electronics

Article

Use of UAVs and Deep Learning for Beach Litter Monitoring

Roland Pfeiffer [1], Gianluca Valentino [1], Sebastiano D'Amico [2], Luca Piroddi [2,*], Luciano Galone [2], Stefano Calleja [1], Reuben A. Farrugia [1] and Emanuele Colica [2]

[1] Department of Communications & Computer Engineering, Faculty of Information and Communication Technology, University of Malta, MSD 2080 Msida, Malta
[2] Department of Geosciences, Faculty of Science, University of Malta, MSD 2080 Msida, Malta
* Correspondence: lucapiroddi@yahoo.it or luca.piroddi@um.edu.mt

Abstract: Stranded beach litter is a ubiquitous issue. Manual monitoring and retrieval can be cost and labour intensive. Therefore, automatic litter monitoring and retrieval is an essential mitigation strategy. In this paper, we present important foundational blocks that can be expanded into an autonomous monitoring-and-retrieval pipeline based on drone surveys and object detection using deep learning. Drone footage collected on the islands of Malta and Gozo in Sicily (Italy) and the Red Sea coast was combined with publicly available litter datasets and used to train an object detection algorithm (YOLOv5) to detect litter objects in footage recorded during drone surveys. Across all classes of litter objects, the 50%–95% mean average precision (mAP50-95) was 0.252, with the performance on single well-represented classes reaching up to 0.674. We also present an approach to geolocate objects detected by the algorithm, assigning latitude and longitude coordinates to each detection. In combination with beach morphology information derived from digital elevation models (DEMs) for path finding and identifying inaccessible areas for an autonomous litter retrieval robot, this research provides important building blocks for an automated monitoring-and-retrieval pipeline.

Keywords: beach litter; object detection; drone surveys; unmanned aerial vehicles (UAVs); deep learning; yolov5; geolocation; litter monitoring; beach cleaning; digital elevation models; unmanned aircraft systems

check for updates

Citation: Pfeiffer, R.; Valentino, G.; D'Amico, S.; Piroddi, L.; Galone, L.; Calleja, S.; Farrugia, R.A.; Colica, E. Use of UAVs and Deep Learning for Beach Litter Monitoring. *Electronics* **2023**, *12*, 198. https://doi.org/10.3390/electronics12010198

Academic Editors: Umberto Papa, Marcello Rosario Napolitano, Giuseppe Del Core and Salvatore Ponte

Received: 1 December 2022
Revised: 23 December 2022
Accepted: 28 December 2022
Published: 31 December 2022

1. Introduction

Marine litter has been identified as a ubiquitous issue [1–4] with ecological [5] and socioeconomic impacts [6–9]. Marine litter is increasing in amount and results in a number of negative effects on marine flora and fauna [5], and research suggests more than 250,000 tons of plastic litter can be found in the world's oceans [1]. It follows various pathways, with one final sink for litter being the seafloor [10,11], where litter might accumulate either in its original form or in smaller pieces, which may result in the creation of microplastics due to fragmentation (or when already at a small initial size as primary microplastics) [12]. Alternatively, if the buoyancy of the litter remains high enough, it can accumulate along beaches and coastlines [13].

Monitoring and cleaning large areas repeatedly requires a substantial availability of personnel and a large number of person hours [14,15] and might not be possible in hard-to-reach areas. Therefore, airborne monitoring and the automatic retrieval of litter are important steps to streamline detection and mitigation efforts, reduce personnel costs and cover different types of terrain. This research was conducted within the scope of the BIOBLU project ("Robotic BIOremediation for coastal debris in BLUE Flag beach and in a Maritime Protected Area"), one part of which consisted of research establishing and evaluating the necessary components for a pipeline that automates these steps, while focusing on the aspects of litter detection using artificial intelligence and the geolocation of the detected items. This paper presents three essential components of this approach: drone surveys, object detection and geolocation of detected litter objects.

Drones or “unmanned aerial vehicles” (UAVs) can record footage at a much higher resolution than what can be achieved using aeroplane- or satellite-based surveys, as flights are conducted at a low altitude with high-resolution RGB cameras (20–48 MP). Therefore, UAV surveys can capture smaller objects that are usually not detected by aeroplane- or satellite-based surveys. In addition, UAV-based surveys can drastically cut costs when compared with these methods, but they come at the cost of lower coverage [16].

2. Materials and Methods

2.1. Artificial Intelligence for Object Detection

Object detection is the task of detecting the type as well as location (and maximum extent) of objects on an image [17]. Object detection algorithms (or models) can be binary (only detecting one type of object, e.g., “car”) or multiclass (detecting multiple object categories, e.g., “person”, “car”, “traffic light”).

One type of artificial intelligence algorithm commonly used for object detection tasks is the Convolutional Neural Network (CNN) [17]. A CNN consists of a series of convolutional and max pooling layers, which can extract features from images. These features are then typically passed on to a fully connected network. In order to train the object detection algorithm for the task of automatically detecting the objects of interest, a labelled dataset is required—i.e., a set of images in which the objects of interest have been labelled with rectangular boxes (“bounding boxes”) corresponding to the object category.

In order to evaluate the performance of the trained model, this dataset is usually split into three parts: a training set, validation set and test set. The training set is used to train the model, and the performance of the trained model is then evaluated against the validation set. Depending on the training setup, the validation set may be used multiple times, and therefore the test set serves as a reference to evaluate the algorithm performance against data that it has never encountered before during training or validation. This gives an overview of how well the algorithm is able to “generalise”—i.e., to apply the behaviour learned during the training phase to new, unseen instances. If the performance on the test set is lower than that on the validation set, the algorithm suffers from “overfitting”, which occurs when the model has been optimised to cater too closely to the characteristics of the validation data, and therefore struggles with new data that does not exhibit these same characteristics.

2.2. Object Detection Training Dataset Creation

The dataset used in the course of this research contained images from a variety of sources. Footage from drone surveys conducted on beaches in Malta and Gozo, Sicily, as well as along the Red Sea coast were used (see Figure 1 and Table 1) in addition to versions of existing datasets, including the TACO dataset [18] with manually adjusted classes and manually labelled versions of Kaggle datasets [19,20], as well as litter objects manually photographed by the authors using a Nokia X10 mobile phone with a 48 MP camera [21].

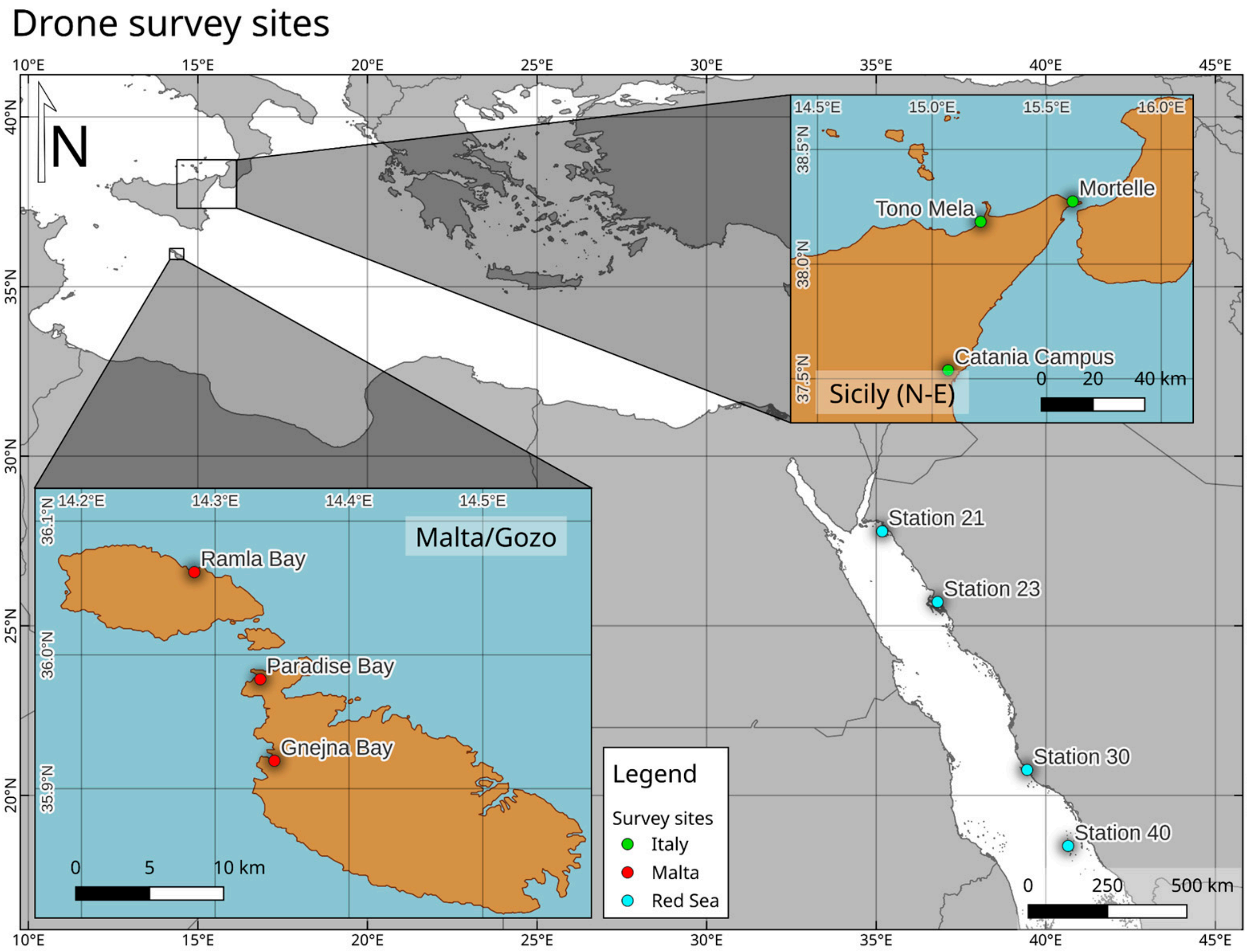

Figure 1. Main drone survey sites on Malta/Gozo (red dots) and Sicily (green dots), as well as survey locations of additional drone footage used for algorithm training that was recorded by [22] along the Red Sea coast (Coordinate Reference System: WGS 84 (EPSG:4326), background shapefile provided by [23]).

Table 1. Locations of survey sites (coordinates in WGS 84 (EPSG:4326)).

Location	Latitude	Longitude	Region	Reference/Source
Paradise Bay	35.981757	14.33372	Malta	Survey
Gnejna Bay	35.920815	14.344291	Malta	Survey
Ramla Bay	36.061839	14.284407	Malta	Survey
Tono Mela	38.185146	15.211505	Italy	Survey
Mortelle	38.273681	15.613148	Italy	Survey
Catania Campus	37.5369902	15.0698772	Italy	Survey
Station 21	27.785	35.1792	Red Sea	Martin et al. [22]
Station 23	25.7008	36.8118	Red Sea	Martin et al. [22]
Station 30	20.7501	39.4539	Red Sea	Martin et al. [22]
Station 40	18.5069	40.663	Red Sea	Martin et al. [22]

Surveys in Malta and Gozo (see Figure 2) were conducted using a DJI Phantom 4 Pro 2.0 (P4P2) drone equipped with a 20 MP RGB camera. Surveys in Italy were conducted using a DJI Mavic 2 Enterprise Advanced (M2EA) drone, equipped with a 48 MP RGB camera, and surveys at the Red Sea coast were conducted using a DJI Phantom 4 Pro [22]. Surveys were flown at a 10 m altitude, and footage was recorded in the form of still images (Malta, Gozo, Red Sea) or video (Italy). A 3D model of the survey site at Ramla Bay (Gozo) can be found at [24].

Figure 2. Setting up the DJI Phantom 4 Pro 2.0 (P4P2) drone for surveys in Ramla Bay (Gozo).

Since YOLOv5 requires images for training, still frames were extracted from drone survey videos at an interval of 1 image per second. In cases where multiple images showing the same location with little or no difference (e.g., when the drone was travelling at a slow speed), only one of those images was used in training in order to avoid duplicates.

Image Processing and Labelling

As using the drone survey images at full resolution for training initially exceeded the memory limits of the graphics processing unit (GPU) used for training, the images obtained from the drone surveys were cut into six square or near-square tiles (two rows, three columns of tiles) so that the resolution did not need to be reduced in the training process. The tile aspect ratio depends on the aspect ratio of the original image, and the maximum tile edge length was 1824 pixels.

Images were manually screened, and litter objects were labelled using the labelme software [25]. In addition, objects that were not litter but still prominent in the images were labelled as well to reduce ambiguity during training. In total, 67 classes were used for categorising labels. A table containing all 67 classes can be found in the Supplementary Materials in Table S1. For simplification and visualisation purposes, metaclasses were assigned based on their material and common waste separation schemes, and categories not aligning with these groups were classified as "Other". Instances that occurred in less than 10 images in the total dataset were assigned the metaclass "N img < 10". Grouped instance counts of different litter types can be seen in Figure 3. The number of annotations and images per class can be found in the Supplementary Materials in Table S1.

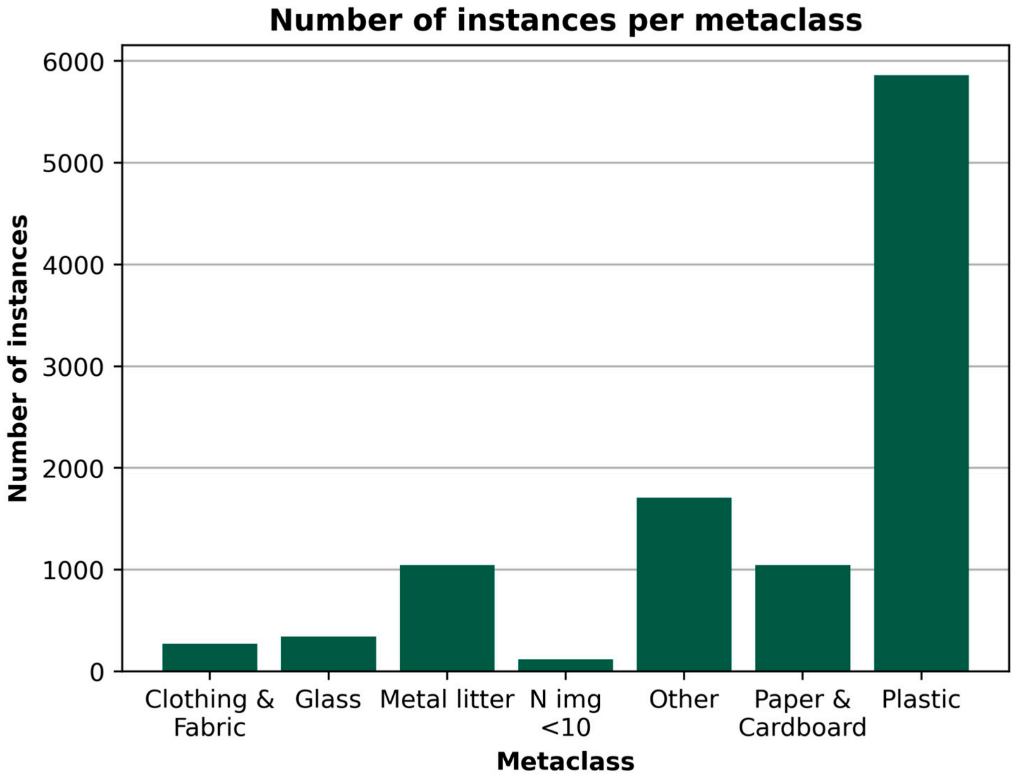

Figure 3. Number of annotations (i.e., instances) per material group. Bars depict meta groups that the 67 label classes were assigned to based on litter materials. Classes that were not litter objects were categorised as "Other", and small classes that were present on fewer than 10 images were labelled "N img < 10".

After image selection and labelling, the dataset consisted of a total of 4126 images and 10,611 annotations, with 1154 images showing no litter objects ("background images"). The dataset was split into a training, validation and test portion of the proportions of 0.6, 0.2 and 0.2, respectively. For the number of images and annotations in each set, see Table 2.

Table 2. Annotation and image counts for the training, validation and test set.

Set	Images	Annotations	Background Images
Training	2476	6124	701
Validation	825	2190	229
Test	825	2297	224
TOTAL	4126	10611	1154

2.3. Object Detection Algorithm Training

Common algorithms for object detection tasks are Convolutional Neural Networks (CNNs). In this paper, the YOLOv5 [26] architecture was used (derived from the original YOLO algorithm developed in 2016 [27]), as its single-stage architecture allows for faster detection speeds than other commonly used two-stage detectors (e.g., Faster R-CNN) [28]. Two-stage detectors first produce region proposals indicating regions of interest, and then they conduct object detection on those regions in a second step, while single-stage detectors perform both tasks in one neural net. The YOLO algorithm—instead of using region proposals—handles the full image, covers it with a grid and lets each cell handle those predictions whose BBOX centres fall within that cell [27]. The YOLOv5 network uses a CSP-Darknet53 as the backbone network, i.e., a Darknet53 Convolutional Neural Network following a Cross Stage Partial (CSP) Network strategy [26]. This backbone part of the algorithm is mainly used for extracting features from the input image. The next stage, the neck of the YOLOv5 algorithm, aggregates the features and allows the algorithm to generalise well across different scales, and uses fast Spatial Pyramid Pooling (SPPF) and CSP-PAN, i.e., a Cross Stage Partial Path Aggregation Network (PAN) with BottleneckCSP [26]. The last part of the YOLOv5 network—the head—uses a YOLOv3 head and is responsible for producing the final output of the predictions: the predicted classes, the corresponding bounding boxes, and the confidence per prediction [26]. For an overview of the general YOLOv5 architecture, see Figure 4.

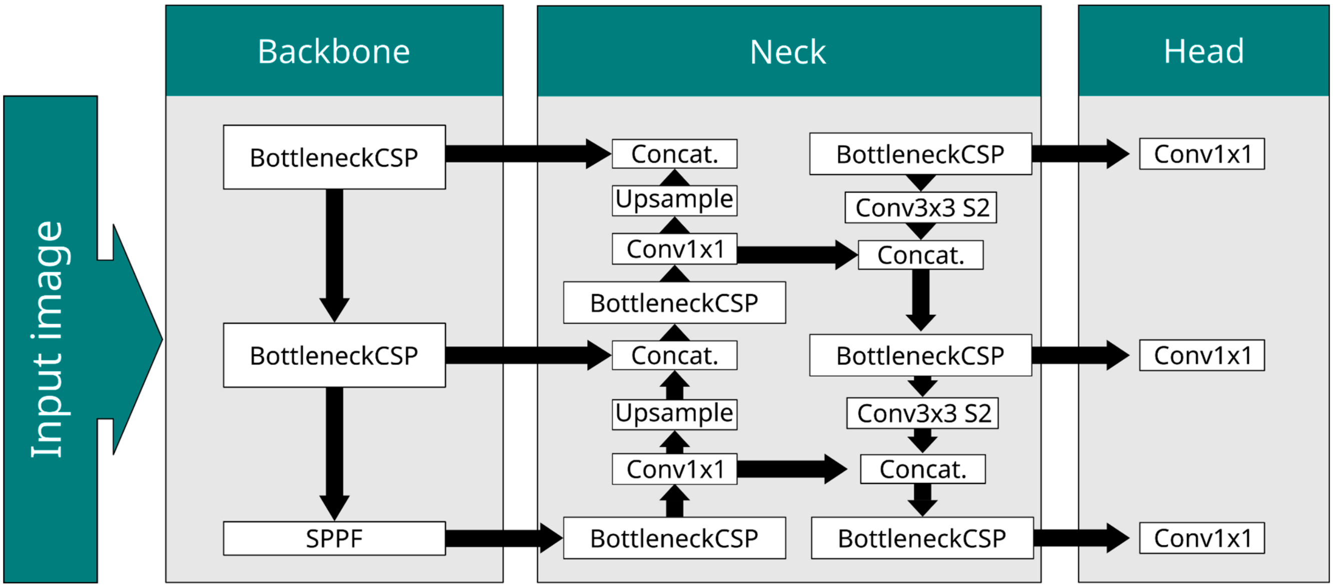

Figure 4. General architecture of the YOLOv5l network. The backbone consists of a Cross Stage Partial Darknet53 Convolutional Neural Network (CSP-Darknet53), responsible mainly for feature extraction. Spatial Pyramid Pooling (SPPF) provides feature pyramids that are used in the neck by a Cross Stage Partial Path Aggregation Network for feature aggregation. The head consists of a YOLOv3 head and provides the final output of the detector: the prediction classes, bounding boxes and confidence values.

Training was conducted on a cluster node running Ubuntu 20.04 LTS, CUDA version 11.4, NVIDIA driver version 470.141.03, utilising a NVIDIA A100 GPU with 80 GB GPU memory. For training, the yolov5l6.pt pretrained weights were used. The maximum number of epochs was set to 5000, the batch size was set to 16 and the image size was set to 1856 pixels, which allowed for the efficient use of the available GPU memory. All other settings were left at default values (specified in the YOLOv5 file "hyp.scratch-low.yaml" [26]).

The performance of the detection was measured in terms of precision (P), recall (R) and mean average precision (mAP). P is a measure of the likelihood of a detected object to have been detected correctly (i.e., is a measure of the accuracy of the predictions). R describes the proportion of objects that have been detected out of all objects that should have been detected (i.e., gives an estimate of the coverage of the algorithm).

P and *R* are calculated using ratios of True Positive (*TP*), False Negative (*FN*) as well as False Positive (*FP*) values (see Equations (1) and (2)). *TP* describes the proportion of correctly detected litter objects, *FN* describes the proportion of objects that were erroneously classified as the background (i.e., "missed" objects) and *FP* describes irrelevant background features that were erroneously detected by the algorithm.

$$P = TP/(TP + FP) \tag{1}$$

$$R = TP/(TP + FN) \tag{2}$$

Usually, P decreases with an increasing R [29]. If P is plotted as a function of R (a so-called precision–recall curve), the area under the curve (AUC) is a useful metric of assessing P over increasing R. In order to assess the performance of multiple classes, the average AUC across classes is calculated, resulting in the mAP metric. mAP values depend on the threshold of the overlap between the prediction and the ground truth box. This overlap is calculated using the Intersection-over-Union (IoU) ratio. IoU is calculated as depicted in Equation (3), where A and B are the ground-truth box and the prediction box, respectively.

$$IoU(A, B) = A \cap B / A \cup B \tag{3}$$

Predictions with bounding boxes that have an IoU value above a set threshold are regarded as correctly capturing the object, while boxes with an IoU below the threshold are considered FP. In this paper, we used the mAP metric of mAP@50-95 (average of mAPs of thresholds from 50% to 95%, in steps of 5%) to compare the performance of different classes.

A single metric that combines both P and R values is the $F1$ score, which is the harmonic mean between P and R and is calculated using Equation (4) [17]. As such, the $F1$ score is also a measure of whether P and R are both similarly high and penalises high differences between P and R [17].

$$F1 = 2/((1/P) + (1/R)) \tag{4}$$

2.4. Geolocation of Object Detections

In order to deliver useful information to a robot for debris retrieval, predictions made by the YOLOv5 algorithm need to be geolocated so that their coordinates can be communicated to the robot. In order to be able to retrieve coordinates for the predictions, the original footage needs to come with the GPS coordinates of the drone at the time of recording either embedded in the metadata (in case of image footage) or contained in a metadata-subtitle file (.srt, in the case of video footage).

After running the predictions, the results can then, in combination with the GPS information from the image metadata or video subtitle file, be georeferenced so that every prediction comes with a corresponding GPS coordinate. The geolocation of prediction boxes incorporates the following steps:

1. Calculation of pixel size of the footage.
2. Calculation of the distance (in m) between Meridians and Parallels at the latitude of recording.
3. Calculation of the horizontal and vertical distance of the prediction box centre from the image centre.
4. Transformation of the prediction distance to the real-world distance and the calculation of the prediction coordinates.

2.4.1. Pixel Size

A calculation of the real-world pixel size of the footage, also named the ground sampling distance (GSD), was conducted using Equation (5) provided by [30], which incorporates the drone camera's sensor width in millimetres (Sw), the flight altitude in metres (a), the focal length of the camera in millimetres (f, real focal length, not 35 mm equivalent) as well as the width of the recorded image in pixels (imW):

$$GSD = (Sw \cdot a \cdot 100)\ (f \cdot imW) \tag{5}$$

2.4.2. Distance between Meridians and Parallels

The distance between the Meridians and Parallels depends on the latitude, due to the spheroid shape of the Earth. Calculations for the geolocation are based on the WGS84 spheroid [31] with a semimajor axis of 6,378,137.0 metres (a) and first eccentricity of $8.1819190842622 \times 10^{-02}$ (e). From these values, the radii of curvature in metres along the Meridians (M) and Parallels (N) at latitude ϕ can be calculated using Equations (6) and (7), respectively, provided by [32]:

$$M = (a(1 - e^2))/(1 - e^2 \cdot sin^2\ \Phi)(3/2) \tag{6}$$

$$N = a/(1 - e^2 \cdot sin^2\ \Phi)(1/2) \tag{7}$$

From these, the distance in metres between Meridians (d_{lon}) and Parallels (d_{lat}) can be calculated using Equations (8) and (9), respectively:

$$d_{lon} = (N \cdot \pi)/180 \tag{8}$$

$$d_{lat} = (M \cdot \pi)/180 \tag{9}$$

2.4.3. Latitude and Longitude of Prediction Box

The horizontal and vertical offset in pixels of the prediction box centre (which is provided with the prediction from the algorithm) from the image centre along the latitude and longitude directions can be calculated using trigonometry, Euclidean distance calculations as well as d_{lat}, d_{lon} and GSD values (see Figure 5). For a visualisation of the workflow outlined above, see Figure 6.

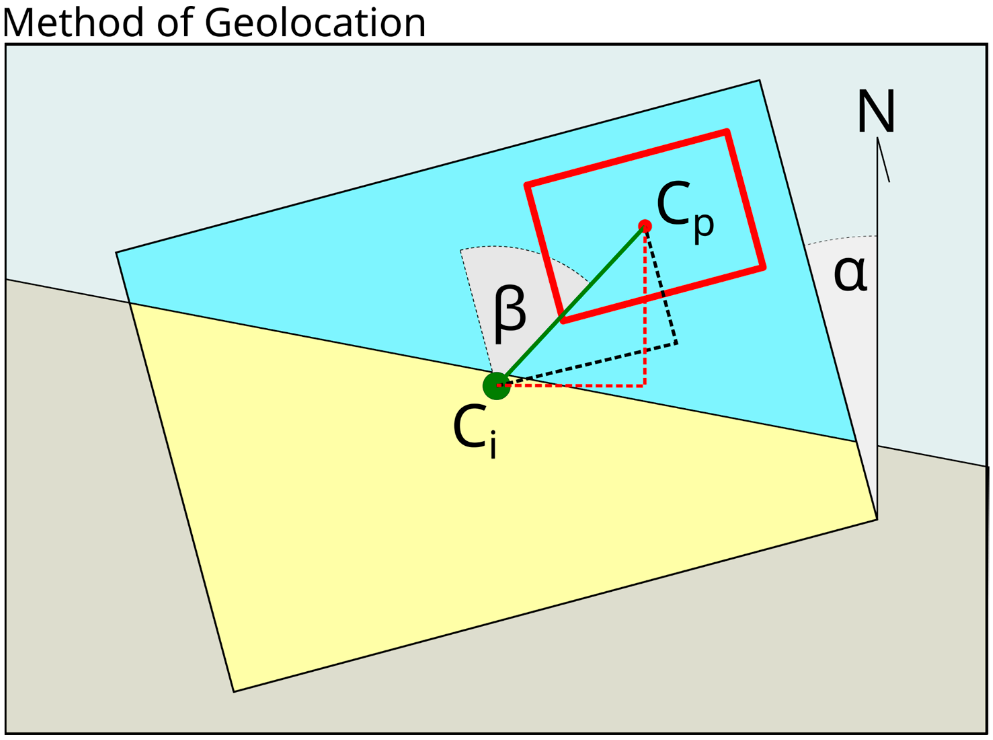

Figure 5. Schematic description of calculation of the location of the centre (Cp, red dot) of the prediction BBOX (red rectangle). The coordinates of the image centre (Ci, green dot) are known from image metadata. Horizontal and vertical offset of the prediction box centre (black dashed line) within the image are calculated from Cp and the image dimensions. Real-world offset of the BBOX centre along latitude and longitude (red dashed lines) is calculated using the Euclidean distance between Cp and Ci, the angle (β) of the Euclidean distance combined with the angle of drone yaw (α) and the previously calculated *GSD*-, d_{lat}- and d_{lon} values.

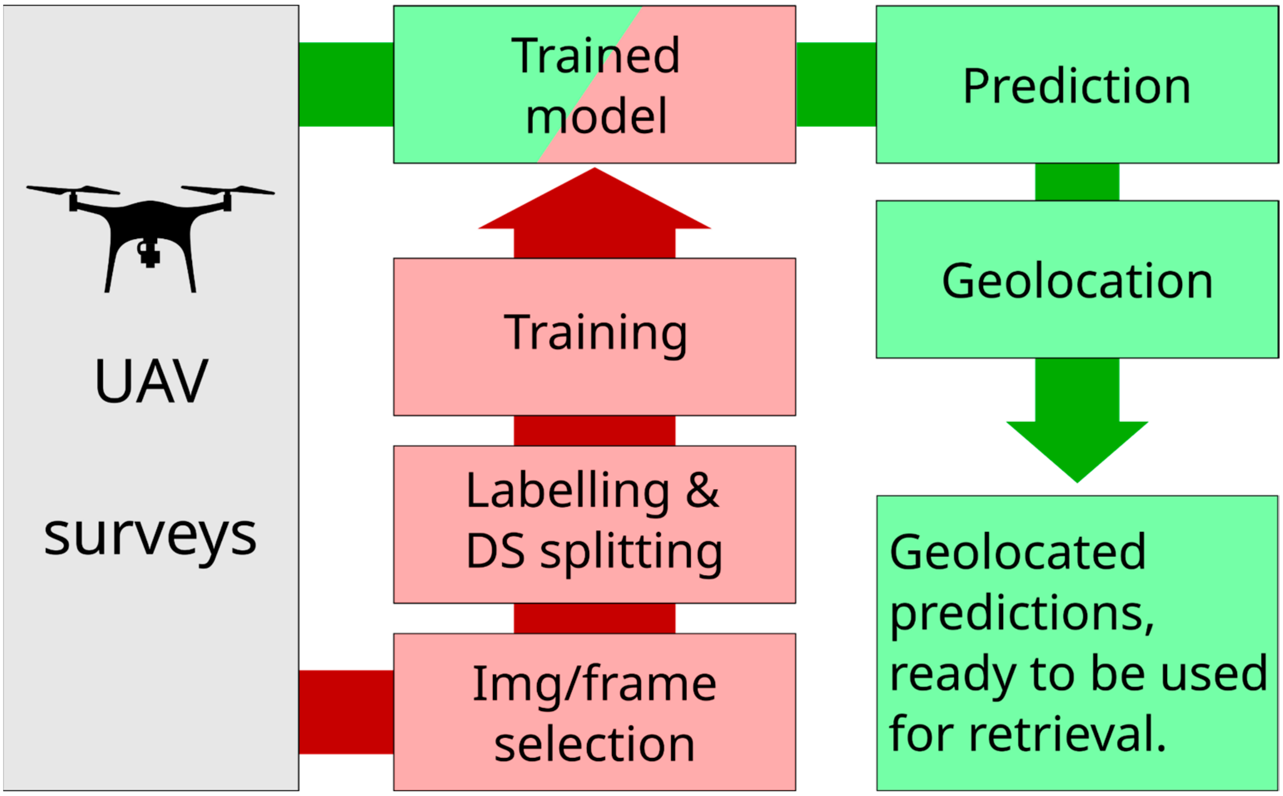

Figure 6. Visualisation of the general workflow with the two main pipelines of training (**red**) and prediction (**green**). For training, image footage from unmanned aerial vehicle (UAV) surveys was used (either as separate images or as frames from video file), and a subselection of image files was used to avoid objects appearing on multiple images. Images were subsequently labelled manually, and the algorithm was trained on the dataset, leading to a trained model. For prediction (**green**), this trained model could then be used on UAV survey footage (images or video footage) to predict litter objects. The predictions were then geolocated, leading to a set of geolocated predictions that are ready to be used in automated retrieval operations.

3. Results

Regarding the validation of the trained YOLOv5 algorithm against the test set of the labelled dataset, the performance metrics of the trained network across all classes were precision = 0.695, recall = 0.288, mAP50 = 0.314 and mAP50-95 = 0.252. Figure 7 provides a breakdown of the results, grouped by common waste separation categories. The overall *F1* score of the trained algorithm was 0.32 at a confidence of 0.235.

Considering that small classes are prone to overfitting [17] and the fact that we therefore focused on more abundant classes that appeared in more than 100 images, the top ten classes on which the algorithm performed most reliably were the categories “plastic bottle” (mAP50-95: 0.674), “metal can” (“mAP50-95: 0.516), “plastic bottlecap” (mAP50-95: 0.483), “plastic container” (mAP50-95: 0.447), “shoe” (mAP50-95: 0.43), “cardboard” (mAP50-95: 0.369), “pop tab” (mAP50-95: 0.366), “rope & string” (mAP50-95: 0.337), “wood” (mAP50-95: 0.324) and “glass bottle” (mAP50-95: 0.317). For the precision, recall and mAP50 values for these as well as the other classes, see Table S1 in the Supplementary Materials.

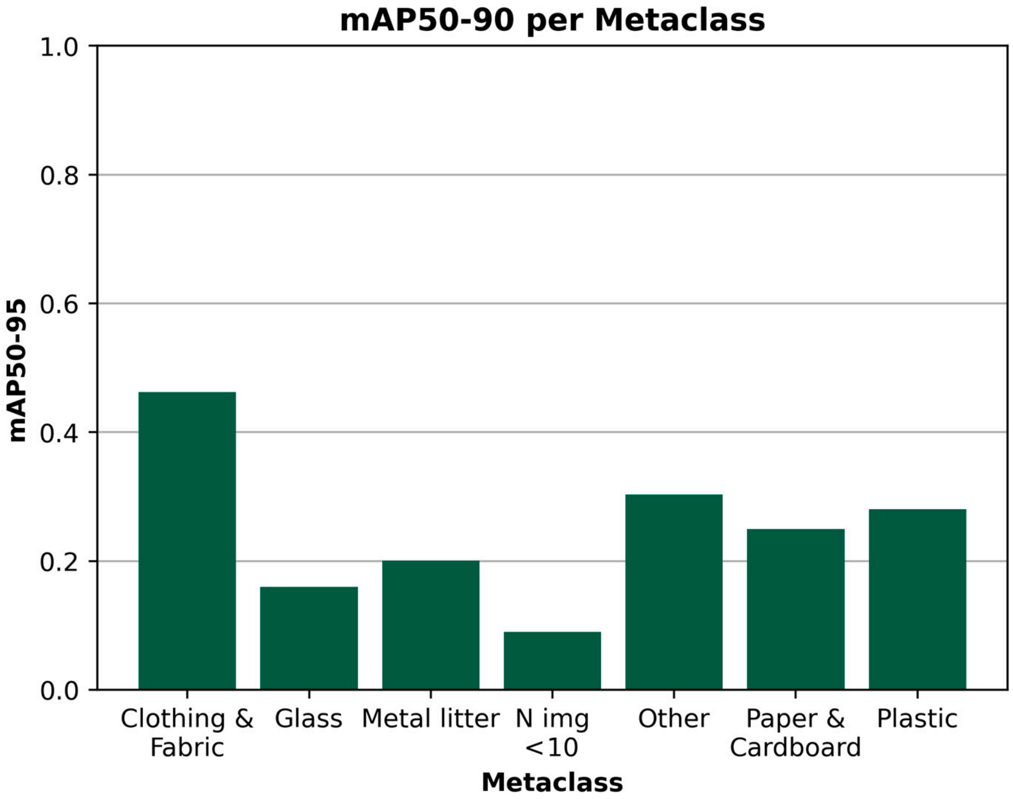

Figure 7. Mean average precision across intersect-over-union thresholds between 50% and 95% (mAP50-95) scores for grouped litter classes. Original classes were assigned metaclasses based on common waste separation protocols. Classes that were not common litter classes were grouped into "Other". Classes that occurred on fewer than 10 images in the dataset were assigned the metaclass "N img < 10".

For comprehensive, per-class mAP50-95 values, see Figure 8. Additionally, a table with the corresponding values can be found in the Supplementary Materials in Table S1. The per-class confusion matrix from the validation against the test set confirmed the above mentioned results and showed that the majority of the mislabelled objects were found in the "background" category, indicating False Negatives during the prediction (see Figure S1).

After the detections were geolocated, each detection was associated with latitude and longitude values. For an example of geolocated predictions from an image set recorded at Paradise Bay, see Figure 9.

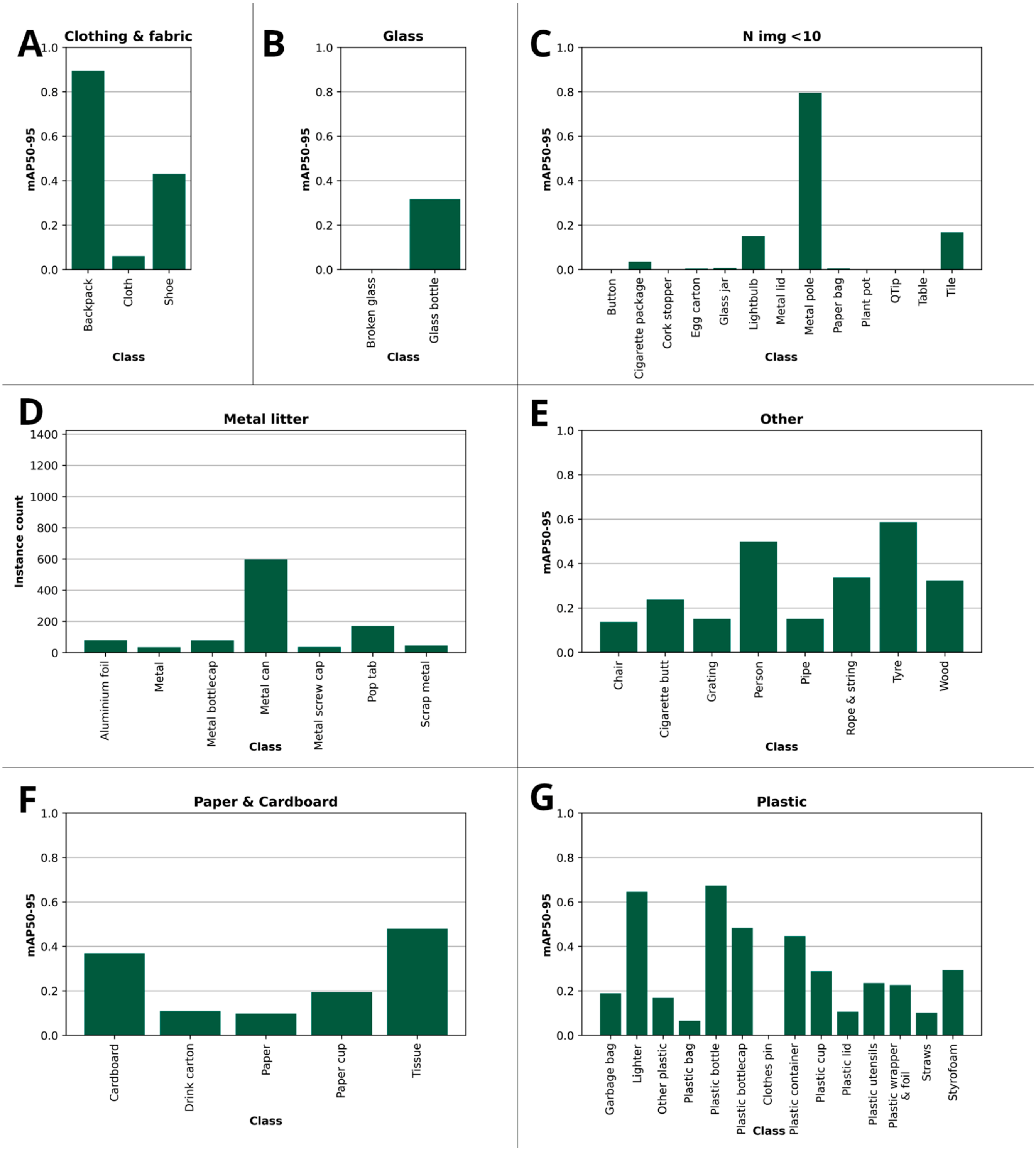

Figure 8. Per-class performance in subplots for each metaclass (**A**–**G**). X axes depict class name, y axes depict mean average precision across intersect-over-union thresholds between 50% and 95% (mAP50-95). Classes that were not litter objects were categorised as "Other", and small classes that were present on fewer than 10 images were labelled "N img < 10".

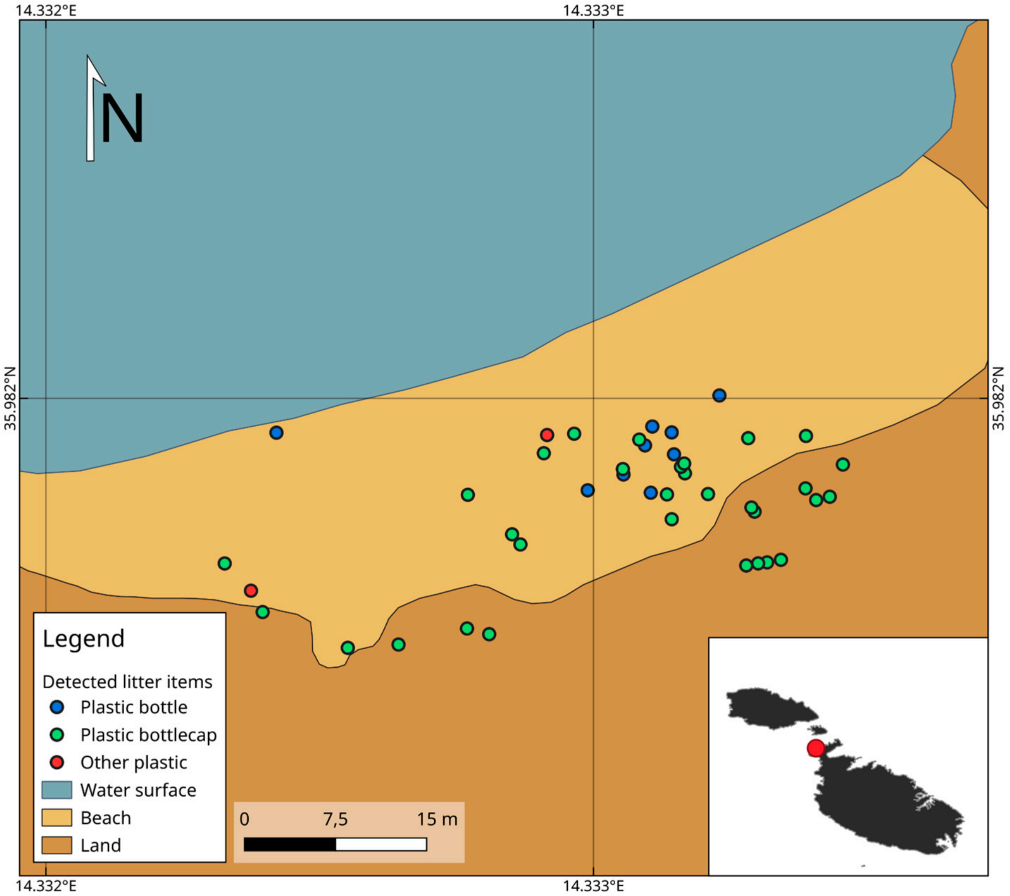

Figure 9. Geolocated object detections, colour-coded by their corresponding class, on Paradise Bay, at the northwestern coast of Malta. Coordinate Reference System: WGS 84 (EPSG:4326), overview shapefile provided by [23].

4. Discussion

The object detection and geolocation approach outlined in the sections above provides essential information that can be used for automatic retrieval. High-performance metrics were observed for classes that were frequent but also for some that were infrequent in the dataset. Since deep learning algorithms require large amounts of training data in order to be reliable and able to generalise [17], a recommendation would be to increase the amount of training data to bring classes that have a small number of images and/or instances in the training dataset to an even level compared to the larger classes. This would also make a comparison between the (previously imbalanced) classes more informative.

While drone surveys are mostly unaffected by ground morphology (except when encountering cliffs, for example, or flying at very low altitudes), automatic retrieval via robots requires detailed information about beach morphology, and in order to conduct ground surveys in a safe way, it is essential to identify those areas which are accessible to the robot and to distinguish them from inaccessible ones. This includes features such as boulders or rocky terrain, runoff channels or break-off edges due to erosion. Since these features can be subject to change over time, due to seasonality, weather events or long-term topography changes [33,34], it is important to collect up-to-date beach morphology information before deploying a robot for automatic retrieval missions.

UAVs are a useful platform for collecting beach morphology information. Common UAV-based methods include Lidar [35], a laser-based technique that has been shown to perform well for beach morphology monitoring and constructing digital elevation models (DEMs) [36,37], as well as Structure-from-Motion (SfM) photogrammetry, which utilises RGB images to recreate the 3D structure of the surveyed area and is commonly used for beach landform analyses and has been shown to perform well in comparison to aeroplane-based Lidar surveys by building high-resolution digital elevation models (DEMs) [38]. Furthermore, photogrammetry allows researchers to generate a high-resolution orthomosaic alongside the DEM, which facilitates the correct interpretation of the latter (see Figure 10).

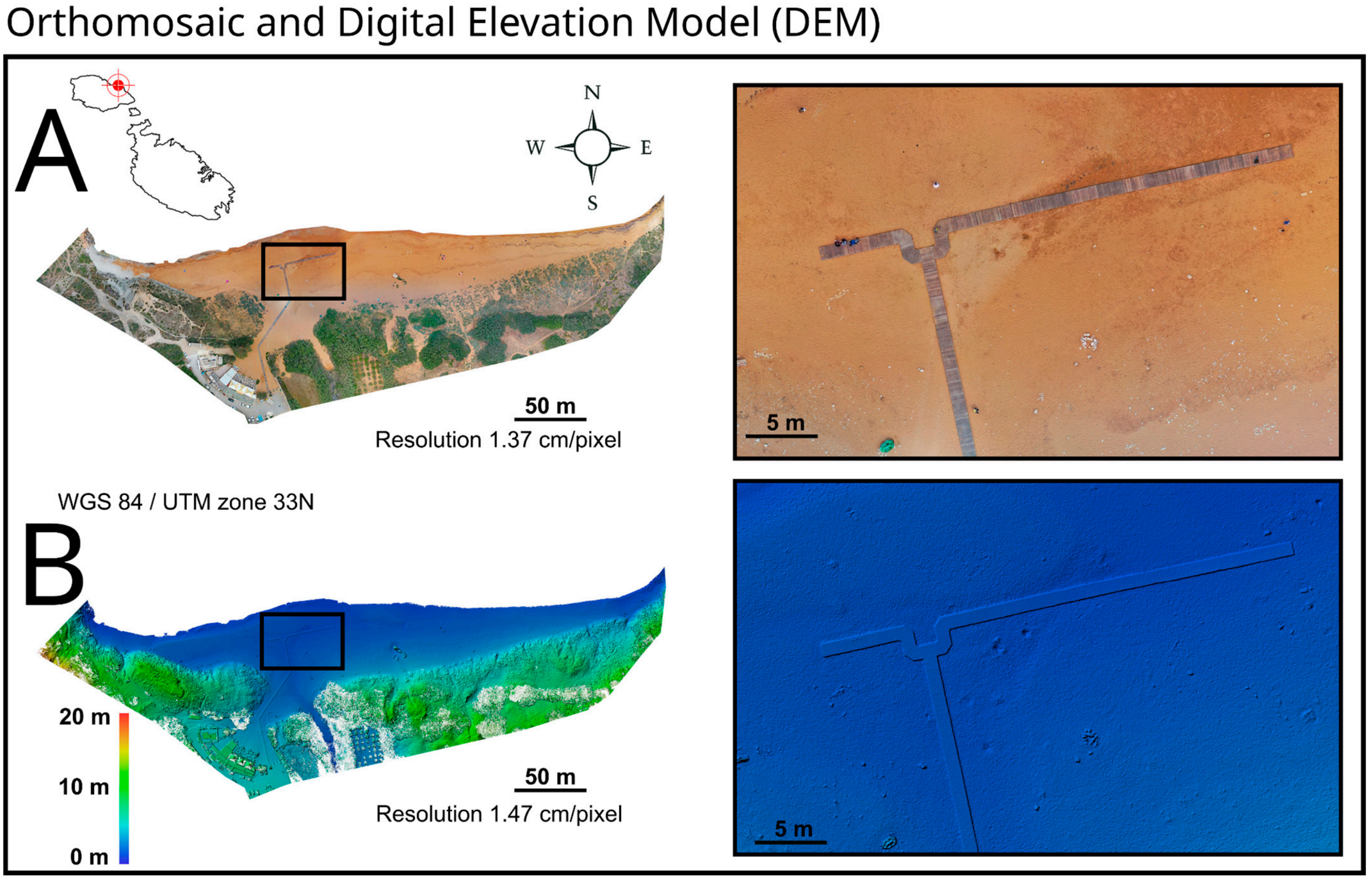

Figure 10. Examples of orthomosaics (**A**) and digital elevation models (**B**). Complete overviews on the left and zoomed-in sections on the right. Footage was collected in Ramla Bay Beach, Gozo.

The correct interpretation of the DEM is crucial when deploying autonomous robots for litter retrieval. The output derived from SfM photogrammetry can be used to detect inaccessible and/or impassable areas and optimise their path [36,37,39]. In addition, when surveys are repeated over time, DEMs and orthomosaics can be used to analyse beach morphology changes over time and assess local trends such as net losses or gains.

One example of using SfM photogrammetry for beach monitoring is a study by Colica et al. [38]. They used a DJI Phantom 4 Pro drone, equipped with a camera with a resolution of 20 Megapixels and 1″ Exmor R CMOS image sensor, to create a high-resolution DEM of Ramla Bay beach (Gozo). The acquisition interval of the images and the flight plan were programmed through the DJI Ground Station app, and the set parameters were an 85% forward and 70% side overlap of the images with a pixel resolution of 5472 × 3648 acquired at a flight altitude of about 60 m above sea level. The dataset includes 1021 nadiral images that were processed using the commercial software Agisoft Metashape [40], which allows one to set different parameters to control the photogrammetric reconstruction process including the accuracy parameter during image alignment, which controls the size and resolution of the images on which the software will detect the key points that are useful to the image alignment. In this phase, with the accuracy parameters set to the highest, a "sparse" point cloud containing approximately 722,000 points was produced. Subsequently, the 56 Ground Control Points (GCPs) measured with the Topcon HiPer HR

DGNSS receivers [41] in a Base + Rover configuration showed a horizontal accuracy of 3 mm ± 0.1 part per million and a vertical accuracy of 3.5 mm ± 0.4 part per million. Subsequently, the dense point cloud (about 94 million points) and depth maps with ultrahigh quality and mild filtering mode were calculated. From these, the DEM was then generated with a resolution of 1.47 cm/px and the orthomosaic with a resolution of 1.37 cm/px.

From the DEM, accessibility to a ground robot can be derived, e.g., by conducting a traversability analysis and providing a 2D costmap [39]. The costmap can be fed into a path-planning algorithm such as a D* algorithm [42] and then provide a series of waypoints to cover the accessible areas of interest as efficiently as possible (at a minimum cost, based on the cost map), as demonstrated by [39]. In addition, when surveys are repeated over time, DEMs should be constructed repeatedly as well in order to account for beach topology changes and identify newly inaccessible areas, for example. These repeatedly collected DEMs can also be used to analyse beach morphology changes over time and assess local trends such as net losses or gains.

Geolocating single frames from video footage or overlapping images from an image set will result in multiple detections of the same objects, as YOLOv5 does not natively allow for the tracking of objects. One possible approach to remedy these "double detections" is to cluster the geolocated points, e.g., by using clustering algorithms such as OPTICS or DBSCAN clustering [43,44], but clustering might not be possible on beaches where a high density of litter objects is present. In this case, using an algorithm that is capable of tracking objects across different images or video frames might be a worthwhile approach.

Clustering should also be considered when running object detection on drone video footage where the GPS recording frequency is not matched to the framerate of the recorded footage, as a mismatch of recording frequencies between the GPS and camera can distort GPS positioning along the UAVs flight direction.

5. Conclusions and Outlook

The object detection algorithm trained in the context of this paper performed well on recognising common beach litter categories such as plastic bottles, metal cans and plastic bottle caps, and the geolocation provided necessary information for later automatic retrieval when merged with additional information about accessibility derived from DEMs. For further improvement of the performance and reliability of the algorithm, more instances should be added to the classes that were underrepresented in the current dataset to reach a more balanced number of annotations and images per class. In order to reduce the number of double detections on overlapping footage, clustering and/or object tracking algorithms should be explored.

Supplementary Materials: The following supporting information can be downloaded at: https://www.mdpi.com/article/10.3390/electronics12010198/s1, Table S1: Name, number of annotations per class, number of images in which each class occurs, precision, recall, mean average precision at 50% intersect-over-union (IoU) threshold (mAP50) and at 50% to 95% IoU (in 5% steps, mAP50-95) values per class. Empty fields indicate classes that did not appear in the test set due to the low number of images available in the training set and the metaclass; Figure S1: Confusion matrix from the validation against the test set. The x-axis represents the actual categories of objects, while the y-axis represents the categories of the detections as predicted by the algorithm. Values in the matrix represent the proportion of predictions for detections of each category. Perfect predictions with no misclassifications will lead to a single line of 1.0 values running across the matrix from the top left to the bottom right. The fact that many misclassifications have been predicted as "background" and are therefore accumulating at the bottom of the matrix indicates False Negative (FN) predictions (i.e., missed objects that should have been detected).

Author Contributions: Conceptualization, G.V. and S.D.; methodology, G.V. and R.P.; software, R.P.; validation, R.P. and G.V.; formal analysis, R.P. and G.V.; investigation, R.P., G.V. and E.C.; resources, S.D., G.V. and E.C.; data curation, R.P. and S.C.; writing—original draft preparation, R.P.; writing—review and editing, R.P., S.D., G.V., L.P., L.G. and E.C.; visualization, R.P., E.C. and L.G.; supervision, G.V., S.D. and R.A.F.; project Administration, G.V.; funding acquisition, S.D., G.V. and R.A.F. All authors have read and agreed to the published version of the manuscript.

Funding: This research was financially supported by the BIOBLU project—Robotic BIOremediation for coastal debris in BLUE Flag beach and in a Maritime Protected Area (Code Area C2). Grant Number: J49C20000060007. Program INTERREG V A Italia-Malta 2014 2020 (Interreg V-A Cross Border Cooperation Italia-Malta projects). This work was partially supported by the projects "Coastal Satellite-Assisted Governance (tools, technique, models) for Erosion" (Coastal SAGE, grant agreement number SRF-2020-1S1) and "Satellite Investigation to study POcket BEach Dynamics" (SIPOBED, grant agreement number SRF-2021-2S1) financed by the Malta Council for Science & Technology (MCST, https://mcst.gov.mt/ (accessed on 30 November 2022)), for and on behalf of the Foundation for Science and Technology, through the Space Research Fund.

Data Availability Statement: The python scripts used in this study are publicly available under https://dsrg-ict.research.um.edu.mt/gianluca/bioblu (accessed on 30 November 2022). Trained weights for the object detection algorithm, as well as the labelled dataset and digital elevation models can be provided upon request but are not publicly available due to large file sizes.

Acknowledgments: The authors would like to thank Mario Vitti, Dario Guastella, Giovanni Cicceri, Giuseppe Sutera and Francesco Cancelliere for conducting drone surveys and collecting footage in Sicily, and for the valuable exchange of experience.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Eriksen, M.; Lebreton, L.C.M.; Carson, H.S.; Thiel, M.; Moore, C.J.; Borerro, J.C.; Galgani, F.; Ryan, P.G.; Reisser, J. Plastic Pollution in the World's Oceans: More than 5 Trillion Plastic Pieces Weighing over 250,000 Tons Afloat at Sea. *PLoS ONE* **2014**, *9*, e111913. [CrossRef] [PubMed]
2. Tekman, M.B.; Krumpen, T.; Bergmann, M. Marine Litter on Deep Arctic Seafloor Continues to Increase and Spreads to the North at the HAUSGARTEN Observatory. *Deep Sea Res. Part I Oceanogr. Res. Pap.* **2017**, *120*, 88–99. [CrossRef]
3. Bergmann, M.; Lutz, B.; Tekman, M.B.; Gutow, L. Citizen Scientists Reveal: Marine Litter Pollutes Arctic Beaches and Affects Wild Life. *Mar. Pollut. Bull.* **2017**, *125*, 535–540. [CrossRef] [PubMed]
4. Eriksson, C.; Burton, H.; Fitch, S.; Schulz, M.; van den Hoff, J. Daily Accumulation Rates of Marine Debris on Sub-Antarctic Island Beaches. *Mar. Pollut. Bull.* **2013**, *66*, 199–208. [CrossRef] [PubMed]
5. Kühn, S.; Bravo Rebolledo, E.L.; van Franeker, J.A. Deleterious Effects of Litter on Marine Life. In *Marine Anthropogenic Litter*; Bergmann, M., Gutow, L., Klages, M., Eds.; Springer International Publishing: Cham, Switzerland, 2015; pp. 75–116. ISBN 978-3-319-16510-3.
6. Mouat, J.; Lopez Lozano, R.; Bateson, H. *Economic Impacts of Marine Litter*; Kommunernes Internationale Miljøorganisation (KIMO): Lerwick, UK, 2010; 117p.
7. Ballancea, A.; Ryanb, P.G.; Turpieb, J.K. How Much Is a Clean Beach Worth? The Impact of Litter on Beach Users in the Cape Peninsula, South Africa. *S. Afr. J. Sci.* **2000**, *96*, 210–213.
8. Botero, C.M.; Anfuso, G.; Milanes, C.; Cabrera, A.; Casas, G.; Pranzini, E.; Williams, A.T. Litter Assessment on 99 Cuban Beaches: A Baseline to Identify Sources of Pollution and Impacts for Tourism and Recreation. *Mar. Pollut. Bull.* **2017**, *118*, 437–441. [CrossRef]
9. Williams, A.; Rangel-Buitrago, N.; Anfuso, G.; Cervantes, O.; Botero, C.-M. Litter Impacts on Scenery and Tourism on the Colombian North Caribbean Coast. *Tour. Manag.* **2016**, *55*, 209–224. [CrossRef]
10. García-Rivera, S.; Lizaso, J.L.S.; Millán, J.M.B. Composition, Spatial Distribution and Sources of Macro-Marine Litter on the Gulf of Alicante Seafloor (Spanish Mediterranean). *Mar. Pollut. Bull.* **2017**, *121*, 249–259. [CrossRef]
11. Ioakeimidis, C.; Zeri, C.; Kaberi, H.; Galatchi, M.; Antoniadis, K.; Streftaris, N.; Galgani, F.; Papathanassiou, E.; Papatheodorou, G. A Comparative Study of Marine Litter on the Seafloor of Coastal Areas in the Eastern Mediterranean and Black Seas. *Mar. Pollut. Bull.* **2014**, *89*, 296–304. [CrossRef]
12. Woodall, L.C.; Sanchez-Vidal, A.; Canals, M.; Paterson, G.L.J.; Coppock, R.; Sleight, V.; Calafat, A.; Rogers, A.D.; Narayanaswamy, B.E.; Thompson, R.C. The Deep Sea Is a Major Sink for Microplastic Debris. *R. Soc. Open Sci.* **2014**, *1*, 140317. [CrossRef]
13. Kusui, T.; Noda, M. International Survey on the Distribution of Stranded and Buried Litter on Beaches along the Sea of Japan. *Mar. Pollut. Bull.* **2003**, *47*, 175–179. [CrossRef]

14. Cruz, C.J.; Muñoz-Perez, J.J.; Carrasco-Braganza, M.I.; Poullet, P.; Lopez-Garcia, P.; Contreras, A.; Silva, R. Beach Cleaning Costs. *Ocean Coast. Manag.* **2020**, *188*, 105118. [CrossRef]
15. Martin, C.; Parkes, S.; Zhang, Q.; Zhang, X.; McCabe, M.F.; Duarte, C.M. Use of Unmanned Aerial Vehicles for Efficient Beach Litter Monitoring. *Mar. Pollut. Bull.* **2018**, *131*, 662–673. [CrossRef]
16. Muchiri, N.; Kimathi, S. A Review of Applications and Potential Applications of UAV. In Proceedings of the 2016 Annual Conference on Sustainable Research and Innovation, Juja, Kenya, 4 May 2016; p. 4.
17. Géron, A. *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow*, 2nd ed.; O'Reilly Media: Sebastopol, ON, Canada, 2019; ISBN 978-1-4920-3264-9.
18. Proença, P.F.; Simões, P. TACO: Trash Annotations in Context for Litter Detection. *arXiv* **2020**, arXiv:2003.06975.
19. Abid, M. Bottles and Cans Images. Available online: https://www.kaggle.com/datasets/moezabid/bottles-and-cans (accessed on 25 October 2022).
20. Abla, M. Garbage Classification (12 Classes). Available online: https://www.kaggle.com/datasets/126ab2c7f7e22add276bc29e44b97f635e3f6a04368afb20130a83518a9056b9 (accessed on 25 October 2022).
21. Nokia Nokia X10 Mobile. Available online: https://www.nokia.com/phones/en_int/nokia-x-10 (accessed on 25 October 2022).
22. Martin, C.; Zhang, Q.; Zhai, D.; Zhang, X.; Duarte, C.M. Enabling a Large-Scale Assessment of Litter along Saudi Arabian Red Sea Shores by Combining Drones and Machine Learning. *Environ. Pollut.* **2021**, *277*, 116730. [CrossRef]
23. US National Geospatial-Intelligence Agency. Administrative Boundaries World 1995. Available online: https://earthworks.stanford.edu/catalog/tufts-worldboundaries95 (accessed on 21 October 2022).
24. UM_GeoLab Ramla Bay May 2019—3D Model by UM_GeoLab (@UM_Geo_Lab). Available online: https://sketchfab.com/models/f0d48f607b634fe4a2c8ab16d66c86ea/embed?autostart=1 (accessed on 26 November 2022).
25. Wada, K. Wkentaro/Labelme. Available online: https://github.com/wkentaro/labelme (accessed on 3 June 2022).
26. Ultralytics YOLOv5. Available online: https://github.com/ultralytics/yolov5 (accessed on 3 June 2022).
27. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. *arXiv* **2016**, arXiv:1506.02640.
28. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *arXiv* **2016**, arXiv:1506.01497. [CrossRef]
29. Padilla, R.; Netto, S.L.; da Silva, E.A.B. A Survey on Performance Metrics for Object-Detection Algorithms. In Proceedings of the 2020 International Conference on Systems, Signals and Image Processing (IWSSIP), Niterói, Brazil, 1–3 July 2020; pp. 237–242.
30. Pix4D Support. TOOLS—GSD Calculator. Available online: http://support.pix4d.com/hc/en-us/articles/202560249-TOOLS-GSD-calculator (accessed on 16 June 2022).
31. US National Geospatial-Intelligence Agency. World Geodetic System 1984, Its Definition and Relationships with Local Geodetic Systems. Version 1.0.0; National Geospatial-Intelligence Agency (NGA) Standardization Document; 2014. Available online: https://earth-info.nga.mil/php/download.php?file=coord-wgs84 (accessed on 16 June 2022).
32. Bugayevskiy, L.M.; Snyder, J. *Map Projections: A Reference Manual*, 1st ed.; Taylor & Francis: London, UK, 1995; ISBN 978-0-429-15984-8.
33. Taveira-Pinto, F.; Silva, R.; Pais-Barbosa, J. Coastal Erosion Along the Portuguese Northwest Coast Due to Changing Sediment Discharges from Rivers and Climate Change. In *Global Change and Baltic Coastal Zones*; Schernewski, G., Hofstede, J., Neumann, T., Eds.; Coastal Research Library; Springer: Dordrecht, The Netherlands, 2011; pp. 135–151. ISBN 978-94-007-0400-8.
34. Bird, E.; Lewis, N. Causes of Beach Erosion. In *Beach Renourishment*; Bird, E., Lewis, N., Eds.; SpringerBriefs in Earth Sciences; Springer International Publishing: Cham, Switzerland, 2015; pp. 7–28. ISBN 978-3-319-09728-2.
35. National Oceanic and Atmospheric Administration What Is LIDAR. Available online: https://oceanservice.noaa.gov/facts/lidar.html (accessed on 8 November 2022).
36. Stockdonf, H.F.; Sallenger, A.H., Jr.; List, J.H.; Holman, R.A. Estimation of Shoreline Position and Change Using Airborne Topographic Lidar Data. *J. Coast. Res.* **2002**, *18*, 502–513.
37. Sallenger, A.H.; Krabill, W.B.; Swift, R.N.; Brock, J.; List, J.; Hansen, M.; Holman, R.A.; Manizade, S.; Sontag, J.; Meredith, A.; et al. Evaluation of Airborne Topographic Lidar for Quantifying Beach Changes. *J. Coast. Res.* **2003**, *19*, 125–133.
38. Colica, E.; Micallef, A.; D'Amico, S.; Cassar, L.F.; Galdies, C. Investigating the Use of UAV Systems for Photogrammetric Applications: A Case Study of Ramla Bay (Gozo, Malta). *Xjenza Online* **2017**, *5*, 125–131. [CrossRef]
39. Guastella, D.C.; Cantelli, L.; Melita, C.D.; Muscato, G. A Global Path Planning Strategy for a UGV from Aerial Elevation Maps for Disaster Response. In Proceedings of the 9th International Conference on Agents and Artificial Intelligence, Porto, Portugal, 24–26 February 2017; SCITEPRESS—Science and Technology Publications: Porto, Portugal, 2017; pp. 335–342.
40. Agisoft LLC Agisoft Metashape Professional Edition 2021. Available online: https://www.agisoft.com/features/professional-edition/ (accessed on 25 November 2022).
41. Topcon HiPer HR. Available online: https://www.topconpositioning.com/na/gnss-and-network-solutions/integrated-gnss-receivers/hiper-hr (accessed on 26 November 2022).
42. Stentz, A. *The D*Algorithm for Real-Time Planning of Optimal Traverses*; The Robotics Institute, Carnegie Mellon University: Pittsburgh, PA, USA, 1994; p. 34.

43. Ankerst, M.; Breunig, M.M.; Kriegel, H.-P.; Sander, J. OPTICS: Ordering Points to Identify the Clustering Structure. *SIGMOD Rec.* **1999**, *28*, 49–60. [CrossRef]
44. Ester, M.; Kriegel, H.-P.; Sander, J.; Xu, X. A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. *Knowl. Discov. Data Min.* **1996**, *96*, 1996.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.