*remote sensing*

MDPI

*Article*

# Airborne Streak Tube Imaging LiDAR Processing System: A Single Echo Fast Target Extraction Implementation

**Yongji Yan [1], Hongyuan Wang [2], Boyi Song [1], Zhaodong Chen [1], Rongwei Fan [1], Deying Chen [1] and Zhiwei Dong [1,*]**

[1] National Key Laboratory of Science and Technology on Tunable Laser, Harbin Institute of Technology, Harbin 150001, China
[2] Research Center for Space Optical Engineering, Harbin Institute of Technology, Harbin 150001, China
* Correspondence: dong19809@163.com

**Abstract:** In this paper, a ground target extraction system for a novel LiDAR, airborne streak tube imaging LiDAR (ASTIL), is proposed. This system depends on only a single echo and a single data source, and can achieve fast ground target extraction. This system consists of two modules: Autofocus SSD (Single Shot MultiBox Detector) and post-processing. The Autofocus SSD proposed in this paper is used for object detection in the ASTIL echo signal, and its prediction speed exceeds that of the original SSD by a factor of three. In the post-processing module, we describe in detail how the echoes are processed into point clouds. The system was tested on a test set, and it can be seen from a visual perspective that satisfactory results were obtained for the extraction of buildings and trees. The system $mAP^{IoU=0.5}$ is 0.812, and the FPS is greater than 34. The results prove that this ASTIL processing system can achieve fast ground target extraction based on a single echo and a single data source.

**Keywords:** airborne streak tube imaging LiDAR (ASTIL); ground target extraction; object detection; single-shot multibox detector (SSD)



**Citation:** Yan, Y.; Wang, H.; Song, B.; Chen, Z.; Fan, R.; Chen, D.; Dong, Z. Airborne Streak Tube Imaging LiDAR Processing System: A Single Echo Fast Target Extraction Implementation. *Remote Sens.* **2023**, *15*, 1128. https://doi.org/10.3390/rs15041128

Academic Editors: Silvia Liberata Ullo, Alfonso Farina, Yu Yao, Harun Taha Hayvaci and Pia Addabbo

Received: 19 December 2022
Revised: 15 February 2023
Accepted: 15 February 2023
Published: 18 February 2023



## 1. Introduction

Airborne LiDAR is an active ground observation system, which has the advantages of round-the-clock, strong penetration, accurate range finding, and short production cycle [1,2]. Since its ability to quickly collect 3D terrain data in local areas [3], it has been widely used in urban 3D modeling [4,5], forestry resources survey [6,7], power facility monitoring [8,9], disaster assessment [10,11], and other applications [12–14]. Airborne streak tube imaging LiDAR (ASTIL) is a novel LiDAR that was originally applied for underwater object detection [15], and is still not widely available.

It is generally known that the data format obtained by conventional LiDAR based on single-point scanning is generally a point cloud. Most post-processing algorithms based on point clouds generally demand a certain scale of points to obtain reliable results. Taking the filtering algorithm based on adaptive TIN (triangular irregular network) proposed by Axelsson [16] as an example, the local lowest point within a user-defined grid is selected as the seed point, and the grid size should not be smaller than the size of the largest structure. This requirement for the number of points makes the real-time processing of a single echo signal from conventional airborne LiDAR a difficult problem to solve.

Compared with the conventional LiDAR, the laser footprint from ASTIL is shaped into a strip that is hundreds of meters long, and a streak tube is used to collect the echo signal [17]. These configurations result in the ASTIL having the advantages of wide field of view and high info acquisition efficiency. Because of the idiosyncratic working mechanism of ASTIL, its raw echo signal is a two-dimensional single-channel digital image. This kind of echo signal is rich in semantic information and is capable of reflecting the cross-section

information of surface objects in the irradiated area [18]. Therefore, it has the potential to directly identify the target only by relying on a single echo and a single data source. However, the direct application of the ASTIL raw echo for ground objects extraction has not received sufficient attention [17,18].

In terms of data processing, S. Zhang [19] presented the solution that geospatial artificial intelligence applies deep learning techniques to help solve complex detection and classification problems. Currently, the state-of-the-art deep learning benchmark frameworks in the field of object detection mainly include Faster RCNN, YOLO (You Only Look Once) [20], SSD (Single Shot MultiBox Detector) [21], and their derivatives [22]. Faster RCNN is the first deep learning framework to implement end-to-end object detection [23]. The most prominent contribution of this framework is the proposed RPN (Region Proposal Network), which associates the proposal region generation and the convolutional network through the anchor mechanism, with high accuracy [24,25]. This framework with RPN is called two-stage algorithms, and the prediction accuracy is higher because of the introduction of RPN, however, at the cost of a decrease in inference speed [26]. One-stage algorithms abandon the time-consuming component RPN and treat the detection task as a regression problem, such as YOLO, SSD [27]. SSD is a new object detection framework proposed by W. Liu et al. [21] in 2015. It is considered as the second one-stage object detection framework in the deep learning era [28]. SSD predicts objects of different scales from feature maps of different scales, achieving high detection accuracy. It uses small convolutional filters applied to feature maps to predict category scores and box offsets for a fixed set of default bounding boxes. On the Pascal VOC2007 test dataset, the inference speed of the SSD300 model is as high as 59 FPS, which significantly outperforms YOLOV1 in terms of speed and accuracy. However, the above frameworks were all proposed for RGB real scene images. For ASTIL echo signals, the semantic information is simpler and has fewer categories, and ASTIL has higher real-time requirements for signal processing. As a result, the original SSD network cannot be suitable for the fast processing of ASTIL echo, and needs to be structurally optimized (see Section 3.1).

In this paper, we propose an ASTIL processing system. This system only needs a single echo and a single data source to achieve fast ground target extraction, which gives ASTIL the potential for real-time ground target extraction. This system consists of two modules: an object detection module and a post-processing module. In the object detection module, we structurally optimize the original SSD according to the ASTIL signal characteristics, and propose a Autofocus SSD to speed up the prediction speed. In the post-processing module, we show in detail how the signal is processed into a point cloud. Then, we conducted an experiment to compare the performance of the Autofocus SSD and other state-of-the-art networks. An ablation experiment was executed, and the optimal base network structure was explored. Finally, we tested the overall performance of the system.

## 2. Background Knowledge

### *2.1. Airborne Streak Tube Imaging LiDAR*

ASTIL adopts the "pendulum" scanning mode, as shown by the yellow dotted line in Figure 1a, and the scanning trajectory is zigzag along the flight direction of the flight vehicle. Figure 1b shows the scanning trajectory of the ASTIL laser footprint in a certain scene. Some trees and buildings are located within the scanned area. Here, $F_1$ to $F_4$ respectively represent laser footprints irradiated on different objects. Echo signals of footprint $F_1$ to $F_4$ are shown in Figure 2.

As can be seen from Figure 2, the echo morphological features from buildings and trees in the ASTIL echo signal are obviously different. The echo from a building is generally stronger and has two scarps adjacent to the echo from the ground. The echo from trees is usually low in intensity and appears as a diffuse point cloud. Therefore, two kinds of ground objects in the ASTIL echo signal can be extracted according to their different echo morphological features.

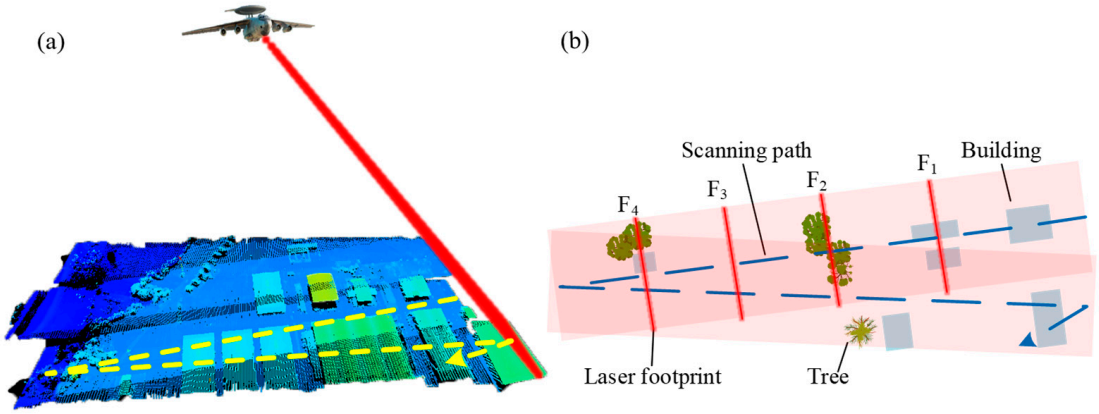


**Figure 1.** (**a**) ASTIL scanning mechanism [18]; (**b**) ASTIL footprint scanning trajectory.

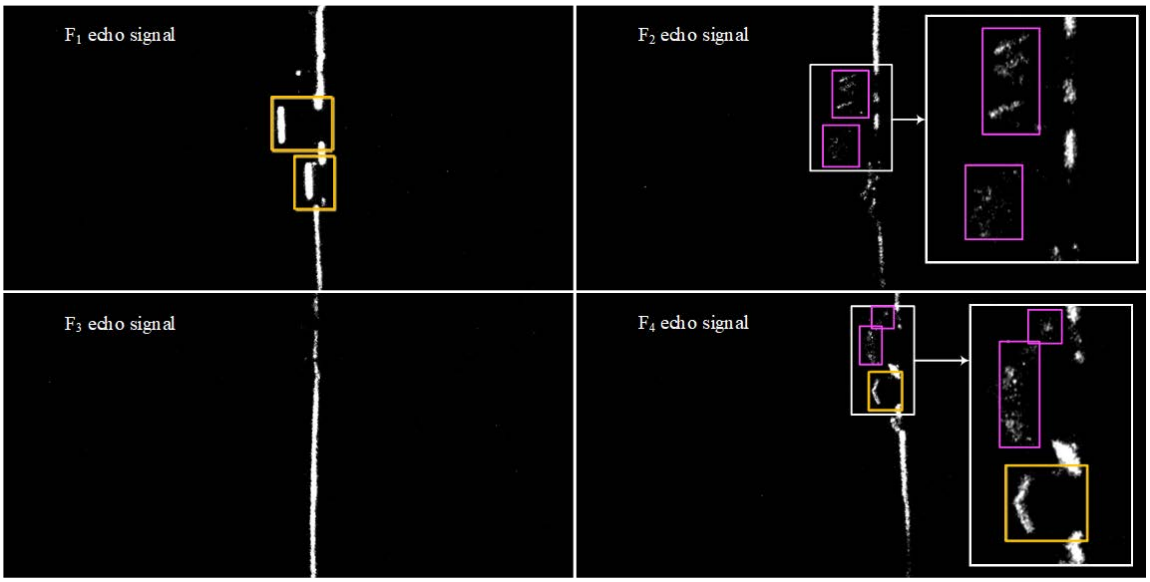


**Figure 2.** Echo signals in the scene of Figure 1b; The boxes with white border show the local details of the signal; Signals representing trees and buildings are marked by boxes with pink and orange borders, respectively.

### *2.2. Data Collection and Annotation*

The data used for training and test sets were collected near Hanzhong City, Shaanxi Province, China. Figure 3 shows a typical scene of the location. In this scene, many buildings and trees are irregularly staggered, and this scene can represent the distribution pattern of ground objects in most urban low-rise residential areas. As a result, the collected data has a good representativeness for the urban low-rise residential area.

At the phase of data collection, the aircraft was flying at an altitude of about 3000 m above sea-level, and ASTIL's laser repetition frequency was 1000 Hz. More details of the data collection are shown in Table 1. The ASTIL irradiated laser footprint on the ground surface was about 130 m long and 0.5 m wide. After data collection, we annotated these data using the tool, LabelImg. The specific details of various data sets are shown in Table 2. The class-imbalance is caused by the large difference in the number of categories of buildings and trees in the raw training set. To address this, we flipped the echo signal containing only trees up and down, and down sampled the signal containing buildings to get the final training set.

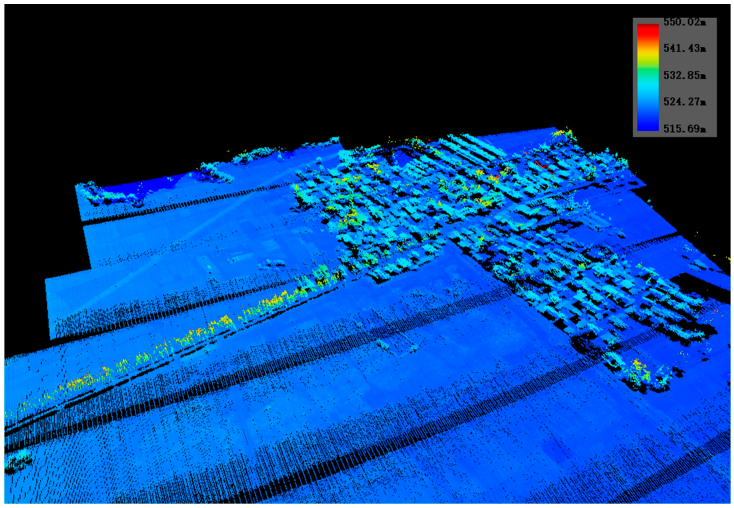


**Figure 3.** Point cloud of data collection area.

**Table 1.** Data collection details.

| Item | Configuration Info |
|---|---|
| acquisition platform | Harbin Y-12 (fixed-wing aircraft) |
| laser wavelength | 532 nm |
| echo type | waveform sampling |
| training set acquisition time | 2014.8.18 14:00:00 |
| test set acquisition time | 2014.8.18 13:41:38 |

**Table 2.** Details of various data sets.

| Objects | Raw Training Set | Training Set | Test Set |
|---|---|---|---|
| tree | 9249 | 11,223 | 18,537 |
| building | 28,124 | 11,086 | 14,018 |

## 3. ASTIL Echo Signal Fast-Processing System

The overall framework of the ASTIL fast-processing system is shown in Figure 4, which mainly includes two modules: object detection and post-processing. More details are given in the following sections.

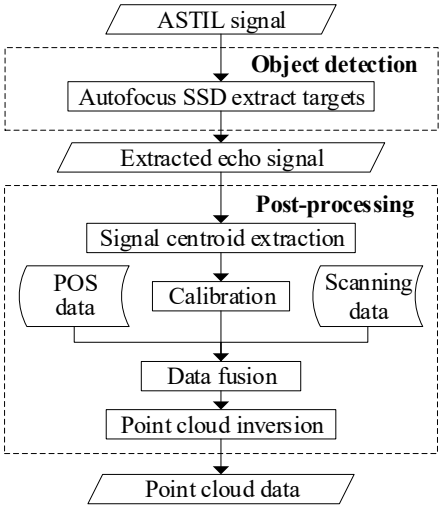


**Figure 4.** ASTIL fast-processing system.

### *3.1. Autofocus SSD Network*

#### 3.1.1. Hierarchical Setting of Default Box Size Based on K-Means

In the SSD network, default boxes with suitable sizes can not only shorten the regression time, but also improve the prediction accuracy of the model. The default box size applied in the original SSD is specifically designed for real scene images. However, the objects in these real scene images are not the same size as those in the ASTIL echo signals. The k-means clustering was used to calculate default box sizes. We write $g_i^u$ as an indicator for the $u$-th parameter of the $i$-th ground truth box size. Here, ground truth box parameters only contain height and width, $n = 2$. The distance measure is Euclidean distance, as defined in (1).

$$dist(g_i, g_j) = \sqrt{\sum_{u=1}^{n} \left| g_i^u - g_j^u \right|^2} \tag{1}$$

Ground truth box sizes is clustered into eight clusters, and the results are shown in Figure 5. After rounding the clustering results, sizes of the default boxes are (30, 32), (49, 49), (56, 75), (96, 67), (63, 111), (123, 101), (83, 302), and (71, 174).

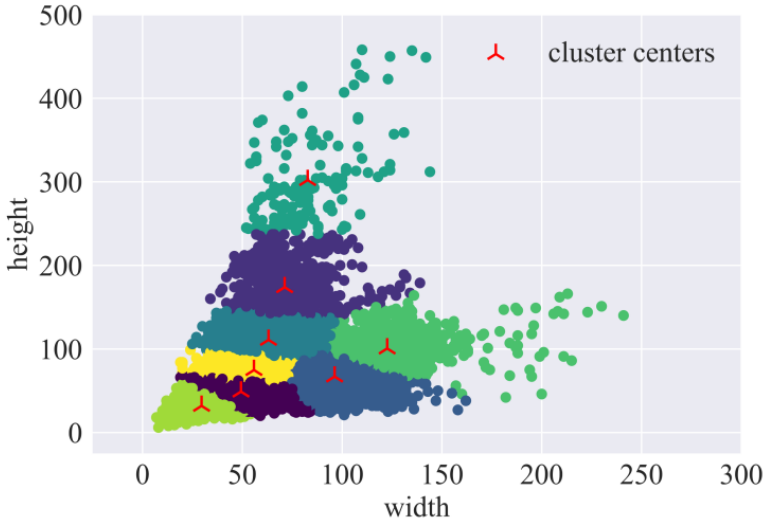


**Figure 5.** Clustering results of ground truth boxes. The clusters obtained by clustering are distinguished by different colors.

#### 3.1.2. Architecture

Different from real scene images, ASTIL echo signals have the characteristics of concentrated echo distribution and straightforward semantic information. In view of these characteristics, we improve the SSD network and propose a Autofocus SSD network. Figure 6 shows the overall architecture of the network. Here, "IRB" represents the inverted residual module in the MobileNetV2 [29]. 'CONV', 'Maxpool', 'FC', and 'BN' denote convolutional layer, maximum pooling layer, fully connected layer, and batch normalization layer, respectively. F1 to F4 represent feature maps. 'LOC' and 'CONF' denote the location and confidence prediction modules of the model, respectively. 'SRE' is the signal region extractor and 'SRE backbone' is the feature extraction network, and these two modules constitute the signal region extraction unit proposed in this paper.

In order to meet the demand of fast processing of echo signals, we improved the original SSD by adding a signal region extraction unit and simplifying the base network, which improves the prediction speed.

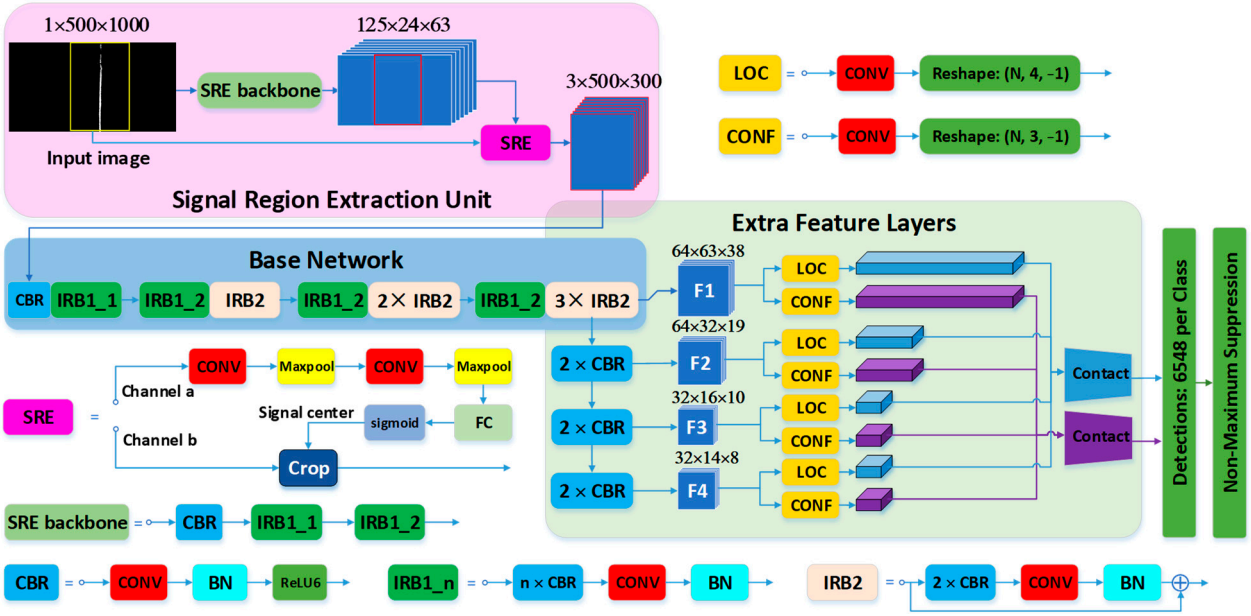


**Figure 6.** Autofocus SSD network architecture. 'CONV', 'Maxpool', 'FC', and 'BN' denote convolutional layer, maximum pooling layer, fully connected layer, and batch normalization layer, respectively. F1 to F4 represent feature maps. 'LOC' and 'CONF' denote the location and confidence prediction modules of the model, respectively. 'SRE' is the signal region extractor and 'SRE backbone' is the feature extraction network, and these two modules constitute the signal region extraction unit proposed in this paper.

#### 3.1.3. Signal Region Extraction Unit

The original SSD network extracts feature maps by the base network and some CBR modules, and then predicts position and confidence on each cell of each output feature map. Write the convolution filter as $\mathrm{Conv}(c_{\mathrm{input}}, c_{\mathrm{output}}, k, s, p)$, where $c_{\mathrm{input}}$, $c_{\mathrm{output}}$, $k$, $s$ denote the input channel, output channel, kernel size, and stride, respectively, and $p$ refers to the number of zeros padded around the input tensor.

For input channel $c$, the number of default boxes on the feature map $n_{\mathrm{dbox}}$, the location prediction module 'LOC' is written as $\mathcal{L}$, can be expressed as (2). Here, $\mathcal{T}_{\xi}$ is defined as a tensor dimensional transformation mapping shown in (3). Suppose the single-scale feature map (such as F1 in Figure 6) $\boldsymbol{F}_{\mathrm{s}} \in \mathbb{R}^{N\times c\times h\times w}$, because there are four regression parameters for ground-truth box, here $\xi$ is set to 4. $\mathcal{L}(\boldsymbol{F}_{\mathrm{s}})$ is also considered as a linear transformation as shown in (4).

$$\mathcal{L} = [\mathrm{Conv}(c, 4\cdot n_{\mathrm{dbox}}, (3,3), 1, 1) \circ \mathcal{T}_4] \tag{2}$$

$$\mathcal{T}_{\xi} : \mathbb{R}^{N\times c\times h\times w} \to \mathbb{R}^{N\times \xi\times \frac{c\cdot h\cdot w}{\xi}} \tag{3}$$

$$\mathbb{R}^{N\times c\times h\times w} \to \mathbb{R}^{N\times 4n_{\mathrm{dbox}}\times h\times w} \to \mathbb{R}^{N\times 4\times hwn_{\mathrm{dbox}}} \tag{4}$$

Similarly, for the categories number $cls$, the confidence prediction module 'CONF' $\mathcal{C}$, can be expressed as (5). $\mathcal{C}(\boldsymbol{F}_{\mathrm{s}})$ is a linear transformation as shown in (6).

$$\mathcal{C} = [\mathrm{Conv}(c, (cls+1)\cdot n_{\mathrm{dbox}}, (3,3), 1, 1) \circ \mathcal{T}_{cls+1}] \tag{5}$$

$$\mathbb{R}^{N\times c\times h\times w} \to \mathbb{R}^{N\times (cls+1)\cdot n_{\mathrm{dbox}}\times h\times w} \to \mathbb{R}^{N\times (cls+1)\times hwn_{\mathrm{dbox}}} \tag{6}$$

Therefore, for multi-scale feature maps $\boldsymbol{F} = (\boldsymbol{F}_1, \boldsymbol{F}_2, \boldsymbol{F}_3, \boldsymbol{F}_4)$, $\boldsymbol{O}_{\mathrm{loc}}$, $\boldsymbol{O}_{\mathrm{conf}}$ denotes the predicted output tensor with respect to the location and confidence, then (7) and (8) are established.

$$\boldsymbol{O}_{\mathrm{loc}} \in \mathbb{R}^{N\times 4\times n_{\mathrm{dbox}}\cdot \sum_{i=1}^{4} h_i w_i} \tag{7}$$

$$\boldsymbol{O}_{\text{conf}} \in \mathbb{R}^{N\times(cls+1)\times n_{\text{dbox}}\cdot\sum_{i=1}^{4} h_i w_i} \tag{8}$$

The raw echo signal (Figure 2) size of ASTIL is 500 × 1000 (height × width). However, as the signal region extraction in Figure 6 shows, those streaks indicating the signal, called signal streaks, are concentrated in only one part of the image (yellow box), and the information in other areas is not substantially useful for the processing of the echo signal. In view of this, the signal region extraction unit (SREU) was proposed.

The SREU is composed of a signal region extraction backbone (SRE backbone) and a signal region extractor (SRE). The SRE backbone is intercepted from the first three layers of the MobileNetV2 network, however, we adjusted the stride of the third layer to (2,2). It is specifically composed of CBR, IRB1_1, and IRB1_2 (See the SRE backbone in Figure 6). In the other hand, the SRE (see SRE in Figure 6) was set up with two channels, the input of channel a is the feature map extracted from the SRE backbone, and the input of channel b is the raw echo signal. The feature map entering channel a is extracted features by two convolutional and pooling layers, after which the predicted signal center is output by a fully connected layer and a sigmoid activation function. Then, the SRE expands 150 pixels left and right based on the predicted signal center, and the signal region is obtained. Table 3 shows the specific configuration of the SRE. The signal center value is clamped to [150, 850] to prevent cropping beyond the image.

**Table 3.** SRE specific configuration.

| Operator | $c_{\text{input}}$ | $c_{\text{output}}$ | $k$ | $s$ | $p$ |
|---|---|---|---|---|---|
| CONV | 24 | 16 | (3, 3) | 1 | none |
| Maxpool | – | – | (3, 3) | – | – |
| CONV | 16 | 8 | (3, 3) | 1 | none |
| Maxpool | – | – | (3, 3) | – | – |
| FC | 624 | 1 | – | – | – |

After the input image is processed by SREU, the size is cropped from raw 500 × 1000 to 500 × 300 with almost no information loss. After that, suppose the input $\boldsymbol{I} \in \mathbb{R}^{N\times c\times h\times w}$, output $\boldsymbol{O}_{\text{loc}}^{\text{SREU}}$ and $\boldsymbol{O}_{\text{conf}}^{\text{SREU}}$, which are processed by SREU, become (9) and (10).

$$\boldsymbol{O}_{\text{loc}}^{\text{SREU}} \in \mathbb{R}^{N\times 4\times \frac{3}{10} n_{\text{dbox}}\cdot\sum_{i=1}^{4} h_i w_i} \tag{9}$$

$$\boldsymbol{O}_{\text{conf}}^{\text{SREU}} \in \mathbb{R}^{N\times(cls+1)\times \frac{3}{10} n_{\text{dbox}}\cdot\sum_{i=1}^{4} h_i w_i} \tag{10}$$

According to the SSD working mechanism, $\boldsymbol{O}_{\text{loc}}^{\text{SREU}}$ and $\boldsymbol{O}_{\text{conf}}^{\text{SREU}}$ are used for post-processing processes such as non-maximum suppression. Compared to and $\boldsymbol{O}_{\text{loc}}$, and $\boldsymbol{O}_{\text{conf}}$, the data size of $\boldsymbol{O}_{\text{loc}}^{\text{SREU}}$ and $\boldsymbol{O}_{\text{conf}}^{\text{SREU}}$ is reduced to 3/10 of original outputs.

#### 3.1.4. Streamlining the Base Network

The original SSD is mainly applied to real scene images, and its base network is the truncated VGG16. However, the ASTIL echo signal possesses simple semantic information and small pattern space compared with real scene images, and the use of complex VGG networks does not achieve fast prediction. Therefore, the base network of the framework proposed adopted the truncated MobileNetV2 with simple structure and low number of parameters for fast signal processing. The specific configuration is shown as the base network in Figure 6. We only intercepted the first 11 layers of MobileNetV2 and adjusted the stride of the convolutional filter in the second CBR in the seventh inversed residual structure to (1, 1) to get feature maps at the appropriate scale. For the exploration of the base network structure, see Section 4.3.3.

3.1.5. Loss Function

Let $x_{ij}^k = \{0, 1\}$ denotes the $i$-th default box matching to the $j$-th ground truth box of category $k$. When $x_{ij}^k$ is 1, it means that the both have a matching relationship and vice versa. In this research, the overall objective loss function is a weighted sum of the localization loss, the confidence loss, and the SREU prediction loss:

$$\begin{aligned} &L(x, s, l, g, \hat{c}^{\text{sig}}, c^{\text{sig}}) \\ &= \frac{1}{N_{\text{dbox}}}(L_{\text{loc}}(x, l, g) + L_{\text{conf}}(x, s)) + 10 \cdot L_{\text{SREU}}(\hat{c}^{\text{sig}}, c^{\text{sig}}) \end{aligned} \tag{11}$$

where $L_{\text{loc}}(x, l, g)$ is the localization loss between the predicted bounding box ($l$) and ground truth box ($g$), and $L_{\text{conf}}(x, s)$ is the confidence loss. $s$ denotes the confidence score. $\hat{c}^{\text{sig}}$ and $c^{\text{sig}}$ are the signal center predicted by SREU and the ground truth signal center, respectively. $N_{\text{dbox}}$ is the number of matched default boxes. Similar to SSD, we regress the offsets for the center $(\text{cx}, \text{cy})$ of the default bounding box ($d$) and for its width (w) and height (h), refer to Equation (12). $l_i^m$ is the regression parameters predicted by the $i$-th bounding box. $\hat{g}_j^m$ is the regression parameters generated by the $i$-th bounding box matching to the $j$-th ground truth box.

$$\begin{aligned} &L_{\text{loc}}(x, l, g) = \sum_{i \in \text{Pos}}^{N_{\text{dbox}}} \sum_{m \in \{\text{cx,xy,w,h}\}} x_{ij}^k \text{smooth}_{\text{L1}}(l_i^m - \hat{g}_j^m) \\ &\hat{g}_j^{\text{cx}} = (g_j^{\text{cx}} - d_i^{\text{cx}})/d_i^{\text{w}} \quad \hat{g}_j^{\text{cy}} = (g_j^{\text{cy}} - d_i^{\text{cy}})/d_i^{\text{h}} \\ &\hat{g}_j^{\text{w}} = \log(g_j^{\text{w}}/d_i^{\text{w}}) \quad \hat{g}_j^{\text{h}} = \log(g_j^{\text{h}}/d_i^{\text{h}}) \end{aligned} \tag{12}$$

Especially, the confidence loss of object recognition is

$$L_{\text{conf}}(x, s) = -\sum_{i \in \text{Pos}}^{N_{\text{dbox}}} x_{ij}^k \log(\hat{s}_i^k) - \sum_{i \in \text{Neg}} \log(\hat{s}_i^0) \tag{13}$$

where $\hat{s}_i^k = \exp(s_i^k)/\sum_k \exp(s_i^k)$.

The prediction loss of the SREU is shown in Equation (14).

$$L_{\text{SREU}}(\hat{c}^{\text{sig}}, c^{\text{sig}}) = \left| \hat{c}^{\text{sig}} - c^{\text{sig}} \right| \tag{14}$$

*3.2. Post-Processing of the Echo Signal*

The ASTIL echo signal only reflects the cross-sectional outline of the surface of the ground object within the laser irradiation area (150 m × 0.5 m). For common ground objects, multiple echo signals are required to complete the detection of the object as a whole. In order to correctly map the complete surface morphology and geographic location of objects, it is necessary to convert the echo signals into a point cloud format. This section proposes a pipeline for converting ASTIL echo signal to a point cloud.

3.2.1. Signal Centroid Extraction

The horizontal coordinate of each pixel of the signal streak in the ASTIL echo signal reflects the distance information of the target, whereas the vertical coordinate of the pixel reflects the spatial location information of the target. Only by accurately obtaining the distance and position information of the pixels representing the target can they be inverted into point clouds using the imaging model combined with pos information.

Signal centroid extraction is similar to the skeleton of signal streak, as shown in Algorithm 1. The method of extracting the signal centroid for each row is as follows. First, find the position ($index_{max}$) and its value ($value_{max}$) of the pixel with the largest value in that row. If there is more than one equal maximum value, the leftmost pixel is used as the maximum pixel. Second, the start and end positions ($pos_{start}$ and $pos_{end}$) of the signal streak larger than the signal threshold ($threshold_{sig}$) are found by traversing to the left and right

with the $index_{max}$, and the signal width ($width_{sig}$) is calculated. If the $value_{max}$ is greater than the $threshold_{sig}$ and the $width_{sig}$ is greater than the $threshold_{width}$, the signal centroid in this row can be calculated as $(pos_{start} + pos_{end})/2$. Otherwise, no signal exists in this row.

**Algorithm 1**: Signal centroid extraction

**Input:** Extracted echo signal $\boldsymbol{I} \in \mathbb{R}^{500\times1000}$
**Output:** Echo streak centroid $\boldsymbol{O} \in \mathbb{R}^{500}$

**function** calc_line(*line_data*, $threshold_{sig}$, $threshold_{width}$):
  $index_{max} \leftarrow$ **argmax**(*line_data*)
  $value_{max} \leftarrow$ **max**(*line_data*)
  $index_{start} \leftarrow 0$
  $index_{end} \leftarrow 1000$
  **for** *index* = $index_{max}$ **to** $index_{start}$ **do**
    **if** *line_data*[*index*] > $threshold_{sig}$ **then**
      $pos_{start}$ = index
    **else**
      **break**
    **end if**
  **end for**
  **for** *index* = $index_{max}$ **to** $index_{end}$ +1 **do**
    **if** *line_data*[*index*] > $threshold_{sig}$ **then**
      $pos_{end}$ = index
    **else**
      **break**
    **end if**
  **end for**
  $width_{sig} = pos_{end} - pos_{start} + 1$
  **if** $value_{max} > threshold_{sig}$ **and** $width_{sig} > threshold_{width}$ **then**
    $out\_x = (pos_{start} + pos_{end})/2$
    **return** *out_x*
  **end if**
  **function** main($\boldsymbol{I}$, $threshold_{sig}$, $threshold_{width}$):
  $\boldsymbol{O} \leftarrow$ [ ]
  **foreach** *line_data* $\in \boldsymbol{I}$ **do**
    $\boldsymbol{O}$.**append**(calc_line(*line_data*, $threshold_{sig}$, $threshold_{width}$))
  **end foreach**
  **return** $\boldsymbol{O}$

### 3.2.2. Calibration

Due to the structural characteristics of STIU, the signal it collects is distorted. In application scenarios, it is necessary to use a calibration matrix to reduce the errors introduced by the distortion. The implementation details of calibration are as Equation (15). Here, $\boldsymbol{I}^{\text{cali}} \in \mathbb{R}^{\iota\times2}$ is the input matrix, $\boldsymbol{O}^{\text{cali}} \in \mathbb{R}^{\iota\times2}$ is the corresponding output matrix, $\iota \in [0, 500)$. $\mathbf{A}^{\text{cali\_x}}, \mathbf{A}^{\text{cali\_y}} \in \mathbb{R}^{500\times1000}$ are the horizontal and vertical calibration matrices, respectively. $A_{i,j}$ denotes the $i, j$ elements of matrix $\boldsymbol{A}$, and $\boldsymbol{A}_{i,:}$ denotes the $i$-th row of matrix $\boldsymbol{A}$.

$$\boldsymbol{O}^{\text{cali}}_{\varphi,:} = \begin{cases} \left(\mathrm{A}^{\text{cali\_y}}_{\text{index}}, \mathrm{A}^{\text{cali\_x}}_{\text{index}}\right) & \left\lceil I^{\text{cali}}_{\varphi,1} \right\rceil = \left\lfloor I^{\text{cali}}_{\varphi,1} \right\rfloor \\ \left(\mathrm{A}^{\text{cali\_y}}_{\text{index}}, \text{lerp}(\boldsymbol{I}^{\text{cali}}_{\varphi,:}, \mathbf{A}^{\text{cali\_x}})\right) & \left\lceil I^{\text{cali}}_{\varphi,1} \right\rceil \neq \left\lfloor I^{\text{cali}}_{\varphi,1} \right\rfloor \end{cases} \tag{15}$$
$$\text{index} = \boldsymbol{I}^{\text{cali}}_{\varphi,:}$$

$$\text{lerp}((j,i), \boldsymbol{A}) = A_{j,\lfloor i \rfloor} + (A_{j,\lceil i \rceil} - A_{j,\lfloor i \rfloor}) \cdot \frac{i - \lfloor i \rfloor}{\lceil i \rceil - \lfloor i \rfloor} \tag{16}$$

3.2.3. Data Fusion

Data fusion refers to the operation of fusing echo signals with global positioning system (GPS), inertial measurement unit (IMU), and scan angle data. This information is an indispensable part of the point cloud inversion. However, these data originate from devices with different acquisition frequencies. In this experiment, the acquisition frequency of GPS and IMU is 200 Hz, and the counterpart of ASTIL is 1000 Hz. For the data fusion, POS were interpolated to coincide with the ASTIL acquisition frequency. The POS data acquired per second can be expressed as:

$$\boldsymbol{D} = \left(\boldsymbol{d}^{\mathrm{t}_0}, \cdots \boldsymbol{d}^{\mathrm{t}_{99}}\right)^{\mathrm{T}},\ \boldsymbol{D} \in \mathbb{R}^{200\times 7}$$
$$\boldsymbol{d}^t = (t, d_t^{\text{longitude}}, d_t^{\text{latitude}}, d_t^{\text{altitude}}, d_t^{\text{roll}}, d_t^{\text{pitch}}, d_t^{\text{yaw}})$$

where $t$ indicates the acquisition moment and the meanings of other variables are shown in the superscripts. The interpolation method is as in Equation (17). Here, $t^{\text{echo}}$ is the acquisition moment of the echo signal, $\boldsymbol{D}_{\varphi,i}$ denotes the $i$-th element of the $\varphi$-th element in $\boldsymbol{D}$. The relationship between $t^{\text{echo}}$, $\boldsymbol{D}_{\varphi}$, and $\boldsymbol{D}_{\varphi+1}$ is as in Equation (18).

$$\text{lerppos}(\boldsymbol{D}_{\varphi}, \boldsymbol{D}_{\varphi+1}, t^{\text{echo}}) = \frac{t^{\text{echo}} - \boldsymbol{D}_{\varphi,0}}{\boldsymbol{D}_{\varphi+1,0} - \boldsymbol{D}_{\varphi,0}} \cdot \left(\boldsymbol{D}_{\varphi+1} - \boldsymbol{D}_{\varphi}\right) + \boldsymbol{D}_{\varphi} \tag{17}$$

$$t^{\text{echo}} \in \left[\boldsymbol{D}_{\varphi,0}, \boldsymbol{D}_{\varphi+1,0}\right) \tag{18}$$

## 4. Experiment and Results

The ASTIL echo signal fast-processing system was implemented with Python, PyTorch, OpenCV, NumPy, and Open3D. The experimental platform configuration is Windows 10, AMD Ryzen 5 5600G, 32 GB of RAM, and Nvidia RTX A4000.

### *4.1. Network Training Strategy*

4.1.1. Autofocus SSD

The base network for the Autofocus SSD accepts the MobileNetV2 loaded with pre-trained weights provided by PyTorch. Unlike original SSD, no data augmentation techniques were used during training. Two default boxes were set per feature layer. Specifically, default boxes sizes in the first feature layer are (30, 32) and (49, 49), and the second layer are (56, 75) and (96, 67), the third layer are (63, 111) and (123, 101), and the fourth layer are (83, 302) and (71, 174). The training was divided into two stages. First, the base network and extra feature layers were frozen, and the SREU branch was trained for 10 epochs. At this time, the loss is only the $L_{\text{SREU}}$ component in (11). After that, the branch SREU, base network, and extra feature layers were trained for 50 epochs. The total loss at this stage is (11). At each stage, the Autofocus SSD was trained end-to-end using the stochastic gradient descent algorithm with a batch size of 32. The momentum was fixed to 0.9, and the weight decay was selected to be 0.0005. The initial learning rate is set to 0.005, and learning rate becomes half of the original after every five epochs.

4.1.2. Other Networks

For Faster RCNN, original SSD and YOLOV5s, the training batch was set to 16, and the size of the fed image was (3, 500, 1000). Their backbones adopted all feature extraction layers in MobileNetV2. The learning rate decreasing schedule was consistent with Autofocus SSD. Data augmentation and other hyperparameter configurations were left as-were.

4.1.3. Evaluation Metrics

Some evaluation metrics were adopted to analyze the extraction performance of the proposed system: mean average precision (mAP), frames per second (FPS), multiply-accumulate computations (MACs), and precision of SREU ($\text{precision}_{\text{SREU}}$). $\text{mAP}^{\text{IoU}=\delta}$

represents the mAP of the network when the IoU between the predicted bounding box and the ground truth is greater than $\delta$ as the positive sample. MACs consist of one multiplication operation and one addition operation, which is approximately equal to two floating point operations [30]. In this paper, we employ it to evaluate the computational cost of the model. $\text{precision}_{\text{SREU}}$ is defined as follows:

$$\text{precision}_{\text{SREU}} = \frac{\left|\{x_i | \text{IoU}(\hat{S}^i, S^i) > 0.85\}\right|}{|\chi|} \tag{19}$$

where $\chi = \{x_1, x_1, \cdots x_m\}$ is the validation set, $\hat{S}^i$ and $S^i$ are the predicted signal region and the ground truth signal region of $\mathbf{x}_i$. $|\cdot|$ represents the number of elements in a set.

### *4.2. Echo Signal Detection Result*

The ASTIL echo signals were predicted using the Autofocus SSD we proposed, and the results are shown in Figure 7. From a visual perspective, this model is able to detect objects effectively. It's $\text{mAP}^{\text{IoU}=0.5}$ on the validation set is 0.832 and the FPS is 84.54 (Table 4). $\text{precision}_{\text{SREU}}$ is 0.933, indicating that SREU can accurately extract the signal region. The parameter quantity of this model is only 2.36% of the original SSD, and the FPS is 317.94% of the original SSD.

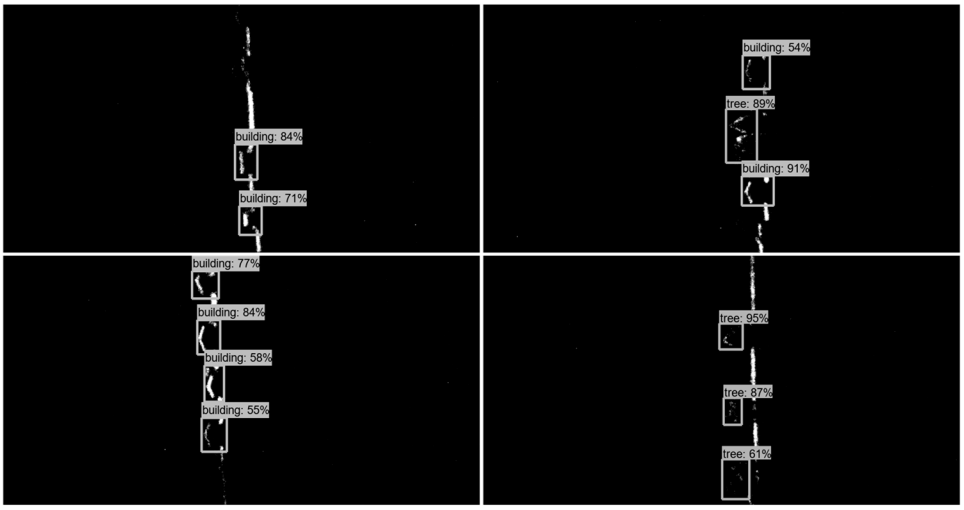


**Figure 7.** Detecting objects on ASTIL raw echo signals using Autofocus SSD.

**Table 4.** Network performance comparison.

| Method | $\text{mAP}^{\text{IoU}=0.5}$ | Params | FPS | $\text{precision}_{\text{SREU}}$ |
|---|---|---|---|---|
| Faster RCNN | 0.827 | 82.32 M | 45.35 | – |
| SSD | 0.842 | 13.57 M | 26.59 | – |
| YOLOV5s | 0.787 | 2.92 M | 87.72 | – |
| Autofocus SSD | 0.832 | 0.32 M | 84.54 | 0.933 |

### *4.3. Autofocus SSD Analysis*

#### 4.3.1. Compared with Baseline Methods

Some state-of-the-art object detection networks and the network we propose were tested, as shown in Table 4. It can be seen here that among these networks, although YOLOV5s has the fastest prediction speed, its mAP is the worst. The SSD has the best mAP, but the FPS is only 26.59. Faster RCNN also has a satisfactory detection result, with a mAP of 0.827. However, the parameters of the model are enormous, with params of 82.32 M. Combining the trade-off between prediction speed and prediction accuracy, Autofocus SSD is the optimal framework. Its prediction accuracy is only 1% lower than the original SSD,

but it has as high as 84.54 FPS, which is comparable to the prediction speed of YOLOV5s. In addition, its parameter size is 0.32 M, only about one tenth of YOLOV5s counterpart, which is more conducive to its deployment on many mobile and embedded applications. It is important to note here that the speed of inference for the first three models in Table 4 is not consistent with our common sense. It is well known that we always think that the prediction speed of one-stage networks is always faster than that of two-stage networks. However, the prediction speed of Faster RCNN is higher than that of SSD, mainly due to the following reasons:

1. The feature extraction networks of these models were replaced with the same networks;
2. The input image sizes for these models were all set to (3, 500, 1000);
3. The data enhancement techniques for these models were retained and were not set to be identical;
4. ASTIL has fewer foreground targets, and Faster RCNN extracted only a small number of proposals, reducing its prediction elapsed time.

#### 4.3.2. Ablation Study

Table 5 shows the ablation studies of Autofocus SSD. Depth indicates how deep the MobileNetV2 framework is used as the base network. The maximum depth of the feature extraction section in the MobileNetV2 network is 19. The study shows that under the same base network conditions, the SREU module can improve mAP by 5.7% with reducing the computational cost by 62.35%. In the circumstance that both have SREU modules, the depth of the base network will also have an impact on the network prediction performance. The mAP at network depth 11 is 9.2% higher than that at depth 19. The parameters and computing cost of the former are only 12.85% and 88.28% of the latter, respectively, and the FPS from the former is 16.48% higher than the latter.

**Table 5.** Ablation study of Autofocus SSD.

| **Components** | SREU | × | ✓ | ✓ |
|---|---|---|---|---|
| | Depth | 19 | 19 | 11 |
| **Metrics** | **mAP$^{\mathbf{IoU=0.5}}$** | 0.683 | 0.740 | 0.832 |
| | **mAP$^{\mathbf{IoU=0.75}}$** | 0.148 | 0.205 | 0.364 |
| | Params | 2.99 M | 2.49 M | 0.32 M |
| | FPS | 72.93 | 74.10 | 86.31 |
| | MACs | 3.40 G | 1.28 G | 1.13 G |

#### 4.3.3. Base Network Structure Selection

Here we conduct two groups of controlled experiments to explore the effects of network depth and feature map size on model prediction performance. The first group maintains a consistent feature map size by adjusting the stride of the second convolutional layer of the 7th and 14th inverse residual blocks of MobileNetV2 to 1. These models are called Modified MobileNetV2. The second group does not change any parameters in the original MobileNetV2, they are called original MobileNetV2. Their specific details are shown in Tables 6 and 7.

Figure 8 shows the relationship between the depth of the base network and the prediction performance of Autofocus SSD. In terms of Modified MobileNetV2, it can be seen that both mAP$^{IoU=0.5}$ and mAP$^{IoU=0.75}$ have been improved as the network depth increases, and the performance reaches saturation when the network depth is 11. Naturally, FPS tends to decrease with the network depth increasing. Comparing the two groups of data, Modified MobileNetV2 and Original MobileNetV2, the latter's mAP$^{IoU=0.5}$ and mAP$^{IoU=0.75}$ are both lower than the former. In the case where the weight parameters are not changed (Figure 8c), the main factor of performance degradation is the reduction in the feature map size from the (63, 38) to (32, 19). This performance degradation is even more pronounced when the depth is 15, when the feature map size is scaled from (32, 19) to

(16, 10). In terms of prediction speed, when the network depth is greater than 10, the FPS of the two groups of models is roughly the same.

**Table 6.** Relationship between depth and Autofocus SSD prediction performance in Modified MobileNetV2 group.

| Network Depth | $mAp^{IoU=0.5}$ | $mAp^{IoU=0.75}$ | Feature Map Size | Params (M) | FPS |
|---|---|---|---|---|---|
| 5 | 0.658 | 0.184 | | 0.098 | 82.61 |
| 6 | 0.733 | 0.241 | | 0.113 | 95.57 |
| 7 | 0.781 | 0.292 | | 0.128 | 90.57 |
| 8 | 0.811 | 0.316 | | 0.154 | 88.38 |
| 9 | 0.821 | 0.338 | (63, 38), | 0.209 | 87.25 |
| 10 | 0.831 | 0.350 | (32, 19), | 0.263 | 86.45 |
| Autofocus SSD | 0.832 | 0.364 | (16, 10), | 0.317 | 86.31 |
| 12 | 0.831 | 0.358 | (14, 8), | 0.389 | 83.74 |
| 13 | 0.836 | 0.365 | | 0.507 | 82.82 |
| 14 | 0.835 | 0.358 | | 0.625 | 80.64 |
| 15 | 0.813 | 0.324 | | 0.791 | 80.08 |
| 16 | 0.824 | 0.340 | | 1.110 | 79.20 |

**Table 7.** Relationship between depth and Autofocus SSD prediction performance in Original MobileNetV2 group.

| Depth | $mAp^{IoU=0.5}$ | $mAp^{IoU=0.75}$ | Feature Map Size | Params (M) | FPS |
|---|---|---|---|---|---|
| 9 | 0.799 | 0.297 | (32, 19), (16, 10), (8, 5), (6, 3) | 0.209 | 93.13 |
| 11 | 0.804 | 0.300 | (32, 19), (16, 10), (8, 5), (6, 3) | 0.317 | 87.91 |
| 13 | 0.799 | 0.295 | (32, 19), (16, 10), (8, 5), (6, 3) | 0.507 | 84.18 |
| 15 | 0.725 | 0.191 | (16, 10), (8, 5), (4, 3), (2, 1) | 0.791 | 80.69 |

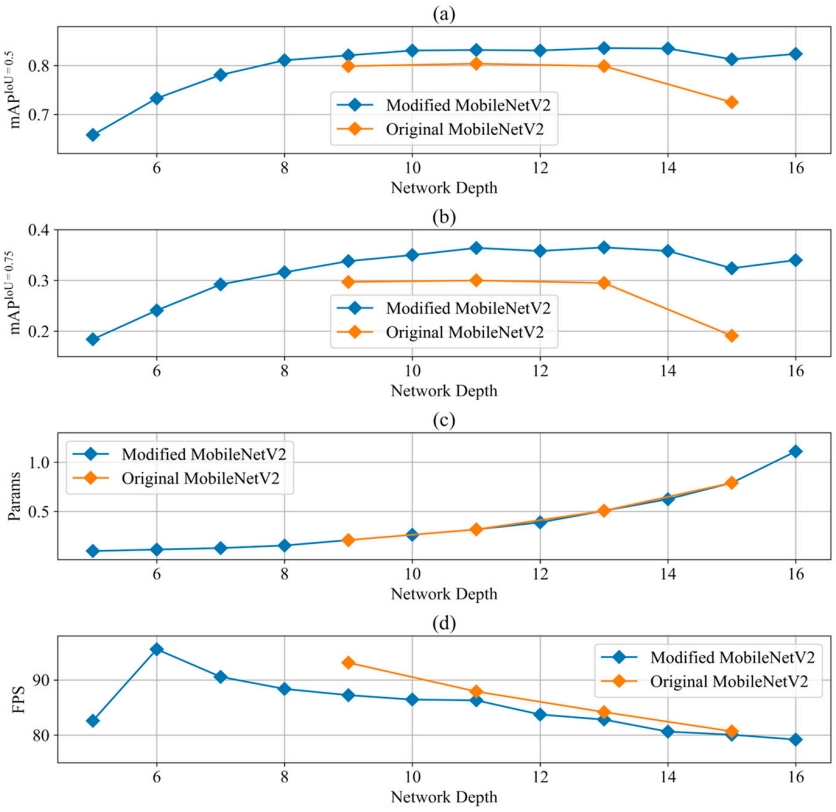


**Figure 8.** Relationship between base network depth and Autofocus SSD prediction performance: (**a**) $mAP^{IoU=0.5}$, (**b**) $mAP^{IoU=0.75}$, (**c**) Params, (**d**) FPS.

Therefore, according to the above results, the network with a depth of 11 in the Modified MobileNetV2 group is the optimal base network for the Autofocus SSD.

### *4.4. ASTIL Fast-Processing System Evaluation*

In this section we used the framework shown in Figure 4 to process ASTIL raw echo signals. We employed nine processes to accelerate the data processing and applied Open3D to display the data in real time.

We tested the system on the test set and the results are shown in Figure 9. Two regions ((a), (d)) were tested, and it is evident that the system is able to roughly extract targets from a visual perspective. Comparing (a) and (b), it can be seen that the system can even extract the rough outlines of trees under complex scene conditions. However, the extraction of buildings is not satisfactory. It can be seen from (c) and (f) that there are some interstices in the extraction results of buildings, which may be partly due to incomplete echo signals of ASTIL, and partly due to insufficient representation of the training set.

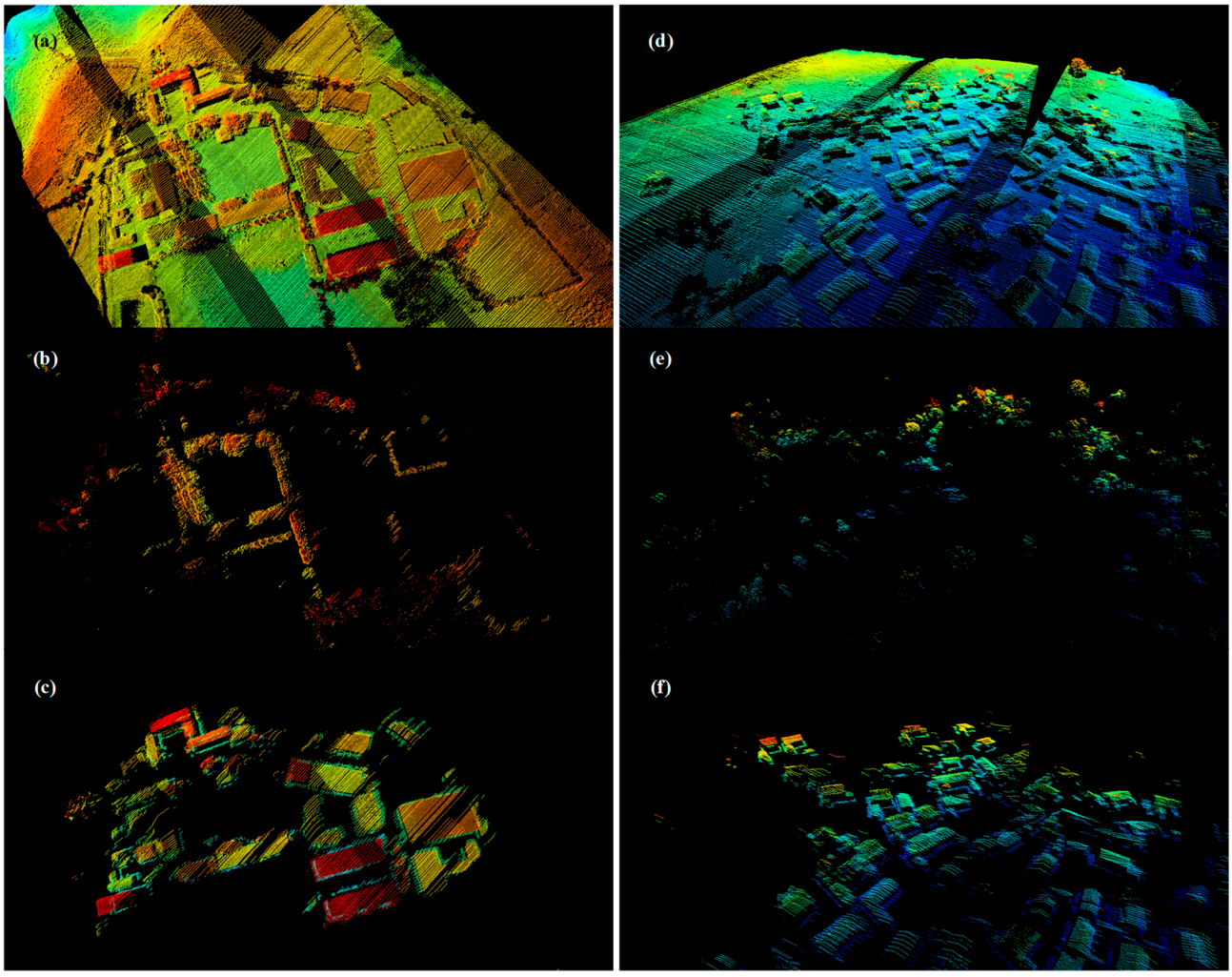


**Figure 9.** Target extraction results of ASTIL fast-processing system. (**b**,**c**) are the tree extraction result and the building extraction result for the area (**a**), respectively. (**e**,**f**) are the tree extraction result and the building extraction result for the area (**d**), respectively.

See Table 8 for more statistical information. The gap between the $mAP^{IoU=0.5}$ obtained on the test set and that obtained on the validation set is within an acceptable range, indicating that the model, Autofocus SSD, has good generalization ability. However, the system FPS is lower than the model FPS, which is only 40.34% to 45.44% of the model, mainly because part of the time overhead is spent on the post-processing. In addition, the

extraction efficiency of the system for different targets is also different, which is mainly due to the difference in the number of ground objects in the measured area, and the prediction rate is higher with fewer ground objects.

**Table 8.** ASTIL fast processing system performance evaluation.

| Sys $mAP^{IoU=0.5}$ | Sys $mAP^{IoU=0.75}$ | Model FPS | Sys FPS | |
|---|---|---|---|---|
| | | | Building | Tree |
| 0.812 | 0.295 | 86.31 | 34.82 | 39.22 |

## 5. Conclusions

In this paper, we propose a fast-processing system for the novel LiDAR ASTIL. This system has the advantage of only depending on a single echo and a single data source to achieve fast ground target extraction, which contributes to ASTIL having the potential for real-time ground target extraction.

The system mainly includes two modules: object detection and post-processing. For object detection, in order to achieve the purpose of fast extracting echo signals, we have carried out structural optimization on the SSD, and proposed an Autofocus SSD, which can achieve $mAP^{IoU=0.5}$ up to 0.832 and FPS up to 84.54. The prediction speed is more than three times faster than the original SSD. For post-processing, we show in detail how to process ASTIL echo signals into point clouds for real-time display.

The system is tested on the test set, system $mAP^{IoU=0.5}$ is 0.812, system FPS is greater than 34, which shows that the system can satisfactorily achieve fast ground target extraction from ASTIL echo signals.

Nevertheless, there are still some potential improvements that can be made. First, for network prediction, we have not adopted any hardware optimization techniques, such as ONNX Runtime inference accelerator, TensorRT inference optimizer. Second, for the implementation of the system, we only enabled multi-process acceleration, but the utilization rate of each CPU is only about 10%, and there is still a lot of potential for improvement. Consequently, in our future work, we are going to apply the above two techniques to accelerate our system for real-time processing of ASTIL echo signals.

**Author Contributions:** Conceptualization, Y.Y. and Z.D.; methodology, Y.Y.; investigation, Y.Y. and B.S.; resources, R.F.; writing—original draft preparation, Y.Y. and Z.D.; writing—review and editing, H.W. and Z.C.; project administration, R.F.; funding acquisition, D.C. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by the National Natural Science Foundation of China (grant no. 62192774), and the National Key Scientific Instrument and Equipment Development Projects of China (grant no. 2012YQ040164).

**Data Availability Statement:** The data presented in this study are available on request from the corresponding author. The data are not publicly available due to funder regulations.

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

## References

1. Hang, R.L.; Li, Z.; Ghamisi, P.; Hong, D.F.; Xia, G.Y.; Liu, Q.S. Classification of Hyperspectral and LiDAR Data Using Coupled CNNs. *IEEE Trans. Geosci. Remote Sens.* **2020**, *58*, 4939–4950. [CrossRef]
2. Zhao, X.D.; Tao, R.; Li, W.; Li, H.C.; Du, Q.; Liao, W.Z.; Philips, W. Joint Classification of Hyperspectral and LiDAR Data Using Hierarchical Random Walk and Deep CNN Architecture. *IEEE Trans. Geosci. Remote Sens.* **2020**, *58*, 7355–7370. [CrossRef]
3. Zhou, R.Q.; Jiang, W.S. A Ridgeline-Based Terrain Co-registration for Satellite LiDAR Point Clouds in Rough Areas. *Remote Sens.* **2020**, *12*, 2163. [CrossRef]
4. Huang, J.; Stoter, J.; Peters, R.; Nan, L.L. City3D: Large-Scale Building Reconstruction from Airborne LiDAR Point Clouds. *Remote Sens.* **2022**, *14*, 2254. [CrossRef]
5. Liu, X.J.; Ning, X.G.; Wang, H.; Wang, C.G.; Zhang, H.C.; Meng, J. A Rapid and Automated Urban Boundary Extraction Method Based on Nighttime Light Data in China. *Remote Sens.* **2019**, *11*, 1126. [CrossRef]

6. Pirotti, F. Analysis of full-waveform LiDAR data for forestry applications: A review of investigations and methods. *iForest* **2011**, *4*, 100–106. [CrossRef]
7. Li, X.; Chen, W.Y.; Sanesi, G.; Lafortezza, R. Remote Sensing in Urban Forestry: Recent Applications and Future Directions. *Remote Sens.* **2019**, *11*, 1144. [CrossRef]
8. Guo, B.; Li, Q.Q.; Huang, X.F.; Wang, C.S. An Improved Method for Power-Line Reconstruction from Point Cloud Data. *Remote Sens.* **2016**, *8*, 36. [CrossRef]
9. Arastounia, M.; Lichti, D.D. Automatic Object Extraction from Electrical Substation Point Clouds. *Remote Sens.* **2015**, *7*, 15605–15629. [CrossRef]
10. Huang, X.M.; Gong, J.; Chen, P.F.; Tian, Y.Q.; Hu, X. Towards the adaptability of coastal resilience: Vulnerability analysis of underground gas pipeline system after hurricanes using LiDAR data. *Ocean Coast. Manag.* **2021**, *209*, 105694. [CrossRef]
11. Liu, Q.R.; Ruan, C.Q.; Guo, J.T.; Li, J.; Lian, X.H.; Yin, Z.H.; Fu, D.; Zhong, S. Storm Surge Hazard Assessment of the Levee of a Rapidly Developing City-Based on LiDAR and Numerical Models. *Remote Sens.* **2020**, *12*, 3723. [CrossRef]
12. Wang, H.Z.; Glennie, C. Fusion of waveform LiDAR data and hyperspectral imagery for land cover classification. *ISPRS J. Photogramm. Remote Sens.* **2015**, *108*, 1–11. [CrossRef]
13. Singh, K.K.; Vogler, J.B.; Shoemaker, D.A.; Meentemeyer, R.K. LiDAR-Landsat data fusion for large-area assessment of urban land cover: Balancing spatial resolution, data volume and mapping accuracy. *ISPRS J. Photogramm. Remote Sens.* **2012**, *74*, 110–121. [CrossRef]
14. Wang, Y.L.; Li, M.S. Urban Impervious Surface Detection From Remote Sensing Images A review of the methods and challenges. *IEEE Geosci. Remote Sens. Mag.* **2019**, *7*, 64–93. [CrossRef]
15. Nevis, A.J. Automated processing for Streak Tube Imaging Lidar data. In Proceedings of the Society of Photo-Optical Instrumentation Engineers, Orlando, FL, USA, 11 September 2003; pp. 119–129. [CrossRef]
16. Axelsson, P. DEM Generation from Laser Scanner Data Using Adaptive TIN Models. *Int. Arch. Photogramm. Remote Sens.* **2000**, *33*, 110–117.
17. Dong, Z.W.; Yan, Y.J.; Jiang, Y.G.; Fan, R.W.; Chen, D.Y. Ground target extraction using airborne streak tube imaging LiDAR. *J. Appl. Remote Sens.* **2021**, *15*, 16509. [CrossRef]
18. Yan, Y.J.; Wang, H.Y.; Dong, Z.W.; Chen, Z.D.; Fan, R.W. Extracting suburban residential building zone from airborne streak tube imaging LiDAR data. *Measurement* **2022**, *199*, 111488. [CrossRef]
19. Zhang, S.; Bogus, S.M.; Lippitt, C.D.; Kamat, V.; Lee, S. Implementing Remote-Sensing Methodologies for Construction Research: An Unoccupied Airborne System Perspective. *J. Constr. Eng. Manag.* **2022**, *148*, 03122005. [CrossRef]
20. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788. [CrossRef]
21. 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 (ECCV), Amsterdam, The Netherlands, 8–16 October 2016; pp. 21–37. [CrossRef]
22. Li, Z.; Wang, Y.C.; Zhang, N.; Zhang, Y.X.; Zhao, Z.K.; Xu, D.D.; Ben, G.L.; Gao, Y.X. Deep Learning-Based Object Detection Techniques for Remote Sensing Images: A Survey. *Remote Sens.* **2022**, *14*, 2385. [CrossRef]
23. Hou, B.; Ren, Z.; Zhao, W.; Wu, Q.; Jiao, L. Object Detection in High-Resolution Panchromatic Images Using Deep Models and Spatial Template Matching. *IEEE Trans. Geosci. Remote Sens.* **2020**, *58*, 956–970. [CrossRef]
24. Fan, Q.C.; Chen, F.; Cheng, M.; Lou, S.L.; Xiao, R.L.; Zhang, B.; Wang, C.; Li, J. Ship Detection Using a Fully Convolutional Network with Compact Polarimetric SAR Images. *Remote Sens.* **2019**, *11*, 2171. [CrossRef]
25. Alganci, U.; Soydas, M.; Sertel, E. Comparative Research on Deep Learning Approaches for Airplane Detection from Very High-Resolution Satellite Images. *Remote Sens.* **2020**, *12*, 458. [CrossRef]
26. Salari, A.; Djavadifar, A.; Liu, X.R.; Najjaran, H. Object recognition datasets and challenges: A review. *Neurocomputing* **2022**, *495*, 129–152. [CrossRef]
27. Tong, K.; Wu, Y.Q. Deep learning-based detection from the perspective of small or tiny objects: A survey. *Image Vis. Comput.* **2022**, *123*, 104471. [CrossRef]
28. Kaur, J.; Singh, W. Tools, techniques, datasets and application areas for object detection in an image: A review. *Multimed. Tools Appl.* **2022**, *81*, 38297–38351. [CrossRef]
29. Sandler, M.; Howard, A.G.; Zhu, M.; Zhmoginov, A.; Chen, L.-C. MobileNetV2: Inverted Residuals and Linear Bottlenecks. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 4510–4520. [CrossRef]
30. Liang, Y.; Ge, C.; Tong, Z.; Song, Y.; Wang, J.; Xie, P. Not All Patches are What You Need: Expediting Vision Transformers via Token Reorganizations. Available online: https://ui.adsabs.harvard.edu/abs/2022arXiv220207800L (accessed on 1 February 2022).