Abstract: Unmanned Surface Vehicles (USVs) are commonly equipped with multi-modality sensors. Fully utilized sensors could improve object detection of USVs. This could further contribute to better autonomous navigation. The purpose of this paper is to solve the problems of 3D object detection of USVs in complicated marine environment. We propose a 3D object detection Depth Neural Network based on multi-modality data of USVs. This model includes a modified Proposal Generation Network and Deep Fusion Detection Network. The Proposal Generation Network improves feature extraction. Meanwhile, the Deep Fusion Detection Network enhances the fusion performance and can achieve more accurate results of object detection. The model was tested on both the KITTI 3D object detection dataset (A project of Karlsruhe Institute of Technology and Toyota Technological Institute at Chicago) and a self-collected offshore dataset. The model shows excellent performance in a small memory condition. The results further prove that the method based on deep learning can give good accuracy in conditions of complicated surface in marine environment.

Keywords: multi-modality sensors; fusion; object detection; unmanned surface vehicle
3D object detection based on point cloud generated by lidar is time-consuming. It cannot meet the real-time requirement of USVs.

This paper proposes a 3D object detection Depth Neural Network (DNN) based on multi-modality data of USVs. This new model contains two significant features. Firstly, this model converts the raw point cloud to Bird’s Eye View (BEV). This reduces the computation requirements of the algorithm and can work smoothly in lower memory spaces of computer. This model contains a deep network that can generate area proposals to overcome the problems of low-resolution object in images and BEV. In this deep network, a modified Resnet [7] is utilized to extract the feature map of images and BEV. Secondly, this model includes a deep fusion detection network. This fusion detection network improves the accuracy of object detection on the sea surface.

We tested our model on the KITTI dataset and a self-built datasets. Experimental results show that our model can achieve better results than other methods did in the USV’s operating environment. The final performance of object detection on the sea surface has been greatly improved. The new model has apparent advantages in performance and accuracy, even in small memory conditions.

The rest of the paper is organized as follows: after reviewing the related work in Section 2, we introduce the proposed object detection architecture in Section 3. Then, we provide a thorough experimental result in Section 4. Finally, the paper is concluded in Section 5.

2. Related Work

The proposed model uses the BEV and the image as input. A two-stage detection network is constructed to detect objects on the sea surface. This network consists of proposal generation network and deep fusion network.

2.1. Point-Cloud-Based Method vs. View-Based Method

In network input, the Point-cloud-based method and the View-based method are two generally used 3D object detection methods. The Point-cloud-based method could be further divided into two types. The first type takes raw 3D point cloud or its varieties as input. It directly takes original 3D point cloud as input [8,9]. For example, PointNet [9] takes an unordered set of points directly into a network. This network uses a symmetric function to select the points of interest. Then, it predicts the label of each point. These methods are time-consuming. The second type encodes the point cloud to some varieties, typically voxel grid [10,11]. For example, voxelnet [11] divides the point cloud into isometric 3D voxels, which are input into the network for learning. These methods involve 3D convolution operation, which is inefficient for the network operation speed in applications of USVs.

The view-based method converts the 3D point cloud into a view, such as Bird’s Eye View (BEV). By transforming 3D point cloud into BEV [6,12–16], these methods extract effective features from depth information. It significantly improves the computing speed by using the fusion learning of different views of image and point cloud. In this paper, the 3D point cloud is encoded as multi-channel BEV to extracting features together with images for object detection.

2.2. Two-Stage Detection Network vs. Single-Stage Detection Network

In network structures, there are two methods for object detection network. One is the single-stage detection method, such as SSD [17] and YOLO [18]. For the single-stage method, the location detection and object classification are fully integrated into a single network. The border position is designed as a regression parameter, which is directly obtained by network regression. As there is no specific region proposal, they are not as accurate as the two stages in detecting small objects.

Two-stage detection implements detection through two procedures, such as Fast R-CNN [19], faster RCNN [20] and mask RCNN [21]. This method adopts Region Proposal Network (RPN). The RPN shares global convolution features with detection network and generates region. The separated proposal generation and classification improve accuracy in small object detection. We adopted
the two-stage detection network scheme to generate proposals by using RPN and made accurate classification in the deep fusion network.

2.3. Deep Fusion Network

The deep fusion network achieves better results than the shallow network in multi-model tasks. The fusion of multi-source data has practical significance in many fields. Giuseppe et al. [22,23] proposed and compared the fusion methods of multiple classifiers. In the application of USVs object detection, we need to fuse the multi-mode features. In [24], information combination in detection is only a simple fusion in the early stage. Such simple fusion sets a latitude. Deeply-Fused Nets [25] pioneered the concept of deep fusion to learn about multi-scale representation. Deep and shallow basic networks learn together and can benefit from each other. Drop-path proposed by Fractalnet [26] refers to randomly dropping subpaths. It can prevent overfitting and improve the performance of the model. We integrated these ideas and chose a deep fusion network with drop-path. In our network, we designed a Resnet-like module to integrate BEV feature graph and image feature. As each module iterates, we selectively drop paths. Using the deep fusion network, the depth expanding of the new network in this paper introduces more parameters that increase expression ability.

3. Method

3.1. The Object Detection Architecture

As shown in Figure 1, our two-stage detection network consists of two parts, namely, the proposal generation network and the deep fusion detection network. The proposal generation network takes BEV and image as input. The feature map is extracted from the inputs to generate 3D proposals. In the Deep Fusion Detection Network, feature maps are deeply fused. It uses proposals to regress the 3D bounding box to predict the bounding box and the direction of the 3D object.

Figure 1. The object detection architecture, where the blue part is proposal generation network and the red part is the deep fusion detection network.
3.2. Proposal Generation Network

3.2.1. Bird’s Eye View Generating

A BEV encodes 3D point cloud in terms of height, intensity, and density. Mv3D [6] transforms the point cloud data into slices to obtain the height map and then connects the height map with the intensity map and density map to obtain the multi-channel feature map. The height map calculates the maximum altitude of points in each grid. The intensity and density are the reflectivity of the highest point in each cloud and the number of points in each grid, respectively. Instead of massive slices, we encoded the point cloud into a six-channel BEV map based on [12]. The resolution of each two-dimensional grid is 0.1 m. The first five channels of BEV are height maps calculated based on the average slices cut along the z-axis. The z-axis threshold is between 0 and 2.5 m. The final sixth channel is the density map. We normalized the density information as \( \min(1.0, \frac{\log(N+1)}{\log 16}) \), where \( N \) is the number of midpoints in the grid. As shown in Figure 2, we visualized the six channels of BEV.

![Figure 2. From top to bottom and left to right are the six channels of BEV.](image)

3.2.2. Feature Map Extraction

Most target objects in the marine scenario are far away from the USVs. It is difficult to extract rich features from results in a small object on the image and BEV. Therefore, we modified the Resnet-50 model for feature extraction, as shown in Figure 3. The network was cut off at conv3, and the number of channels was halved. Thus, the input size is \( M \times N \times D \), and the output of the feature map is \( \frac{M}{8} \times \frac{N}{8} \times D \). Since the resolution is 0.1 m, small objects may occupy tiny pixels in the feature map. There may be some difficulties for object proposal procedure. We introduced deconvolution [27,28] as an up-sampling method to restore small image data to the original size. After a deep neural network of feature extraction, we used transpose convolution for sampling. At the outset, the network contained a series of convolution operations of Resnet-50, the input is down-sampled at \( 8 \times \). After the deconvolution layers, the input goes back to half the size of the original image. It provides a high representational ability.
3.2.3. Proposal Generating

- **3D Box Description:** In two-stage 2D object detection, the proposal is generated from a series of prior boxes. In [6,12], an encoding method of 3D prior boxes is proposed. Each 3D prior box is parameterized to \((x,y,z,l,w,h,r)\), where \((x,y,z)\) is the center of the mass coordinate of box, \((l,w,h)\) is the size of box, and \((r)\) is the direction of the box. The prior box is generated from BEV. The sampling interval \((x, y)\) is 0.5 m, while the ground height of the sensor determines parameter \(z\).

- **Crop and Resize Instead of ROI Pooling:** We chose to use crop and resize \([12,29]\) to extract the box’s corresponding feature map from a particular view instead of the ROI-pooling. Because ROI-pooling uses nearest neighbor interpolation, it may lose spatial symmetry. Nearest neighbor interpolation means that ROI-pooling adopts rounding, which is equivalent to selecting the Nearest point to the target point when the scaled coordinate cannot be an exact integer. To keep the symmetry of the space, crop and resize uses bilinear interpolation to resample an image to a fixed size.

- **Drop Path:** In the fusion of RPN stage, we added the drop path method. The extracted feature channel is randomly discarded. Then, the elements were evenly fused. FractalNet used drop path method to normalize collaborative adaptive neutron paths in fractal structures Larsson et al. [26]. Through this normalization, it is able to prove that the answer given by the shallow subnetwork is faster, while the answer given by the deep subnetwork is more accurate.

- **3D Proposal Generating:** We used \(1 \times 1\) convolution instead of the full connection layer and then used a multi-task loss to classify the object/background and computed the regression of proposal boxes. To sort out the background/object, we used cross-entropy loss, while, for the regression loss of the proposal boxes, we selected smooth L1 loss. When computing boxes regression, we ignored the background anchor point. The background anchor point is determined by the IoU overlap between ground truth and the anchor point in BEV. Overlap above 0.7 is considered as the background. If the overlap is less than 0.5, it is considered as the target. To reduce redundancy, Non-Maximum Suppression (NMS) is applied in BEV with a threshold of 0.8.

3.3. Deep Fusion Detection Network

3.3.1. Deep Fusion

By projecting the 3D proposal box onto the BEV and previously extracted image feature map, we can get two response regions. To conduct in-depth fusion afterwards, the regions are computed by crop and resize. We obtained an equidistant vector.

To fuse different information, we propose an improved deep fusion method. It is based on the method by [6]. Chen et al compared the differences between early fusion and late fusion [6].
That method increases the interaction between the middle layer features of different views. The original fusion process is as follows:

\[
\begin{align*}
    f_0 &= f_{BV} \oplus f_{FV} \oplus f_{RGB} \\
    f_l &= H_{BV}^l(f_{l-1}) \oplus H_{FV}^l(f_{l-1}) \oplus H_{RGB}^l(f_{l-1}) \\
    \forall & l = 1, \ldots, L
\end{align*}
\]  

(1)

where \( H_l, l = 1, \ldots, L \) is a feature transformation function and \( \oplus \) is the join operation (for example, concatenation or sum). We improved the process by eliminating the front view. Since only the BEV and image were available, we removed the fusion of the front views to accommodate our network structure. We also used the element-wise mean join operation for fusion. Our design is as follows:

\[
\begin{align*}
    f_0 &= f_{BV} \oplus f_{RGB} \\
    f_l &= H_{BV}^l(f_{l-1}) \oplus H_{RGB}^l(f_{l-1}) \\
    \forall & l = 1, \ldots, L
\end{align*}
\]  

(2)

In the implementation process, we used a design that is similar to the block in Resnet to make the fusion effect better.

3.3.2. Generating Detection Network

- **3D Bounding Box Regression:** As shown in Figure 4, traditional axis aligned coding uses centers of mass and axes. Chen et al. [6] used eight-corner box coding. We used a more simplified four-corner coding. Considering that the bottom four corners must be aligned with the top, the physical constraint of the 3D bounding box is added. We encoded the border as four corners and two heights. Thus, the original regression target of the 24-dimensional vector of the eight-corner box is changed to \((x_1 \ldots x_2, y_1 \ldots y_2, h_1, h_2)\). To reduce the amount of calculation and to improve the speed of calculation, we selected the four-corner encoding and introduced orientation estimation. From the orientation estimation, we extracted four possible directions of the border box and explicitly computed the box regression loss.

![Figure 4. Modified Resnet-50 feature extractor.](image)

- **Generating Detection:** The fused features are input to the full connection layer. Box regression and classification operation are carried out after the three fully-connected layers of size 2048.
Similarly, Smooth L1 loss is still used for box regression, and cross-entropy loss is used for classification. To eliminate overlap, we used IoU threshold of 0.01 for NMS.

3.4. Training

RPN and detection network are jointly trained in an end-to-end method. For each batch, an image of 512 or 1024 ROIs is used. We used ADAM optimizer to perform 120,000 iterative training with a learning rate of 0.001.

4. Experiments and Results

We evaluated our methods on the KITTI [30] validation set and an offshore marine environments dataset. The experiments were executed on different workstations. The graphics card was one TITAN X or two 1080TI.

4.1. Evaluation on the KITTI Validation Set

We evaluated our network against the KITTI 3D object detection benchmarks. The training set provided 7481 images, and the test set provided 7518 images. We followed Chen et al. [31] to divide the training set into a training set and validation set in a 1:1 ratio. We concentrated our experiments on the category of cars. To facilitate the evaluation, we followed KITTI's simple, moderate, and difficult classification methods to evaluate and compare our network.

1. Evaluation of Object Detection

For 3D overlap standard, in the radar-based method, we used the IoU threshold of 0.7 to conduct the 3D object detection. Because our model focuses on BEV and images, we also compared it with f-point net, while comparing MV3D and AVOD. The results are shown in Table 1. The results are significantly more than 10% higher than MV3D in terms of average precision. The results are also superior to AVOD in the easy and moderate mode. On the hard mode, however, the results are weaker. However, it can be tolerated in practice.

We also carried out 3D detection in the BEV, and the results are shown in Table 2. AVOD did not release its results, thus we could not compare our results with AVOD’s results. Our model was superior to MV3D in all aspects. Our model also performed better than f-point net in easy and hard mode.

Inspired by AVOD, we also included orientation estimation in the network. We compared the Average Orientation Similarity (AOS). In AVOD, they are called Average Heading Similarity (AHS). As shown in Table 3, our model had better results than AVOD.

### Table 1. 3D detection performance: Average precision (AP) (in %) for 3D boxes.

<table>
<thead>
<tr>
<th>Method</th>
<th>Easy</th>
<th>Moderate</th>
<th>Hard</th>
</tr>
</thead>
<tbody>
<tr>
<td>MV3D</td>
<td>71.29</td>
<td>62.68</td>
<td>56.56</td>
</tr>
<tr>
<td>F-Point</td>
<td>83.76</td>
<td>70.92</td>
<td>63.65</td>
</tr>
<tr>
<td>AVOD</td>
<td>84.41</td>
<td>74.44</td>
<td>68.65</td>
</tr>
<tr>
<td>Ours</td>
<td>88.19</td>
<td>74.54</td>
<td>65.86</td>
</tr>
</tbody>
</table>

### Table 2. Bird’s eye view detection performance: Average precision (AP) (in %) for BEV boxes.

<table>
<thead>
<tr>
<th>Method</th>
<th>Easy</th>
<th>Moderate</th>
<th>Hard</th>
</tr>
</thead>
<tbody>
<tr>
<td>MV3D</td>
<td>86.55</td>
<td>78.10</td>
<td>76.67</td>
</tr>
<tr>
<td>F-Point</td>
<td>88.16</td>
<td>84.02</td>
<td>76.44</td>
</tr>
<tr>
<td>Ours</td>
<td>88.73</td>
<td>79.07</td>
<td>77.89</td>
</tr>
</tbody>
</table>
Table 3. Average Orientation Similarity (AOS) (in %) at 0.7 3D IoU.

<table>
<thead>
<tr>
<th>Method</th>
<th>Easy</th>
<th>Moderate</th>
<th>Hard</th>
</tr>
</thead>
<tbody>
<tr>
<td>Deep3DBox</td>
<td>5.84</td>
<td>4.09</td>
<td>3.83</td>
</tr>
<tr>
<td>MV3D</td>
<td>52.74</td>
<td>43.75</td>
<td>39.86</td>
</tr>
<tr>
<td>AVOD</td>
<td>84.19</td>
<td>74.11</td>
<td>68.28</td>
</tr>
<tr>
<td>Ours</td>
<td>84.62</td>
<td>74.66</td>
<td>72.69</td>
</tr>
</tbody>
</table>

2. Analysis of Detection Results

Figure 5 shows the detection results in the six channels of BEV. Our model worked well in medium- and short-range cars. Although the longer distance cars have fewer points, our model still performed well. We were surprised to find that the car’s directional regression was also excellent. These results prove the effectiveness of our model. Figure 6 demonstrates the detection results projected onto the 2D images.

![Figure 5](image-url)

*Figure 5. (Top) Image to be tested; (Middle) from left to right are the first three height feature maps; and (Bottom) the first two are height feature maps, and the last one is density feature map.*
Figure 6. (Top) The detection results are projected onto 2D images (the values are detection accuracy and IoU); and (Bottom) the 3D box is projected onto the image.

3. Runtime and Memory Requirements

We used several parameters to assess the computational efficiency and the memory requirements of the proposed network. Our object detection architecture employed roughly 52.04 million parameters. We significantly reduced the number of parameters compared with the method in [6], the second stage detection network of which has three fully connected layers. Because we chose the Resnet-50 with more layers, our parameter number was higher than the method in [12]. Each frame was processed in 0.14 s on Titan X and 0.12 s in 2080TI. The inference time of the network for each picture was 90 ms at 2080TI.

4. Ablation Studies

As shown in Tables 4 and 5, we first compared our proposed Deep Fusion Detection Network with early fusion method. In the same case that our modified Resnet-50 feature extractor was used, the average precision of our Deep Fusion Detection Network in 3D boxes and BEV boxes was higher than that of the earlier fusion method. Especially in BEV boxes, our networks performed best, about 10% better. To study the contribution of our proposed feature extractor, we replaced the Resnet-50 with the VGG16 used in [6,12] for comparison. Our test results were also slightly higher than VGG16.

<table>
<thead>
<tr>
<th>Method</th>
<th>Easy</th>
<th>Moderate</th>
<th>Hard</th>
</tr>
</thead>
<tbody>
<tr>
<td>Modified Resnet-50 + early</td>
<td>85.43</td>
<td>72.22</td>
<td>57.79</td>
</tr>
<tr>
<td>VGG16+ Deep Fusion</td>
<td>84.37</td>
<td>73.09</td>
<td>57.78</td>
</tr>
<tr>
<td>Modified Resnet-50+Deep Fusion</td>
<td>88.19</td>
<td>74.54</td>
<td>65.86</td>
</tr>
</tbody>
</table>
Table 5. Average precision (AP) (in %) for BEV boxes for ablation studies.

<table>
<thead>
<tr>
<th>Method</th>
<th>Easy</th>
<th>Moderate</th>
<th>Hard</th>
</tr>
</thead>
<tbody>
<tr>
<td>Modified Resnet-50 + early</td>
<td>75.55</td>
<td>66.41</td>
<td>58.77</td>
</tr>
<tr>
<td>VGG16 + Deep Fusion</td>
<td>88.29</td>
<td>77.49</td>
<td>76.41</td>
</tr>
<tr>
<td>Modified Resnet-50 + Deep Fusion</td>
<td>88.73</td>
<td>79.07</td>
<td>77.89</td>
</tr>
</tbody>
</table>

4.2. Evaluation on the Offshore Marine Environments Dataset

We captured many images and point cloud data from a USV in a real offshore marine environment. We divided the sample data into training set and verification set in a 1:1 ratio. Training set data included cruise ships, sailboats and fishing boats. After marking the ground truth of each image, random sampling was applied to generate positive samples and negative samples. When the IoU of random sampling was greater than 0.5, a positive sample was generated. Similarly, negative samples were generated in the same way, while IoU was less than 0.3 and had basic facts. Considering the symmetry of the object, mirror samples were generated for each positive sample and negative sample. For validation set data, the scenarios that USV faces in a real-world natural environment were included. Finally, we tested our algorithms on these challenging datasets to demonstrate the efficiency and accuracy of our network.

We applied the network to our own dataset of maritime ships, as shown in Figure 7. In the case of 300 proposals in one figure, we drew recall under different IoU thresholds. In particular, when the IoU was 0.25, recall reached 84%. When the IoU was 0.5, our recall reached 67%, which indicates that the image and radar fusion learning network had a huge advantage in the case of large numbers of invalid marine image information.

![Figure 7](image-url) From left to right: recall vs. IoU; and recall vs. proposals at IoU threshold of 0.25 and 0.5.

It can be concluded from the BEV picture (Figure 8) and 3D detection picture (Figure 9) that our method could obtain accurate three-dimensional position, size and direction. Finally, we compared our network with DPM algorithm with regard to accuracy and time, as shown in Table 6. The statistical results show that our network had higher accuracy rate and lower false alarm rate, while maintaining higher efficiency.

Table 6. Comparison of accuracy and time.

<table>
<thead>
<tr>
<th>Method</th>
<th>Accuracy</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>DPM</td>
<td>54%</td>
<td>16 s</td>
</tr>
<tr>
<td>Ours</td>
<td>78%</td>
<td>120 ms</td>
</tr>
</tbody>
</table>
5. Conclusions

We propose a 3D object detection Deep Neural Network for USVs based on multi-modality sensor fusion. Our network utilizes lidar point cloud and image data to generate 3D proposals. The Deep Neural Network is used for feature extraction. Then, deep fusion and object detection are carried out. Our network is significantly superior to the traditional approaches in the dataset that are collected in offshore waters. The evaluation on the challenging KITTI benchmark demonstrated improvement as well. In the future, we will use more datasets to improve and test the performance of the algorithm. Furthermore, we will improve our network to carry out distributed work to adapt to the working environment of USVs.

Author Contributions: Data curation, T.L.; Project administration, H.L.; Resources, H.Q.; Software, H.Q.; Supervision, Z.W.; Writing—original draft, Y.W.
**Funding:** This research was funded by Department of Science and Technology of Shandong Province grant number 2016ZDJS09A01, Qingdao Municipal Science and Technology Bureau grant number 17-1-1-3-jch.

**Acknowledgments:** We sincerely thank Qingdao Beihai Shipbuilding Heavy Industry Co. for assisting in the construction of USV. We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan X Pascal GPU used for this research.

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

**References**


© 2019 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 (http://creativecommons.org/licenses/by/4.0/).