones-a case study. *Science of the Total Environment*, *613*, 530–537.

Jabari, O., Ayalew, Y., & Motshegwa, T. (2021). Semi-automated x-ray transmission image annotation using data-efficient convolutional neural networks and cooperative machine learning. *Proceedings of the 2021 5th International Conference on Video and Image Processing*, 205–214.

Jain, S., Smit, A., Ng, A. Y., & Rajpurkar, P. (2021). Effect of radiology report labeler quality on deep learning models for chest x-ray interpretation. *arXiv preprint arXiv:2104.00793*.

Jaiswal, A., Babu, A. R., Zadeh, M. Z., Banerjee, D., & Makedon, F. (2020). A survey on contrastive self-supervised learning. *Technologies*, *9*(1), 2.

Jakovljevic, G., Govedarica, M., & Alvarez-Taboada, F. (2020). A deep learning model for automatic plastic mapping using unmanned aerial vehicle (uav) data. *Remote Sensing*, *12*(9), 1515.

Kaandorp, M. L., Lobelle, D., Kehl, C., Dijkstra, H. A., & van Sebille, E. (2023). Global mass of buoyant marine plastics dominated by large long-lived debris. *Nature Geoscience*, *16*(8), 689–694.

Kako, S., Morita, S., & Taneda, T. (2020). Estimation of plastic marine debris volumes on beaches using unmanned aerial vehicles and image processing based on deep learning. *Marine Pollution Bulletin*, *155*, 111127.

Kataoka, T., & Nihei, Y. (2020). Quantification of floating riverine macro-debris transport using an image processing approach. *Scientific reports*, *10*(1), 2198.

Kingma, D. P., & Welling, M. (2013). Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A. C., Lo, W.-Y., et al. (2023). Segment anything. *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 4015–4026.

Kylili, K., Artusi, A., & Hadjistassou, C. (2021). A new paradigm for estimating the prevalence of plastic litter in the marine environment. *Marine Pollution Bulletin*, *173*, 113127.

Kylili, K., Hadjistassou, C., & Artusi, A. (2020). An intelligent way for discerning plastics at the shorelines and the seas. *Environmental Science and Pollution Research*, *27*(34), 42631–42643.

Kylili, K., Kyriakides, I., Artusi, A., & Hadjistassou, C. (2019). Identifying floating plastic marine debris using a deep learning approach. *Environmental Science and Pollution Research*, *26*, 17091–17099.

Lavitas, L., Redfield, O., Lee, A., Fletcher, D., Eck, M., & Janardhanan, S. (2021). Annotation quality framework-accuracy, credibility, and consistency. *NEURIPS 2021 Workshop for Data Centric AI*, *3*.

Lebreton, L. C., Van Der Zwet, J., Damsteeg, J.-W., Slat, B., Andrady, A., & Reisser, J. (2017). River plastic emissions to the world's oceans. *Nature communications*, *8*(1), 15611.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *nature*, *521*(7553), 436–444.

Li, R., Zeng, X., Sigmund, S. E., Lin, R., Zhou, B., Liu, C., Wang, K., Jiang, R., Freyberg, Z., Lv, H., et al. (2019). Automatic localization and identification of mitochondria in cellular electron cryo-tomography using faster-rcnn. *BMC bioinformatics*, *20*(3), 75–85.

Li, Y., Wang, H., Duan, Y., Xu, H., & Li, X. (2022). Exploring visual interpretability for contrastive language-image pre-training. *arXiv preprint arXiv:2209.07046*.

Liang, M., & Hu, X. (2015). Recurrent convolutional neural network for object recognition. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 3367–3375.

Lin, F., Hou, T., Jin, Q., & You, A. (2021). Improved yolo based detection algorithm for floating debris in waterway. *Entropy*, *23*(9), 1111.

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., & Zitnick, C. L. (2014). Microsoft coco: Common objects in context. *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, 740–755.

Liu, G., Shi, H., Kiani, A., Khreishah, A., Lee, J., Ansari, N., Liu, C., & Yousef, M. M. (2021a). Smart traffic monitoring system using computer vision and edge computing. *IEEE Transactions on Intelligent Transportation Systems*, *23*(8), 12027–12038.

Liu, R.-p., Li, Z.-z., Liu, F., Dong, Y., Jiao, J.-g., Sun, P.-p., & El-Wardany, R. (2021b). Microplastic pollution in yellow river, china: Current status and research progress of biotoxicological effects. *China Geology*, *4*(4), 585–592.

Liu, X., Zhang, F., Hou, Z., Mian, L., Wang, Z., Zhang, J., & Tang, J. (2021c). Self-supervised learning: Generative or contrastive. *IEEE transactions on knowledge and data engineering*, *35*(1), 857–876.

Liu, Z., Hu, H., Lin, Y., Yao, Z., Xie, Z., Wei, Y., Ning, J., Cao, Y., Zhang, Z., Dong, L., et al. (2022a). Swin transformer v2: Scaling up capacity and resolution. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 12009–12019.

Liu, Z., Mao, H., Wu, C.-Y., Feichtenhofer, C., Darrell, T., & Xie, S. (2022b). A convnet for the 2020s. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 11976–11986.

Loshchilov, I., & Hutter, F. (2016). Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*.

Maharjan, N., Miyazaki, H., Pati, B. M., Dailey, M. N., Shrestha, S., & Nakamura, T. (2022). Detection of river plastic using uav sensor data and deep learning. *Remote Sensing*, *14*(13), 3049.

Malli, R. C. (2019). Keras-squeezenet.

Marin, I., Mladenović, S., Gotovac, S., & Zaharija, G. (2021). Deep-feature-based approach to marine debris classification. *Applied Sciences*, *11*(12), 5644.

Martin, C., Almahasheer, H., & Duarte, C. M. (2019). Mangrove forests as traps for marine litter. *Environmental Pollution*, *247*, 499–508.

Martin, C., Parkes, S., Zhang, Q., Zhang, X., McCabe, M. F., & Duarte, C. M. (2018). Use of unmanned aerial vehicles for efficient beach litter monitoring. *Marine pollution bulletin*, *131*, 662–673.

Martin, C., Zhang, Q., Zhai, D., Zhang, X., & Duarte, C. M. (2021). Enabling a large-scale assessment of litter along saudi arabian red sea shores by combining drones and machine learning. *Environmental Pollution*, *277*, 116730.

McCann, E., Li, L., Pangle, K., Johnson, N., & Eickholt, J. (2018). An underwater observation dataset for fish classification and fishery assessment. *Scientific data*, *5*(1), 1–8.

Meijer, L. J., van Emmerik, T., van der Ent, R., Schmidt, C., & Lebreton, L. (2021). More than 1000 rivers account for 80% of global riverine plastic emissions into the ocean. *Science Advances*, *7*(18), eaaz5803.

Mifdal, J., Longépé, N., & Rußwurm, M. (2021). Towards detecting floating objects on a global scale with learned spatial features using sentinel 2. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 285–293.

Miller, J. D., Arasu, V. A., Pu, A. X., Margolies, L. R., Sieh, W., & Shen, L. (2022). Self-supervised deep learning to enhance breast cancer detection on screening mammography. *arXiv preprint arXiv:2203.08812*.

Misra, I., & Maaten, L. v. d. (2020). Self-supervised learning of pretext-invariant representations. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 6707–6717.

Motamedi, M., Sakharnykh, N., & Kaldewey, T. (2021). A data-centric approach for training deep neural networks with less data. *arXiv preprint arXiv:2110.03613*.

Moy, K., Neilson, B., Chung, A., Meadows, A., Castrence, M., Ambagis, S., & Davidson, K. (2018). Mapping coastal marine debris using aerial imagery and spatial analysis. *Marine pollution bulletin*, *132*, 52–59.

Munari, C., Scoponi, M., Sfriso, A. A., Sfriso, A., Aiello, J., Casoni, E., & Mistri, M. (2021). Temporal variation of floatable plastic particles in the largest italian river, the po. *Marine Pollution Bulletin*, *171*, 112805.

Musić, J., Kružić, S., Stančić, I., & Alexandrou, F. (2020). Detecting underwater sea litter using deep neural networks: An initial study. *2020 5th International Conference on Smart and Sustainable Technologies (SpliTech)*, 1–6.

Mustafah, Y. M., Azman, A. W., & Ani, M. H. (2013). Object distance and size measurement using stereo vision system. *Advanced Materials Research*, *622*, 1373–1377.

Neupane, D., & Seok, J. (2020). A review on deep learning-based approaches for automatic sonar target recognition. *Electronics*, *9*(11), 1972.

Nguyen, T.-H., & Dang, M. (2024). Automated marine litter investigation for underwater images using a zero-shot pipeline. *Environmental Modelling & Software*, *177*, 106065.

Noroozi, M., & Favaro, P. (2016). Unsupervised learning of visual representations by solving jigsaw puzzles. *European conference on computer vision*, 69–84.

Oquab, M., Darcet, T., Moutakanni, T., Vo, H., Szafraniec, M., Khalidov, V., Fernandez, P., Haziza, D., Massa, F., El-Nouby, A., et al. (2023). Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*.

Padilla, R., Netto, S. L., & Da Silva, E. A. (2020). A survey on performance metrics for object-detection algorithms. *2020 international conference on systems, signals and image processing (IWSSIP)*, 237–242.

Pan, S. J., & Yang, Q. (2009). A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, *22*(10), 1345–1359.

Panwar, H., Gupta, P., Siddiqui, M. K., Morales-Menendez, R., Bhardwaj, P., Sharma, S., & Sarker, I. H. (2020). Aquavision: Automating the detection of waste in water bodies using deep transfer learning. *Case Studies in Chemical and Environmental Engineering*, *2*, 100026.

Papakonstantinou, A., Batsaris, M., Spondylidis, S., & Topouzelis, K. (2021). A citizen science unmanned aerial system data acquisition protocol and deep learning techniques for the automatic detection and mapping of marine litter concentrations in the coastal zone. *Drones*, *5*(1), 6.

Pham, H. H., Le, T. T., Tran, D. Q., Ngo, D. T., & Nguyen, H. Q. (2021). Interpreting chest x-rays via cnns that exploit hierarchical disease dependencies and uncertainty labels. *Neurocomputing*, *437*, 186–194.

Politikos, D. V., Fakiris, E., Davvetas, A., Klampanos, I. A., & Papatheodorou, G. (2021). Automatic detection of seafloor marine litter using towed camera images and deep learning. *Marine Pollution Bulletin*, *164*, 111974.

Proença, P. F., & Simoes, P. (2020). Taco: Trash annotations in context for litter detection. *arXiv preprint arXiv:2003.06975*.

Punjani, A., & Fleet, D. J. (2021). 3d variability analysis: Resolving continuous flexibility and discrete heterogeneity from single particle cryo-em. *Journal of structural biology*, *213*(2), 107702.

Putra, F. F., & Prabowo, Y. D. (2021). Low resource deep learning to detect waste intensity in the river flow. *Bulletin of Electrical Engineering and Informatics*, *10*(5), 2724–2732.

Qin, X., Luo, X., Wu, Z., & Shang, J. (2021). Optimizing the sediment classification of small side-scan sonar images based on deep learning. *IEEE Access*, *9*, 29416–29428.

Recht, B., Roelofs, R., Schmidt, L., & Shankar, V. (2018). Do cifar-10 classifiers generalize to cifar-10? *arXiv preprint arXiv:1806.00451*.

Reddy, Y., Viswanath, P., & Reddy, B. E. (2018). Semi-supervised learning: A brief review. *Int. J. Eng. Technol*, *7*(1.8), 81.

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 779–788.

Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, *28*, 91–99.

Rocamora, C., Puerto, H., Abadía, R., Brugarolas, M., Martínez-Carrasco, L., & Cordero, J. (2021). Floating debris in the low segura river basin (spain): Avoiding litter through the irrigation network. *Water*, *13*(8), 1074.

Roebroek, C. T., Laufkötter, C., González-Fernández, D., & van Emmerik, T. (2022). The quest for the missing plastics: Large uncertainties in river plastic export into the sea. *Environmental pollution*, *312*, 119948.

Roy, D., Pagliara, S., & Palermo, M. (2021). Experimental analysis of structures for trapping sars-cov-2-related floating waste in rivers. *Water*, *13*(6), 771.

Ruf, P., Madan, M., Reich, C., & Ould-Abdeslam, D. (2021). Demystifying mlops and presenting a recipe for the selection of open-source tools. *Applied Sciences*, *11*(19), 8861.

Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L.-C. (2018). Mobilenetv2: Inverted residuals and linear bottlenecks. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 4510–4520.

Schmidt, C., Krauth, T., & Wagner, S. (2017). Export of plastic debris by rivers into the sea. *Environmental science & technology*, *51*(21), 12246–12253.

Schreyers, L., Van Emmerik, T., Nguyen, T. L., Castrop, E., Phung, N.-A., Kieu-Le, T.-C., Strady, E., Biermann, L., & van Der Ploeg, M. (2021). Plastic plants: The role of water hyacinths in plastic transport in tropical rivers. *Frontiers in Environmental Science*, *9*, 686334.

Schreyers, L. J., Bui, K., van Emmerik, T., Biermann, L., Uijlenhoet, R., Nguyen, H. Q., & van der Ploeg, M. J. (2023). Discontinuity in fluvial plastic transport increased by floating vegetation. *Authorea Preprints*.

Shorten, C., & Khoshgoftaar, T. M. (2019). A survey on image data augmentation for deep learning. *Journal of big data*, *6*(1), 1–48.

Singh, B., & Davis, L. S. (2018). An analysis of scale invariance in object detection snip. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 3578–3587.

Singvi, M., Dasgupta, A., & Routray, A. (2012). A real time algorithm for detection of spectacles leading to eye detection. *2012 4th International Conference on Intelligent Human Computer Interaction (IHCI)*, 1–6.

Sit, M., Demiray, B. Z., Xiang, Z., Ewing, G. J., Sermet, Y., & Demir, I. (2020). A comprehensive review of deep learning applications in hydrology and water resources. *Water Science and Technology*, *82*(12), 2635–2670.

Song, K., Jung, J.-Y., Lee, S. H., & Park, S. (2021). A comparative study of deep learning-based network model and conventional method to assess beach debris standing-stock. *Marine Pollution Bulletin*, *168*, 112466.

Subramanian, M., Shanmugavadivel, K., & Nandhini, P. (2022). On fine-tuning deep learning models using transfer learning and hyper-parameters optimization for disease identification in maize leaves. *Neural Computing and Applications*, 1–18.

Sun, C., Shrivastava, A., Singh, S., & Gupta, A. (2017). Revisiting unreasonable effectiveness of data in deep learning era. *Proceedings of the IEEE international conference on computer vision*, 843–852.

Sun, X., Gu, J., & Sun, H. (2021). Research progress of zero-shot learning. *Applied Intelligence*, *51*, 3600–3614.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. (2016). Rethinking the inception architecture for computer vision. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2818–2826.

Tang, Z., Tian, E., Wang, Y., Wang, L., & Yang, T. (2020). Nondestructive defect detection in castings by using spatial attention bilinear convolutional neural network. *IEEE Transactions on Industrial Informatics*, *17*(1), 82–89.

Tasseron, P., Zinsmeister, H., Rambonnet, L., Hiemstra, A.-F., Siepman, D., & van Emmerik, T. (2020). Plastic hotspot mapping in urban water systems. *Geosciences*, *10*(9), 342.

Tharani, M., Amin, A. W., Rasool, F., Maaz, M., Taj, M., & Muhammad, A. (2021). Trash detection on water channels. *Neural Information Processing: 28th International Conference, ICONIP 2021, Sanur, Bali, Indonesia, December 8–12, 2021, Proceedings, Part I 28*, 379–389.

Thiagarajan, S., & Satheesh Kumar, G. (2019). Machine learning for beach litter detection. *Machine intelligence and signal analysis*, 259–266.

Tian, M., Li, X., Kong, S., Yu, J., et al. (2021). Pruning-based yolov4 algorithm for underwater gabage detection. *2021 40th Chinese Control Conference (CCC)*, 4008–4013.

Tramoy, R., Gasperi, J., Colasse, L., Silvestre, M., Dubois, P., Noûs, C., & Tassin, B. (2020). Transfer dynamics of macroplastics in estuaries–new insights from the seine estuary: Part 2. short-term dynamics based on gps-trackers. *Marine Pollution Bulletin*, *160*, 111566.

Valdenegro-Toro, M. (2016). Submerged marine debris detection with autonomous underwater vehicles. *2016 International Conference on Robotics and Automation for Humanitarian Applications (RAHA)*, 1–7.

van Calcar, C. v., & van Emmerik, T. v. (2019). Abundance of plastic debris across european and asian rivers. *Environmental Research Letters*, *14*(12), 124051.

van Emmerik, T., de Lange, S., Frings, R., Schreyers, L., Aalderink, H., Leusink, J., Begemann, F., Hamers, E., Hauk, R., Janssens, N., et al. (2022a). Hydrology as a driver of floating river plastic transport. *Earth's Future*, *10*(8), e2022EF002811.

van Emmerik, T., Kieu-Le, T.-C., Loozen, M., van Oeveren, K., Strady, E., Bui, X.-T., Egger, M., Gasperi, J., Lebreton, L., Nguyen, P.-D., et al. (2018a). A methodology to

characterize riverine macroplastic emission into the ocean. *Frontiers in Marine Science*, *5*, 372.

van Emmerik, T., Kieu-Le, T.-C., Loozen, M., Van Oeveren, K., Strady, E., Bui, X.-T., Egger, M., Gasperi, J., Lebreton, L., Nguyen, P.-D., et al. (2018b). A methodology to characterize riverine macroplastic emission into the ocean. *Frontiers in Marine Science*, *5*, 372.

van Emmerik, T., Mellink, Y., Hauk, R., Waldschläger, K., & Schreyers, L. (2022b). Rivers as plastic reservoirs. *Frontiers in Water*, *3*, 786936.

van Emmerik, T., & Schwarz, A. (2020). Plastic debris in rivers. *Wiley Interdisciplinary Reviews: Water*, *7*(1), e1398.

van Emmerik, T., Seibert, J., Strobl, B., Etter, S., Den Oudendammer, T., Rutten, M., bin Ab Razak, M. S., & van Meerveld, I. (2020). Crowd-based observations of riverine macroplastic pollution. *Frontiers in earth science*, *8*, 298.

van Emmerik, T., Strady, E., Kieu-Le, T.-C., Nguyen, L., & Gratiot, N. (2019a). Seasonality of riverine macroplastic transport. *Scientific reports*, *9*(1), 13549.

van Emmerik, T., Tramoy, R., Van Calcar, C., Alligant, S., Treilles, R., Tassin, B., & Gasperi, J. (2019b). Seine plastic debris transport tenfolded during increased river discharge. *Frontiers in Marine Science*, *6*, 642.

van Emmerik, T., Vriend, P., & Copius Peereboom, E. (2022c). Roadmap for long-term macroplastic monitoring in rivers. *Frontiers in Environmental Science*, *9*, 802245.

van Emmerik, T. H. M., Janssen, T. W., Jia, T., Bui, T.-K. L., Taormina, R., Nguyen, H.-Q., & Schreyers, L. J. (2024). Water hyacinths as riverine plastic pollution carriers. *EGUsphere*, *2024*, 1–24.

van Emmerik, T. H., Frings, R. M., Schreyers, L. J., Hauk, R., de Lange, S. I., & Mellink, Y. A. (2023). River plastic transport and deposition amplified by extreme flood. *Nature Water*, 1–9.

van Lieshout, C., van Oeveren, K., van Emmerik, T., & Postma, E. (2020). Automated river plastic monitoring using deep learning and cameras. *Earth and space science*, *7*(8), e2019EA000960.

van der Walt, S., Schönberger, J. L., Nunez-Iglesias, J., Boulogne, F., Warner, J. D., Yager, N., Gouillart, E., & Yu, T. (2014). Scikit-image: Image processing in python. *PeerJ*, *2*, e453.

Vriend, P., Roebroek, C. T., & van Emmerik, T. (2020a). Same but different: A framework to design and compare riverbank plastic monitoring strategies. *Frontiers in water*, *2*, 563791.

Vriend, P., Van Calcar, C., Kooi, M., Landman, H., Pikaar, R., & van Emmerik, T. (2020b). Rapid assessment of floating macroplastic transport in the rhine. *Frontiers in Marine Science*, *7*, 10.

Walker, T. R. (2022). Calling for a decision to launch negotiations on a new global agreement on plastic pollution at unea5. 2. *Mar. Pollut. Bull*, *176*(11344710.1016).

Wambugu, N., Chen, Y., Xiao, Z., Tan, K., Wei, M., Liu, X., & Li, J. (2021). Hyperspectral image classification on insufficient-sample and feature learning using deep neural networks: A review. *International Journal of Applied Earth Observation and Geoinformation*, *105*, 102603.

Wang, J., Perez, L., et al. (2017). The effectiveness of data augmentation in image classification using deep learning. *Convolutional Neural Networks Vis. Recognit*, *11*(2017), 1–8.

Watanabe, J.-I., Shao, Y., & Miura, N. (2019). Underwater and airborne monitoring of marine ecosystems and debris. *Journal of Applied Remote Sensing*, *13*(4), 044509.

Wei, Y., Xia, W., Huang, J., Ni, B., Dong, J., Zhao, Y., & Yan, S. (2014). Cnn: Single-label to multi-label. *arXiv preprint arXiv:1406.5726*.

Weideman, E. A., Perold, V., & Ryan, P. G. (2020). Limited long-distance transport of plastic pollution by the orange-vaal river system, south africa. *Science of the Total Environment*, *727*, 138653.

Wenneker, B., & Oosterbaan, L. (2010). Guideline for monitoring marine litter on the beaches in the ospar maritime area. edition 1.0.

Wojke, N., Bewley, A., & Paulus, D. (2017). Simple online and realtime tracking with a deep association metric. *2017 IEEE international conference on image processing (ICIP)*, 3645–3649.

Wolf, M., van den Berg, K., Garaba, S. P., Gnann, N., Sattler, K., Stahl, F., & Zielinski, O. (2020). Machine learning for aquatic plastic litter detection, classification and quantification (aplastic-q). *Environmental Research Letters*, *15*(11), 114042.

Wu, Y.-C., Shih, P.-Y., Chen, L.-P., Wang, C.-C., & Samani, H. (2020). Towards underwater sustainability using rov equipped with deep learning system. *2020 International Automatic Control Conference (CACS)*, 1–5.

Wu, Y., Kirillov, A., Massa, F., Lo, W.-Y., & Girshick, R. (2019). Detectron2.

Xue, B., Huang, B., Chen, G., Li, H., & Wei, W. (2021a). Deep-sea debris identification using deep convolutional neural networks. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, *14*, 8909–8921.

Xue, B., Huang, B., Wei, W., Chen, G., Li, H., Zhao, N., & Zhang, H. (2021b). An efficient deep-sea debris detection method using deep neural networks. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, *14*, 12348–12360.

Yosinski, J., Clune, J., Bengio, Y., & Lipson, H. (2014). How transferable are features in deep neural networks? *Advances in neural information processing systems*, *27*.

Zhang, C., Pan, X., Li, H., Gardiner, A., Sargent, I., Hare, J., & Atkinson, P. M. (2018). A hybrid mlp-cnn classifier for very fine resolution remotely sensed image classification. *ISPRS Journal of Photogrammetry and Remote Sensing*, *140*, 133–144.

Zhang, G., Shang, B., Chen, Y., & Moyes, H. (2017). Smartcavedrone: 3d cave mapping using uavs as robotic co-archaeologists. *2017 International Conference on Unmanned Aircraft Systems (ICUAS)*, 1052–1057.

Zhou, X., Wang, Y., Zhu, Q., Mao, J., Xiao, C., Lu, X., & Zhang, H. (2019). A surface defect detection framework for glass bottle bottom using visual attention model and wavelet transform. *IEEE Transactions on Industrial Informatics*, *16*(4), 2189–2201.

Zhu, X., Vondrick, C., Fowlkes, C. C., & Ramanan, D. (2016). Do we need more training data? *International Journal of Computer Vision*, *119*(1), 76–92.

Curriculum Vitæ

Tianlong Jia

28-12-1994 Born in Jiamusi, China.

Education

2013–2017 Bachelor in Harbours, Water Channels and Coast Engineering
Harbin Engineering University
Harbin, China

2017–2020 Master of Science in Hydraulic Engineering
Huazhong University of Science and Technology
Wuhan, China

2020–2025 PhD Candidate in Sanitary Engineering
Delft University of Technology
Delft, The Netherlands

Experience

2025–present Postdoc researcher
Karlsruhe Institute of Technology
Karlsruhe, Germany

Awards

2019 Chinese National Scholarship
Ministry of Education of the People's Republic of China, China

2019, 2017 First-class Academic Postgraduate Scholarship
Huazhong University of Science and Technology, China

2019 Second-class Zhixing Scholarship
Huazhong University of Science and Technology, China

2019 Merit Student
Huazhong University of Science and Technology, China

List of Publications

Journal papers related to the PhD project

5. **Jia, T.**, Taormina, R., de Vries, R., Kapelan, Z., van Emmerik, T. H. M., Vriend, P. & Okkerman, I. (2025). *A Semi-supervised Learning-Based Framework For Quantifying Litter Fluxes in River Systems* (submitted).

4. **Jia, T.**, de Vries, R., Kapelan, Z., van Emmerik, T. H. M., & Taormina, R. (2024). *Detecting floating litter in freshwater bodies with semi-supervised deep learning.* Water Research, 266, 122405.

3. **Jia, T.**, Vallendar, A. J., de Vries, R., Kapelan, Z., & Taormina, R. (2023). *Advancing deep learning-based detection of floating litter using a novel open dataset.* Frontiers in Water, 5, 1298465. (Jia, T. and Vallendar, A. J. contributed equally)

2. **Jia, T.**, Kapelan, Z., de Vries, R., Vriend, P., Peereboom, E. C., Okkerman, I., & Taormina, R. (2023). *Deep learning for detecting macroplastic litter in water bodies: A review.* Water Research, 231, 119632.

1. van Emmerik, T. H. M., Janssen, T. W., **Jia, T.**, Bui, T. K. L., Taormina, R., Nguyen, H. Q., & Schreyers, L. J. (2025). *River plastic consistently trapped by water hyacinths along the Saigon River* (submitted).

Other publications

6. **Jia, T.**, Yu, J., Sun, A., Wu, Y., Zhang, S., & Peng, Z. (2025). *Semi-supervised learning-based identification of the attachment between sludge and microparticles in wastewater treatment.* Journal of Environmental Management, 375, 124268.

5. **Jia, T.**, Piaggio, A. L., Yu, J., Peng, Z., & de Kreuk, M. K. (2024).*Detecting the interaction between micro particles and biomass in biological wastewater treatment process with Deep Learning method.* Science of the Total Environment, 951, 175813.

4. Wu, Y., Ma, X., Guo, G., **Jia, T.**, Huang, Y., Liu, S., Fan, J. & Wu, X. (2024). *Advancing Deep Learning-Based Acoustic Leak Detection Methods Towards Application for Water Distribution Systems from a Data-centric Perspective.* Water Research, 261, 121999.

3. Chen, G., Zhang, K., Wang, S., & **Jia, T.** (2023). *PHyL v1.0: A parallel, flexible, and advanced software for hydrological and slope stability modeling at a regional scale.* Environmental Modelling and Software, 72, 105882.

2. Yildizli, T., **Jia, T.**, Langeveld, J., & Taormina, R. (2025). *Self-supervised learning approach for automatic sewer detection* (submitted).

1. Wu, Y., **Jia, T.**, Guo, G., Liu, S., & Kapelan, Z. (2025). *Enhancing acoustic leak detection in water distribution systems with semi-supervised deep learning under limited labeled data* (submitted).

CONFERENCE

8. **Jia, T.**, Taormina, R., de Vries, R., Kapelan, Z., van Emmerik, T. H. M., Vriend, P. & Okkerman, I. Quantifying Floating Litter Fluxes with a Semi-Supervised Learning-Based Framework, EGU General Assembly 2025, Vienna, Austria (Poster).

7. **Jia, T.**, de Vries, R., Kapelan, Z., & Taormina, R. Detecting Floating Macroplastic Litter with Semi-Supervised Deep Learning, EGU General Assembly 2024, Vienna, Austria (PICO presentation).

6. **Jia, T.**, de Vries, R., Kapelan, Z., & Taormina, R. Detecting Floating Macroplastic Litter with Semi-Supervised Deep Learning, AGU Annual Meeting 2023, San Francisco, the United States (Poster).

5. Vallendar, A., **Jia, T.**, de Vries, R., Kapelan, Z., & Taormina, R. An open source dataset for Deep Learning-based visual detection of floating macroplastic litter, EGU General Assembly 2023, Vienna, Austria (Oral presentation).

4. **Jia, T.**, de Vries, R., Kapelan, Z., & Taormina, R. A robust deep learning methodology to detect floating macro-plastic litter in rivers, EGU General Assembly 2022, Vienna, Austria (Oral presentation).

3. Yildizli, T., **Jia, T.**, Langeveld, J., & Taormina, R. Self-supervised learning approach for automatic sewer defect detection, 13th Urban Drainage Modelling Conference, Innsbruck, Austria.

2. Yildizli, T., **Jia, T.**, Langeveld, J., & Taormina, R. Self-supervised learning approach for sewer defect detection, 6th International Conference on Water Economics, Statistics and Finance and 10th Leading Edge Conference for Strategic Asset Management (LESAM), Pafos, Cyprus.

1. Yildizli, T., **Jia, T.**, Langeveld, J., & Taormina, R. Self-supervised learning approach for automatic sewer defect detection, 16th International Conference on Urban Drainage 2024, Delft, the Netherlands.