Vol. 06, No. 01, pp. 50 –60 (2025)
ISSN: 2708-0757

JOURNAL OF APPLIED SCIENCE AND TECHNOLOGY TRENDS

www.jastt.org

Hybrid Deep Learning Approach for Marine Debris Detection in Satellite Imagery Using UNet with ResNext50 Backbone

Marada Srinivasa Rao

Department of Information Technology, MVGR College of Engineering (A), Vizianagaram, Andhra Pradesh, India,
srinivas.marada22@gmail.com

Abstract

Marine debris is persistent solid stuff in the water. Oceans include several varieties of organic marine debris, but massive levels of man-made marine trash threaten their biological equilibrium. Manually scanning the ocean for garbage is time-consuming and inefficient, making it uneconomical. Deep learning, which is more efficient than manual methods, is used to detect marine debris in satellite imagery in our work. Deep learning algorithms have been successful in semantic segmentation, however marine debris detection using satellite imagery has been underexplored. The lack of comprehensive marine debris datasets until recently and the complexity of multispectral satellite photos are to blame. Our segmentation method using the UNet architecture and a ResNext50 backbone exceeds the existing state of the art on the Marine Debris Archive Dataset (MARIDA), a dataset of 11 band sentinel 2 Satellite image patches. The hybrid solution combines ResNext50's increased feature extraction with UNet's global and local context preservation, which is crucial in satellite photos of floating bodies due to marine debris' movement pattern. We achieved benchmark mean pixel accuracy, IoU, and F1 scores. We achieved an 88% recall, a 10% improvement over the state of the art, in categorizing marine trash pixels in photos. This work attempts to advance deep learning algorithms for remote sensing and move closer to cleaner oceans.

Keywords: *UNet, ResNext, Encoder Blocks, Decoder Blocks, Marine Debris, ResUNext*
Received: April 8th, 2025 / Revised: June 10th, 2025 / Accepted: June 21st, 2025 / Online: June 24th, 2025

I. INTRODUCTION

In simple terms, marine debris refers to any man-made material that is present in aquatic bodies. These materials are dumped on the shores by humans either intentionally or unintentionally and eventually reache into the deep sea because of tidal currents, winds, and other factors. Marine debris includes different types of objects from industrial-grade materials, microplastics, fishnets, etc. Oceans naturally contain debris such as seeds and plant waste. But these do not necessarily cause any ecological harm and hence are mostly excluded from studies of marine debris. Although they do add hindrance when detecting man-made marine debris. It's important to note that a significant part of marine debris is enormous clusters of plastic waste, which include nanoplastics, microplastics and macroplastics that reach the oceans because of irresponsible dumping by industries and human littering. Apart from these storm drains from factories and container spillage of oil are also active contributors to the increasing marine debris in oceans [1]. The ecological impact of these materials manifests in various ways, mostly harmful. We discuss the environmental impacts of marine debris in the following section.

Various plastics make up most maritime garbage. Marine plastics look like weed, jellyfish, and other organic things due to their shape, size, and color [2]. Many marine animals that feed smaller species mistake marine debris for prey and eat it. Not just aquatic organisms are affected. Albatross and other predatory seabirds like pelicans mistake plastics for prey [3]. The organisms' digestive systems cannot digest these non-biodegradable compounds, resulting in nutritional loss, intestinal blockage, starvation, and death [4]. These trash change the surface of aquatic species' habitats, creating navigational problems that disrupt marine life's biological equilibrium [5]. Marine organisms also consume these wastes. These elements also impact humans. Unvisible microplastics enter fish, crabs, and other species that humans eat [6]. These microplastics can cause hormone changes, intestinal issues, cancer, and more [7]. Marine garbage causes enormous numbers of fish to migrate or

doi: 10.38094/jastt61243

die, which hurts humans economically. Marine debris damages indigenous populations that depend on the sea [8].

A. Challenges in marine debris detection

- Lack of benchmark datasets for marine debris detection.
- Economic limitations of manual and drone-based methods.
- Data hungry nature of deep learning models as the complexity of dataset increases.
- Lack of significant work in working with multispectral satellite images in deep learning domain.

B. Problem Statement

Marine debris is a significant ecological concern considering the effect it has on water based organisms, birds and the significance on humans both in terms of safety and economic harm. While there have been efforts to remove marine debris, there are significant hurdles in the first step of removal itself, that is detection of the said debris. Current approaches to marine debris detection like manual surveys, boats and drones are expensive, slow and hence not scalable. Remote sensing techniques do offer a promising alternative as they are economically viable and offer wide coverage, by leveraging the plethora of satellite data available. But these approaches come with additional complexities as analysis of multispectral data requires specialised expertise in different fields and complicated algorithms to analyse the same. Using deep learning techniques can help to an extent but despite the popularity of deep learning architectures in object detection and segmentation, they have not been utilised to much extent. To bridge this gap, we aim to employ a hybrid approach to utilize the complexity found in multispectral data for efficient feature extraction and semantic segmentation to identify marine debris.

C. Objectives

The objective of this paper is to develop a deep learning based architecture by hybrid approach to advance the detection of marine debris and semantic segmentation of the same from multispectral satellite imagery. More specifically the paper aims to

- Leverage the MARIDA [9] dataset to train machine learning models to advance the detection of marine debris.
- Implement and Test different hybrid architectures for enhanced feature extraction from complex multispectral data.
- Evaluate the proposed model against current state of the art on the MARIDA dataset over metrics like Intersection Over Union IoU, mean Pixel Accuracy, F1 score etc.
- Improve upon the current state of the art by achieving better evaluation metric scores on the test dataset.
- Use the findings of the work to contribute to the advancement of marine debris detection and usage of deep learning for remote sensing in general.

II. LITERATURE SURVEY

Pujie et al. [10] proposed a dual stream real-time detector that synergizes visible and infrared images to enhance detection in challenging scenarios at night. An important contribution in this study was a cross-modal feature enhancement module that uses attention mechanisms to make small target detection by preventing information loss during the processing of images by the network. It also consisted of a three-stage fusion strategy that integrated features across spatial, channel and overall dimensions which ensured that the model delivered robust performance. Although, this study had the drawback of increased computational resources because of the dual stream architecture that was used along with complex fusion strategies.

Kunhao et al [11] proposed using a novel approach of the Multi-Channel Water body detection Network. This was a deep convolutional neural network that integrated the architecture of multi-channel fusion module, Enhanced Atrous Spatial pyramid pooling layers and Space to depth operations to process multispectral data from sentinel 2 imagery. Although the model demonstrated good performance in identifying water bodies, and detecting small water features the study lacked insights on detecting non aquatic objects present in water bodies which have more applications. Uehara et al.[12] proposed a feature extraction technique called Multi Channel Higher order Local Autocorrelation for enhancing object detection using multispectral satellite images. The method is an extension of the conventional Higher Order Local Correlation as it incorporates spectral relationships along with the spatial relationships present in the HLAC. This way it is able to fully utilize the multispectral nature of satellite image to extract data. The study demonstrated that the proposed model effectively captured complex patterns through analysis of pixel intensities within neighbourhood across all the channels together which improved its detection performance thus indicating that cross channel feature extraction is a promising technique. The model outperformed GLCM in identifying golf courses in multi spectral satellite images. However, due to increased computation, it may lead to higher computation time even after training which can affect real time capabilities.

Hu et al. [13] did a critical study on the Spectral interpretations of satellite images. Their work showed that with the mixed band resolutions along with very low sub pixel presence of debris in the images, there were chances for spectral distortions to occur which could lead to misclassification of the pixels if the distortions were treated as sign for floating materials. The study performed simulations and MSI analysis which highlighted the necessity of using both pixel averaging and subtraction techniques while developing algorithms for spectral analysis and to enhance the accuracy of marine debris detection through sensors.

Rubwurm et al. [14] proposed a deep learning-based detector which identifies coastal marine debris using sentinel 2 satellite imagery. The study consisted of a deep segmentation model that outputs pixel level probabilities which depicted marine debris presence. The model was trained on annotated datasets and then tested in real time environments with high likelihood of plastic pollution. The model was successfully able to demonstrate its effectiveness thanks to the data centric approach where negative

examples were sampled extensively. However, due to high reliance on annotated datasets it could make it hard for the model to generalize to different environments. Addressing these limitations is critical to go in the direction of real time applications of deep learning for marine debris detection.

S. S. S. R. Anjaneyulu et al. [15] performed a study on evolution of methods used for satellite image interpretation. The study traced the progress in the field. It highlighted the evolution from the traditional statistical models to advanced machine learning techniques. It also highlights the impact of ML in transforming the field of satellite image analysis especially in sub domains like classification of land covers, monitoring agricultural trends and vegetation patterns and also in urban planning. It also highlighted how with the onset of further advanced techniques like Convolutional Neural networks in deep learning, the tasks of feature extraction which were supposed to be done manually and required lots of resources and time could be automated in these deep learning models reducing need for human intervention. The study also discusses the integration of Synthetic Aperture Radar SAR technology and emphasized its ability to acquire high resolution images independent of weather conditions and whether its daytime or nighttime which is important to monitor dynamic conditions like oil spills, deforestation etc. The study also provided an insight into the various challenges in the field like increased computational demands required to process multispectral and information rich satellite data and lesser reliability due to variability in image quality due to atmospheric conditions and sensor limitations. They propose development of robust algorithms which can be capable of handling these issues which can drastically advance the field of application of geospatial data and analysis and fully utilize these technologies.

An inexpensive technique for finding underwater debris using improved underwater photos is presented by Zhao et al., [16]. Image clarity was improved and object detection accuracy was enhanced using a customized YOLOv8 model (SFD-YOLO) and super-resolution reconstruction (SRR) techniques, particularly the RDN model. An effective method for tracking trash in the ocean, even under harsh conditions, this method attained a high degree of accuracy (91.2% mAP).

Đuraš et al. [17] presents the Seaclear Marine Debris Dataset, the inaugural publicly accessible dataset for underwater debris detection employing instance segmentation and object detection. The dataset comprises 8,610 photos obtained from ROVs in shallow waters, sourced from diverse locations and cameras, annotated for 40 object types, including debris, fauna, flora, and robotic components. The baseline results from Faster RCNN and YOLOv6 underscore the difficulty of generalizing detection models to novel contexts because of domain shift. The dataset seeks to facilitate the creation of more resilient and versatile underwater object identification methods.

A novel approach to coral categorization is put out by Ma et al. [18] utilizing a portable Speed Sea Scanner (SSS-P) in conjunction with point cloud semantic segmentation based on deep learning. Using Structure from Motion (SfM) to generate 3D point clouds and high-resolution coral pictures, the technique successfully recognizes corals in low-light, complicated underwater environments. It provides a useful tool for coral conservation and study, as experiments demonstrate it performs far better than conventional image-based methods.

Shen et al. [19] improves maritime debris detection by the integration of YOLOv7 and attention mechanisms. Of the evaluated models, CBAM attained the highest performance, achieving a 77% F1 score in box identification and a 73% score in mask evaluation. Although the Bottleneck Transformer exhibited inferior overall scores, it identified garbage overlooked by humans and demonstrated superior performance on large items, indicating potential for particular applications.

Nivedita et al. [20] conducted the inaugural investigation utilizing Sentinel-2 satellite optical data to differentiate floating macroplastics from seaweed. Employing a specialized Floating trash Index (FDI) alongside a Naive Bayes machine learning algorithm, researchers detected plastic trash in Brazilian coastal regions with an accuracy of 87.25%. The system also monitored the temporal mobility of plastics and can be utilized to assess marine plastic contamination in various global regions.

Konstantinos Topouzelis et al. [21] have provided a comprehensive overview of detection of marine debris using optical remote sensing. In their study, they have highlighted the need for high resolution multispectral satellite data to monitor extensive marine areas in scale which can offer significant advantages compared to the manual observation methods. The review categorizes different existing detection techniques, and it analyses their methodologies and provides valuable insights into the different approaches used for monitoring floating marine debris. The author further discusses as part of their future scope, foundations of space borne floating marine litter detection systems and show the feasibility along with complexities in accurately identifying and quantifying marine debris from space based platforms. Despite the promising capabilities of remote sensing, the study acknowledges limitations like poor availability of satellite sensor specifications due to confidentiality which can reduce detection accuracy. The authors suggest that refining the methodological processing chain could significantly enhance the future precision of plastic detection from space.

III. METHODOLOGY

A. Dataset

The MARIDA dataset contains Sentinel-2 satellite patch pictures. Each image has a 264*264 resolution and 11 bands, including RGB. Each pixel in the image represents a 10m*10m region. All pixels in the dataset are tagged with one of the 15 classes. For each image, a 256*256 pixel segmentation mask is used. Pixels are labeled with class numbers (1–15). Unannotated pixels are 0. The pixel values in each channel are not between 0-255, unlike in RGB images. Reflectance is their worth. A percentage of that wavelength range's light returned. Both are normalized between 0 and 1. The images are kept in geoTiff format. Popular geographic data format.

Some of the challenge in the dataset were as follows:

- There is also a class imbalance problem in the dataset because the majority of pixels (75%) belong to marine

water. Thus, when designing the architecture we need to accommodate for the imbalance.

- The minority class are extremely small in number compared to the majority class. For example, 40% of the total annotated pixels in the dataset is annotated as sediment water while only 0.41% is annotated as marine debris.
- Further many pixels have been annotated with low confidence due to lack of clarity in cloudy areas or turbulent weather conditions.
- There is noise in some images due to moving ships and turbid water around it. But these are very small in number and would not impact the model significantly.

The following preprocessing steps were done to prepare the data to be fed to the model.

- The images were transformed from CWH format to WHC format for compatibility with pytorch.
- Data augmentation is done to induce diversity in the dataset by using random rotation and flipping.
- Min-max normalisation performed across bands for uniform spread.
- Missing values were replaced with average pixel values for the respective band.
- The classification masks labels were shifted from 0-15 to -1 - 14 for easier accessibility.
- Furthermore, 4 classes Wakes, clouds,Waves, Mixwater all representing water were aggregated into one single Marinewater superclass.
- The dataset was split into 50%, 25%, 25% for training, testing and validation sets respectively.

B. Model Architecture

The model architecture is a hybrid architecture consisting of a mix of two architectures namely UNet [22] and ResNext50 [23].

a) UNet Architecture

The UNet is a popular encoder decoder type architecture that is built of convolution blocks. It consists of a contacting path and an expanding path. The crux of the UNet architecture is that the encoder and decoder, that is, the contracting part and expanding part are connected to each other through skip connections. This helps the model to preserve the spatial information from the encoder layers and also allows better propagation of high resolution features from the earlier layers. This helps to counter the vanishing gradient problem.

b) ResNext Architecture

ResNext is a convolutional neural network architecture that is based on the residual networks architecture. Unlike ResNet that uses two parallel paths,one sequential and the other as skip connections, ResNext uses multiple parallel paths grouped together. The parallel paths allow the network to learn a broader and more diverse set of features. ResNext uses a balance of depth and width to allow the network to find correlation among different features while also not losing the properties of individual features.

c) ResUNext Hybrid Architecture

The ResUNext architecture is a hybrid of the two mentioned models. UNet and ResNext. Specifically, in this paper we use the ResNext50_34d variant. The ResNext is used in the encoder blocks of the architecture as the backbone. This allows the model to successfully extract complex features and also extract spatial relationships. It also helps the model to generalize better by capturing a more diverse set of features.

The Hybrid ResUNext model consists of:

- An input Block consisting of Convolution + BatchNorm+ReLU layer to increase the number of channels from 11 to 64. And a MaxPool layer to reduce the spatial resolution by half. That is from 256*256 to 64*64.
- Three encoder blocks having skip connections to corresponding decoder blocks. With each block consisting of 3, 4 and 6 layers of ResNext groups stacked together. The encoder is described in detail in the following section.
- Three decoder blocks that take the previous blocks along with skip connections' output as input and upsample it to its original resolution step by step.
- Lastly, a Segmentation head that converts the output of the previous layer to a 11*256*256 segmentation map with each channel corresponding to each class.

The encoder block has skip connections in every group. That is, input is added to the outputs in all the layers.

Fig.1, From a bird's eye view suggests, each encoder block doubles the channel width and reduces the spatial width by half. Also, each encoder block's output is concatenated to the decoder block's input which is a basic characteristic of UNet. The decoder blocks increase the spatial dimensions step by step while concatenating its input with output of the corresponding encoder block. Thus extracting the features of higher resolution activation maps too. Lastly a segmentation head covers the activation map to original dimension and a 11*256*256 activation map with logits is returned by the model. Let us look at the blocks in detail now.

Fig.2 depicts an encoder block of the ResNext model. The input of 11*256*256 is passed through a convolution block to increase the channels from 11 to 64 and reduce the spatial dimension by half. Next a maxpool Layer further halves the spatial dimension. So the input to the first encoder block has spatial resolution of 256/4 i.e 64.

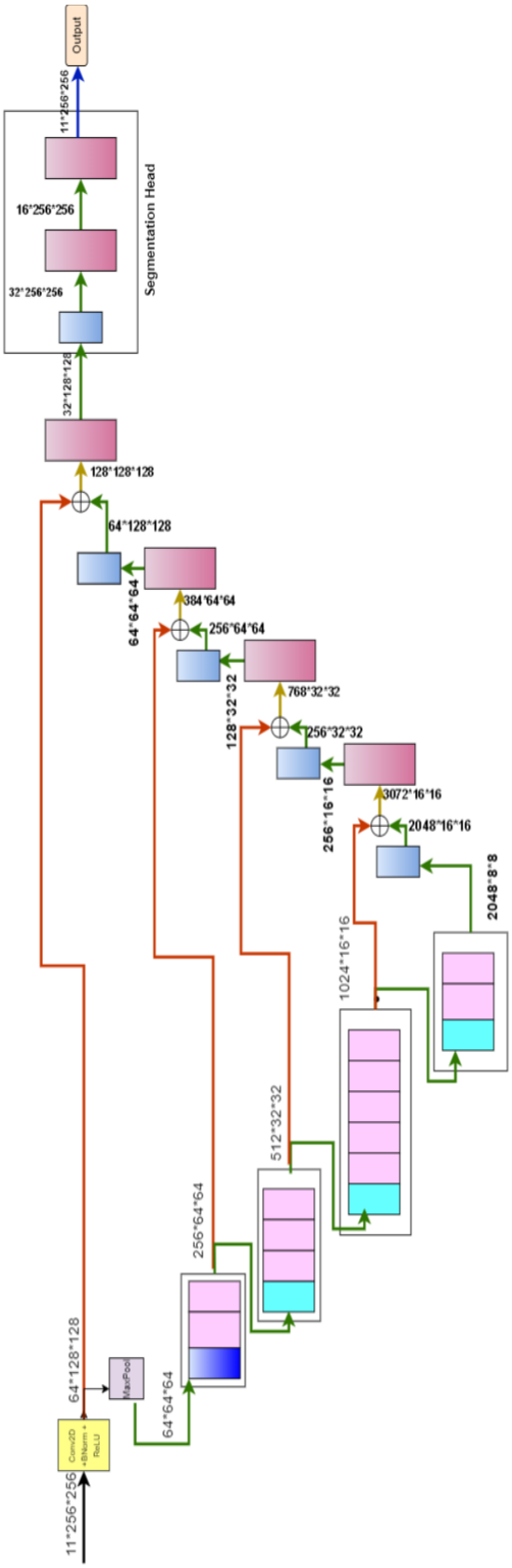

Fig. 1. The hybrid ResUNext architecture depicting the four encoder and encoder blocks along with skip connections and a segmentation head.

The first encoder block has a slightly different structure than the rest of the encoder blocks. The first group in this block takes input from the maxpool layer and first, it increases the channel dimension to 128 and then passes it to the 32 parallel path layers. This layer takes the 128 channels and splits it into 32 parts for each parallel parallel group to extract features from 4 channels each. This parallel extraction allows independent features to be extracted separately thus they are extracted in a more refined manner. Further splitting them into parallel paths is also efficient computationally compared to an equivalent resnet block. The parallel paths are discussed in detail in the next section. Note that spatial resolution doesn't change in this layer in the first encoder block. In all the other blocks, the parallel paths group has a stride of 2 but in this block the stride is 1. Another convolution layer doubles the number of channels. Further, a skip goes from maxpool layer to a convolution block. This convolution block is present to match the maxpool layer input to output of the first group. These are then added as done in residual blocks. Lastly this is passed through a ReLU layer to introduce non linearity. The other two groups in this encoder are similar in structure. They take input from the previous group, reduce the channel width by half, pass it through a parallel path and lastly, double it again to get the same channel width as input. The difference in these blocks is that the input of the previous layer is directly added to output as it is of the same dimension. The output of this block has double the channel and half the spatial resolution of the first convolution block's output.

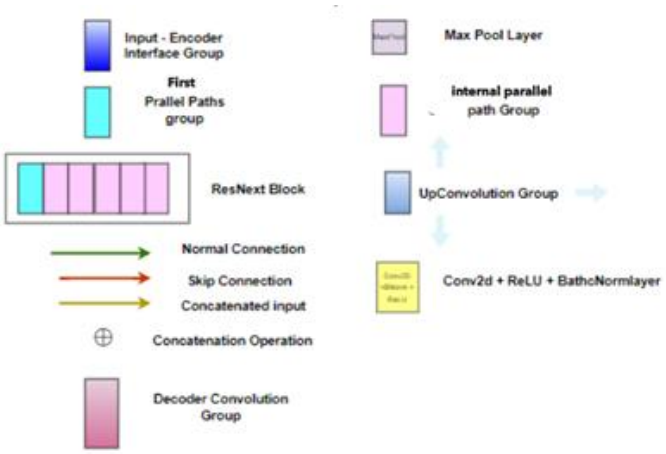

Fig. 2. Legend for architecture diagram

The terminology used for describing diagrams is as follows. There are 4 encoding blocks and 4 decoding blocks. Each encoding block consists of 3,4 and 6 groups respectively. Further each group consists of different layers like Convolution, ReLU etc. In the decoding blocks, each block has 2 groups, which are just an UpConvolution layer for upsampling and a Convolution layer that decreases the number of channels for the concatenated input. The segmentation head consists of an UpConv block to resize the image and two Convolution layers to reduce the number of channels to 11. The output obtained is 11*256*256 segmentation map.

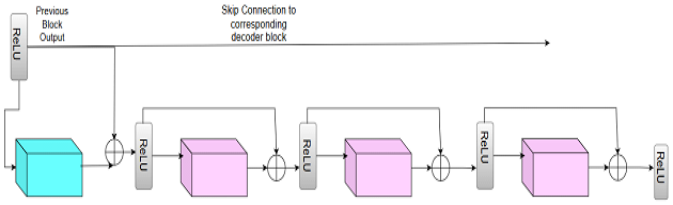

Fig. 3. ResNext Encoder block with 4 groups

The rest of the encoder blocks have a slightly different structure. The first group in those blocks are similar except the first convolution does not change the number of channels in the activation map. And also, the parallel paths have a stride of two in the first group of these (2nd, 3rd ,4th) encoder blocks and hence are responsible for reducing the spatial resolution by half. The last convolution in the initial group is the same as in the first block. It doubles the channel width. And the output is concatenated with the skip connection input after passing it through a convolution layer to match the dimension of input and output, just like in the first block. Next this is passed through a ReLU and the activation map is passed as input to the next layer and also as skip connection to the decoder block for concatenation later. Refer to Fig.3 to get a visual idea of the use of residual connections in the architecture.

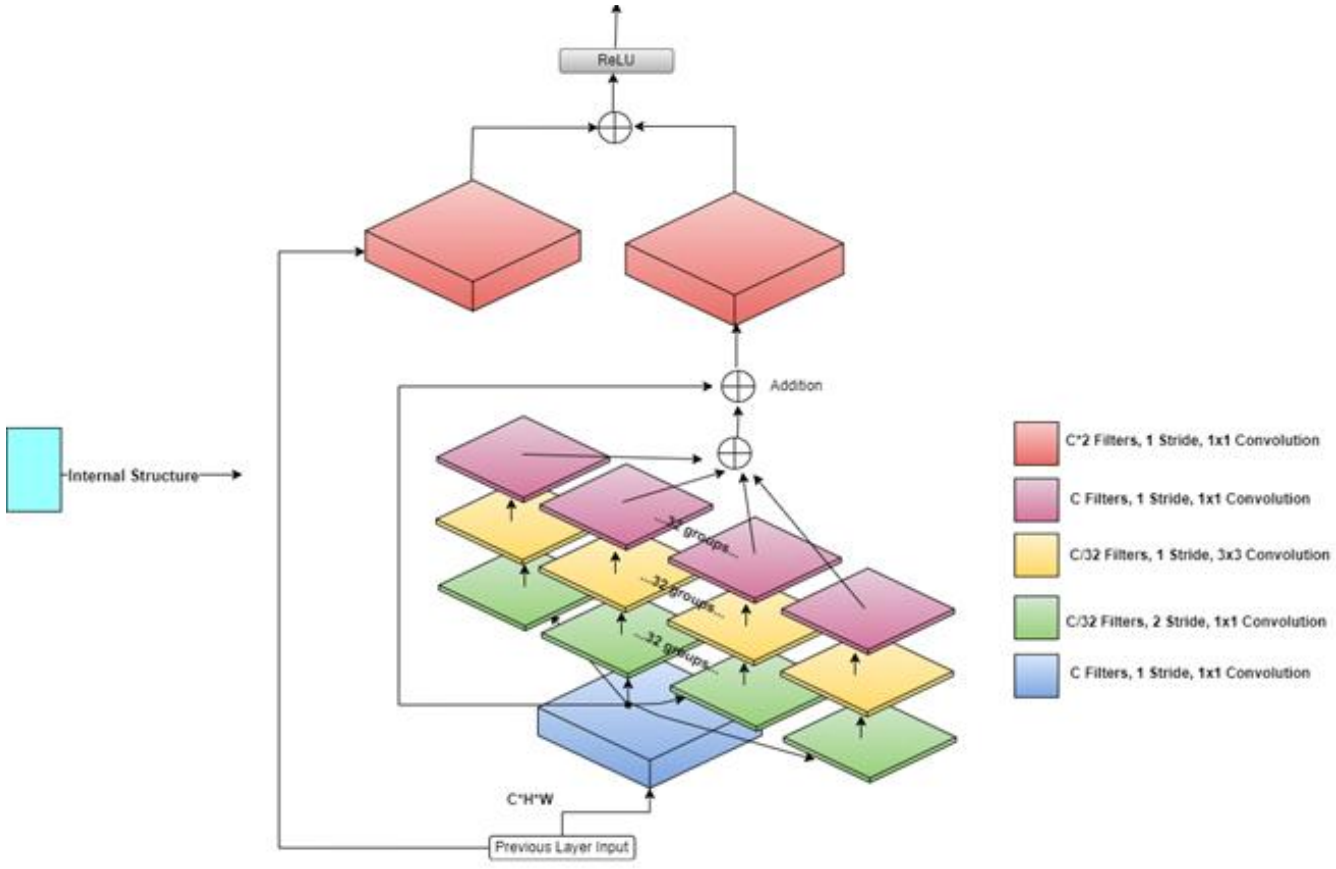

Fig 4. First group of a ResNext block.

Fig.4 gives a much more detailed insight to the parallel paths and its connection to the convolution layer. This entire structure can be called the building blocks of encoders. The green, yellow and cyan blocks represent the 32-group parallel path. The parallel path helps in extraction of features as in the following steps. The first layer of the parallel paths (Green Layers in Fig 4) each take the entire activation map of dimension C*W*H where C refers to channel width and W and H refer to width and height in number of pixels. This layer is a 1*1 convolution and apply C/32 filters on it parallelly. Whose main task is to take the full activation map and though not directly but intuitively it distributes the C channels into 32 groups with each parallel path extracting features of the C/32 channels in that path. Each of the parallel paths processes the input independently, Note that all the encoder blocks' first group except the first encoder block have a stride of 2 while the rest have a stride of 1. Indicating that spatial features extraction happens in this layer.

The next layer is essentially the core feature extraction layer. Each of the 32 parallel path in this layer (yellow block in Fig4) takes the C/32 channel input of their corresponding previous layer and performs a 3x3 convolution operation using C/32 kernels on them. Then it sends these to the third layer in the parallel path. (Cyan color in Fig 4). The final layer in the parallel path (Cyan in Fig4). This is required to restore the original number of channels.This layer consists of 1x1 convolution kernels with stride of 1. Lastly, the output of the parallel paths are added together and the residue is added to it and the output is sent into the next layer.

The decoder blocks are relatively simpler in structure compared to the encoder blocks. Each decoder block has an UpConvolution Layer and a Conv2d layer. The UpConvolution layer takes input from the previous layer and performs a deconvolution operation that essentially produces an activation map of higher resolution from a latent space of lower resolution. It is essentially the reverse of a convolution operation. Once the spatial dimension of the feature map is doubled, it is concatenated with the feature map from the corresponding encoder layer and then they are passed through a convolution layer to reduce the channel width to restore the original image step by step.

Segmentation Head is the final block in the network. It takes in 32*128*128 activation map as input, increases the spatial resolution of the image by a factor of 2*2 by applying a UpConvolution layer and restores the image to 11 channels by using two convolution layers. The final output is a 11*256*256 segmentation map where each channel corresponds to the logits representing the probability of a particular pixel belonging to that class.

d) Loss function

In this paper we have used weighted cross entropy loss [24]. This was used to handle the class imbalance in the dataset. While we experimented with using focal loss [25] for the network, we chose to stick with weighted cross entropy because while focal loss did classify hard to classify examples more accurately, it had problems with classifying easy to classify examples. This could be due to large number of classes to be classified in the dataset and further the imbalance in the dataset in not distributed in one class but several different classes.

$$L = -\frac{1}{m_w}\sum_{i=1}^{m} y_i \cdot \log(\hat{y}_i) \quad (1)$$

Here,
m_w : percentage of pixels of class w.
m: Classes 1 to 11
y_i: Probability of class i (True probability)
$\hat{y}_i$: Probability of class i predicted by model

e) Optimization

The model was trained on the training data consisting of 694 images in geotiff format for 100 epochs. Validation set was used after each epoch to record improvement which contained 328 images. We tested with batch sizes of 5, 20 and 30 and best results were obtained for batch size 20 both in terms of training time and convergence. Learning rate of $2*10^{-4}$ was used. ADAM [26] optimizer was used for faster convergence and its versatility with multiple types of layers.

f) Experimental Setup

The experiment was performed in a google colab environment by using free offered resources. It consists of a Linux kernel OS with a T4 Tpu and 12 GB RAM and 75 GB disk memory. The T4 GPU consists of ~2500 cores that can perform operations parallelly.

Libraries used:

Rasterio: Converting geoTIFF files to numpy format

PyTorch: Primary deep learning framework.

Numpy: Dealing with image matrix data.

Tensorboard: Visualising the evaluation and performance metrics.

g) Training Strategy

The dataset was split into 50%, 25% and 25% percent for training testing and validation sets respectively. The training set consisted of 694 images; the testing set had 359 images and the validation set had 329 images respectively.

Model training was done in parts as training time sometimes exceeded google colab limits. The model weights were saved for every epoch for checkpointing to be able to easily resume training in case of failure in between. The training time was between 35 to 40 minutes.

The following metrics were monitored during training:

a. *Macro Recall:* This is the average recall across all classes. Does not account for class frequency.

$$\text{Macro Recall} = \frac{\sum_{i=1}^{n} \text{Recall}_i}{n} \quad (2)$$

b. *Micro Recall /Accuracy:* This is the global recall calculated for all classes in one go.

$$\text{Micro Recall} = \frac{\text{Total Correct Predictions (TP)}}{\text{Total Instances (TP + FN)}} \quad (3)$$

c. *Weighted Recall:* This is the weighted average recall value across all classes.

$$\text{Weighted Recall} = \frac{\sum_{i=1}^{n}(w_i \cdot \text{Recall}_i)}{\sum_{i=1}^{n} w_i} \quad (4)$$

d. *Micro Precision:* Calculated globally across all instances; emphasizes overall instance-level performance.

e. *Macro Precision:* Average precision across all classes; treats all classes equally, regardless of size.

f. *Weighted Precision:* Weighted average precision, where each class's contribution is proportional to its size.

g. *Intersection Over Union:* This metric is specifically used in image segmentation tasks. It is a measure of the number of pixels correctly classified among total classified pixels.

$$\text{IoU} = \frac{\text{Area of Overlap}}{\text{Area of Union}} = \frac{\text{Intersection}}{\text{Union}} \quad (5)$$

IV. RESULTS AND DISCUSSIONS

This section analyses the training and testing process to provide valuable insights. It discusses the epoch wise training trends, model wise comparison and a discussion on the outputs obtained. Model training time averaged down to 30 minutes. The comparison section clearly demonstrates the superiority of the ResUNext hybrid among other models. For reference we have also tested with UNet with a focal loss variant which gave subpar results. For deeper insights we have also calculated the normalized confusion matrix in the form of heatmap for easy visualisation. Lastly 12 patches from the test dataset along with the outputs given by the model are shown and a discussion is provided on the same.

Fig. 5. Training and testing loss per epoch.

The above Fig.5 suggests that while there was significant loss during the first 20 epochs, there was a sharp decline in the loss after 20- epochs. There are few spikes in epochs 21 and 75, these can be attributed to sudden gradient changes. This can be limited using gradient clipping [27] but since it did not have an affect on the overall loss, we decided not to use clipping. A pattern that can be noticed from this plot is the loss keeps reducing and after it reaches a local minima, the loss again spikes. This is followed by gradual decrease in the loss until it reaches a lower local minima than the last one. This suggests that prolonged training times could improve the accuracy without much risk of overfitting. But for the sake of optimum usage of resources we trained it for 100 epochs. Eventually by the 95th epoch we can also observe that the training loss and testing loss have converged to ~0.15. Further discussion on evaluation metrics are given in the following section.

We recorded the loss per epoch for training and testing along with other metrics like F1 Macro, IoU and Accuracy. The model was able to converge both the training and testing losses which indicates that the model is not prone to overfitting, This suggests that the model has good generalization to unseen data. The oscillatory behaviour of the loss such that the loss decreases then spikes before settling at a lower minimum also suggests that the optimizer is escaping saddle points and poor local minima. This aligns with momentum based optimization techniques like Adam which help to deal with sharp loss landscapes. Based on the monitoring of validation loss we also observed that early stopping would have caused the model to be stuck at local minima. Lastly, after 80 epochs it is observed that the loss reduction is minimal. Based on this we can conclude that beyond 100 epochs training might yield diminishing returns and the loss would almost plateau. Hence in order to comply with the resource constraints we decided to use the model by training it upto 100 epochs.

A. Model Comparison and performance

Below is a graphical illustration showing comparison between different models

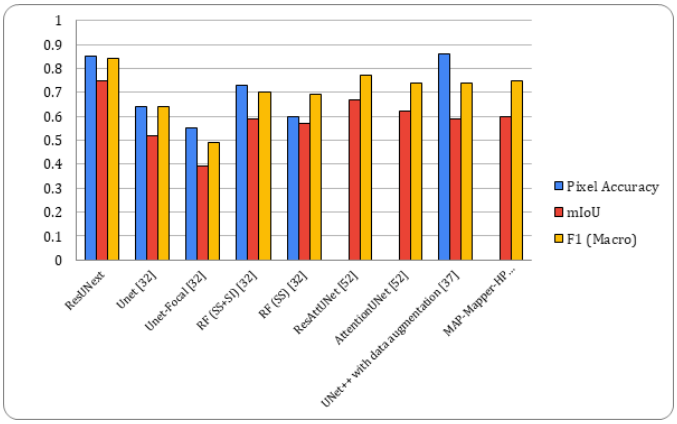

Fig. 6. Comparison of ResUNext with different models.

TABLE I. EVALUATION METRICS OF DIFFERENT MODELS

Model	Pixel Accuracy	mIoU	F1 (Macro)
ResUNext	0.85	0.75	0.84
UNet [9]	0.64	0.52	0.64
UNet-Focal [9]	0.55	0.39	0.49
RF (SS+SI) [9]	0.73	0.59	0.7
RF (SS) [9]	0.6	0.57	0.69
ResAttUNet [29]	-	0.67	0.77
AttentionUNet [33]	-	0.62	0.74
UNet++ with data augmentation [14]	0.86	0.59	0.74
MAP-Mapper-HP [28]	-	0.60	0.75

As we can see in TABLE I, ResUNext performed better by a margin of 21% compared to UNet and 12% compared to the RR(SS+SI) Random forest with spectral signatures and spectral indices. The improvement is further visible if we compare the mIOU values which shows 23% improvement compared to UNet and 16% improvement compared to RF(SS+SI).

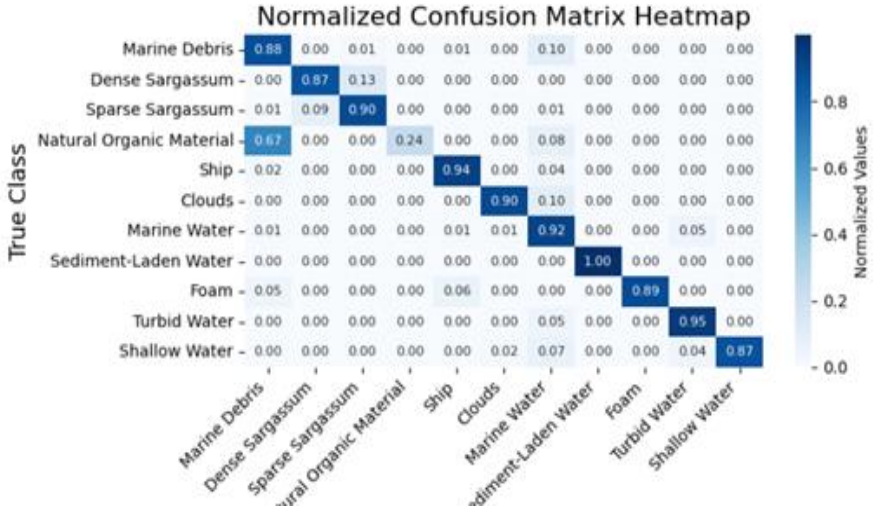

Fig.7. Normalised Confusion matrix depicting predicted and actual classes.

Notice that a large chunk of natural organic material is classified as marine debris shown in Fig 7. This is in line with expected results as both have similar characteristics [29]. The aim of the study was to utilise a hybrid approach to improve the detection of marine debris. Based on the results obtained, it is observed that the hybrid model ResUNext has the highest IoU, F1 and pixel accuracy. Random forest does have metrics that are better than both UNet with focal loss and Vanilla UNet but Random forest, being a machine learning model, requires significant feature extraction. Here the results obtained use Spectral signatures and spectral Indices [30] which require extra computational costs. Further, random forests are in general black boxes as we cannot interpret anything from intermediate results [31]. Whereas neural networks like the UNets can be analysed from between by looking at their activation maps. Secondly, random forests, because of their inherent simple nature, cannot infer spatial relations in images, which are much more practical in Real time than calculating spectral signatures [32]. Because of these, if images are slightly rotated, or transformed while a deep learning based architecture would not have much effect while random forest will fail to classify them correctly.

Observing the confusion matrix, it is noticeable that a large portion of marine debris is being classified as natural organic material. Despite natural organic material being a minority class. This can be attributed to the similar nature of marine debris and organic materials like seaweed [33] etc. Further, though not as visible as the former, dense sargassum and sparse sargassum samples are being classified as each other a few times. This is again, obvious because of their similarity. The Network is quite successful in handling the class imbalance with weighted cross entropy loss. Talking about the epoch wise results it can be observed that a local minima is reached at 80th epoch after which the loss spikes up. The loss converges well enough by the end of the 100th epoch. This indicates low overfitting to training data.

B. *Outputs*

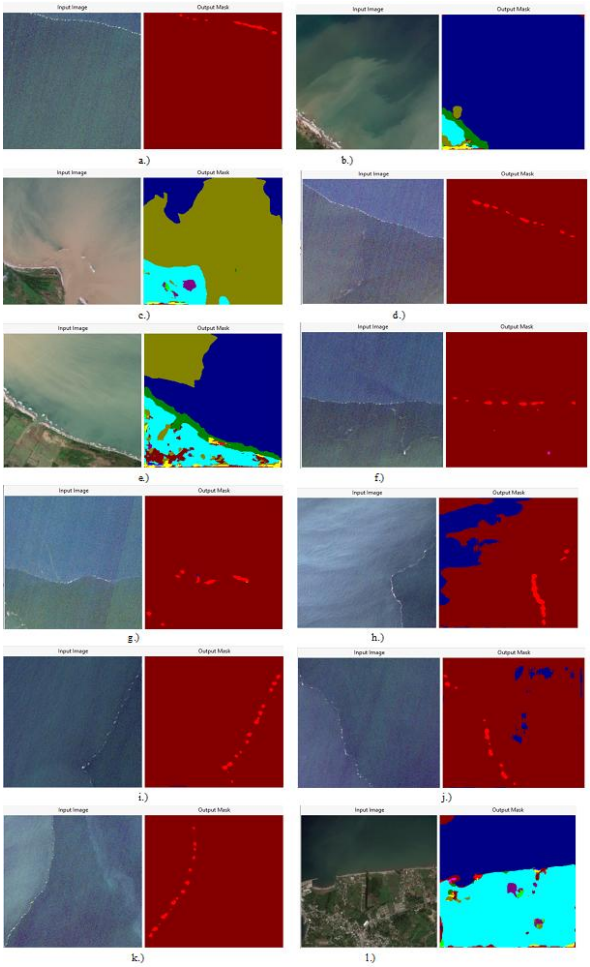

Fig. 8(a to l). Segmentation Outputs along with inputs for the ResUNext Hybrid Model.

Fig. 9. Legend for output segmentations mask

As expected with the dataset majority of the output's pixels were Marine water shown in Fig.8. It's worth noting that the model classifies land as clouds due to lack of available data for land pixels. Further the classifier is also able to classify water bodies surrounded by land as can be seen from the 12th output Fig.9. Majority of the segmentation masks are filled with dark red pixels which are marine water that is the majority class. The light red pixels denote the marine debris class, our primary class of consideration. And all the other classes are also classified with high accuracy as present in the input images. Note that while the images have been displayed in RGB format for visualisation the input images are actually 11 band images. One of the important observations made from these was that if the patches had vegetation cover or land in general, they were classified as clouds. This was because we didn't include a separate land class as there were very few images for the model to be able to learn properly. Although it is to be noted that the misclassification of land did not affect the classification of other pixels. The model was also able to differentiate between turbid water, sediment laden water and marine water respectively showing its efficiency. Instances where land features were misclassified as clouds, particularly in heterogeneous regions, were observed. This can mislead downstream land cover classification. Future strategies to mitigate this include incorporating multi-temporal data, integrating spectral indices or training using auxiliary terrain data to better differentiate between high-reflectance land and cloud cover.

V. Conclusion And Future Work

In this work we take a step further in the direction of solving the problem of marine debris by contributing to improved detection of marine debris using deep learning. We began with the primary goal of utilizing a hybrid approach to improve detection of marine debris specifically in the MARIDA dataset We were able to achieve 10% improvement in overall metrics like IoU, F1 and Pixel Accuracy. Thus demonstrating the better performances of hybrid models for multispectral data. The ability of ResUNext to extract independent features by the use of parallel paths is advantageous by separating unrelated features. Overall, marine debris were classified with 80 percent accuracy. Some classes are still classified incorrectly, indicating further scope for improvement. The limited size of the dataset and its class imbalance is a limiting factor, in the future MARIDA can be augmented with class specific datasets like marine debris specifically to improve the performance. A few limitations can be highlighted in the work which can be contributed towards in the future works. Firstly, despite significantly reducing the training time thanks to the parallelization provided by ResUNext blocks, the training time is still somewhat high. This can be improved by introducing more parallelizable hybrid architecture. Apart from that a huge problem in the work has been to handle the imbalance classes. The imbalance in classes has made it harder to get an accuracy above 90%. This can be worked upon by trying out other loss functions which can handle class imbalance much more robustly like Focal loss by tuning the alpha parameter over several iterations. However overall, the paper explored the potential for marine debris detection through remote sensing and indicates a promising future for the same. In Future Work, will include actionable directions such as Incorporating spectral/spatial attention modules to enhance multispectral feature extraction, evaluating the model on synthetic or simulated datasets for better generalization and exploring contrastive pretraining to improve feature representation with limited labeled data.

References

[1] Perumal, K. et al. (2021) 'Sources, spatial distribution, and abundance of marine debris on Thondi Coast, Palk Bay, Southeast Coast of India', *Environmental Sciences Europe*, 33(1). doi:10.1186/s12302-021-00576-x.

[2] Schuyler, Q.A. et al. (2014) 'Mistaken identity? Visual similarities of marine debris to natural prey items of sea turtles', *BMC Ecology*, 14(1). doi:10.1186/1472-6785-14-14.

[3] Ryan, P.G. (2015) 'Birds and plastic pollution: recent advances', *Marine Pollution Bulletin*, 105(1), pp. 25–26. doi:10.1016/j.marpolbul.2016.02.011.

[4] Hidayaturrahman, H. and Lee, T.G. (2019) 'Microplastics in Digestive System of Little-black cormorant (Phalacrocorax sulcirostris) in Pulau Rambut Sanctuary', *Marine Pollution Bulletin*, 149, p. 110566. doi:10.1016/j.marpolbul.2019.110566.

[5] Thompson, R.C. et al. (2011) 'Marine Debris as a Global Environmental Problem: Introducing a solutions based framework focused on plastic', *STAP Information Document*, Global Environment Facility. doi : hegef.org sites/default/files/publications/STAP_MarineDebris_-_website_1.pdf.

[6] Barboza, L.G.A. et al. (2018) 'Microplastics in Fish and Fishery Products and Risks for Human Health: A Review', *Environmental International*, 114, pp. 200–212. doi:10.1016/j.envint.2018.02.017.

[7] Chae, Y. and An, Y.J. (2017) 'Health Effects of Microplastic Exposures: Current Issues and Perspectives in South Korea', *Environmental Research*, 158, pp. 754–762. doi:10.1016/j.envres.2017.06.002.

[8] Raymond-Yakoubian, J. et al. (2017) 'An Indigenous approach to ocean planning and policy in the Bering Strait region of Alaska', *Marine Policy*, 97, pp. 101–108. doi:10.1016/j.marpol.2018.05.007.

[9] Kikaki, K. et al. (2022) 'MARIDA: A Benchmark for Marine Debris Detection from Sentinel-2 Remote Sensing Data', *PLOS ONE*, 17(1), e0262247. doi:10.1371/journal.pone.0262247.

[10] Zhao, P. et al. (2024) 'Object Detection in Multispectral Remote Sensing Images Based on Cross-Modal Cross-Attention', Sensors, 24(13), p. 4098. doi:10.3390/s24134098.

[11] Yuan, K. et al. (2021) 'Deep-Learning-Based Multispectral Satellite Image Segmentation for Water Body Detection', *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 14, pp. 7422–7434. doi:10.1109/JSTARS.2021.3098678.

[12] Uehara, K. et al. (2017) 'Multi-Channel Higher-Order Local Autocorrelation for Object Detection in Multispectral Imagery', Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 123–130. doi:10.1109/CVPRW.2017.123.

[13] Hu, C. (2022) 'Remote Detection of Marine Debris Using Sentinel-2 Imagery: A Cautious Note on Spectral Interpretations', Remote Sensing, 14(5), p. 1123. doi:10.3390/rs14051123.

[14] Rußwurm, M. et al. (2023) 'Large-scale Detection of Marine Debris in Coastal Areas with Sentinel-2', ISPRS Journal of Photogrammetry and Remote Sensing, 192, pp. 49–60. doi:10.1016/j.isprsjprs.2023.04.012.

[15] Anjaneyulu, S.S.S.R. et al. (2020) 'An Overview of Technological Revolution in Satellite Image Analysis', Journal of the Indian Society of Remote Sensing, 48, pp. 497–513. doi:10.1007/s12524-019-01068-7.

[16] Zhao, F., Huang, B., Wang, J., Shao, X., Wu, Q., Xi, D., Liu, Y., Chen, Y., Zhang, G., Ren, Z., Chen, J., & Mizuno, K. (2025) 'Seafloor debris detection using underwater images and deep learning-driven image restoration: A case study from Koh Tao, Thailand', *Marine Pollution Bulletin*, *214*, 117710. https://doi.org/10.1016/j.marpolbul.2025.117710.

[17] Đuraš, A., Wolf, B. J., Ilioudi, A., Palunko, I., & De Schutter, B. (2024) 'A dataset for detection and segmentation of underwater marine debris in shallow waters', *Scientific Data*, *11*(1). https://doi.org/10.1038/s41597-024-03759-2.

[18] Ma, B., Zhao, F., Xi, D., Wang, J., Shao, X., Wang, S., Tabeta, S., & Mizuno, K. (2024) 'A New Coral Classification Method Using Speed Sea Scanner-Portable and Deep Learning-Based Point Cloud Semantic Segmentation', In *OCEANS 2024 - Halifax, Halifax, NS, Canada, 2024* (pp. 1–4). https://doi.org/10.1109/oceans55160.2024.10753899.

[19] Shen, A., Zhu, Y., Angelov, P., & Jiang, R. (2024) 'Marine debris detection in satellite surveillance using attention mechanisms', *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, *17*, 4320–4330. https://doi.org/10.1109/jstars.2024.3349489.

[20] Nivedita, V., Begum, S. S., Aldehim, G., Alashjaee, A. M., Arasi, M. A., Sikkandar, M. Y., Jayasankar, T., & Vivek, S. (2024) 'Plastic debris detection along coastal waters using Sentinel-2 satellite data and machine learning techniques', *Marine Pollution Bulletin*, *209*, 117106. https://doi.org/10.1016/j.marpolbul.2024.117106

[21] Topouzelis, K. et al. (2019) 'Floating Marine Litter Detection Algorithms and Techniques Using Optical Remote Sensing Data: A Review', Marine Pollution Bulletin, 145, pp. 429–442. doi:10.1016/j.marpolbul.2019.06.011.

[22] Ronneberger, O., Fischer, P., and Brox, T. (2015) 'U-Net: Convolutional Networks for Biomedical Image Segmentation', *Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, pp. 234–241. doi:10.1007/978-3-319-24574-4_28.

[23] Xie, S. et al. (2017) 'Aggregated Residual Transformations for Deep Neural Networks', *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5987–5995. doi:10.1109/CVPR.2017.634.

[24] Phan, T.H. and Yamamoto, K. (2020) 'Resolving Class Imbalance in Object Detection with Weighted Cross Entropy Losses', *arXiv preprint*, arXiv:2006.01413. doi: https://arxiv.org/pdf/2006.01413.

[25] Lin, T.Y. et al. (2017) 'Focal Loss for Dense Object Detection', *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 2980–2988. doi:10.1109/ICCV.2017.324.

[26] Kingma, D.P. and Ba, J. (2015) 'Adam: A Method for Stochastic Optimization', *arXiv preprint*, arXiv:1412.6980. doi: /arxiv.org/abs/1412.6980.

[27] Qian, J. et al. (2021) 'Understanding Gradient Clipping in Incremental Gradient Methods', Proceedings of the 24th International Conference on Artificial Intelligence and Statistics (AISTATS), pp. 3645–3683.

[28] Booth, H., Ma, W. and Karakuş, O. (2023) 'High-precision density mapping of marine debris and floating plastics via satellite imagery', Scientific Reports, 13(1). doi:10.1038/s41598-023-33612-2.

[29] Schaum, A. (2009) 'Remote spectral detection using a laboratory signature', 2009 First Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing, pp. 1–4. doi:10.1109/whispers.2009.5289061.

[30] Piryonesi, S.M. (2019) 'The Application of Data Analytics to Asset Management: Deterioration and Climate Change Adaptation in Ontario Roads', *Doctoral dissertation*, University of Toronto. Available at: https://tspace.library.utoronto.ca/handle/1807/97601.

[31] Piryonesi, S.M. and El-Diraby, T.E. (2021) 'Using Machine Learning to Examine Impact of Type of Performance Indicator on Flexible Pavement Deterioration Modeling', *Journal of Infrastructure Systems*, g27(2), 04021005. doi:10.1061/(ASCE)IS.1943-555X.0000602.

[32] Smith, P.F., Ganesh, S., and Liu, P. (2013) 'A Comparison of Random Forest Regression and Multiple Linear Regression for Prediction in Neuroscience', *Journal of Neuroscience Methods*, 220(1), pp. 85–91. doi:10.1016/j.jneumeth.2013.08.024.

[33] Azhan, Mohammed. (2022) 'ResAttUNet: Detecting Marine Debris using an Attention activated Residual UNet.', *ArXiv* abs/2210.08506 (2022).