




Article

# A Novel YOLOv3 Algorithm-Based Deep Learning Approach for Waste Segregation: Towards Smart Waste Management

Saurav Kumar <sup>1</sup>, Drishti Yadav <sup>1</sup>, Himanshu Gupta <sup>1</sup>, Om Prakash Verma <sup>1</sup>, Irshad Ahmad Ansari <sup>2</sup>  
and Chang Wook Ahn <sup>3,\*</sup>

<sup>1</sup> Department of Instrumentation and Control Engineering, Dr. B. R. Ambedkar National Institute of Technology Jalandhar, Punjab 144011, India; srvnitj@gmail.com (S.K.); drish131196@gmail.com (D.Y.); guptah.nitj@gmail.com (H.G.); vermaop@nitj.ac.in (O.P.V.)

<sup>2</sup> Electronics and Communication Engineering, PDPM Indian Institute of Information Technology, Design and Manufacturing, Jabalpur 482005, India; irshad@iiitdmj.ac.in

<sup>3</sup> AI Graduate School, Gwangju Institute of Science and Technology (GIST), Gwangju 61005, Korea

\* Correspondence: cwan@gist.ac.kr

**Abstract:** The colossal increase in environmental pollution and degradation, resulting in ecological imbalance, is an eye-catching concern in the contemporary era. Moreover, the proliferation in the development of smart cities across the globe necessitates the emergence of a robust smart waste management system for proper waste segregation based on its biodegradability. The present work investigates a novel approach for waste segregation for its effective recycling and disposal by utilizing a deep learning strategy. The YOLOv3 algorithm has been utilized in the Darknet neural network framework to train a self-made dataset. The network has been trained for 6 object classes (namely: cardboard, glass, metal, paper, plastic and organic waste). Moreover, for comparative assessment, the detection task has also been performed using YOLOv3-tiny to validate the competence of the YOLOv3 algorithm. The experimental results demonstrate that the proposed YOLOv3 methodology yields satisfactory generalization capability for all the classes with a variety of waste items.

**Keywords:** Convolutional Neural Network (CNN); deep learning; object detection; waste management; waste segregation; YOLOv3 algorithm



**Citation:** Kumar, S.; Yadav, D.; Gupta, H.; Verma, O.P.; Ansari, I.A.; Ahn, C.W. A Novel YOLOv3 Algorithm-Based Deep Learning Approach for Waste Segregation: Towards Smart Waste Management. *Electronics* **2021**, *10*, 14. <https://dx.doi.org/10.3390/electronics10010014>

Received: 30 November 2020

Accepted: 17 December 2020

Published: 24 December 2020

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



**Copyright:** © 2020 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

The rapid explosion in industrialization, urbanization, and global population rate is an attention-grabbing concern, pertaining to environmental degradation. With the global population expanding at an alarming rate, there has been terrific degradation of the environment, resulting in its dreadful condition. As per the published report (2019), India annually generates more than 62 million tons (MT) of solid waste, out of which only 43 MT of the waste is collected, 11.9 MT is treated and almost 31 MT is dumped in landfill sites [1]. Attributable to the existing environmental concerns and improper management of waste, the world encounters gargantuan deleterious effects on the economy, public health and, essentially, the environment. This has shifted the above-all focus towards the worldwide progression of smart cities for guaranteeing effective and smart urban waste management. Moreover, the recycling of waste opens the gateway for research and development and introduces waste to the wealthy business model for sustainable development. However, concerns have been raised towards the requisite of segregation of waste based on its biodegradable and non-biodegradable behavior.

Usually, in the Indian context, wastes consist of paper, plastic, rubber, metal, glass, textiles, organics, sanitary products, electricals and electronics, hazardous substances (paint, spray and chemical) and infectious materials (hospital and clinical), which can be broadly classified as biodegradable (BD) and nonbiodegradable (NBD) waste with their respective share of 52% and 48% [2]. Further, according to the recent Indian government reports, the

most common things that have been thrown in the garbage are paper, paper boxes, food, and glass [3]. These things constitute more than 99.5% of total garbage collected, which clearly indicates that people throw dry and wet waste away together. Efficacious waste segregation would assist in the proper disposal and recycling of these wastes based on their biodegradability. Thus, the present era dictates the evolution of a smart waste segregation system to allude to the aforementioned causes of ecological ruin. The segregation of waste is, consequently, seeking attention from various researchers and academicians across the globe [4,5].

## 2. Motivation

The appropriate classification and organization of wastes into various categories (such as recyclable, biodegradable, nonbiodegradable, organic, harmful, etc.) helps in the proper utilization and disposal of wastes. For waste segregation, computer vision may provide cost effective solutions to identify, classify, and separate out the waste from the humungous dumps of garbage and trash. Owing to the unparalleled developments in the arena of computer vision, waste segregation has been made possible via the identification and detection of wastes through the effective utilization of various approaches for object detection. Usually, the drawbacks of classical object detection techniques (based on Haar cascade classifier, SVMs (Support Vector Machines) or Sliding Window methods) are overcome by deep learning models, for instance, deep convolutional neural networks (DCNNs). Attributable to the ground-breaking achievements in object detection and image classification, researchers across the globe are focusing on the advancements in deep learning approaches for difficult object detection tasks [6–10]. Various deep learning techniques (including CNN, i.e., Convolutional Neural Network [11], R-CNN (Region-based Convolutional Neural Network) family, SSD (Single-shot detector) family [12,13], YOLO (You Only Look Once) [14], etc.) and other classification techniques (SVM and, MLP, i.e., Multi-layer perceptron) [15] have demonstrated their efficiency in performing complex and sophisticated object detection tasks [16,17]. In the context of waste segregation, researchers have shown the competencies of various deep learning strategies in the detection of garbage and its classification [4,5,18,19]. These approaches include CNN and MLP [20], and Faster R-CNN. However, owing to the slow computational capability and slow response time of these approaches, on account of their pipeline execution framework [21], YOLOv3, an improved version of the YOLO [22] algorithm, based on CNNs, emerges to be one of the proficient and promising deep learning algorithms for object detection tasks. The YOLOv3 algorithm yields exceptional results in terms of real-time response rate. Hence, YOLOv3 has established its supremacy over other existing algorithms, along with the members of YOLO family (YOLO, YOLOv2, YOLOv2-tiny and YOLOv3-tiny). Additionally, the efficiency of YOLOv3 in waste segregation is still unexplored. Furthermore, the YOLOv3-tiny Algorithm has been used for various object-detection tasks [23]. Even though the speed of detection is improved by YOLOv3-tiny, some of the detection accuracy is lost. Under the umbrella of the above framework, the present work proposes a maiden attempt towards waste segregation via the effective demonstration of the application of the YOLOv3 algorithm. The efficient capability of the YOLOv3 algorithm in precise object detection with proximate real-time performance has been endorsed in the present investigation. Furthermore, the effectiveness of the YOLOv3 algorithm has been validated by comparing its performance with the YOLOv3-tiny algorithm.

Conclusively, the objective is to replace the waste segregation process, involving human labor by automated waste management processes, wherein the waste will be classified by deep learning approaches. To sum up, the main goal is to be able to successfully identify and classify waste items based on their broad degradability in a reliable manner. To achieve this goal, the work presented in this paper makes the following key contributions:

- The main contribution of the present investigation is to endorse the efficiency of machine learning and/or deep learning techniques (particularly, YOLO family) for waste segregation based on the broad biodegradable properties of garbage, as these

techniques have never been incorporated in this regard, to the best knowledge and belief of authors.

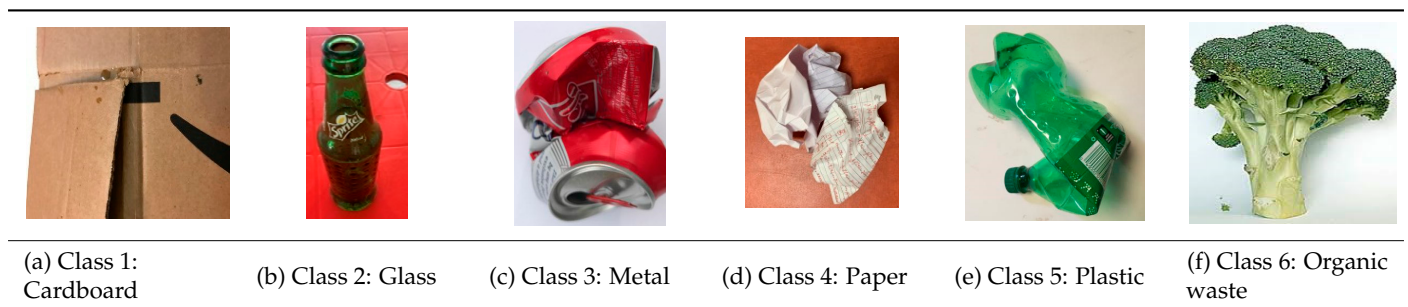
- The other subcontribution of the present investigation is the development of a garbage image dataset, consisting of 6437 images and distributed among six classes (cardboard, glass, plastic, paper, metal, and organic waste), usually visible in household garbage.

The remainder of this paper is organized as follows. Section 3 describes the dataset utilized in the present investigation. A brief sketch of the YOLOv3 algorithm is presented in Section 4. Then, details of the system specifications and parametric settings used to train the model are presented in Section 5. In Section 6, the experimental results are presented and discussed. To end, Section 7 gives the concluding remarks of the present work.

### 3. Dataset

The present investigation emphasizes urban wastes in the vicinity of public areas which are frequently disposed of by commuters, pedestrians and occasionally during commercial events. Here, we examine a number of waste items commonly encountered in the surroundings, including BD and NBD items. However, since this was the very first attempt to segregate these items based on the biodegradable property of the material, therefore, a garbage dataset consisting of the most commonly seen waste items need to be developed. For this purpose, 7826 images were acquired in JPEG format using the camera of an Apple iPhone XR (64GB) with  $1280 \times 960$  pixel resolution. After the preprocessing and cleaning of the collected data, 6437 (82%) images were utilized to form a self-made real time dataset wherein each image was labeled with the name of the class to which it belonged and its type (BD/NBD). In the present investigation, these cleaned sample images were grouped into six classes, namely cardboard, glass, plastic, paper, metal and organic waste, as illustrated in Table 1. Furthermore, a detailed description of the waste items assigned to these defined classes, along with their volume size, is provided in Table 2, and their class distribution is represented in Figure 1.

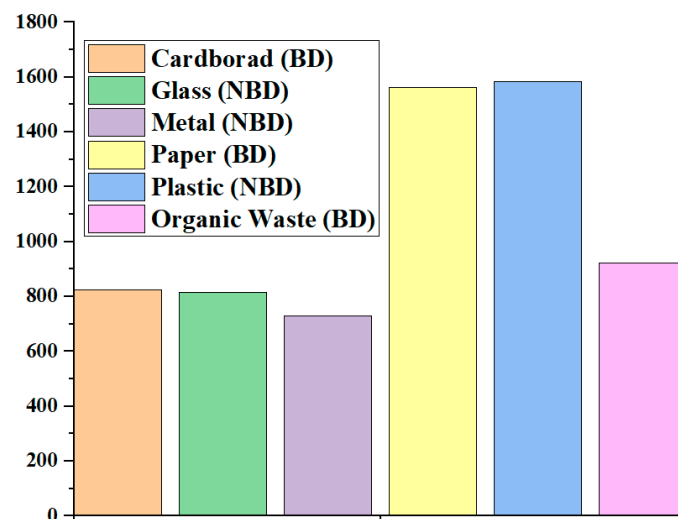
**Table 1.** Illustration of sample images with their respective class.



**Table 2.** Description and statistics of Garbage Images in the dataset.

Class (Type)	Items			Quantity
Cardboard (BD)	Pizza box	Mailing box	Paperboard box	825
	Gift box	Packing box		
Glass (NBD)	Glass Jar	Glass cup	Mirror	816
	Soft drink bottle	Wine Bottle	Window Glass	
Metal (NBD)	Soft Drink Cans	Beer Cans	Blades	730
	Water Bottles			
Paper (BD)	Newspapers	Paper Glasses	Notebooks	1561
	Teacups	Books	Posters	
Plastic (NBD)	Water Bottles	Milk Bottles	Polybags	1583
	Plastic Jars			
Organic Waste (BD)	Fruits	Vegetables		922

BD: Biodegradable, NBD: Non-Biodegradable.

**Figure 1.** Statistical analysis of class distribution of dataset.

Further, in general, most of the waste items presented in Table 2 belong to only one class. However, it has been observed that some captured images of a few items, such as pizza box, have been made by combining a very thin plastic coating and cardboard, but to efficiently differentiate between these with- and without-plastic coating boxes even with human eyes is very challenging. Moreover, the incorporation of these special cases significantly enhances the complexity and computational overhead. Therefore, in the present analysis, these objects have been considered as objects of the parent class. Another issue that arises with the development of the present garbage dataset is that, in some of the acquired images, one class itself contains one or many other classes; for example, polybags which may contain other class items within them. However, because of the limitation of visual object recognition, these kinds of issues have been resolved by considering only the visible object and classifying accordingly.

#### 4. Methodology: YOLOv3 Algorithm

YOLO (You Only Look Once) is one of the most prominent state-of-the-art deep learning techniques [22] which enables simultaneous object detection and classification. To accomplish the object-detection task, earlier techniques (R-CNN and its variations) employed a pipeline execution architecture, which involves multiple steps. Due to the pipeline architecture and the necessity of the separate training of each individual component, slow speed is achieved along with the increased complexity in optimization. These drawbacks



are overcome by YOLO, which transforms object detection into a single regression problem. This performs the simultaneous prediction of multiple bounding boxes and their class probabilities. Unlike sliding window and region proposal-based techniques, the training in YOLO is carried out on full images, thereby directly optimizing the detection performance. However, the real-time speed, end-to-end training capability, along with high average precision and generalization capability of YOLOv3 substantiates its efficiency in performing complex object detection tasks, including significantly small objects.

In general, the YOLOv3 algorithm (as illustrated in Figure 2) simply takes an input image, passes it through a neural network (similar to CNN) to produce an output vector of bounding boxes and class predictions. YOLOv3 extracts a single image, which is then resized to  $416 \times 416$ , that serves as the input to the YOLOv3 neural network. The architecture of the YOLOv3 neural network employed in the darknet-53 framework is illustrated in Figure 3. It consists of convolutional layers, residual layers, upsampling layers, and skip (shortcut) connections. The comprehensive details about the architecture of YOLOv3 are available in an extensive volume of literature works [24].

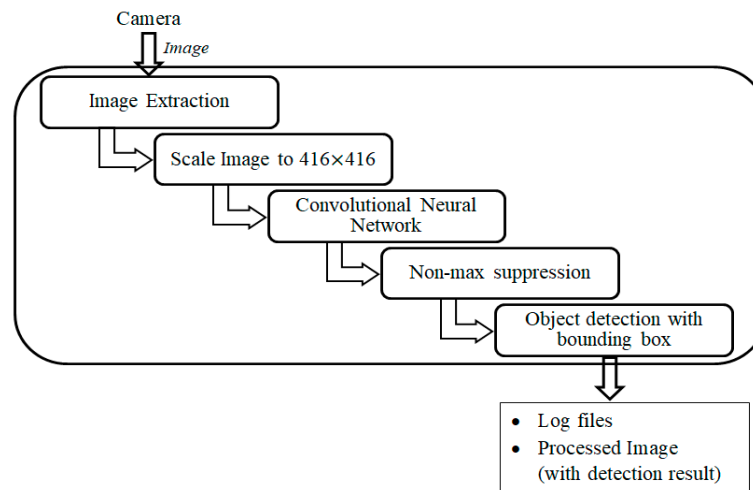


Figure 2. Outline of YOLOv3 algorithm.

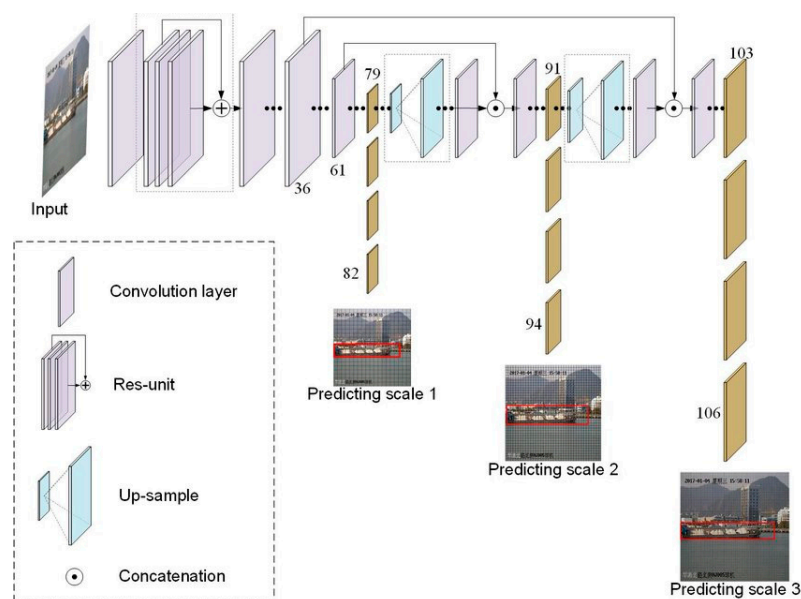


Figure 3. YOLOv3 architecture.

The YOLOv3 neural network takes an input image to return an output vector (Figure 4). The output vector consists of the following parameters:

- Prediction Probability (Pc): A probability that each bounding box contains a detectable object.
- Bounding box properties: Width (Bw), Height (Bh) and Cartesian position (x and y) of the box inside the image (Bx, By).
- Class probabilities (C1, C2, C3, C4, C5, C6): Probabilities that each object within its bounding box is associated with a specific class.

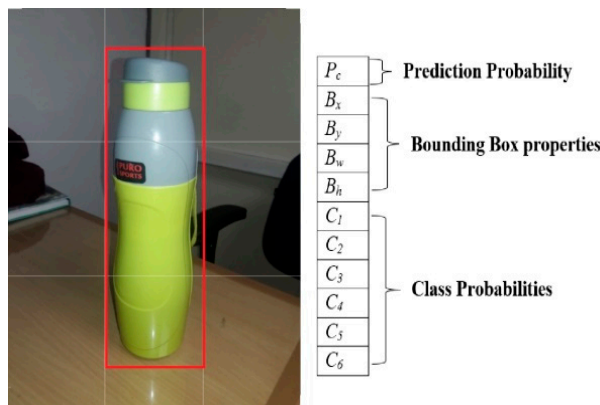


Figure 4. Prediction of Bounding Box.

In YOLOv3, the prediction of bounding boxes is carried out by utilizing the dimension clusters as anchor boxes. Four coordinates ( $t_x, t_y, t_w, t_h$ ) are predicted for each bounding box by the YOLOv3 neural network. From the top left corner of the image, in this case, the cell is offset by ( $C_x, C_y$ ) and the width and height of the bounding box prior are ( $P_w, P_h$ ), then the corresponding predictions are expressed as  $B_x, B_y, B_w$  and  $B_h$ , respectively, as demonstrated in Figure 4. Moreover, if  $\hat{t}^*$  denotes the ground truth corresponding to a certain coordinate prediction, then the difference of the ground truth value (calculated via the ground truth box) and the estimated prediction (i.e.,  $\hat{t}^* - t^*$ ) is the gradient. By inverting the equations mentioned in Figure 5, the ground truth value can be calculated. In YOLOv3, the score of an object for each bounding box is predicted by utilizing logistic regression. The score of the object is 1 if the overlapping of the bounding-box prior is greatest among all other bounding-box priors w.r.t. the ground truth object. The bounding-box priors, other than the best one, are ignored from prediction, even if their overlapping is greater than the threshold (0.5, in this case). Only one bounding box is assigned for each ground truth object in YOLOv3.

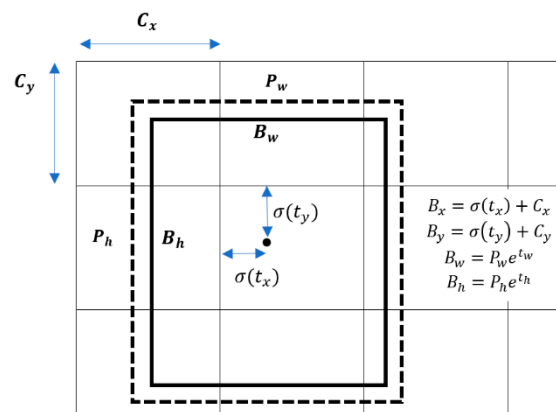


Figure 5. Location prediction via bounding box.

#### 4.1. Performance Parameter Indices

The present investigation examines some of the fundamental key values [24] throughout the training phase to investigate the performance of YOLOv3 in waste segregation. These fundamental key values are as follows.

##### 4.1.1. Precision

Precision is defined in terms of the ratio of the number of objects detected correctly to the number of total objects detected. Mathematically, precision can be computed as expressed by Equation (1).

$$\text{Precision} = \frac{N_{TP}}{N_{TP} + N_{FP}} \quad (1)$$

##### 4.1.2. Recall

Recall is evaluated in terms of the percentage of the number of objects which are correctly detected to the number of ground truth objects. Recall can be evaluated using Equation (2):

$$\text{Recall} = \frac{N_{TP}}{N_{TP} + N_{FN}} \quad (2)$$

where  $N_{TP}$  = Number of True Positives, i.e., number of objects detected correctly;

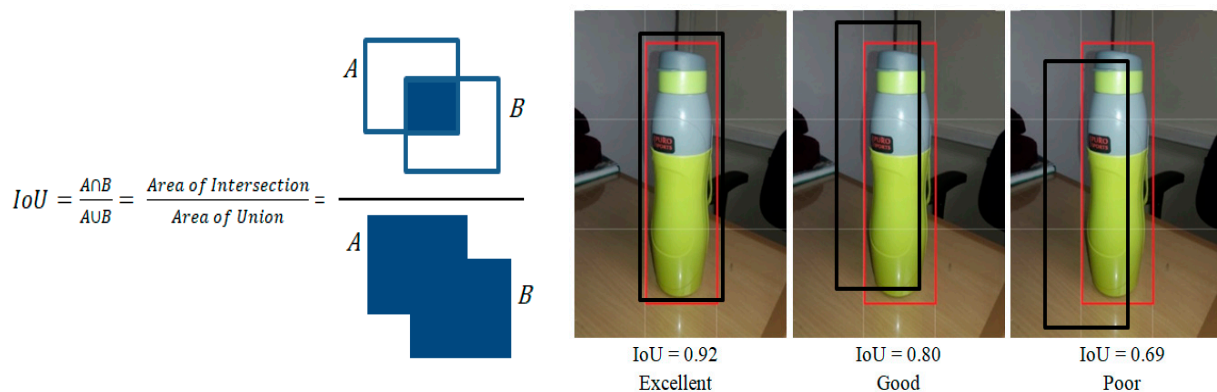
$N_{FP}$  = Number of False Positives, i.e., number of detected objects which could not correspond to the ground truth objects;

$N_{FN}$  = Number of False Negatives, i.e., number of ground truth objects that could not be detected.

##### 4.1.3. Intersection Over Union (IoU)

IoU is a well-known evaluation metric in object detection tasks, which is mathematically represented by Equation (3) and illustrated in Figure 6.

$$\text{IoU} = \frac{A \cap B}{A \cup B} \quad (3)$$



**Figure 6.** Illustration of the concept of IoU (Intersection Over Union).

Here,  $A$  and  $B$  represent the bounding boxes of prediction and ground truth, respectively.

#### 4.1.4. Average Precision (AP)

For a specified threshold value of IoU, a precision–recall curve can be drawn after the identification of the values of precision and recall. The area under the precision–recall curve is referred to as the Average Precision (AP), which can be expressed by Equation (4):

$$AP = \int_0^1 p(r) dr \quad (4)$$

#### 4.1.5. Mean Average Precision (mAP)

This signifies the mean of average precisions of all classes defined in the test model and is expressed by Equation (5) for N number of classes.

$$mAP = \frac{\sum_{i=1}^N AP_i}{N} \quad (5)$$

#### 4.1.6. Loss Function

In the course of training, the sum of squared error loss [22] is used. The computation of the value of the loss function is one of the important criteria in evaluating the performance of YOLOv3 on the test model. Usually, the loss function is defined by Equation (6).

$$Loss = Error_{coord} + Error_{IoU} + Error_{cls} \quad (6)$$

Here,  $Error_{coord}$  is the coordinate prediction error, which is expressed by Equation (7).

$$Error_{coord} = \lambda_{coord} \sum_{i=1}^{S^2} \sum_{j=1}^B 1_{ij}^{obj} \left[ \left\{ (x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 \right\} + \left\{ (w_i - \hat{w}_i)^2 + (h_i - \hat{h}_i)^2 \right\} \right] \quad (7)$$

Here,  $(\hat{x}_i, \hat{y}_i, \hat{w}_i, \hat{h}_i)$  denote the values of the coordinate position, width and height of the predicted bounding box, respectively, and  $x_i, y_i, w_i$  and  $h_i$  signify the true or actual values. Further,  $\lambda_{coord}, S^2$ , and  $B$  represents coordinate error weight, number of grids in the input image, and number of bounding boxes generated by each grid, respectively. The value of  $1_{ij}^{obj}$  will be 1 if the object falls into the  $j$ th bounding box in grid  $i$ . In Equation (6),  $Error_{IoU}$  refers to the IoU error expressed by Equation (8).

$$Error_{IoU} = \sum_{i=1}^{S^2} \sum_{j=1}^B 1_{ij}^{obj} (C_i - \hat{C}_i)^2 + \lambda_{noobj} \sum_{i=1}^{S^2} \sum_{j=1}^B 1_{ij}^{noobj} (C_i - \hat{C}_i)^2 \quad (8)$$

where  $\lambda_{noobj}, C_i$ , and  $\hat{C}_i$  represent IoU error weight and predicted and true confidence, respectively.

Additionally,  $Error_{cls}$  denotes the classification error and is usually expressed by Equation (9). The  $Error_{cls}$  corresponding to the  $i$ th grid is the addition of classification errors associated with all the objects within that grid.

$$Error_{cls} = \sum_{i=1}^{S^2} \sum_{j=1}^B 1_{ij}^{obj} \sum_{c \in classes} (p_i(c) - \hat{p}_i(c))^2 \quad (9)$$

The notation utilized in Equation (9) is described as below.

- $c$  The specified class to which the detected object belongs;
- $p$  true probability that the object belonging to class  $c$  is in grid  $i$ ;
- $\hat{p}_i$  predicted value.

In the YOLOv3 algorithm, the input image is distributed into a grid of cells of dimensions  $S \times S$ , where each grid cell can predict three bounding boxes. Usually, YOLOv3 predicts the bounding boxes at three different scales. For determining the bounding box priors, the k-means clustering is used. In the present investigation, nine clusters and three

scales were selected. Further, these clusters were evenly divided across scales and they were distributed as  $(27 \times 34)$ ,  $(56 \times 66)$ ,  $(118 \times 177)$ ,  $(132 \times 332)$ ,  $(225 \times 234)$ ,  $(220 \times 354)$ ,  $(349 \times 285)$ ,  $(302 \times 356)$ ,  $(376 \times 367)$ .

#### 4.2. Anchor Boxes

In the case that the midpoints of multiple objects fall in the same grid cell, then the detection of those multiple objects becomes impossible. To avoid this issue, each object in the same grid was assigned with an anchor box. For instance, if we take three anchor boxes, then three predictions can be associated in a single grid cell. Each object is assigned to that anchor box, which has the highest IoU (Intersection over Union). If the IoU is less than the threshold value (here, set to 0.5), then that particular object will not be considered for detection. Thus, the detection of multiple objects in a single grid cell becomes possible by using the sole idea of anchor boxes, as illustrated in Figure 7.

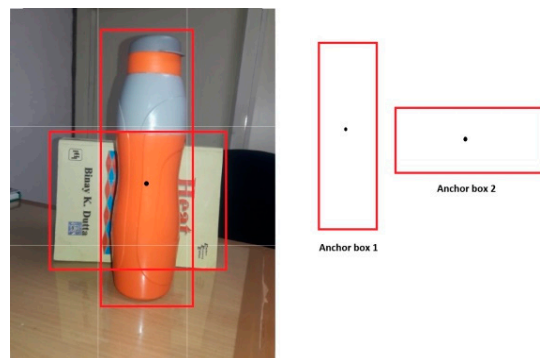


Figure 7. Illustration of anchor boxes.

#### 4.3. Non-Max Suppression (NMS)

Another problem encountered in object detection is the multiple detections of the same object, rather than detecting an object just once. The one-time detection of an object is feasible by using non-max suppression. The NMS algorithm compares the bounding box (with max  $P_c$ ) with all other bounding boxes intersecting with it in a sequential manner. All the bounding boxes associated with the object with comparatively low  $P_c$  are suppressed, as demonstrated in Figure 8.

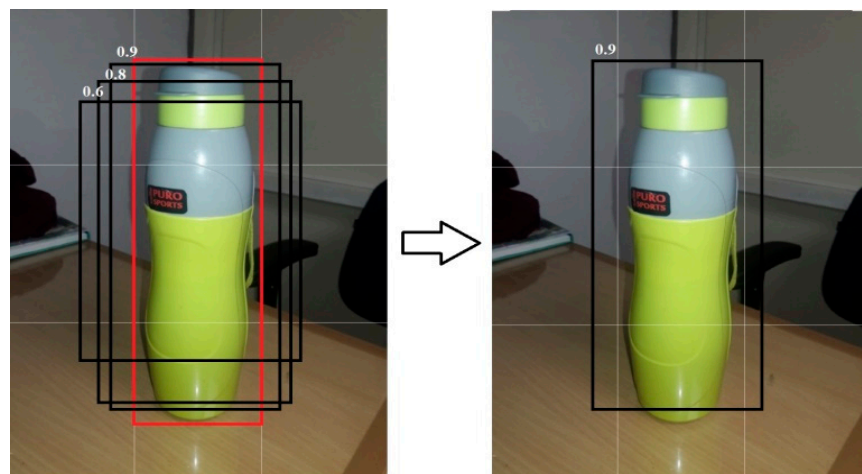


Figure 8. Non-max suppression for filtering multiple detections.



## 5. The Training

In this work, the entire experimental platform configuration utilized for the training and evaluation of the YOLOv3 neural network are presented in Table 3.

**Table 3.** Experimental platform configuration.

Specification	Details
Operating System	Windows, 64-bit Operating System
CPU	Intel(R) Core(TM) i7-9700F CPU @ 3.00 GHz
RAM	8 GB
GPU	MSI Gaming GeForce GTX 1650 Super 128-Bit HDMI/DP/DVI 4GB GDDR6 HDCP Support DirectX 12 Dual Fan VR Ready OC Graphics Card
GPU acceleration library	CUDA10.0, CUDNN7.4

From the cleaned dataset of 6437 images, 80% of the images (5150 images) were used for training purposes and the remaining 20% (1287) were used for testing and validation. Usually, the performance of any deep learning model is highly influenced by the size of the dataset. Generally, training with a small dataset leads to overfitting and to handle such a problem, a transfer learning approach is used [25]. In this approach, a pretrained model is repurposed to accomplish similar detection tasks. This initiates with the training of a base network on a base dataset and task, and then the learned features are transferred to a second target network to be trained on a target dataset and task. This process will tend to work if the features are suitable for both base and target tasks, instead of being specific to the base task. Considering the performance of the YOLO algorithms being trained on the large-scale image dataset (COCO), this work transfers the pre-trained YOLOv3 and YOLOv3-tiny networks on COCO.

At the start of training, the initialization of weights was performed using pretrained weights for the convolutional layers darknet53.conv.74. As discussed earlier, the YOLOv3 neural network is trained for the detection of waste via six classes of objects (cardboard, glass, metal, paper, plastic and organic waste), as discussed in Table 2. Moreover, the performance of the YOLOv3 algorithm has been compared with that of YOLOv3-tiny. For this purpose, the parametric settings used to train the model via the YOLOv3 and YOLOv3-tiny algorithm are tabulated in Table 4. The whole investigation environment uses Visual Studio 2017 for the compilation of the entire script.

**Table 4.** Parameters of CFG used for training our model.

Parameter	Value(s)	
	YOLOv3 Neural Network	YOLOv3-tiny Neural Network
Width	416	416
Height	416	416
Batch	64	64
Subdivisions *	64	16
Channels	3	-
Momentum	0.9	0.9
Decay	0.0005	0.0005
Learning rate	0.001	0.001
Maximum number of Batches *	12,000	12,000
Policy	Steps	Steps
Steps *	4800, 5400	4000, 4500
Scale	0.1, 0.1	0.1, 0.1
Classes *	6	6
Filters *	$(4 + 1 + 6) \times 3 = 33$	$(4 + 1 + 6) \times 3 = 33$

\* Represents the parameters modified in the original YOLOv3 CFG and YOLOv3-tiny CFG, respectively. Note: Filters usually depend on the number of classes, bounding box properties, prediction probability, and the number of masks, i.e., filters = {number of bounding box properties (4) + Prediction probability Pc (1) + Total number of classes (6)} × Number of masks, where mask denotes the indices of anchors (3).

## 6. Performance Evaluation, Results and Discussion

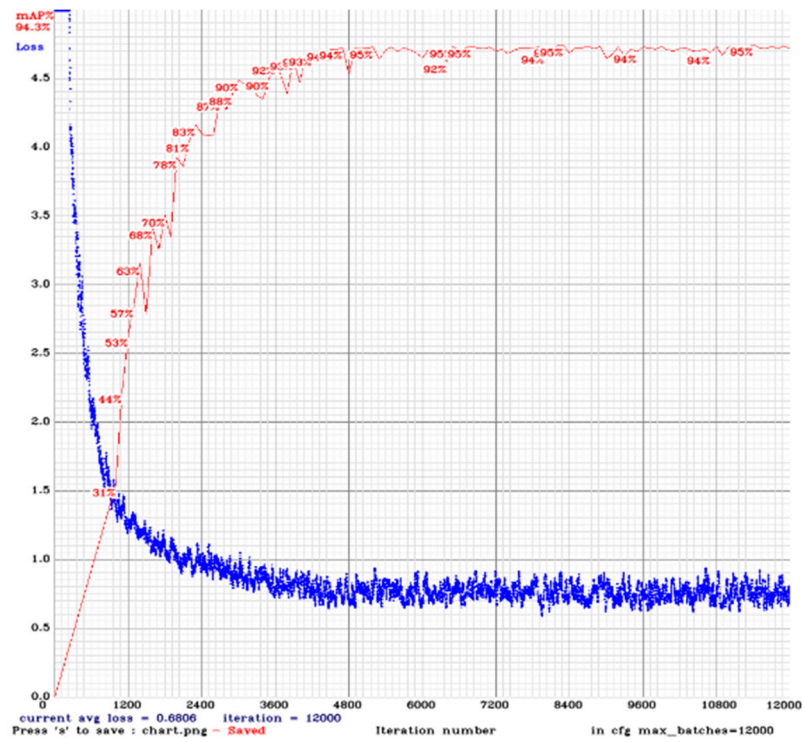
In the present work, the training was accomplished for 12,000 iterations and the total time taken to complete the training simulation was about 48 h on the mentioned platform (illustrated in Table 3) for the YOLOv3 algorithm. During the training simulation, the abovementioned performance parameter indices (including the AP of each class, recall, mAP and average IOU) were examined on a regular time interval. Table 5 presents the results obtained during the training phase with these performance parameter indices for YOLOv3 and YOLOv3-tiny. As evident from Table 5, after the 5000th iteration, mAP reaches around 94% by YOLOv3; however, YOLOv3-tiny yields only 45.96%. Thereafter, the mAP value settled down at approximately the same value (94.99%, best value) for YOLOv3. However, the mAP for the YOLOv3-tiny algorithm for the present experiment does not settle even after 12,000 iterations and attains the best value of 51.95%.

**Table 5.** Comparative study of simulation training results of YOLOv3 with YOLOv3-tiny on the test model.

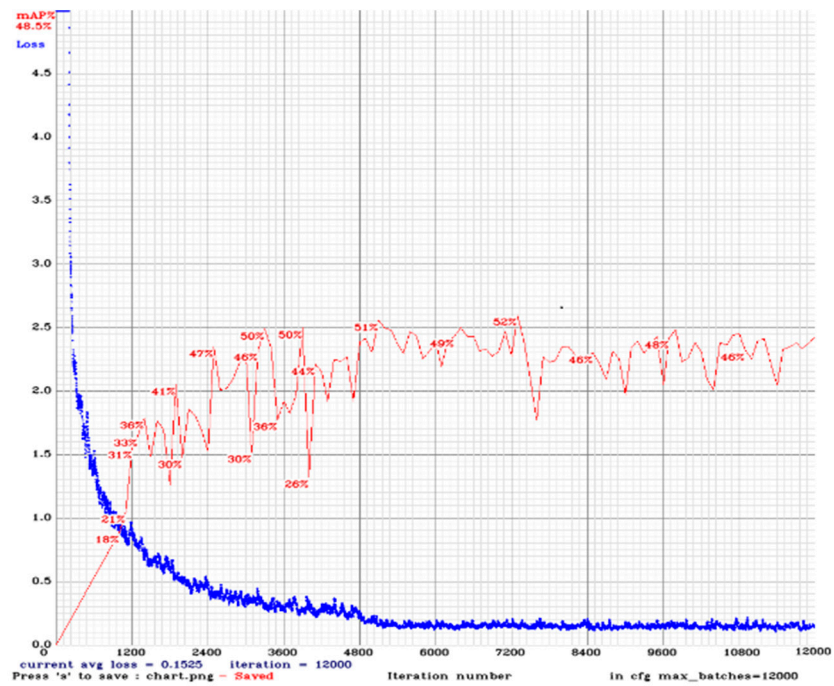
Iterations	YOLO Version	Classes AP (%)						mAP (%)	Recall	Average IoU (%)
		Cardboard (BD)	Glass (NBD)	Metal (NBD)	Paper (BD)	Plastic (NBD)	Organic Waste (BD)			
1000	v3	31.60	29.57	40.19	15.86	10.03	57.58	30.80	0.21	21.63
	v3-tiny	16.54	16.96	17.06	23.68	6.77	15.67	16.11	0.13	14.81
2000	v3	79.31	80.74	88.60	58.27	71.09	91.05	78.16	0.68	46.11
	v3-tiny	33.53	51.39	15.40	21.97	33.82	21.37	29.58	0.33	24.18
3000	v3	91.01	90.54	96.93	80.14	82.13	97.31	89.67	0.81	56.54
	v3-tiny	60.43	50.88	39.28	38.98	21.37	57.87	44.77	0.32	47.50
4000	v3	91.06	87.67	99.56	75.34	86.75	96.00	89.40	0.86	47.45
	v3-tiny	34.81	39.85	16.93	22.61	6.72	39.87	26.40	0.32	22.33
5000	v3	95.75	96.92	99.79	84.21	90.44	98.76	94.31	0.81	70.20
	v3-tiny	51.92	46.36	32.19	48.02	26.89	70.45	45.96	0.44	47.08
6000	v3	93.83	94.35	99.60	81.61	89.97	98.48	92.98	0.89	55.21
	v3-tiny	58.28	52.23	34.12	45.30	23.13	74.16	47.87	0.47	46.04
7000	v3	94.85	97.03	99.75	84.79	89.55	99.06	94.17	0.77	71.45
	v3-tiny	62.22	54.90	27.83	43.89	19.14	69.67	46.28	0.49	40.95
8000	v3	94.59	96.55	99.69	84.98	87.55	99.39	93.79	0.71	70.88
	v3-tiny	56.60	56.14	29.52	45.00	20.37	73.49	46.85	0.50	43.36
9000	v3	95.06	92.11	99.71	82.34	89.91	97.83	92.83	0.89	54.76
	v3-tiny	47.06	50.04	23.28	40.72	14.70	60.13	39.32	0.43	35.46
10,000	v3	95.77	94.92	99.79	84.38	91.15	98.50	94.08	0.90	59.86
	v3-tiny	53.55	53.81	31.77	42.71	18.10	71.57	45.25	0.48	42.61
11,000	v3	96.82	97.43	99.89	84.62	90.43	98.29	94.58	0.88	65.69
	v3-tiny	54.99	53.53	30.36	42.44	18.09	71.36	45.13	0.47	41.54
12,000	v3	95.64	95.96	99.84	84.95	91.19	98.49	94.35	0.90	61.79
	v3-tiny	42.60	52.88	47.88	49.16	27.69	71.08	48.55	0.47	50.35
BEST *	v3	97.27	97.40	99.87	85.28	91.16	98.93	94.99	0.87	67.42
	v3-tiny	62.16	61.79	31.98	48.32	26.15	81.29	51.95	0.52	47.20

\* BEST represents the iteration for which maximum mAP has been observed during training. AP: Average Precision. mAP: Mean Average Precision.

The variations in average loss and mAP values w.r.t the number of iterations during training by YOLOv3 and YOLOv3-tiny algorithm are illustrated in Figure 9a,b, respectively. Additionally, for comparative analysis, training was also carried out using the YOLOv3-tiny algorithm on the same dataset, which took approximately 14 h on a similar system configuration, as shown in Table 3. As is evident from the results illustrated in Figure 9a,b, the average loss function value using YOLOv3 and YOLOv3-tiny are 0.6806 and 0.1525, respectively, after the completion of training (12,000 iterations). Conclusively, our simulation experimental training results indicate that the mAP value for YOLOv3 is 82.85% higher than that of the YOLOv3-tiny model, in reference with the best value, which strengthens and validates our hypothesis.



(a)



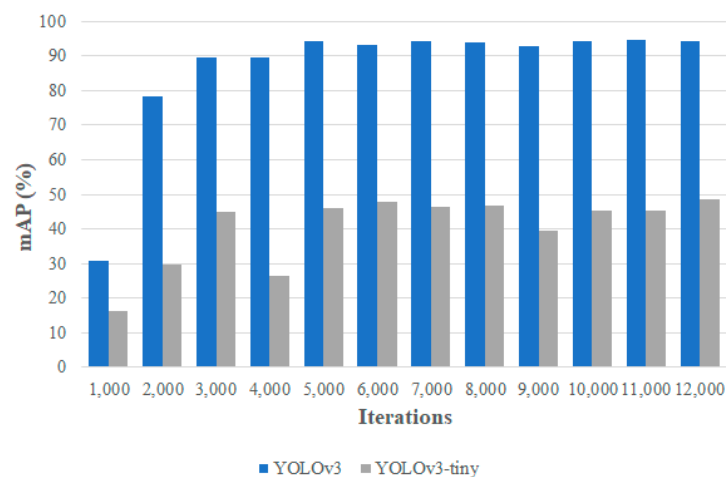
(b)

Figure 9. Variations of loss function value and mAP value w.r.t. number of iterations for (a) YOLOv3 (b) YOLOv3-tiny.

From Figure 10, it is observed that YOLOv3 offers enhanced AP for each class during experimental training simulation. The trends obtained during the training for all the mentioned classes (Table 2) indicate that YOLOv3 is, again, demonstrating outstanding performance as compared to YOLOv3-tiny in terms of AP, as illustrated in Figure 10. For instance, the best AP value (as illustrated in Table 5) attained by YOLOv3 for various classes (cardboard, glass, plastic, paper, metal and organic waste) are 97.27%, 97.40%, 99.87%, 85.28%, 91.16% and 98.93%, respectively; however, YOLOv3-tiny attains 62.16%, 61.79%, 31.98%, 48.32%, 26.15% and 81.29%, respectively, for the same classes. To provide a comparative insight into the performance of these two approaches in terms of mAP (%), a comparative sketch is provided, illustrating the variations in mAP values with the increasing number of iterations during training by YOLOv3 and YOLOv3-tiny. This comparison is demonstrated in the form of chart, as portrayed in Figure 11.



Figure 10. Variation in AP *w.r.t.* number of iterations for various classes via YOLOv3 and YOLOv3-tiny algorithm (a) cardboard (BD) (b) glass (NBD) (c) metal (NBD) (d) paper (BD) (e) plastic (NBD) (f) organic waste (BD).



**Figure 11.** Variation in mAP *w.r.t.* number of iterations via YOLOv3 and YOLOv3-tiny algorithm.

Furthermore, a statistical analysis of the present work in terms of AP, mAP and detection speed (frames per second, i.e., FPS) is presented in Table 6. The data reveal the effectiveness of YOLOv3 over the YOLOv3-tiny algorithm.

**Table 6.** Comparison of YOLOv3 and YOLOv3-tiny algorithms in terms of average precision and detection speed.

Algorithm	Classes AP (%)						mAP (%)	Detection Speed (FPS)	
	Cardboard (BD)	Glass (NBD)	Metal (NBD)	Paper (BD)	Plastic (NBD)	Organic Waste (BD)		CPU *	On Specified Platform (Table 3)
YOLOv3	97.27	97.40	99.87	85.28	91.16	98.93	94.99	0.3	3.0
YOLOv3-tiny	62.16	61.79	31.98	48.32	26.15	81.29	51.95	2.9	21.7

IOU = 0.75; mAP denotes the mean AP. \* CPU represents the detection on Intel(R) Core(TM) i5-4200U CPU @ 1.60GHz 2.30GHz.

After simulating the training, the test images were validated on the trained model. The test images correspond to the image garbage test set developed in this paper, which has 1287 images, including 165 cardboard, 163 glass, 146 metal, 312 paper, 317 plastic and 184 organic waste samples. The obtained experimental results demonstrate that the detection capability and prediction probability of YOLOv3 is significantly higher than YOLOv3-tiny, as visualized in Figure 12 and presented in Table 7. Most of the test images were accurately detected with acceptable prediction probability by YOLOv3. YOLOv3 gives true predictions of all the test images and is capable of detecting even small-size objects. However, YOLOv3-tiny gives false predictions for test images 1, 2, 6, 8 and 9. In addition, YOLOv3-tiny does not predicts test image 7 at all.



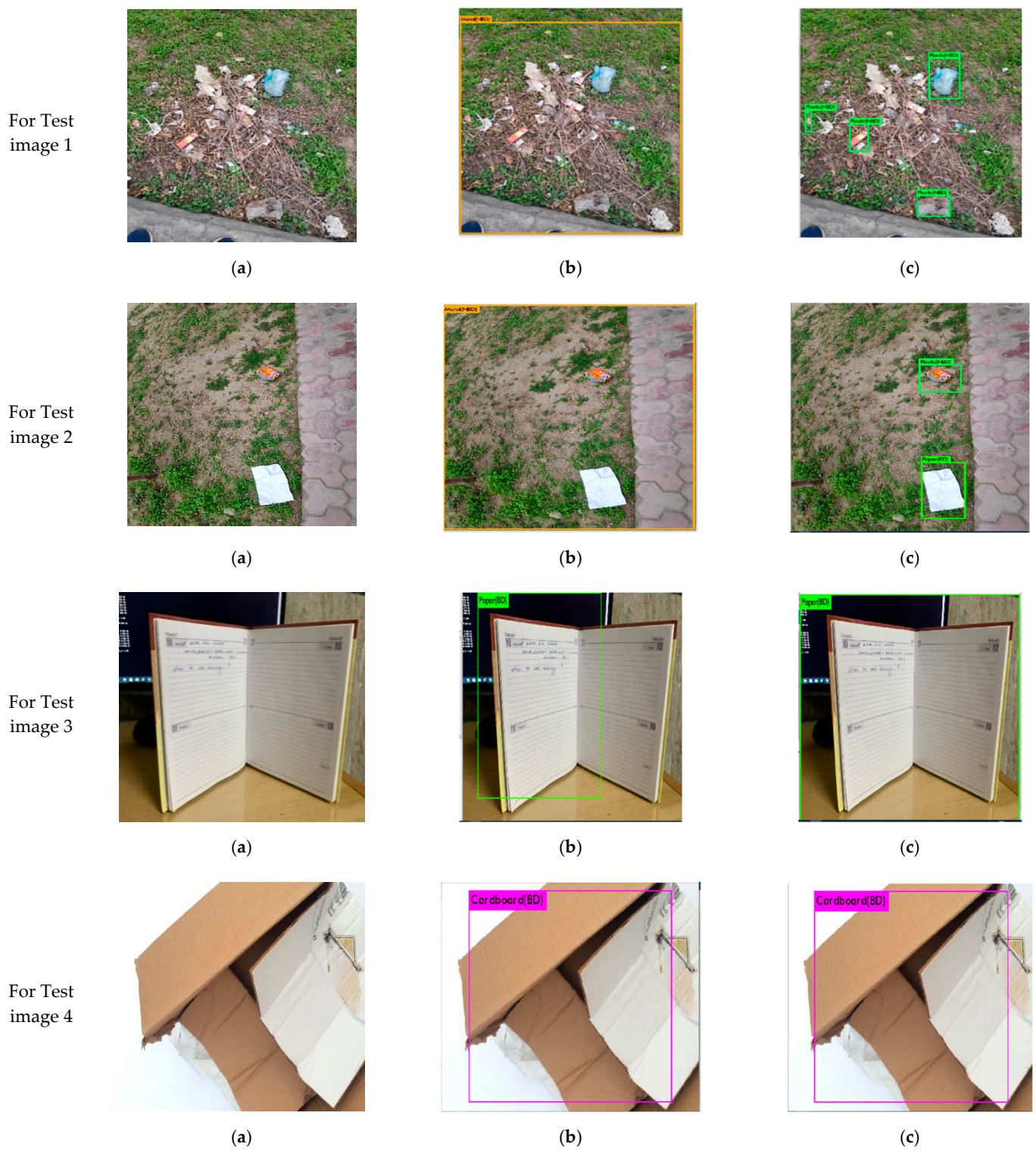


Figure 12. Cont.

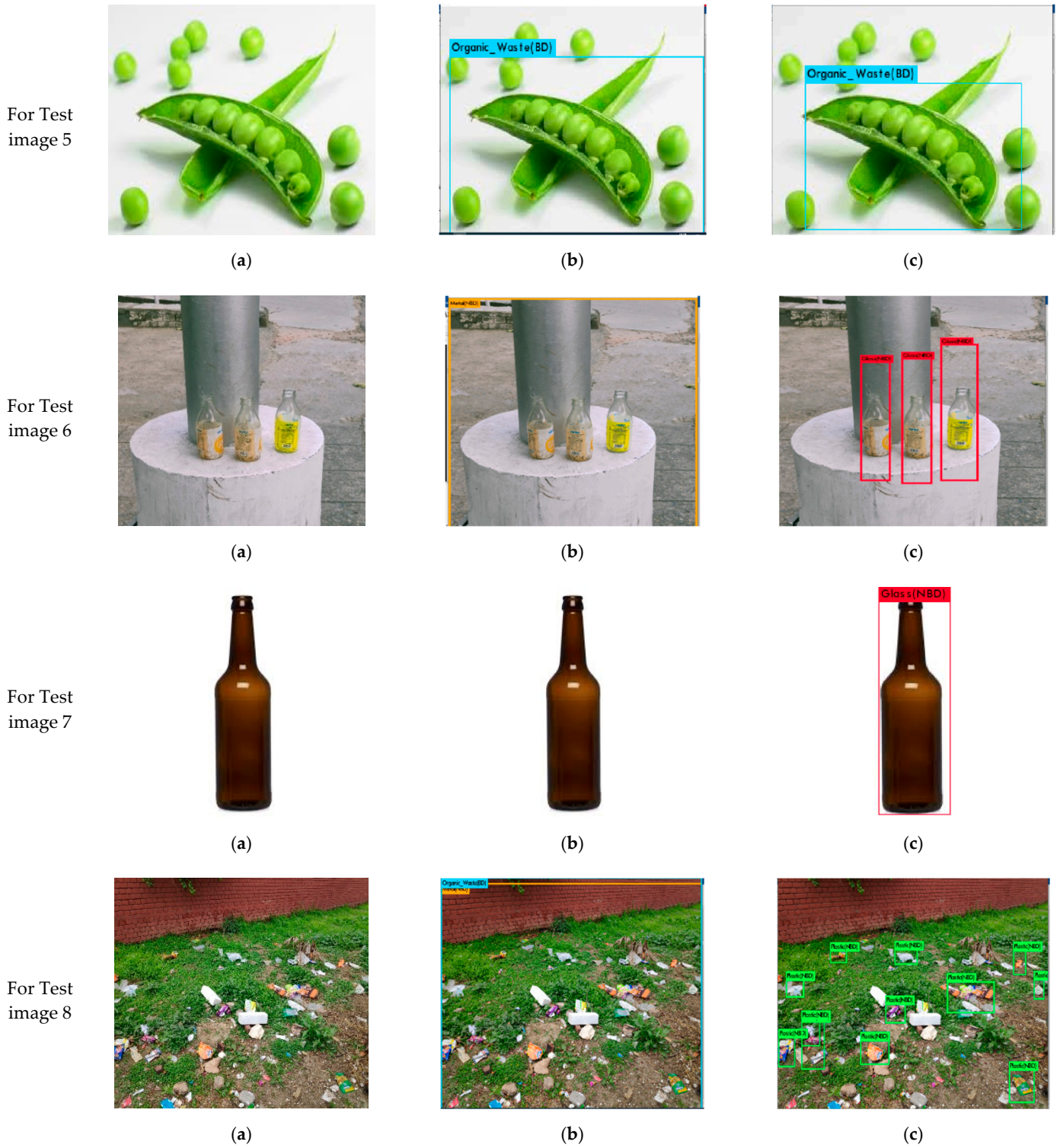


Figure 12. Cont.





**Figure 12.** Experimental results of object detection for waste segregation on various test images (a) original image (b) detection by YOLOv3-tiny (c) detection by YOLOv3.

**Table 7.** Quantitative comparison of the Experimental results of the test model.

Test Image	Items	Prediction Probability (in %)		Prediction Time (in Milliseconds)	
		YOLOv3	YOLOv3-tiny	YOLOv3	YOLOv3-tiny
1	Plastic (NBD)	78		228.30	52.13
	Plastic (NBD)	88	False Detection		
	Plastic (NBD)	92			
	Plastic (NBD)	55			
2	Paper (BD)	98		248.50	51.86
	Plastic (NBD)	89	False Detection		
3	Paper (BD)	100	27	222.42	51.30
4	Cardboard (BD)	100	85	222.57	51.39
5	Organic Waste (BD)	95	89	252.47	51.45
6	Glass (NBD)	94		247.94	51.93
	Glass (NBD)	83	False Detection		
	Glass (NBD)	93			
7	Glass (NBD)	98	No Detection	252.64	52.00
8	Plastic (NBD)	61		261.42	59.91
	Plastic (NBD)	73			
	Plastic (NBD)	62			
	Plastic (NBD)	55			
	Plastic (NBD)	88			
	Plastic (NBD)	99	False Detection		
	Plastic (NBD)	80			
	Plastic (NBD)	78			
	Plastic (NBD)	68			
	Plastic (NBD)	78			
Plastic (NBD)	79				
Plastic (NBD)	82				
9	Plastic (NBD)	88		247.97	51.98
	Paper (BD)	68	False Detection		
	Paper (BD)	72			
	Paper (BD)	92			

Table 8 presents the missed and false detection rates for both algorithms. Evidently, these detection rates are comparatively low (in the case of YOLOv3) when compared to the YOLOv3-tiny algorithm. Furthermore, the test results illustrate that the simulation time in the prediction to classify the objects using YOLOv3-tiny is lowered by approximately four-times more than YOLOv3, as shown in Table 7. This means that the computation speed of YOLOv3-tiny is significantly higher than YOLOv3. Conclusively, although the

computational performance of YOLOv3-tiny is remarkable, the detection capability and prediction probabilities are not acceptable.

**Table 8.** Comparison of missed and false detections with YOLOv3 and YOLOv3-tiny algorithms.

Algorithm	Missed Detection Rate (%)	False Detection Rate (%)	MAP (%)
YOLOv3	0	0.57	94.99
YOLOv3-tiny	4.78	11.58	51.95

Furthermore, to quantify the obtained results, a comparison of detection capability in test images has been made among the models developed by employing YOLOv3 and YOLOv3-tiny. As depicted in Table 9, the trained YOLOv3 model achieved superior detection capability compared to YOLOv3-tiny. It was able to detect most of the objects in the test images with significant prediction probability. Specifically, YOLOv3 achieved 100% detection accuracy for test images 2, 3, 4, 5, 6, and 7; however, YOLOv3-tiny achieved this for very simple test images (3, 4, and 5). Furthermore, the average accuracy for all the test images was been obtained as 85.29% and 26.47% for YOLOv3 and YOLOv3-tiny, respectively. Therefore, YOLOv3 dominates YOLOv3-tiny by a notable margin of 58.82%. However, YOLOv3 also struggles in the accurate detection of objects, particularly under occlusion and complex environmental conditions (test images 1, 8, and 9), as illustrated in Figure 12 and Table 9. This might be because of very small visual appearances and cluttered backgrounds.

**Table 9.** Comparison of detection capability.

Test Image	Ground Truth	Objects Detected	
		YOLOv3	YOLOv3-tiny
1	6	4	1
2	2	2	1
3	1	1	1
4	1	1	1
5	1	1	1
6	3	3	1
7	1	1	0
8	14	12	2
9	5	4	1

## 7. Conclusions

This paper presented a novel application of the YOLOv3 algorithm for waste segregation as an aid to strengthen smart urban waste segregation and management framework. The neural network was trained on a self-made dataset of 6437 images of urban waste products for the detection of six classes of waste items. The obtained experimental and investigational results demonstrate the efficiency of the proposed work in the segregation of waste into two different categories—biodegradable and nonbiodegradable. The near real-time detection of waste was accomplished in this work. The quantitative comparison of the results obtained by YOLOv3 and YOLOv3-tiny endorse the efficacy of YOLOv3 in waste segregation. Furthermore, the improved prediction probability by YOLOv3 demonstrates its effectiveness over YOLOv3-tiny. Conclusively, the comparative analysis between YOLOv3-tiny and YOLOv3 quantified the percentage improvement in speed with reduced accuracy (due to the simplified architecture of YOLOv3-tiny) which, in turn, helped in understanding the accuracy–speed trade-off. Furthermore, the garbage image detection process consisted of many complexities, such as objects made up of more than one type of material and may inherited other-class objects. To deal such real time complexities, objects belonging to parent classes and objects of visible category only were considered; however, this opens the window for further research to more exactly classify garbage, depending

upon the property of material(s). Additionally, the object detection strategy for waste segregation utilized in this work opens the gateway for the effective recycling and disposal of waste. However, the reduction in the time of detection along with exceptionally high prediction probability provides scope for further research. Future work will focus on the optimization of results, along with the prediction probability for other waste items in the real world.

**Author Contributions:** Conceptualization, S.K. and O.P.V.; methodology, S.K. and H.G.; software, S.K., O.P.V. and I.A.A.; validation, S.K., O.P.V. and I.A.A.; formal analysis, H.G., D.Y. and C.W.A.; investigation, S.K., D.Y. and O.P.V.; data curation, S.K., D.Y. and H.G.; writing—original draft preparation, S.K., D.Y. and H.G.; writing—review and editing, O.P.V., I.A.A. and C.W.A.; visualization, H.G. and I.A.A.; supervision, O.P.V. and C.W.A.; funding acquisition, C.W.A. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by the National Research Foundation of Korea (NRF) grant funded by the Korea Government (MSIT) (No. NRF-2019R1I1A2A01057603).

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

**Informed Consent Statement:** Not applicable.

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

## References

1. India's Trash Bomb: 80% of 1.5 Lakh Metric Tonne Daily Garbage Remains Exposed, Untreated—India News. Available online: <https://www.indiatoday.in/india/story/india-s-trash-bomb-80-of-1-5-lakh-metric-tonne-daily-garbage-remains-exposed-untreated-1571769-2019-07-21> (accessed on 20 November 2020).
2. Sharma, K.D.; Jain, S. Overview of Municipal Solid Waste Generation, Composition, and Management in India. *J. Environ. Eng.* **2019**, *145*, 04018143. [CrossRef]
3. SURAT KHUB SURAT-1. Available online: <https://darpg.gov.in/sites/default/files/19.DoortoDoorGarbageCollectionsystem.pdf> (accessed on 13 December 2020).
4. Wang, Y.; Zhang, X. Autonomous garbage detection for intelligent urban management. *MATEC Web Conf.* **2018**, *232*, 01056. [CrossRef]
5. Devi, R.S.S.; Vijaykumar, V.R.; Muthumeena, M. Waste segregation using deep learning algorithm. *Int. J. Innov. Technol. Explor. Eng.* **2018**, *8*, 401–403.
6. Lu, H.; Zhang, M.; Xu, X.; Li, Y.; Shen, H.T. Deep Fuzzy Hashing Network for Efficient Image Retrieval. *IEEE Trans. Fuzzy Syst.* **2020**. [CrossRef]
7. Lu, H.; Li, Y.; Chen, M.; Kim, H.; Serikawa, S. Brain Intelligence: Go beyond Artificial Intelligence. *Mob. Netw. Appl.* **2018**, *23*, 368–375. [CrossRef]
8. Lu, H.; Li, Y.; Mu, S.; Wang, D.; Kim, H.; Serikawa, S. Motor anomaly detection for unmanned aerial vehicles using reinforcement learning. *IEEE Internet Things J.* **2018**, *5*, 2315–2322. [CrossRef]
9. Chen, Z.; Lu, H.; Tian, S.; Qiu, J.; Kamiya, T.; Serikawa, S.; Xu, L. Construction of a Hierarchical Feature Enhancement Network and Its Application in Fault Recognition. *IEEE Trans. Ind. Inform.* **2020**. [CrossRef]
10. Huang, R.; Gu, J.; Sun, X.; Hou, Y.; Uddin, S. A Rapid Recognition Method for Electronic Components Based on the Improved YOLO-V3 Network. *Electronics* **2019**, *8*, 825. [CrossRef]
11. Dundar, A.; Jin, J.; Martini, B.; Culurciello, E. Embedded streaming deep neural networks accelerator with applications. *IEEE Trans. Neural Netw. Learn. Syst.* **2017**, *28*, 1572–1583. [CrossRef] [PubMed]
12. 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 Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*; Springer: Berlin/Heidelberg, Germany, 2016; Volume 9905 LNCS, pp. 21–37.
13. Fu, C.-Y.; Liu, W.; Ranga, A.; Tyagi, A.; Berg, A.C. DSSD: Deconvolutional Single Shot Detector. *arXiv* **2017**, arXiv:1701.06659.
14. Lu, S.; Wang, B.; Wang, H.; Chen, L.; Linjian, M.; Zhang, X. A real-time object detection algorithm for video. *Comput. Electr. Eng.* **2019**, *77*, 398–408. [CrossRef]
15. Dollár, P.; Wojek, C.; Schiele, B.; Perona, P. Pedestrian detection: An evaluation of the state of the art. *IEEE Trans. Pattern Anal. Mach. Intell.* **2012**, *34*, 743–761. [CrossRef] [PubMed]
16. Pathak, A.R.; Pandey, M.; Rautaray, S. Application of Deep Learning for Object Detection. In *Proceedings of the Procedia Computer Science*; Elsevier B.V.: Amsterdam, The Netherlands, 2018; Volume 132, pp. 1706–1717.
17. Zhao, Z.Q.; Zheng, P.; Xu, S.T.; Wu, X. Object Detection with Deep Learning: A Review. *IEEE Trans. Neural Netw. Learn. Syst.* **2019**, *30*, 3212–3232. [CrossRef] [PubMed]
18. Adedeji, O.; Wang, Z. Intelligent waste classification system using deep learning convolutional neural network. In *Proceedings of the Procedia Manufacturing*; Elsevier B.V.: Amsterdam, The Netherlands, 2019; Volume 35, pp. 607–612.



19. Chu, Y.; Huang, C.; Xie, X.; Tan, B.; Kamal, S.; Xiong, X. Multilayer hybrid deep-learning method for waste classification and recycling. *Comput. Intell. Neurosci.* **2018**, *2018*, 5060857. [[CrossRef](#)] [[PubMed](#)]
20. Sung, K.K.; Poggio, T. Example-based learning for view-based human face detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **1998**, *20*, 39–51. [[CrossRef](#)]
21. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; Volume 2016-Decem, pp. 779–788.
22. Redmon, J.; Farhadi, A. YOLOv3: An Incremental Improvement. *arXiv* **2018**, arXiv:1804.02767.
23. Nordin, P.; Lidström, F. Object Detection Using Yolov3 Tiny. 2019. Available online: <http://urn.kb.se/resolve?urn=urn:nbn:se:oru:diva-76453> (accessed on 20 November 2020).
24. Nie, X.; Yang, M.; Liu, R.W. Deep Neural Network-Based Robust Ship Detection Under Different Weather Conditions. In Proceedings of the 2019 IEEE Intelligent Transportation Systems Conference, ITSC 2019, Auckland, New Zealand, 27–30 October 2019.
25. Alom, Z.; Taha, T.M.; Yakopcic, C.; Westberg, S.; Sidike, P.; Nasrin, S.; Hasan, M.; Van Essen, B.C.; Awwal, A.A.S.; Asari, V.K. A State-of-the-Art Survey on Deep Learning Theory and Architectures. *Electronics* **2019**, *8*, 292. [[CrossRef](#)]