



Article

# Object Detection and Classification Based on YOLO-V5 with Improved Maritime Dataset

Jun-Hwa Kim <sup>1</sup> , Namho Kim <sup>1</sup> , Yong Woon Park <sup>2</sup> and Chee Sun Won <sup>1,\*</sup> 

<sup>1</sup> Department of Electrical and Electronic Engineering, Dongguk University-Seoul, 30, Pildong-ro 1-gil, Jung-gu, Seoul 04620, Korea; jhkim414@dongguk.edu (J.-H.K.); namho96@dgu.ac.kr (N.K.)

<sup>2</sup> Department of Autonomous Things Intelligence, Graduate School, Dongguk University-Seoul, 30, Pildong-ro 1-gil, Jung-gu, Seoul 04620, Korea; yongwoon5901@gmail.com

\* Correspondence: cswon@dongguk.edu

**Abstract:** SMD (Singapore Maritime Dataset) is a public dataset with annotated videos, and it is almost unique in the training of deep neural networks (DNN) for the recognition of maritime objects. However, there are noisy labels and imprecisely located bounding boxes in the ground truth of the SMD. In this paper, for the benchmark of DNN algorithms, we correct the annotations of the SMD dataset and present an improved version, which we coined SMD-Plus. We also propose augmentation techniques designed especially for the SMD-Plus. More specifically, an online transformation of training images via Copy & Paste is applied to solve the class-imbalance problem in the training dataset. Furthermore, the mix-up technique is adopted in addition to the basic augmentation techniques for YOLO-V5. Experimental results show that the detection and classification performance of the modified YOLO-V5 with the SMD-Plus has improved in comparison to the original YOLO-V5. The ground truth of the SMD-Plus and our experimental results are available for download.

**Keywords:** object detection; maritime dataset; deep learning; data relabel



**Citation:** Kim, J.-H.; Kim, N.; Park, Y.W.; Won, C.S. Object Detection and Classification Based on YOLO-V5 with Improved Maritime Dataset. *J. Mar. Sci. Eng.* **2022**, *10*, 377. <https://doi.org/10.3390/jmse10030377>

Academic Editor: Marco Cococcioni

Received: 17 January 2022

Accepted: 4 March 2022

Published: 6 March 2022

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Public image datasets such as COCO [1] and Pascal visual object classes (VOC) [2] have made a great contribution to the development of deep neural networks (DNN) for computer vision problems [3–8]. These datasets include many different categories of objects. On the other hand, a domain-specific dataset usually contains only a relatively small number of sub-categories under a parent category. For domain-specific applications, obtaining a sufficient number of annotated images is considered a difficult task. Moreover, most domain-specific datasets suffer from the class-imbalance problem and noisy labels. Thus, to overcome the overfitting problem due to these inherent problems in the domain-specific dataset, a DNN model pre-trained by the public image dataset mentioned above is usually adopted for its fine-tuning.

The application areas that make use of domain-specific datasets have been expanding and now include road condition recognition [9,10], face detection [11,12], and food recognition [13,14], among others. Object recognition [15,16] in maritime environments is another important domain-specific problem for various security and safety purposes. For example, an autonomous ship equipped with an Automatic Identification System (AIS) requires safe navigation, which is achieved by the detection of surrounding objects [17]. This is a difficult problem simply because the objects at sea change dynamically due to environmental factors such as illumination, fog, rain, wind, and light reflection. In addition, depending on the viewpoint, the same ship can be shown with quite different shapes. Since the ocean usually has a wide-open view, the ships on the sea can be seen with a variety of sizes and occlusions. That is, large inter-class variances in terms of the size and shape of the maritime objects make the recognition problem very challenging. To tackle these difficulties, we rely on

the recent advancements in DNN. However, the immediate problem of the DNN-based approach is the lack of annotated training data in maritime environments.

Maritime video datasets with annotated bounding boxes and object labels are hardly available. There exist few published datasets, collected especially for object detection in maritime environments [18–20]. Among them, only the Singapore Maritime Dataset (SMD), introduced by Prasad et al. [20], provides sufficiently large video data with labeled bounding boxes for 10 maritime object classes. The SMD consists of onboard and onshore video shots captured by Visual-Optical (VIS) and Near Infrared (NIR) sensors, which can be used for tracking as well as detecting ships on the sea. Although the SMD can be used for the training and testing of DNNs, it is hard to find completely reproducible results published with the SMD for comparative studies. This is due to the fact that the SMD has the following problems. First, there are bounding boxes in the ground truth of the SMD with inaccurate object boundaries. Some of their bounding boxes are too loose to include the background as well as the whole object. Additionally, some of them are too tight to have only a part of the object. Since the maritime images are usually taken from a wide-open view, a faraway object can appear as a tiny one. In this case, a small difference at the border of the bounding box can make a big difference in testing the accuracy of object detection. Second, there are incorrectly labeled classes in the ground truth of the SMD. These noisy labels may not be a big problem for distinguishing the foreground object from the background, but they certainly affect the training and testing of the DNN for the object classification problem. Third, there exists a serious class imbalance in the SMD. The class imbalance can cause the biased training of the DNN in favor of the majority classes and deteriorate the generalization ability of the model. Fourth, there is no proper train/test split in the original SMD.

Note that in [15], they split the SMD into training, validation, and testing subsets. Using the split datasets, they also provided the benchmark results for the object detection via the Mask R-CNN model. However, their benchmark results were about object detection, with no further classification for each detected object. In fact, most of the previous research works that used the dataset only dealt with object detection [15,21,22]. However, for applications in maritime security such as in the use of Unmanned Surface Vehicles (USV), we also need to identify the type of the detected object [23]. Since the original SMD includes the class labels of the objects as well as their bounding box information, we may use the SMD for both object detection and classification problems.

Although the SMD provides the class label for each object with a bounding box, as already mentioned, there are still noisy labels. Furthermore, the split dataset provided by [15] suffers from the class-imbalance problem (e.g., no data assigned for some of the object classes such as Kayak and Swimming Person in the training subset). In this paper, by using the SMD as a benchmark dataset for both detection and classification tasks, we fix its imprecisely determined bounding boxes and noisy labels. To alleviate the class-imbalance problem, we discard rare classes such as ‘swimming person’ and ‘flying bird and plane’. In addition, we merge the ‘boat’ and ‘speed boat’ labels and thus propose a modified SMD (coined SMD-Plus) with seven maritime object classes.

Hence, in having the SMD-Plus dataset, we are able to provide benchmark results for the detection and classification (detection-then-classification) problem. That is, based on the YOLO-V5 model [24], we modify its augmentation techniques through the consideration of the maritime environments. More specifically, an Online Copy & Paste is applied to alleviate the imbalance problem in the training process. Likewise, the original augmentation techniques of the YOLO-V5 such as the geometric transformation, mosaic, and mix-up of the YOLO-V5 are adjusted especially for the SMD-Plus.

The contributions of this paper can be summarized as follows:

- (i) We have improved the existing SMD dataset by removing noisy labels and fixing the bounding boxes. It is expected that the improved dataset of the SMD-Plus will be used as a benchmark dataset for the detection and classification of objects in maritime environments.

- (ii) In addition to the YOLO-V5 augmentation techniques, we proposed the Online Copy & Paste and Mix-up methods for the SMD-Plus. Our Online Copy & Paste scheme has significantly improved the classification performance for the minority classes, thus alleviating the class-imbalance problem in the SMD-Plus.
- (iii) The ground truth table for the SMD-Plus and the results of the detection and classification are open to the public and may be downloaded from the following website (accessed on 2 March 2022): <https://github.com/kjunhwa/Singapore-Maritime-Dataset-Plus>.

## 2. Related Work

### 2.1. Maritime Dataset

In domain-specific DNN applications, it is of vital importance to obtain a proper dataset for training. However, for some domain-specific problems, it is quite difficult to obtain publically available datasets. Depending on the target domain, it is often expensive to collect images for specific classes and annotate them. Moreover, security and proprietary rights often prevent the owners from opening their datasets. One such domain-specific dataset is the maritime dataset. Maritime datasets can be classified into three groups [25]: (i) datasets for object detection [19], (ii) datasets for object classification [26], (iii) datasets for both object detection and classification [20]. The dataset for object detection provides the location information of the objects in the image with their bounding boxes, while no class label is given for each object. On the other hand, in the dataset for both object detection and classification, each image includes multiple objects with their bounding boxes and class labels. Finally, there is only a single maritime object in an image from the dataset for object classification.

Although the SMD [20] provides the ground truth of video objects and their class labels for both object detection and classification, there are no benchmark results reported from the SMD. This is due to the fact that the original SMD is not quite ready for training DNN models. Moosbauer et al. [15] analyzed the SMD and proposed the split sub-datasets of ‘train, validation, and test’. After applying Mask R-CNN on their split sub-datasets, they then reported the foreground object detection results. However, for both object detection and classification tasks, their split sub-datasets of train, validation, and test may not be appropriate for training the DNNs. Note that there certainly exist noisy labels in the SMD, which cause no problems in detection but negatively affect the DNN training for the classification. Additionally, due to the class-imbalance problem of the SMD, some of the split sub-datasets in [15] only have a few or even no data in a certain class of the test dataset. The SMD has been combined with other existing maritime datasets to resolve the limitations. For example, to expand the SMD dataset, Shin et al. [22] exploited the public datasets for classification such as MARVEL [18] by pasting copies of the objects in MARVEL into the SMD dataset. Furthermore, in Nalamati et al. [23], the SMD was combined with the SeaShips [19] dataset. However, these combined datasets were only used for detection. Moreover, due to the lack of dataset-combining details, it is hard to reproduce and compare the results. The Maritime Detection Classification and Tracking benchmark (MarDCT) [27] provided maritime datasets for detection, classification, and tracking separately. Therefore, it is inappropriate to use them for the classification of detected objects with bounding boxes.

### 2.2. Object Detection Models

Although improved versions of R-CNN [3], such as Faster R-CNN [4] and cascade R-CNN [28], were proposed to speed up the inference, the two-stage architectures of the R-CNN generically limit the processing speed. This has motivated researchers to develop one-stage DNNs such as YOLO [29], SSD [8], and RetinaNet [7] for object detection. Unlike the R-CNN, YOLO performs classification and bounding box regression at the same time, thus reducing the processing time. To further improve the accuracy and speed performance, the first YOLO has been refined to YOLO-V3 [6], YOLO-V4 [30], and YOLO-V5 [24]. The SSD [8] is another model of the one-stage object detector. For the anchor box of the YOLO, the SSD uses a predefined default box and has a scale-invariant feature by using a number

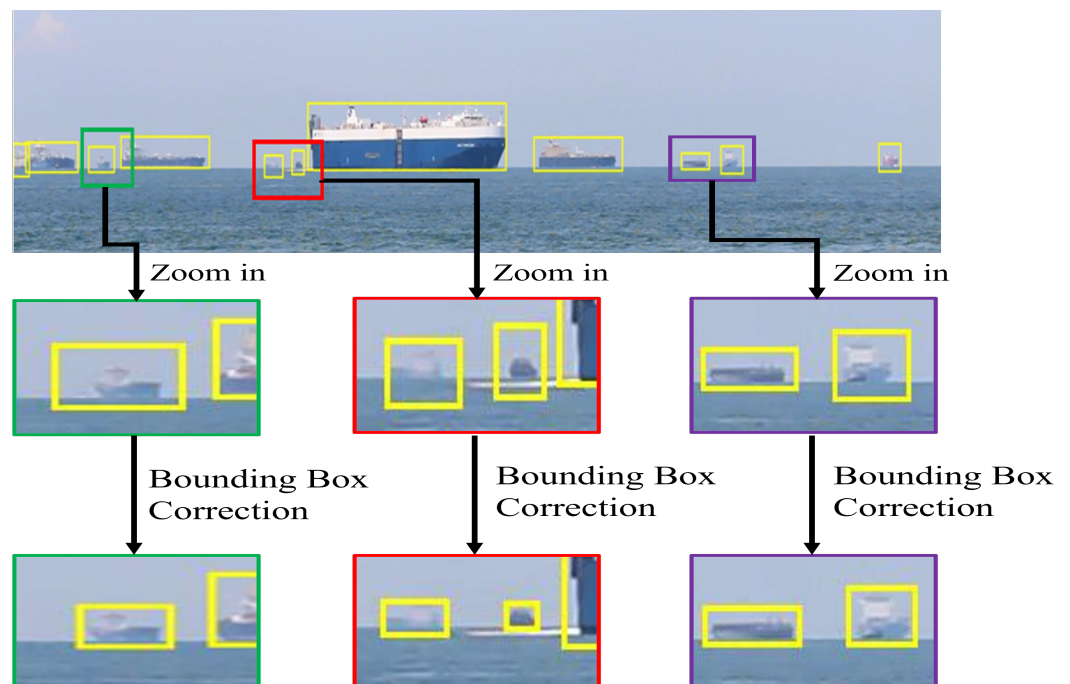
of feature maps obtained from the middle layer of the backbone. RetinaNet [7] also adopts the one-stage framework with a modified focal loss, which assigns small weights to easily detectable objects but large weights to objects that are difficult to handle.

The detectors based on anchor boxes have the disadvantage of being sensitive to hyper-parameters. To solve this problem, anchor-free methods such as FCOS [31] have been proposed. However, since FCOS [31] performs pixel-wise bounding box prediction, it takes more time to execute the detection-then-classification task. Since the real-time requirement is essential for autonomous surveillance, we focus on using the fast one-stage method of YOLO-V5 [24] as the baseline object detection model.

### 3. Improved SMD: SMD-Plus

The SMD provides high-quality videos with ground truth for 10 types of objects in marine environments. Since the ground truth of the SMD was created by non-expert volunteers, it includes some label errors and imprecise bounding boxes. Those ambiguous and incorrect class labels in the ground truth make it difficult to use the SMD as a benchmark dataset for maritime object classification. Therefore, most of the researches making use of the SMD only deal with object detection, with no classification of the detected objects. To make use of the SMD for the detection-then-classification purpose, our first task was to revise and improve its imprecise annotations.

To train a DNN for object detection, we needed the location and size information of the bounding boxes. Note that unlike the datasets with general objects, the background regions of sea and sky in the maritime datasets, similar to the SMD, usually take up much larger areas in the image than the target objects of ships. Therefore, the precise bounding box annotations for the small maritime objects are of importance, and even a small mislocation of the bounding box for the small object can make a huge difference in the training and testing of the DNNs. Figure 1 shows examples of inaccurate bounding boxes in the original SMD. More specifically, the yellow bounding boxes within the zoomed red, green, and purple boxes in the top image of Figure 1 are too loose and mislocated. These bounding boxes are refined in the bottom part of the figure.



**Figure 1.** The original bounding boxes of the original SMD in the top image are refined in those at the bottom.

The ground truth annotation of the SMD for each maritime object provides one of ten class labels as well as its bounding box information of location and size. However, there are quite a few noisy labels in the SMD. In addition, there are indistinguishable classes that need to be merged. For example, as shown in Figure 2, the two ships from the apparently identical class are assigned the different labels of ‘Speed boat’ and ‘Boat’. Therefore, in our improved version of the SMD-Plus, we are going to merge the two classes of ‘Speed boat’ and ‘Boat’ into a single class of ‘Boat’. Another motivation to combine these two classes is that the number of image data for the two classes is not sufficient for training and testing.

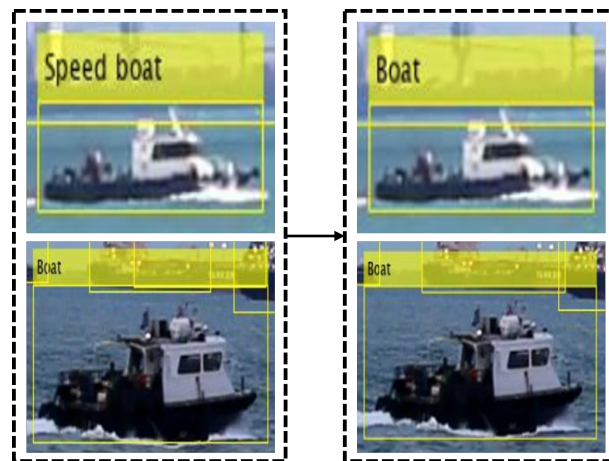


Figure 2. Integration of ‘Speed boat’ into ‘Boat’.

The similar-looking ships in the top part of Figure 3b have two different labels of ‘Speed boat’ and ‘Ferry’, and one of them must be incorrect. In the SMD, most of the ships labeled as ‘Ferry’ are the ones that can carry many passengers, as shown on Figure 3a. By this definition of ‘Ferry’, we can correct the class label of ‘Ferry’ into ‘Boat’, as seen in the bottom part of Figure 3b.

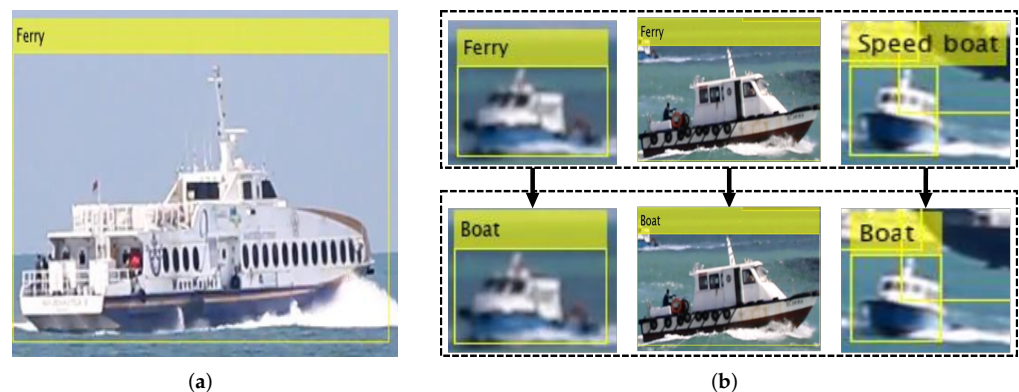


Figure 3. Example of noisy label correction in the SMD: (a) A typical image for ‘Ferry’, (b) Noisy labels in the top and their corrected ones at the bottom.

Next, we point out the problem of the ‘Other’ classification in the SMD. We noticed that the SMD included a clearly identifiable ‘Person’ in the ‘Other’ class, as seen in Figure 4a, as well as blurred unidentifiable objects, as seen in Figure 4b. This makes the definition of the label ‘Other’ rather fuzzy. Therefore, we assigned the ‘Other’ classification only to unidentifiable objects, excluding rare objects such as the ‘Person’ from the class.



**Figure 4.** Examples of the ‘Other’ class in the SMD: (a) Deleted object from the ‘Other’ class, (b) Remained objects in the ‘Other’ class.

Since there exist no actual labeled objects for the ‘Flying bird and plane’ and ‘Swimming person’ classes in the SMD, we discarded these two classes. Therefore, putting all the above modifications together, we can summarize the criteria for our SMD revisions as follows:

- (i) ‘Swimming person’ class is empty and is deleted;
- (ii) Non-ship ‘Flying bird and plane’ class is deleted;
- (iii) Visually similar classes of ‘Speed boat’ and ‘Boat’ are merged;
- (iv) Bounding boxes of the original SMD are tightened;
- (v) Some of the missing bounding boxes in ‘Kayak’ are added;
- (vi) According to our redefinitions for the ‘Ferry’ and ‘Other’ classes, some of the misclassified objects in them are corrected.

Our final version of the SMD, coined as SMD-Plus, is quantitatively compared with the original SMD in Table 1.

**Table 1.** The number of objects in each class label for the original SMD and the SMD-Plus.

SMD		SMD-Plus	
Class	Objects(#)	Class	Objects(#)
Boat	1499	Boat	14,021
Speed Boat	7961	Vessel/Ship	125,872
Vessel/Ship	117,436	Ferry	3431
Ferry	8588	Kayak	3798
Kayak	4308	Buoy	3657
Buoy	3065	Sail Boat	1926
Sail Boat	1926	Others	24,993
Others	12,564	Flying bird and plane	-
Flying bird and plane	650	Swimming Person	-
Swimming Person	0		

We needed to split the SMD-Plus into training and testing subsets for the DNNs. Note that the separation of the SMD into train, validation, and test subsets proposed by [15] is good for detection, but not for detection-then-classification. Furthermore, some of the classes in the test subset of the original SMD were empty. Hence, we carefully re-separated the SMD video clips such that they were distributed evenly for all classes in both the train and test subsets as much as possible (see Table 2).



Table 2. Cont.

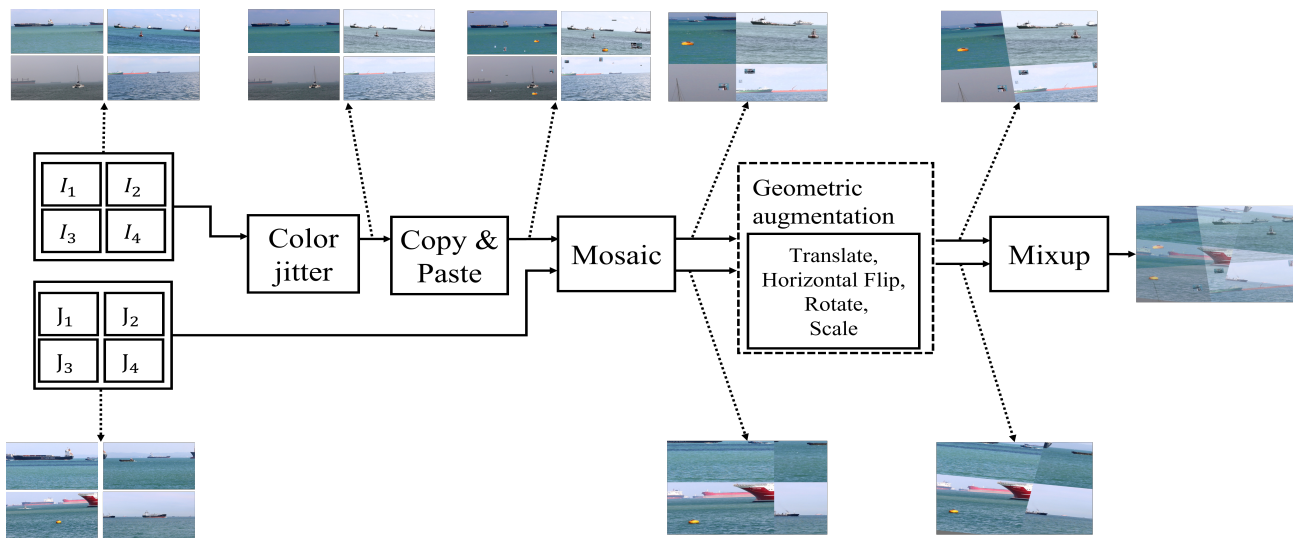
Set	Subset	Video Name	Condition	Number of Objects							
				c1	c2	c3	c4	c5	c6	c7	Total
Test (14)	OnShore (12)	MVI_1469	Daylight	0	600	3600	941	0	0	600	5741
		MVI_1474	Daylight	0	1335	3560	890	0	0	3560	9345
		MVI_1587	Dark/twilight	0	0	6000	600	0	0	586	7186
		MVI_1592	Dark/twilight	0	0	2850	0	683	0	0	3533
		MVI_1613	Daylight	0	0	5750	0	0	0	904	6654
		MVI_1614	Daylight	0	0	5464	582	0	0	934	6980
		MVI_1615	Dark/twilight	0	0	3277	0	0	566	566	4409
		MVI_1644	Daylight	0	0	1008	0	0	0	756	1764
		MVI_1645	Daylight	0	0	3210	0	0	0	0	3210
		MVI_1646	Daylight	0	0	4610	0	0	0	373	4533
		MVI_1448	Hazy	165	0	3624	1590	0	0	19	5398
		MVI_1640	Daylight	302	0	1756	0	0	0	38	2096
	OnBoard (2)	MVI_0799	Daylight	161	0	379	0	0	0	40	580
		MVI_0804	Daylight	0	0	484	0	0	0	980	1464

#### 4. Data Augmentation for YOLO-V5

In this section, we address our detection-then-classification method based on YOLO-V5 with the SMD-Plus dataset. We focus mainly on image augmentation techniques designed especially for the maritime dataset of the SMD-Plus.

Considering the relatively small size and class imbalance problems in the SMD-Plus, data augmentation plays an important role in alleviating the overfitting problem when training the DNNs. As shown in Figure 5, in addition to the basic YOLO-V5 augmentation techniques such as mosaic and geometric transformation, we employ the Online Copy & Paste and Mix-up techniques. That is, to a set of four training images,  $\{I_1, I_2, I_3, I_4\}$ , we first apply color jittering by randomly altering the brightness, hue, and saturation components of the images. Then, the Copy & Paste is performed by inserting the copied objects from other training images into the input images. Next, adding another set of four training images,  $\{J_1, J_2, J_3, J_4\}$ , a random mosaic is applied to both sets of  $\{I_1, I_2, I_3, I_4\}$  and  $\{J_1, J_2, J_3, J_4\}$ . Then, the two mosaic images are geometrically transformed by translation, horizontal flip, rotation, and scaling. Finally, after the geometric transformations, the two images are fused by the Mix-up process. Among the augmentations mentioned previously, the Copy & Paste and the Mix-up are the newly adopted techniques for the basic YOLO-V5 augmentations. Now, we will elaborate on these two techniques in the following subsections.





**Figure 5.** Flow of image augmentations for our YOLO-V5.

#### 4.1. Copy & Paste Augmentation

Copy & Paste augmentation is an effective means of increasing the number of objects for the minority classes, thus alleviating the class-imbalance problem. Here, to enhance the recognition performance for small objects, we can choose smaller objects to be copied as much as possible. To this end, we first divide the objects in the training images into three groups: small (*s*), medium (*m*), and large (*l*). The criterion for the division is given by the size of the rectangular area of the bounding box (see Table 3). Moreover, from Table 1, we can choose more objects from the minority classes for the Copy & Paste to mitigate the class-imbalance problem. Consequently, we first choose the class  $k \in \{1, 2, \dots, K\}$  out of the  $K$  object class with the following probability,  $P_{class}(k)$ :

$$P_{class}(k) = \frac{w_c(k)}{\sum_{i=1}^K w_c(i)} \quad k = 1, 2, \dots, K \quad (1)$$

where  $w_c(k) = N_{min}/N_k$ ,  $N_{min} = \min\{N_1, \dots, N_K\}$ , and  $N_k$  is the number of objects in class  $k$ . By choosing the object to be copied by (1), the minority classes have higher chances of being selected. Once the object from class  $k$  is chosen by (1), we need to select the final object to be copied from one of the three groups of small (*s*), medium (*m*), and large (*l*), determined according to Table 3. The probability of choosing one of the three groups  $P_{size}(k)$  for class  $k$  is given by the following equation:

$$P_{size}(j) = \frac{w_s(j)}{\sum_{i \in \{s, m, l\}} w_s(i)} \quad (2)$$

where  $w_s(j) = \min\{N_k(s), N_k(m), N_k(l)\}/N_k(j)$ , and  $N_k(j)$  is the number of objects for the size of  $j \in \{s, m, l\}$  in the object class  $k$ . Note that  $P_{size}(j)$  in (2) also gives a higher probability for the minority group among small (*s*), medium (*m*), and large (*l*). Since the small-sized (*s*) groups for all class labels usually have the smallest number of objects in the SMD-Plus, the objects in the small-sized group *s* has more chances of being selected than the other groups of *m* and *l*.

**Table 3.** The size criterion for grouping small, medium, and large objects.

	Min Rectangle Area	Max Rectangle Area
Small object	0 × 0	32 × 32
Medium object	32 × 32	96 × 96
Large object	96 × 96	∞ × ∞

In the previous methods, Copy & Paste was executed before training as an offline pre-processing technique. As a consequence, the images pre-processed by the Copy & Paste were used over and over again for every epoch of the training process. To provide more diversified images in training the DNN, for this paper, we apply the Copy & Paste in an on-the-fly manner in order to have an Online Copy & Paste scheme. Now, this Online Copy & Paste creates differently pasted objects for every training epoch, which allows the DNN to be trained with maritime objects of many different sizes and locations.

Next, we need to locate the position in the training image where the copied object is to be pasted, avoiding any overlap between the copied object and the existing ones. This can be performed by calculating the Intersection of Union (IoU) between the candidate position for the paste and the location of the original bounding box. That is, with the equation below, we can check if the IoU for the paste is equivalent to zero. In the object detection area, the IoU measures the overlapping area between the to-be-pasted bounding box  $B_p$  and the existing bounding box  $B_{gt}$  in the ground truth, divided by the area of union between them:

$$IoU = \frac{area(B_p \cap B_{gt})}{area(B_p \cup B_{gt})}. \quad (3)$$

#### 4.2. Mix-up Augmentation

The Mix-up technique [32] is a means of generating a new image by the weighted linear interpolation of two images and their labels. It is known to be effective for mislabeled data because the labels of the two images are mixed, just as their images. More specifically, for the given input images and their label pairs  $(x_i, y_i)$  and  $(x_j, y_j)$  from the training data, the Mix-up can be implemented as follows:

$$\bar{x} = \lambda x_i + (1 - \lambda)x_j \quad (4)$$

$$\bar{y} = \lambda y_i + (1 - \lambda)y_j \quad (5)$$

where  $(\bar{x}, \bar{y})$  are the Mix-up outputs and  $\lambda \in [0, 1]$  is the mixing ratio.

#### 4.3. Basic Augmentations from YOLO-V5

We also use the basic geometric transformations of YOLO-V5 such as flipping, rotation, translation, and scale. Another basic augmentation adopted from YOLO-V5 is the mosaic augmentation. It was first introduced in [30]. The mosaic augmentation mixes four training images into a single training image in order to have four different contexts. According to [30], the mosaic augmentation allows the model to learn how to identify objects on a smaller-than-usual scale, and it is useful for training as it greatly reduces the need for large mini-batch sizes.

### 5. Experiment Results

As explained in the previous section, we revised the SMD in order to obtain the SMD-Plus. As a tool for modifying the ground truth of the SMD, we used the *MATLAB* ImageLabeler tool. The *MATLAB* ImageLabeler provides an application interface to be able to easily create video clips and attach annotations to each object.

Our experiments were conducted on an *Intel I7-9900* Processor with a main memory of 32GB and an *NVIDIA GeForce RTX 2080Ti*. Based on the YOLO-V5, we trained the model with the SMD-Plus. The hyper-parameters for the YOLO-V5 training are as follows: the stochastic gradient descent (SGD) optimizer with a momentum of 0.9, a learning rate of 0.01, and a batch size of 8. We also used the following values for the augmentation parameters:

- For color jittering: hue ranges from 0 to 0.015; saturation, from 0 to 0.7; and brightness, from 0 to 0.4;
- The probability of generating a mosaic is 0.5;
- Translate shifts range from 0 to 0.1;
- The probability of a horizontal flip is 0.5;

- Random rotation within angles from  $-10$  to  $+10$  degrees;
- Random scaling in the range of  $0.5 \times \sim 1.5 \times$ .

Using the same augmentation parameters listed above, for the sake of comparison, we conducted additional experiments with YOLO-V4 [30]. Table 4 compares the detection performance of the SMD and the SMD-Plus. As shown in Table 4, the detection performance of the SMD-Plus compared to the SMD increased by more than 10% for both YOLO-V4 and all versions of YOLO-V5. Here, as in the previous benchmarks [15,21,22], only foreground and background detections were performed. Note that the problem with detecting only the foreground and background is that it can be used to evaluate the accuracy of the bounding box detection, but not the recognition accuracy for the class label. Therefore, we can use the results of Table 4 to verify the bounding box accuracy of the SMD-Plus.

**Table 4.** Comparison of foreground and background detection of the SMD and the SMD-Plus. mAP(0.5) represents the mean average precision (mAP) for IoU = 0.5, while mAP(0.5:0.95) is the averaged mAP for increasing IoU threshold values, from 0.5 to 0.95 by 0.05.

Dataset	Network	mAP(0.5)	mAP(0.5:0.95)
SMD	YOLO-V4	0.704	0.297
	YOLO-V5-S	0.772	0.386
	YOLO-V5-M	0.750	0.403
	YOLO-V5-L	0.766	0.407
SMD-Plus	YOLO-V4	0.847	0.428
	YOLO-V5-S	0.898	0.522
	YOLO-V5-M	0.867	0.528
	YOLO-V5-L	0.878	0.527

Table 5 shows the results of object detection-then-classification task for the train/test split of the SMD, as suggested by [15]. In this train/test split, however, there exist classes with no test data. Therefore, the corresponding classes of columns c1, c5, c7, and c10 are blank. Those non-empty classes for the test set in [15] include ‘Speed boat’, ‘Vessel/ship’, ‘Ferry’, ‘Buoy’, ‘Others’, and ‘Flying bird and Plane’. Fixing the IoU threshold at 0.5, the mAPs for the six non-empty classes are 0.186 for YOLO-V4, 0.22 for YOLO-V5-S, 0.182 for YOLO-V5-M, and 0.304 for YOLO-V5-L.

Next, Table 6 shows the results of the detection-then-classification task for the SMD-Plus. In the table, we can evaluate the performance for the Copy & Paste scheme. More specifically, the detection-then-classification results for ‘No Copy&Paste’, ‘Online Copy&Paste’, and ‘Offline&Paste’ are compared in Table 6. As one can see in the table, our proposed ‘Online Copy&Paste’ outperformed the ‘None’ and ‘Offline Copy&Paste’ methods for YOLO-V4 and all versions of YOLO-V5. Furthermore, the proposed ‘Online&Paste’ has been proven to be quite effective for the minority classes, such as ‘Kayak’ of c6.

**Table 5.** Detection-then-classification results for the SMD dataset: c1: Boat, c2: Speed Boat, c3: Vessel/ship, c4: Ferry, c5: Kayak, c6: Buoy, c7: Sail Boat, c8: Others, c9: Flying bird and plane, c10: Swimming Person.

Dataset	Network	Object Class										mAP (0.5)	mAP (0.5:0.95)
		c1	c2	c3	c4	c5	c6	c7	c8	c9	c10		
SMD	YOLO-V4	-	0.0205	0.657	0.271	-	0.148	-	0.00223	0.000	-	0.186	0.0807
	YOLO-V5-S	-	0.0285	0.657	0.249	-	0.379	-	0.00671	0.000	-	0.22	0.0903
	YOLO-V5-M	-	0.0627	0.706	0.249	-	0.0538	-	0.0213	0.000	-	0.182	0.0817
	YOLO-V5-L	-	0.0879	0.678	0.357	-	0.594	-	0.11	0.000	-	0.304	0.128

**Table 6.** Detection-then-classification results for the SMD-Plus dataset: c1: Ferry, c2: Buoy, c3: Vessel\_ship, c4: Boat, c5: Kayak, c6: Sail\_boat, c7: Others. Columns P and R represent the precision and the recall performance, respectively, for IoU = 0.5.

Dataset	Copy & Paste	Network	Object Class							P	R	mAP	
			c1	c2	c3	c4	c5	c6	c7	0.5	0.5	0.5	0.5:0.95
SMD-Plus	None	YOLO-V4	0.160	0.622	0.868	0.632	0.00995	0.995	0.274	0.476	0.566	0.509	0.258
		YOLO-V5-S	0.372	0.691	0.827	0.569	0.00573	0.995	0.089	0.716	0.517	0.507	0.254
		YOLO-V5-M	0.588	0.882	0.816	0.615	0.00063	0.97	0.111	0.741	0.513	0.569	0.298
		YOLO-V5-L	0.673	0.789	0.846	0.571	0.0123	0.995	0.131	0.803	0.505	0.574	0.286
	Online	YOLO-V4	0.172	0.539	0.868	0.721	0.114	0.995	0.243	0.486	0.621	<b>0.522</b>	<b>0.308</b>
		YOLO-V5-S	0.471	0.864	0.869	0.549	0.162	0.995	0.123	0.650	0.536	<b>0.576</b>	<b>0.291</b>
		YOLO-V5-M	0.588	0.706	0.842	0.607	0.259	0.991	0.123	0.709	0.486	<b>0.588</b>	<b>0.338</b>
		YOLO-V5-L	0.714	0.806	0.828	0.582	0.232	0.995	0.147	0.811	0.534	<b>0.615</b>	<b>0.33</b>
	Offline	YOLO-V4	0.217	0.445	0.881	0.647	0.108	0.995	0.172	0.481	0.610	0.495	0.284
		YOLO-V5-S	0.475	0.386	0.887	0.603	0.0985	0.994	0.152	0.582	0.482	0.514	<b>0.291</b>
		YOLO-V5-M	0.49	0.809	0.852	0.603	0.0592	0.995	0.169	0.724	0.788	0.568	0.309
		YOLO-V5-L	0.618	0.789	0.847	0.667	0.0319	0.995	0.231	0.688	0.541	0.597	0.316

## 6. Conclusions

In this paper, we provided an improved SMD-Plus dataset for future research works on maritime environments. We also adjusted the augmentation techniques of the original YOLO-V5 for the SMD-Plus. In particular, the proposed ‘Online Copy & Paste’ method was proven to be effective in alleviating the class-imbalance problem. Our SMD-Plus dataset and the modified YOLO-V5 are open to the public for future research. We hope that our detection-then-classification model of YOLO-V5 based on the SMD-Plus serves as a benchmark for future research and development initiatives for automated surveillance in maritime environments.

**Author Contributions:** Conceptualization, C.S.W. and J.-H.K.; methodology, J.-H.K.; software, N.K.; validation, J.-H.K. and N.K.; formal analysis, C.S.W.; investigation, J.-H.K. and N.K.; resources, N.K.; data curation, J.-H.K.; writing—original draft preparation, J.-H.K. and N.K.; writing—review and editing, C.S.W.; visualization, J.-H.K. and N.K.; supervision, C.S.W.; project administration, Y.W.P; funding acquisition, Y.W.P. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by the Future Challenge Program through the Agency for Defense Development funded by the Defense Acquisition Program Administration.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** <https://github.com/kjunhwa/Singapore-Maritime-Dataset-Plus> (accessed on 2 March 2022).

**Acknowledgments:** For C.S.W., this work was supported by the Dongguk University Research Fund of 2021.

**Conflicts of Interest:** The authors declare no conflicts of interest.

## References

- Lin, T.Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C.L. Microsoft coco: Common objects in context. In Proceedings of the European Conference on Computer Vision, Zurich, Switzerland, 6–12 September 2014; Springer: Berlin/Heidelberg, Germany, 2014; pp. 740–755.

2. Everingham, M.; Van Gool, L.; Williams, C.K.; Winn, J.; Zisserman, A. The pascal visual object classes (voc) challenge. *Int. J. Comput. Vis.* **2010**, *88*, 303–338. [[CrossRef](#)]
3. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Washington, DC, USA, 23–28 June 2014; pp. 580–587.
4. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. *arXiv* **2015**, arXiv:1506.01497.
5. Cai, Z.; Vasconcelos, N. Cascade R-CNN: High quality object detection and instance segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2019**, *43*, 1483–1498. [[CrossRef](#)] [[PubMed](#)]
6. Redmon, J.; Farhadi, A. Yolov3: An incremental improvement. *arXiv* **2018**, arXiv:1804.02767.
7. Lin, T.Y.; Goyal, P.; Girshick, R.; He, K.; Dollár, P. Focal loss for dense object detection. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2980–2988.
8. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. Ssd: Single shot multibox detector. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 11–14 October 2016; Springer: Berlin/Heidelberg, Germany, 2016; pp. 21–37.
9. Shim, S.; Cho, G.C. Lightweight semantic segmentation for road-surface damage recognition based on multiscale learning. *IEEE Access* **2020**, *8*, 102680–102690. [[CrossRef](#)]
10. Yuan, Y.; Islam, M.S.; Yuan, Y.; Wang, S.; Baker, T.; Kolbe, L.M. EcRD: Edge-cloud Computing Framework for Smart Road Damage Detection and Warning. *IEEE Internet Things J.* **2020**, *8*, 12734–12747. [[CrossRef](#)]
11. Li, X.; Lai, S.; Qian, X. DBCFace: Towards Pure Convolutional Neural Network Face Detection. *IEEE Trans. Circuits Syst. Video Technol.* **2021**, early access. [[CrossRef](#)]
12. Zhang, S.; Chi, C.; Lei, Z.; Li, S.Z. Refineface: Refinement neural network for high performance face detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, *43*, 4008–4020. [[CrossRef](#)] [[PubMed](#)]
13. Zhao, H.; Yap, K.H.; Kot, A.C.; Duan, L. Jdnet: A joint-learning distilled network for mobile visual food recognition. *IEEE J. Sel. Top. Signal Process.* **2020**, *14*, 665–675. [[CrossRef](#)]
14. Won, C.S. Multi-scale CNN for fine-grained image recognition. *IEEE Access* **2020**, *8*, 116663–116674. [[CrossRef](#)]
15. Moosbauer, S.; Konig, D.; Jakel, J.; Teutsch, M. A benchmark for deep learning based object detection in maritime environments. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, Long Beach, CA, USA, 16–17 June 2019; pp. 916–925.
16. Liu, T.; Pang, B.; Zhang, L.; Yang, W.; Sun, X. Sea Surface Object Detection Algorithm Based on YOLO v4 Fused with Reverse Depthwise Separable Convolution (RDSC) for USV. *J. Mar. Sci. Eng.* **2021**, *9*, 753. [[CrossRef](#)]
17. Gao, M.; Shi, G.; Li, S. Online Prediction of Ship Behavior with Automatic Identification System Sensor Data Using Bidirectional Long Short-Term Memory Recurrent Neural Network. *Sensors* **2018**, *18*, 4211. [[CrossRef](#)] [[PubMed](#)]
18. Gundogdu, E.; Solmaz, B.; Yücesoy, V.; Koc, A. Marvel: A large-scale image dataset for maritime vessels. In Proceedings of the Asian Conference on Computer Vision, Taipei, Taiwan, 20–24 November 2016; Springer: Berlin/Heidelberg, Germany, 2016; pp. 165–180.
19. Shao, Z.; Wu, W.; Wang, Z.; Du, W.; Li, C. Seaships: A large-scale precisely annotated dataset for ship detection. *IEEE Trans. Multimed.* **2018**, *20*, 2593–2604. [[CrossRef](#)]
20. Prasad, D.K.; Rajan, D.; Rachmawati, L.; Rajabally, E.; Quek, C. Video processing from electro-optical sensors for object detection and tracking in a maritime environment: A survey. *IEEE Trans. Intell. Transp. Syst.* **2017**, *18*, 1993–2016. [[CrossRef](#)]
21. Zhang, Y.; Li, Q.Z.; Zang, F.N. Ship detection for visual maritime surveillance from non-stationary platforms. *Ocean Eng.* **2017**, *141*, 53–63. [[CrossRef](#)]
22. Shin, H.C.; Lee, K.I.; Lee, C.E. Data augmentation method of object detection for deep learning in maritime image. In Proceedings of the 2020 IEEE International Conference on Big Data and Smart Computing (BigComp), Busan, Korea, 19–22 February 2020; pp. 463–466.
23. Nalamati, M.; Sharma, N.; Saqib, M.; Blumenstein, M. Automated Monitoring in Maritime Video Surveillance System. In Proceedings of the 2020 35th International Conference on Image and Vision Computing New Zealand (IVCNZ), Wellington, New Zealand, 25–27 November 2020; pp. 1–6.
24. YOLO-V5. Available online: [ultralytics.com/yolov5:V3.0](https://ultralytics.com/yolov5/v3.0) (accessed on 13 August 2020).
25. Qiao, D.; Liu, G.; Lv, T.; Li, W.; Zhang, J. Marine Vision-Based Situational Awareness Using Discriminative Deep Learning: A Survey. *J. Mar. Sci. Eng.* **2021**, *9*, 397. [[CrossRef](#)]
26. Zhao, R.; Wang, J.; Zheng, X.; Wen, J.; Rao, L.; Zhao, J. Maritime Visible Image Classification Based on Double Transfer Method. *IEEE Access* **2020**, *8*, 166335–166346. [[CrossRef](#)]
27. Bloisi, D.D.; Iocchi, L.; Pennisi, A.; Tombolini, L. ARGOS-Venice Boat Classification. In Proceedings of the 2015 12th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), Karlsruhe, Germany, 25–28 August 2015; pp. 1–6. [[CrossRef](#)]
28. Cai, Z.; Vasconcelos, N. Cascade r-cnn: Delving into high quality object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 6154–6162.

29. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.
30. Bochkovskiy, A.; Wang, C.Y.; Liao, H.Y.M. Yolov4: Optimal speed and accuracy of object detection. *arXiv* **2020**, arXiv:2004.10934.
31. Tian, Z.; Shen, C.; Chen, H.; He, T. Fcos: Fully convolutional one-stage object detection. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Korea, 27–28 October 2019; pp. 9627–9636.
32. Zhang, H.; Cisse, M.; Dauphin, Y.N.; Lopez-Paz, D. mixup: Beyond empirical risk minimization. *arXiv* **2017**, arXiv:1710.09412.