ISBN: 978-981-94-1156-6
2024 the 14th International Workshop on Computer Science and Engineering (WCSE 2024)
doi: 10.18178/wcse.2024.06.002

Real-Time Detection of Floating Debris in Waterways Using YOLOv8

John Paul Q. Tomas [1+], Jan Edgar E. Tupas [2], Miguel T. Soniel [3], Christian Henry Miguel E. Caruz [4] and Darryll B. Babar [5]

[12345] School of Information Technology, Mapúa University, Philippines

Abstract. The urbanization of most waterways has led to a significant amount of pollution in the multiple bodies of water, with mostly floating debris contributing to the environmental threat. The Philippines has contributed to immense discharge of plastic waste into rivers. Furthermore, the prevalence of invasive aquatic plants disrupts the ecosystem of the water bodies and affect economic activities. Traditional methods of monitoring water surface debris are inefficient and resource-demanding. This study proposes the development of an object detection model based on YOLOv8 to identify floating debris on a water surface accurately and in real-time, including garbage and invasive plants. The researchers created a dataset of floating debris. Image preprocessing techniques such as resizing, orientation, resizing, contrast adjustment, and augmentation were done to improve the dataset. The researchers tuned the model in terms of optimizer using Adam and Stochastic Gradient Descent (SGD), and learning rates of 0.01 and 0.001. Upon evaluation, the researchers determined that the model using the SGD optimizer performed better than the model using Adam optimizer in floating debris detection. The researchers further determined that the model performed best when utilizing the best weights from training and a learning rate of 0.001 with the SGD optimizer.

Keywords: Water surface debris, object detection, YOLOv8, floating debris detection, environmental monitoring

1. Introduction

Waterways are essential in the movement of water for purposes such as navigation, irrigation, and drainage. Floating items on the water's surface of rivers degrade slowly or never at all and will pollute the environment [12]. The accumulation of water surface debris over time can impede the normal flow of water in rivers, streams, canals, and other waterways. Recognizing and locating small items on the surface of the water can enhance the capacity to carry out environmental surveillance, reduce water pollution, or guide vessels [11]. Waterways in the Philippines are constantly bombarded with water surface debris such as garbage and invasive plants. The Philippines was responsible for discharging over 356,371 metric tons of plastic waste every year [10]. Due to the lack of resources and research in the Philippines, there is not much information available regarding the monitoring of plastic wastes [9].

Monitoring water surface debris can aid in early detection of pollution, environmental assessment, and detection of debris accumulation. With the advancement of deep learning and computer vision, object detection has become a reliable tool to automate various human tasks. YOLOv8 is a variant of the YOLO algorithms and has proven to be effective and produce better results when applied to various tasks, such as underwater object detection [1], pedestrian tracking [2], road defect detection [4], and UAV object detection [3]. There is a lack of research that focuses on using YOLOv8 for water surface object detection. Thus, the YOLOv8 variant is used and explored in the study.

In this paper, the researchers developed an object detection model for the detection of floating debris on a water surface. The floating debris detected by the researchers were garbage, invasive aquatic plants, branches, and leaves. The researchers leveraged the YOLOv8 detection algorithm since recent object detection works using YOLOv8 yielded promising and research on using the algorithm with floating object detection is limited.

+ Corresponding author. Tel.: + 639178695257
E-mail address: jpqtomas@mapua.edu.ph (J.P.Q. Tomas).

The primary objective of the study is to develop a YOLOv8 model to accurately identify instances of water surface debris within waterways. The specific objectives of the study are the following: (1) To develop a YOLOv8 model that effectively and accurately detects water surface debris using the created dataset; (2) To evaluate and compare the performance of the YOLOv8 model with varying optimizers and learning rates; (3) To assess the detection performance of the YOLOv8 model for each class.

2. Review of Related Literature

This section provides an overview of related research studies and literature conducted by researchers regarding floating debris detection on the water surface through machine learning. The selected and reviewed literature are discussed.

2.1. Negative Impacts of Floating Debris on Waters

In the study of Meijer et al. [10], the authors found that a significant portion of global emissions are caused by small and medium-sized rivers and 98.5% of plastic waste stays trapped in terrestrial settings, where it builds up and gradually contaminates aquatic ecosystems. According to a report by Tekman et al. [18], 88% of 297 marine species were adversely affected by plastic debris due to entanglement, ingestion, smothering, and chemical pollution, resulting in the injury or death of marine life. Invasive aquatic plants also have ecological and economic impacts. Invasive aquatic plants diminish water quality and biodiversity, inhibit the growth of native plants, elevate the risk of extinction for vulnerable species, alter the overall quality of the habitat, disrupt commercial navigation, heighten the occurrence of floods, compromise drinking water quality, and provide a breeding ground for disease-carrying insects [8]. Human resources are primarily used in monitoring water-floating trash, and cleaning floating trash requires significant amounts of both human and material resources but is inefficient [19]. Future developments will favor mechanized salvage methods, with research into floating garbage detection algorithms on the water's surface driving this trend [17].

2.2. YOLO-based Floating Debris Detection

Object detection algorithms are either one-stage or two-stage where two-stage prioritizes accuracy and one-stage focuses on speed [7]. In applying object detection for real-time tasks, one-stage object detection algorithms are utilized. The YOLO algorithm uses end-to-end network structure to achieve real-time requirements with fast detection speed [6]. Studies have focused on utilizing the YOLO series for detecting floating objects on a water surface. In the study of Jiang et al. [19], an improved YOLOv7 with ACanny PConv-ELAN and multi-scale gated attention for adaptive weight allocation (MGA) was proposed to detect floating garbage. The proposed APM-YOLOv7 model outperformed the benchmark YOLOv7, increasing the mean average precision (mAP) by 7.02% and recall by 11.82%. The study of Lin et al. [20] proposed the FMA-YOLOv5 algorithm for detecting floating debris in a waterway. The authors concluded that the proposed model outperforms the YOLOv5s with an increase in mAP by 2.18%. The authors note that the model can be used to monitor floating objects but improvements are needed in the detection of blurred and dense objects. In the study of Zailan et al. [16], the authors utilized YOLOv4 to detect floating debris for a riverine monitoring system. The authors compared the performance of the model with and without transfer learning and concluded that using transfer learning decreases the training time and improves model performance in terms of mAP, f1 score, average IoU, precision, and recall. In a study by Xu et al. [21], a YOLOW algorithm is developed and proposed to automatically detect water objects. With a learning rate of 0.1%, epoch of 300, and batch size of 20, the authors concluded that the proposed YOLOW algorithm outperforms the YOLOv5 variants in terms of precision, recall, and mAP by 6.4%, 2.3%, and 4.3% respectively. To address the issues of traditional image processing techniques for floating debris detection, Qiao et al [13] proposed an improved YOLOv5 model. The authors compared the performance of the proposed model to Faster R-CNN, SSD, YOLOv3, and YOLOv5s, and concluded that the proposed model outperforming the other models. The studies show the effectiveness of YOLO for object detection tasks.

2.3. Comparing YOLOv8 with YOLO Variants

YOLOv8 processes objectness, classification, and regression tasks separately using an anchor-free model with a decoupled head [5]. By allowing each branch to concentrate on its specific task, this design enhances the overall accuracy of the model. In the work of Maity et al. [15], the authors compared the performance of recent YOLO variants, namely YOLOv5, YOLOv7, and YOLOv8 for vehicle detection in terms of precision, recall, and mAP. With the JUVDsi v1 dataset, the YOLOv8 outperformed the YOLOv5 and YOLOv7 by a small margin but all three models performed well with a mAP50 score of 0.755, 0.816, and 8.817 for YOLOv5, YOLOv7, and YOLOv8 respectively. In the IRUVD dataset, all models were very effective but the YOLOv7 model performed the best with a mAP50 score of 0.960 compared with that of YOLOv5 with 0.893 and YOLOv8 with 0.946. The study of Adegun et al. [14] evaluated object detection algorithms for detecting objects in remote sensing statellite images. The authors compared the model performance of Detectron2, YOLOv5, YOLOv6, YOLOv7, and YOLOv8. The YOLOv8, YOLOv7, YOLOv6, YOLOv5, and Detectron2 models obtained a precision score of 68%, 54.5%, 53.2%, 53.4%, and 50% respectively, a recall score of 60%, 46.2%, 47.4%, 49.7%, and 32.7% respectively, a mAP50 score of 43%, 34.1%, 32.1%, 27%, and 16% respectively, and a speed of 0.2ms, 0.3ms, 0.4ms, 0.5ms, and 0.9ms respectively. The authors concluded that the YOLOv8 model had superior performance over other models. YOLOv8 has proven to be effective and produce better results when applied to various tasks, including underwater object detection [1], pedestrian tracking [2], road defect detection [4], and UAV object detection [3]. YOLOv5 and YOLOv7 were commonly used in the reviewed studies relating to floating object detection and YOLOv8 is yet to be utilized. This provides an opportunity for exploring and implementing YOLOv8 for floating debris detection.

3. Methodology

This section provides an overview of the materials and methods used in the study. The researchers collected and curated their own dataset of floating water debris. Image preprocessing techniques were applied to the dataset. The processed data were fed into an object detection model for training and testing. The researchers optimized the model to obtain the best performance. The methodology encompasses the following: data collection, data preprocessing, model training and validation, model testing, and model performance evaluation. The conceptual framework of the study is illustrated in Fig. 1.

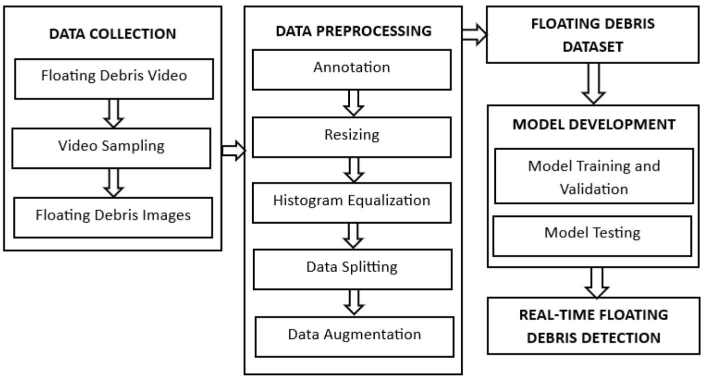

Fig. 1: Conceptual framework of the study

3.1. Data Collection

The researchers curated and created their own dataset for the study. The video of floating debris on the water surface of the Pasig River was taken in Globe Circuit Event Grounds in the Philippines. The 24.1 megapixel CMOS Canon M50 Mk II camera was used to capture the video in 1080p Full HD resolution. To ensure that the floating objects can be clearly seen, the video was taken during daytime. The camera was positioned at a height of 162 cm from the ground, a distance of 10 feet from the water, and at an angle of 45 degrees all throughout the 19-minute session. The result was a 19-minute video that captured floating debris on the water surface of Pasig River in a consistent manner.

3.2. Dataset

The researchers utilized the 19-minute video to create their own dataset. The prominent objects identified were garbage, aquatic plants, branches, and leaves. Hence, these were the classes used in the study. Roboflow was used to sample the video by 1 frame per second. This resulted in a total of 1144 images. The instances of each object class in each image were manually counted and totaled and is shown in Table 1. The dataset had a significant imbalance where branches and invasive aquatic plants were a minority class. To alleviate the impact of this issue, augmentation was performed resulting in a total of 2746 images.

Table 1: Total instances of each class

Debris Type	**Total Instance**
Garbage	7284
Leaves	2830
Branches	169
Aquatic Plants	149
Total	10432

3.3. Data Preprocessing

The researchers used Roboflow in implementing the data preprocessing methods. The researches first annotated the images based on the classes of garbage, leaves, branches, and invasive aquatic plants. The images were auto-oriented and resized to 640 x 640 to ensure better optimization and compatibility with the model. Auto-adjusting contrast through histogram equalization was performed on the images to improve the visibility of details and features in the images. The data was split into training, validation, and testing with a ratio of 70:20:10. Data augmentation was performed on the training dataset to artificially increase its size to help in addressing the dataset imbalance, model generalization, robustness, and reduces overfitting. The augmentation techniques performed are shown in Table 2. The dataset consisted of 2403 images for the train set, 229 images for the valid set, and 114 images for the test set for a total of 2746 images.

Table 2: Data augmentation techniques

Augmentation Technique	**Type**
Flip	horizontal, vertical
90° Rotation	clockwise, counter-clockwise, upside down
Rotation	-15° to +15°
Shear	+-10° horizontal, +-10° vertical
Saturation	-25% to +25%
Brightness	-15% to +15%
Exposure	-10% to +10%
Noise	up to 0.1% pixels

3.4. Model Training and Validation

Google colab with a T4 GPU was used to develop the object detection model. The YOLOv8 object detection algorithm was used in this study for floating debris detection. With the limited dataset, the researchers opted for a YOLOv8 model pre-trained with the COCO dataset as opposed to training from scratch. The model was custom trained with the created dataset using the pre-trained weights with an epoch of 300 and batch size of 16. The researchers used TensorBoard to note on the performance of the model at various stages. The researches also used patience which allows for the early stopping of training when there are no more improvements made. Training was stopped when no improvements were made for the last 50 epochs. After the initial testing and validation, the researchers performed hyperparameter tuning, focusing on optimizer and learning rate. The process was iterated to determine if the performance of the model could be improved.

3.5. Model Testing

The model was tested after validation and tuning. The model was evaluated on data that it had not yet seen during the training process to assess how well it generalizes to new data and estimate its performance in real-world data. The researchers evaluated the testing performance of the model through mean average precision at an IoU threshold of 0.5 (mAP50), and ranging from 0.5 to 0.95 (mAP50-95), precision (P), recall (R), f1 Score, and confusion matrix.

4. Results and Discussion

This section presents the results produced by the YOLOv8 model in detecting floating debris. The effects of varying optimizers and learning rates on model performance is evaluated. The performance of the model in detecting objects per class is also discussed.

4.1. Impact of Optimizers and Learning Rates

The researchers compared the impact of two optimizers and learning rates to the object detection performance of the YOLOv8. The optimizers evaluated were the Adam Optimizer and Stochastic Gradient Descent (SGD). The learning rates assessed were 0.01 and 0.001. Upon completing model training, YOLOv8 provides the best weights for the model, as well as the weights used in the last epoch during model training, so the best and last weights were also compared. The model was pre-trained with the COCO dataset. The model was trained using 300 epochs, batch size of 16, and patience of 50. These parameters were kept constant to focus on the influence of the optimizers and learning rates. Table 3 shows the validation results of the model based on the optimizer, learning rate, and the weights attained from training.

Table 3: Performance of the YOLOv8 model with varying optimizers, weights, and learning rates

Optimizer	Weights	Learning Rate	mAP50	mAP50-95	Precision	Recall	F1 Score
Validation							
Adam	Best	0.01	0.836	0.405	0.840	0.772	0.805
Adam	Best	0.001	0.861	0.453	0.877	0.816	0.845
SGD	Best	0.01	0.860	0.473	0.841	0.831	0.836
SGD	Best	0.001	0.864	0.470	0.854	0.828	0.841
Testing							
Adam	Best	0.01	0.819	0.419	0.819	0.746	0.781
Adam	Best	0.001	0.854	0.454	0.847	0.797	0.821
Adam	Last	0.01	0.005	0.001	0.002	0.094	0.004
Adam	Last	0.001	0.869	0.453	0.888	0.844	0.865
SGD	Best	0.01	0.850	0.470	0.888	0.826	0.856
SGD	Best	0.001	0.872	0.467	0.882	0.855	0.868
SGD	Last	0.01	0.851	0.467	0.879	0.832	0.855
SGD	Last	0.001	0.859	0.462	0.892	0.812	0.850

In the validation, the performance of the model when using the Adam optimizer improves in all metrics when learning rate is reduced to 0.001. Meanwhile the performance of the model using the SGD optimizer varies according to the metrics when learning rate is reduced. Although the mAP50, precision, and the f1 score of the model is improved, the map50-95 and recall of the model is reduced. In testing, the SGD optimizer obtained better performance scores than the Adam optimizer based on the evaluation metrics. The model with the SGD optimizer with a learning rate of 0.001 and with the best weights obtained from the training obtained the highest scores in mAP50 with 0.872 or 87.2%, recall with 0.855 or 85.5%, and f1 score with 0.868 or 86.8%. But it achieved higher scores in precision using the last epoch weights from training at 0.888 or 88.8%, while it achieved a higher Map50-95 score with the best weights and a learning rate of 0.01 at 0.470 or 47%. Meanwhile, the model using Adam optimizer had best performance with a learning rate of 0.001 and the last epoch weights from training with a mAP50 score of 0.869 or 86.9%, precision score of 0.888 or 88.8%, recall score of 0.844 or 84.4%, and f1 score of 0.865 or 86.5%. It can be observed that decreasing the learning rate improves the model, in terms of the mAP50, mAP50-95, precision, and recall when using the Adam optimizer, whereas improvements in the SGD optimizer are observed in mAP50, recall, and f1 scores when using the best weights from training. The model using SGD outperforms the model using Adam optimizer in almost all cases when both optimizers use the same parameters. It is important to note that the model was run with 300 epochs but performed early stopping when no improvements from the latest 50 epochs are identified. As such, the model training stopped at 170 epochs when using the SGD optimizer 0.01 learning rate and best results were observed at epoch 120. With SGD and a learning rate of 0.001, training stopped at 123 epochs where best results were observed in epoch 73. With Adam and a learning rate of 0.01, training completed with 151 epochs and best results were observed at epoch 101. Lastly, with Adam and a learning rate of 0.001, training was stopped after 152 epochs where the best results were obtained in epoch 102.

4.2. Evaluating the Object Detection per Class

Table 4: Model performance in detecting each class

Class	Best Adam (lr=0.01)					Best SGD (lr=0.01)				
	mAP50	mAP50-95	Precision	Recall	F1 Score	mAP50	mAP50-95	Precision	Recall	F1 Score
Aquatic Plant	0.995	0.634	0.762	1	0.865	0.995	0.749	1	0.95	0.974
Branch	0.936	0.489	0.929	0.812	0.867	0.959	0.534	0.910	1	0.953
Garbage	0.568	0.222	0.791	0.445	0.570	0.639	0.246	0.806	0.558	0.659
Leaf	0.778	0.331	0.796	0.726	0.759	0.807	0.349	0.834	0.797	0.815
Class	Best Adam (lr=0.001)					Best SGD (lr=0.001)				
	mAP50	mAP50-95	Precision	Recall	F1 Score	mAP50	mAP50-95	Precision	Recall	F1 Score
Aquatic Plant	0.995	0.680	0.938	1	0.968	0.995	0.667	1	0.994	0.997
Branch	0.939	0.553	0.830	0.875	0.852	0.991	0.605	0.941	1	0.970
Garbage	0.633	0.236	0.786	0.507	0.616	0.660	0.252	0.760	0.594	0.667
Leaf	0.848	0.348	0.833	0.805	0.819	0.841	0.342	0.828	0.832	0.830
Class	Last Adam (lr=0.001)					Last SGD (lr=0.001)				
	mAP50	mAP50-95	Precision	Recall	F1 Score	mAP50	mAP50-95	Precision	Recall	F1 Score
Aquatic Plant	0.995	0.674	0.992	1	0.996	0.995	0.659	1	0.958	0.979
Branch	0.980	0.556	0.941	0.990	0.965	0.984	0.605	0.912	1	0.954
Garbage	0.635	0.235	0.758	0.561	0.645	0.642	0.238	0.811	0.534	0.644
Leaf	0.865	0.347	0.862	0.825	0.843	0.814	0.347	0.846	0.756	0.798

Table 4 shows the performance of the model in detecting each class based on the specific optimizer and learning rate. It can be observed that the model can effectively detect aquatic plants with both optimizers and with varying learning rates. This can be attributed to the size of the aquatic plants, which aids the model in terms of visibility. The model achieves a constant 0.995 or 99.5% mAP50. The model using SGD outperforms the model using Adam in terms of precision, obtaining a constant 1 or 100%. Meanwhile the model using Adam optimizer outperforms the model using SGD in terms of recall, also obtaining a constant 1 or 100%. The model can also effectively detect branches as shown in the high metric scores. The model using the SGD optimizer with a learning rate of 0.001 and with the best weights from training perform the best in distinguishing branches as shown in the obtained metric scores with an mAP50, mAP50-95, precision, recall, and f1 score of 0.991 or 99.1%, 0.60 or 60.5%, 0.941 or 94.1%, 1 or 100%, and 0.970 or 97% respectively. The model struggles to detect garbage, despite the class with the most number of instances in the dataset. Garbage detection peaked its highest at only 0.660 or 66.6% map50 with the model using SGD optimizer with the best weights and learning a rate of 0.001. The model was able to detect leaves better than garbage as shown. The Adam optimizer with the last epoch weights and a learning rate of 0.001 obtained the highest map50 for detecting leaves at 0.865 or 86.5%. In reviewing the images, the annotated garbage instances were small and blended with the water which may have limited the capacity of the model to detect the garbage instances effectively. Furthermore, a 25% confidence rate was implemented to avoid detection of unlikely images. Because of this, tiny garbage instances, as well as other small objects, did not reach the specified confidence interval to be detected. Fig. 2 shows inference images of the model detecting floating debris using the SGD optimizer.

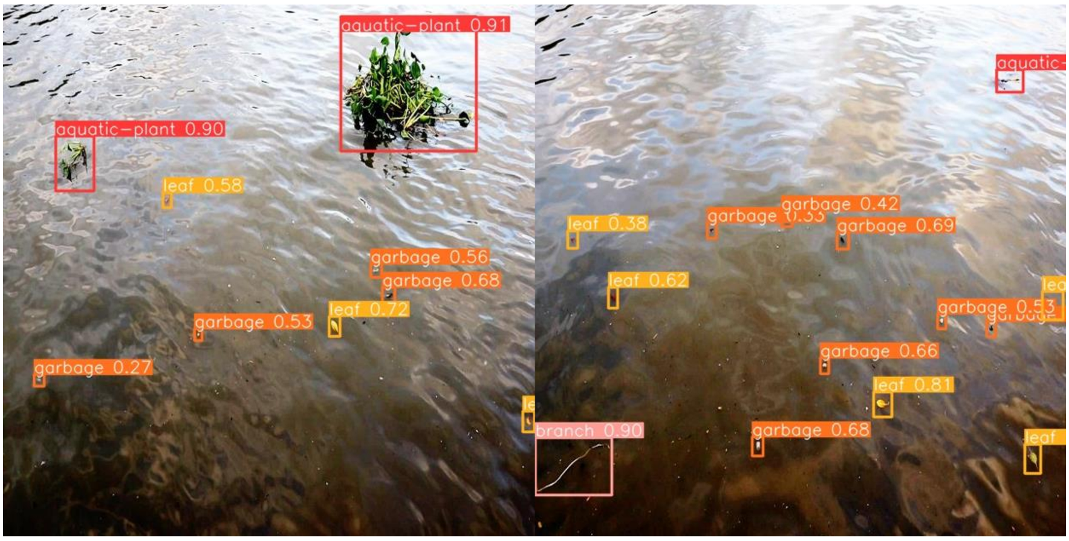

Fig. 2: Floating Debris Detection of YOLOv8 model using SGD Optimizer

	Aquatic Plant	Branch	Garbage	Leaf	Background
Aquatic Plant	25	0	2	0	8
Branch	0	29	0	0	2
Garbage	1	0	623	15	117
Leaf	0	0	25	419	41
Background	1	0	789	128	0

	Aquatic Plant	Branch	Garbage	Leaf	Background
Aquatic Plant	26	0	2	0	5
Branch	0	28	1	0	1
Garbage	0	1	905	18	286
Leaf	0	0	35	477	97
Background	1	0	496	67	0

Fig. 3: Confusion Matrices of YOLOv8 Model using Adam with 0.01 and 0.001 Learning Rates

	Aquatic Plant	Branch	Garbage	Leaf	Background
Aquatic Plant	24	0	2	0	4
Branch	0	29	2	0	2
Garbage	2	0	980	28	359
Leaf	0	0	36	481	100
Background	1	0	419	53	0

	Aquatic Plant	Branch	Garbage	Leaf	Background
Aquatic Plant	26	0	1	0	4
Branch	0	28	0	0	5
Garbage	0	1	980	28	323
Leaf	0	0	30	469	76
Background	1	0	428	65	0

Fig. 4: Confusion Matrices of YOLOv8 Model using SGD with 0.01 and 0.001 Learning Rates

The figures above present the confusion matrices of the model using Adam and SGD with 0.01 and 0.001 learning rates. In Fig. 3, the model accurately detected and classified all instances of aquatic plants. Branches were also identified but some were misclassified as garbage. Leaves were properly detected but some were misclassified as garbage and branch. The model struggles to detect garbage. Although numerous instances were correctly classified, the model misclassifies the other as background, attributed by the tiny instances of the garbage and the influence of the water where the debris floats. In Fig. 4, the model using SGD show better performance. Branches were effectively identified by the model but an instance of the aquatic plant was misclassified, and instances of leaves were misclassified as garbage or background. This can be attributed to the small size and color of leaves which may closely resemble instances of garbage.

Garbage was also misclassified by the model but in lesser instances compared to the model using Adam. The model struggles to differentiate garbage instances with the water which can be attributed to the similarities in color as the water surface of the Pasig River is slightly dark while garbage like plastics is transparent. Furthermore, the waves in the river submerges instances of garbage reducing vision. Overall, the model using SGD optimizer performs better in the object detection task against the Adam optimizer, especially with a learning rate of 0.001 and using the best weights from training.

5. Results and Discussion

5.1. Conclusion

The researchers utilized YOLOv8 to develop a model for floating debris detection on waterways. The researchers created their own floating debris dataset and utilized it to train the object detection model. To improve the dataset, the researchers performed preprocessing and augmentation techniques, improving the quality of the data and increasing the size of the dataset. The researchers acknowledge the limitation of the created dataset due to the imbalance between classes. The researchers successfully compared the performance of the model using SGD and Adam optimizers, and 0.01 and 0.001 learning rates. The researchers also successfully assessed the performance of the model for each surface debris class. The results show that the model using the SGD optimizer performs better overall than the model using the Adam optimizer. The model with the SGD optimizer with a learning rate of 0.001 and with the best weights obtained from the training obtained the highest score in mAP50 with 0.872 or 87.2%.The researchers determined that the model using the SGD optimizer produced less errors in detecting and classifying the floating debris objects. The researchers determined that the capacity of the model to accurately detect floating debris is influenced by the size and color of the object. Tiny objects and those resembling water exhibit low confidence, failing to meet the necessary minimum confidence threshold of 25%. The model with the SGD optimizer, best weights, and a learning rate of 0.001, can effectively be used to detect floating debris on waterways.

5.2. Recommendation

The researchers noted on the limitations of the created dataset, especially due to its significant imbalance. The dataset was collected in a very controlled approach to ensure that images were clear and constant. The dataset also consisted of only four classes. The researchers recommend creating a dataset of floating debris by collecting new data and integrating more classes. It is recommended to collect data with more variations such as taking images from various time of the day, different camera positions and angles, and different lighting. It is also recommended that the classes must be balanced and represented by sufficient images as this will aid in model development. The researchers also recommend exploring more image preprocessing techniques to improve the quality of the dataset, such as using deep learning methods to artificially enhance image quality. In the study, the optimizer and learning rate were the only parameters tuned. It is recommended to fine-tune the object detection model with more hyperparameters such as batch size, anchors, epochs, momentum, and weight decay to improve model performance. It is also recommended to explore different deep learning object detection algorithms for detecting floating debris in waterways and compare the performance with the proposed YOLOv8 model. The model can be still be improved and incorporated in applications to effectively implement real-time detection of floating debris on waterways, especially in areas where floating debris is prevalent.

6. Acknowledgements

The researchers would like to express their gratitude to their advisor Prof. John Paul Tomas for support in this study. His guidance was necessary for this study.

7. References

[1] Minghua Zhang, Zhihua Wang, Wei Song, Danfeng Zhao, and Huijuan Zhao. 2024. Efficient Small-Object Detection in Underwater Images Using the Enhanced YOLOv8 Network. *Applied Sciences 2024*, 14, 1095. DOI:

https://doi.org/10.3390/app14031095

[2] Xin Xiao, and Xinlong Feng. 2023. Multi-Object Pedestrian Tracking Using Improved YOLOv8 and OC-SORT. Sensors 2023, 23, 8439. DOI: https://doi.org/10.3390/s23208439

[3] Gang Wang, Yanfei Chen, Pei An, Hanyu Hong, Jinghu Hu, and Tiange Huang. 2023. UAV-YOLOv8: A Small-Object-Detection Model Based on Improved YOLOv8 for UAV Aerial Photography Scenarios. Sensors 2023, 23, 7190. DOI: https://doi.org/10.3390/s23167190

[4] Xueqiu Wang, Huanbing Gao, Zemeng Jia, and Zijian Li. 2023. BL-YOLOv8: An Improved Road Defect Detection Model Based on YOLOv8. Sensors 2023, 23, 8361. DOI: https://doi.org/10.3390/s23208361

[5] Juan Terven, and Diana Cordova-Esparza. 2024. A Comprehensive Review of YOLO Architectures in Computer Vision: From YOLOv1 to YOLOv8 and YOLO-NAS. Machine Learning and Knowledge Extraction, 5, 4, 1680-1716. DOI: https://doi.org/10.3390/make5040083

[6] Kaiyue Liu, Haitong Tang, Shuang He, Qin Yu, Yulong Xiong, and Nizhuan Wang. Performance Validation of Yolo Variants for Object Detection. In Proceedings of the 2021 International Conference on Bioinformatics and Intelligent Computing, 239-243. DOI: https://doi.org/10.1145/3448748.3448786

[7] Futai Liang, Yan Zhou, Xin Chen, Fengzeng Liu, Chenhao Zhang, and Xinjian Wu. 2021. Review of Target Detection Technology based on Deep Learning. In Proceedings of the 5th International Conference on Control Engineering and Artificial Intelligence (CCEAI '21), 132-135. DOI: https://doi.org/10.1145/3448218.3448234

[8] John Madsen. 2020. Impact of Invasive Aquatic Plants on Aquatic Biology. In: Biology and Control of Aquatic Plants: A Best Management Practices Handbook, 4th edition, Lyn Gettys, William Haller, and David Petty (eds), Aquatic Ecosystem Restoration Foundation (AERF). 1-6.

[9] Mary Louise Gonzaga, M. T. S. Wong, Ariel Blanco, and Jeark Principe. 2021. Utilization of sentinel-2 imagery in the estimation of plastics among floating debris along the coast of manila bay. The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, XLVI-4/W6-2021, 177–184. DOI: https://doi.org/10.5194/isprs-archives-XLVI-4-W6-2021-177-2021

[10] Lourens J. J. Meijer, Tim van Emmerik, Ruud van der Ent, Ruud van der Ent, Christian Schmidt, and Laurent Lebreton. 2021. More than 1000 rivers account for 80% of global riverine plastic emissions into the ocean. Science Advances, 7, 18 (2021), 13 pages. DOI: https://doi.org/10.1126/sciadv.aaz5803

[11] Asif Iqbal, Manuel G. Garcia, Lakshmi Chellappan, and Nicholas Gans. Object Detection and Classification for Small Objects in/on Water. Journal of Electronic Imaging, 31, 3, 18 pages. DOI: https://doi.org/10.1117/1.JEI.31.3.033041

[12] Lili Zhang, Yaxue Wei, Huibin Wang, Yehong Shao, and Jie Shen. 2021. Real-Time Detection of River Surface Floating Object Based on Improved RefineDet. IEEE Access, 9, 81147-81160. DOI: https://doi.org/10.1109/ACCESS.2021.3085348

[13] Guangchao Qiao, Mingxiang Yang, and Hao Wang. 2022. A Detection Approach for Floating Debris Using Ground Images Based on Deep Learning. Remote Sensing, 14, 4161. DOI: https://doi.org/10.3390/rs14174161

[14] Adekanmi Adeyinka Adegun, Jean Vincent Fonou Dombeu, Serestina Viriri, and John Odindi. 2023. State-of-the-Art Deep Learning Methods for Objects Detection in Remote Sensing Satellite Images. Sensors, 23, 5849. DOI: https://doi.org/10.3390/s23135849

[15] Sourajit Maity, Arpan Chakraborty, Pawan Kumar Singh, and Ram Sarkar. 2023. Performance Comparison of Various YOLO Models for Vehicle Detection: An Experimental Study. In: Proceedings of Data Analytics and Management, Lecture Notes in Networks and Systems 787, Abhishek Swaroop, Zdzislaw Polkowski, Sérgio Duarte Correia, and Bal Virdee (eds), 677-684. DOI: https://doi.org/10.1007/978-981-99-6550-2_50

[16] Nur Athirah Zailan, Anis Salwa Mohd Khairuddin, Uswah Khairuddin, and Akira Taguchi. 2021. YOLO-based Network Fusion for Riverine Floating Debris Monitoring System. In Proceedings of the 3rd International Conference on Electrical, Communication and Computer Engineering (ICECCE). DOI: https://doi.org/10.1109/ICECCE52056.2021.9514096

[17] Ning Li, He Huang, Xueyuan Wang, Baohua Yuan, Yi Liu, and Shoukun Xu. 2022. Detection of Floating Garbage on Water Surface Based on PC-Net. Sustainability 2022, 14, 11729. DOI: https://doi.org/10.3390/su141811729

[18] Mine B. Tekman, Bruno Andreas Walther, Corina Peter, Lars Gutow, Melanie Bergmann. 2022. Impacts of plastic pollution in the oceans on marine species, biodiversity and ecosystems. WWF Germany, 1-221. DOI: https://doi.org/10.5281/zenodo.5898684

[19] Zhanjun Jiang, Baijing Wu, Long Ma, Huawei Zhang and Jing Lian. 2023. APM-YOLOv7 for Small-Target Water-Floating Garbage Detection Based on Multi-Scale Feature Adaptive Weighted Fusion. Sensors 2024, 24, 50. DOI: https://doi.org/10.3390/s24010050

[20] Feng Lin, Tian Hou, Qiannan Jin, and Aiju You. 2021. Improved YOLO Based Detection Algorithm for Floating Debris in Waterway. Entropy 2021, 23, 1111. DOI: https://doi.org/10.3390/e23091111

[21] Shufang Xu, Hanqing Tang, Jianna Li, Longbao Wang, Xuejie Zhang, and Hongmin Gao. 2023. A YOLOW Algorithm of Water-Crossing Object Detection. Applied Sciences 2023, 13, 15, 8890. DOI: https://doi.org/10.3390/app13158890