Article

Research on Spaceborne Target Detection Based on Yolov5 and Image Compression

Qi Shi [1,2], Daheng Wang [3], Wen Chen [1,2], Jinpei Yu [1,2,*], Weiting Zhou [4], Jun Zou [4] and Guangzu Liu [4]

1 Innovation Academy for Microsatellites of Chinese Academy of Sciences, Shanghai 201304, China
2 University of Chinese Academy of Sciences, Beijing 100039, China
3 China Satellite Network Group Co., Ltd., Beijing 100000, China
4 School of Electronic and Optical Engineering, Nanjing University of Science and Technology, Nanjing 210094, China
* Correspondence: yujp@microsate.com

Abstract: Satellite image compression technology plays an important role in the development of space science. As optical sensors on satellites become more sophisticated, high-resolution and high-fidelity satellite images will occupy more storage. This raises the required transmission bandwidth and transmission rate in the satellite–ground data transmission system. In order to reduce the pressure from image transmission on the data transmission system, a spaceborne target detection system based on Yolov5 and a satellite image compression transmission system is proposed in this paper. It can reduce the pressure on the data transmission system by detecting the object of interest and deciding whether to transmit. An improved Yolov5 network is proposed to detect the small target on the high-resolution satellite image. Simulation results show that the improved Yolov5 network proposed in this paper can detect specific targets in real satellite images, including aircraft, ships, etc. At the same time, image compression has little effect on target detection, so detection complexity can be effectively reduced and detection speed can be improved by detecting the compressed images.

1. Introduction

With the rapid development of remote sensing space technology, the transmission bandwidth and data rate of remote sensing images continue to increase, resulting in a surge in the amount of image data, which puts pressure on data transmission bandwidth and spaceborne storage. Therefore, remote sensing image compression has become a necessary means to reduce the amount of data and relieve the pressure on transmission bandwidth and storage [1–4]. At present, lossy compression is widely used for high-rate compression coding of remote sensing images. Although this method reduces the amount of image data to a large extent, it also causes a loss of key information in the area of interest of the image. At present, the most commonly used static image compression algorithms are JPEG, JPEG2000, and JPEG-LS [5,6].

JPEG2000 adopts the discrete wavelet transform and EBCOT encoding mechanism and supports lossy and lossless compression. In general, its compression performance is an improvement of 20~40% when compared with JPEG, and in the case of high compression ratios, JPEG2000 has obvious advantages in this regard [5]. It is possible for JPEG2000 to maintain a relatively intact image quality without Mosaic distortion, and it allows users to specify the compression quality of any region and which part of the decompression processing should be given priority. However, its disadvantage is the high complexity of its algorithm. Following the development of the JPEG2000 special compression chip, image data can now be compressed and transmitted by JPEG2000 on board.

On the other hand, the ocean occupies 70% of the Earth's area and in most cases does not contain useful information. Valuable targets in remote sensing images are mainly

airplanes, ships, offshore beaches, and islands. For such targets, the main mission requirements are location annotation and mask segmentation of ships, ports, islands, and aircraft in the image, against the background of a sea surface, a lake surface, or the ground. Aircraft and ship images are different from natural images in scale and resolution, and ports and islands are not obvious because of their rugged contours.

The compression is usually used to decrease the image size and the pressure on the communication link between the satellite and the ground. We can use target detection to further decrease the pressure on the communication link and let the satellite only transmit the image we are interested in.

Traditional target detection generally adopts a rotating target detection method, which can be divided into two main technical routes: one-stage and two-stage detection [7–9].

The one-stage rotating target detection method is based on one-stage horizontal target detection methods such as Yolo [10] and SSD [11]. The target expression in the probe head is replaced by the rotating target expression, and the design for the characteristics of the rotating target is added. For example, TextBoxes++ uses a 3×5 convolution kernel with a larger aspect ratio to extract the features of a rotating target with a larger aspect ratio in the feature extraction stage [12]. Because of the large length and width ratio, the rotation target is arranged more closely than the ordinary target, so TextBoxes++ implements a denser prior box. In RSR [13], the rotation-sensitive convolution layer is used to replace the common convolution layer, and the features with rotation are extracted, which is more suitable for the changeable characteristics of the rotating target. Moreover, an inception module is added to deal with rotating targets with large aspect ratios.

The two-stage rotating target detection method is based on two-stage horizontal target detection methods, such as Faster R-CNN. The traditional methods, such as RR-CNN [14], directly increase the angle prediction in the final regression stage to predict the rotation of the target box. For example, the RRPN [15], R2PN [16], and RR-CNN methods have similar technical routes. Angle parameters are generally added to anchor the frame design, and rotating candidate frames are proposed in the stage of the candidate frame. Then, the features of the rotating candidate frames are extracted by the pooling of the rotating RoI (region of interest). Finally, the regression target boxes and categories are utilized.

Most of the late two-stage rotating target detection methods are based on an RoI transformer [15]. Based on the horizontal anchor frame, the rotating anchor frame is obtained by full connection learning in the candidate frame stage. Unlike RRPN and other methods that set a large number of rotating anchor frames, the RoI transformer greatly reduces the total number of anchor frames, thus reducing the complexity.

Aiming to solve problems such as small targets, unclear identification masks, and fuzzy input images in remote sensing images, we propose a modified Yolov5 network for small target detection. Yolov5 is an excellent neural network in face recognition scenarios with different rotations and sizes. After modifications are made to the Yolov5 network, it can be used to detect small targets with different rotations on a satellite image.

Based on Yolov5 and JPEG2000, an on-board target detection and compression transmission system is designed by combining remote sensing target detection technology with compression storage transmission technology. According to the target set, the system can recognize the image data on the satellite, compress and store the regional images in the target set, and feed back to the ground through the telemetry link, effectively improving the transmission efficiency of high-value remote sensing images between the satellite and the ground.

2. System Design

Figure 1 shows the procedure of the image compression and target detection system on the satellite. The high-data-rate image information is sent to the FPGA for compression using the camera link interface and then the target detection is undertaken on the machine learning platform. Images containing the intended targets are selected and sent to the ground station using the data transmission system. The target detection with the Yolov5

network is implemented on the Jetson Xavier NX, which is produced by NDIVIA and is a popular platform for machine learning [16]. The image compression is undertaken using ADV212, which is produced by ADI [17,18]. ADV212 is a single-chip JPEG2000 codec for video and high-bandwidth image compression applications, enabling it to benefit from the enhanced picture quality and capabilities provided by the JPEG2000 image compression standard. When the camera resolution is high, multiple ADV212 chips can be used in parallel to compress the original image. ADV212 can connect directly to a wide variety of host processors and ASICs using an asynchronous SRAM-style interface, DMA accesses, or streaming mode (JDATA) interface. The ADV212 supports 16- and 32-bit buses for control and 8-/16-/32-bit buses for data transfer. The power consumption of one ADV212 is less than 0.5 W which is very suitable for a satellite with limited energy.
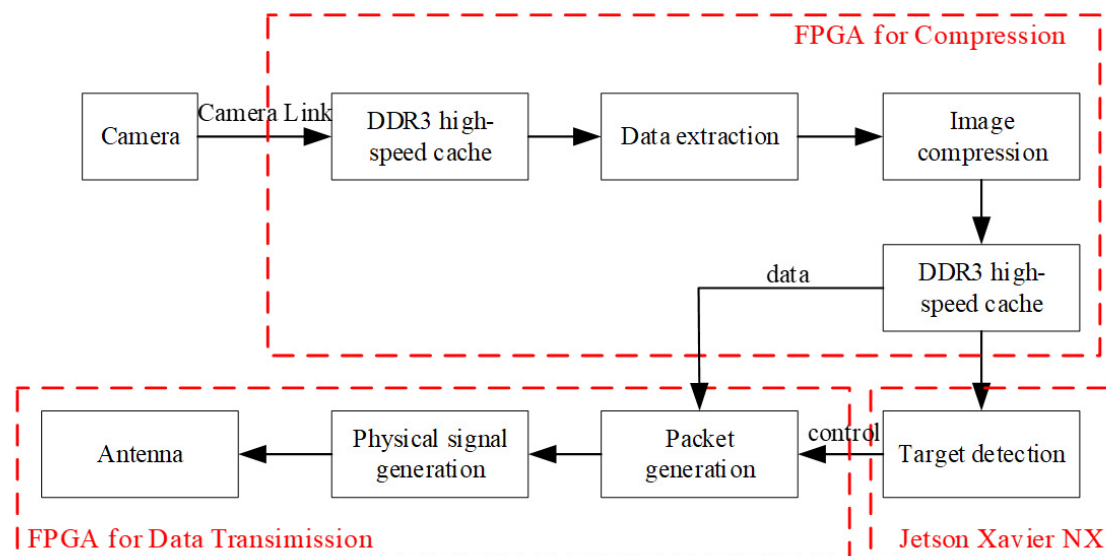

Figure 1. The procedure of an image compression and target detection system on a satellite.

In our design, the original image is compressed before the target detection occurs, which can reduce the complexity and the image size. The original image is first compressed according to the JPEG2000 standard, which can reduce the requirements for the data rate on the interface between the Jetson Xavier NX and the camera. Then, the JP2 image is converted and resized to a JPG image to undergo target detection by the Jetson Xavier NX. An alternative method is detecting the intended targets in the original image data and sending the intended original data to the ADV212 chip to generate the compressed image, which can reduce the task of ADV212, especially in a high-resolution scenario. Although compressed images may affect the detection accuracy, this method can still achieve a high detection rate, which is shown in the simulation results.

The platform based on FPGA and ASIC is the best solution after considering the speed, hardware resources, development cycle, and other factors. As a highly customized chip, ASIC can meet some specific user needs and solve the most difficult core algorithm problems with minimal cost. Therefore, Xilinx 690T and ADV212 are adopted as hardware platforms for image compression in the proposed system.

3. Remote Sensing Target Detection

The existing Yolov5 target detection system is developing rapidly, but there are still some bottlenecks in the detection of small targets, such as in satellite images that have a large field of view and a large size.

This section shows the Yolov5 network and the process of detecting small targets on satellite images.

3.1. Yolov5 Network

Figure 2 is the network structure diagram of Yolov5. The network structure is divided into four parts: input, backbone, neck, and prediction.
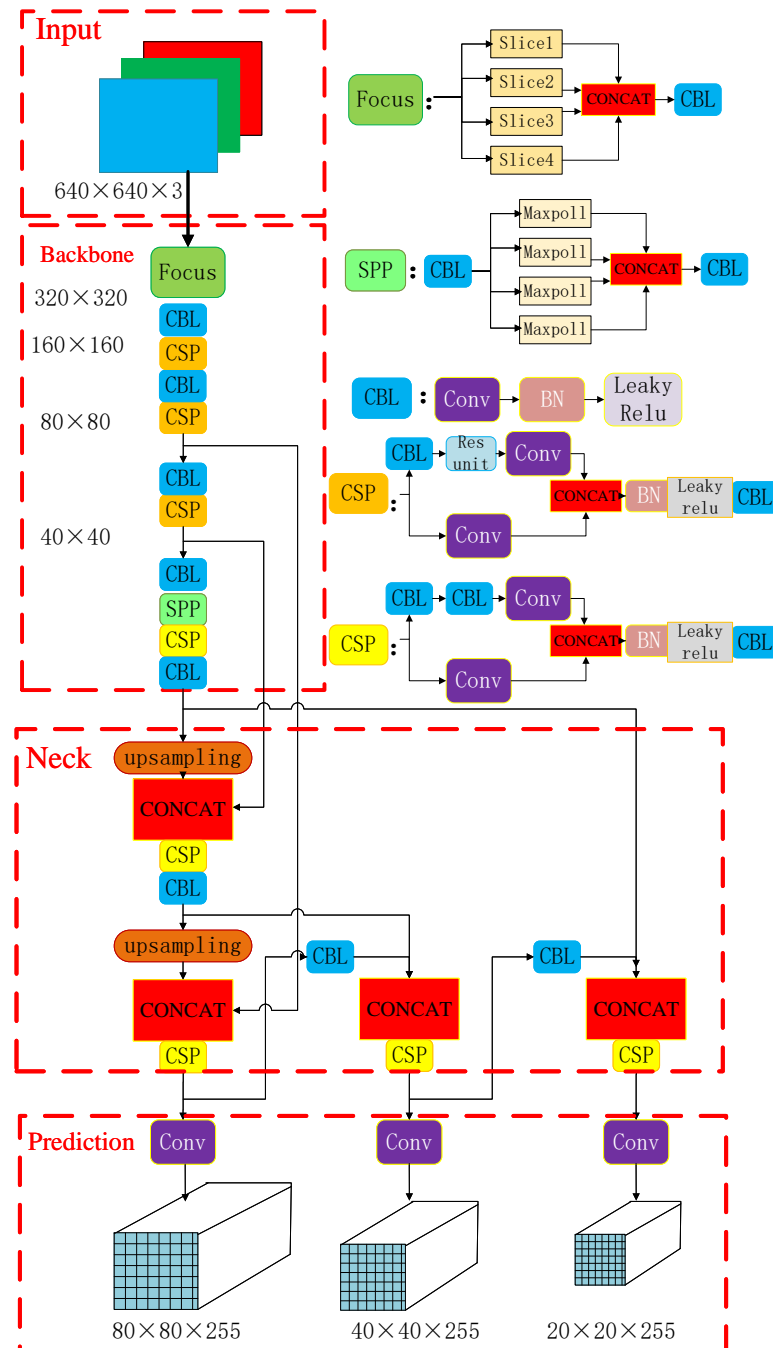

Figure 2. The network structure diagram of Yolov5.

3.1.1. Input Layer

(1). Mosaic data augmentation

The author of the Mosaic data enhancement proposal is also a member of the Yolov5 team. The data enhancement using a stitching effect is created from random zoom, random cropping, and random arrangement, and the detection effect is good for small targets.

(2). Adaptive anchor box calculation

In Yolo's algorithm, there is an anchor box with an initial length and width for different data sets.

In the network training, the network outputs the prediction box on the basis of the initial anchor box, and then compares it with the real box, calculates the gap between the two, and then updates it in reverse to iterate the network parameters. The anchor setting in the model file is the initial anchors box, as shown in Table 1.

Table 1. The anchor setting of Yolov5.

Anchors	Size
8	[10,13,16,30,33,23]
16	[30,61,62,45,59,119]
32	[116,90,156,198,373,326]

(3). Adaptive picture zoom

In the commonly used target detection algorithms, different pictures are different in length and width, so the usually implemented method is to uniformly scale the original pictures to a standard size, such as the widely used 640×640, and to then send them into the detection network.

For rectangular pictures with a different length and width, Yolov5 is processed by minimizing the filling, using gray to fill the scaled picture to scale it to the standard size, and it is then input into the neural network.

3.1.2. Backbone Module

(1). Focus structure

The structure of the focus layer is shown in Figure 3. Taking the structure of Yolov5s as an example, the original 640×640×3 image is input into the Focus structure, using the slice operation, which first becomes a feature map of 320×320×12, and then goes through a convolution operation of 32 convolution cores, and finally becomes a feature map of 320×320×32. The number of convolution kernels for the other models will increase with the model depth.
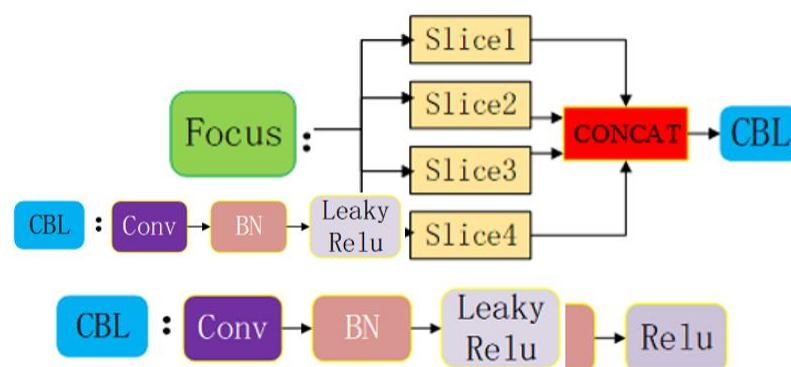

Figure 3. The structure of the focus layer.

Batch normalization (BN) converts the distribution of data in each layer so that it has a zero mean and a variance of 1, so that the distribution of data in each layer is roughly the same, and the training will be easier to converge and prevent overfitting. If only the output data of layer A of the network is normalized and then sent to the next layer (layer B) of the network, the features learned by the network A of the layer will be affected. BN also uses transformation reconstruction and introduces the learnable parameters γ and β, so that the network can learn to restore the original features of the target. The batch normalization is processed as shown in Algorithm 1.

Algorithm 1 The procedure of batch normalization.
Input: Values of x over a mini-batch: $C = \{x_1, x_2, \cdots, x_m\}$;   **Parameters to be learned:** γ, β
Output: $\left\{ y_i = \mathrm{BN}_{\gamma, \beta}(x_i) \right\}$
1. Calculate the mini-batch mean according to $\mu_C = \frac{1}{m} \sum\limits_{i=1}^{m} x_i$;
2. Calculate the mini-batch variance according to $\sigma_C^2 = \frac{1}{m} \sum\limits_{i=1}^{m} (x_i - \mu_C)^2$;
3. Normalize the input according to $\hat{x}_i = \frac{x_i - \mu_C}{\sqrt{\sigma_C^2 + \varepsilon}}$;
4. Scale and shift the input according to $y_i = \gamma \hat{x}_i + \beta \triangleq \mathrm{BN}_{\gamma, \beta}(x_i)$

The Leaky ReLU function is a variant of the classical (and widely used) ReLU activation function whose output has a small slope to the negative input. As the derivative is always non-zero, this reduces the emergence of silent neurons, allowing for gradient-based learning.

$$\mathrm{LeakyReLU}(x) = \begin{cases} x, x > 0 \\ ax, x \le 0 \end{cases}$$

(2). CSP structure

CSPnet (cross stage partial network) is proposed to reduce the calculation of very large networks, mainly from the perspective of network structure design. The CSP module first divides the feature mapping of the base layer into two parts, and then merges them across the stage hierarchy, which can guarantee the accuracy while reducing the calculation amount.

There are two kinds of CSPnet structures in the Yolov5 network. CSP 1 structure is applied to the backbone network, as shown in Figure 4.
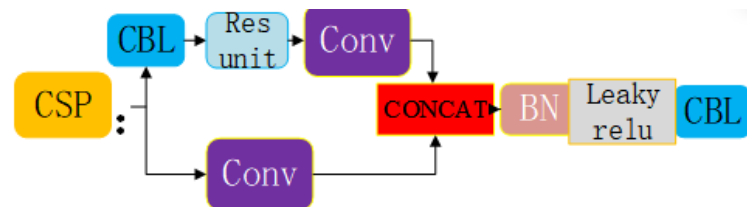

Figure 4. The structure of CSP 1.

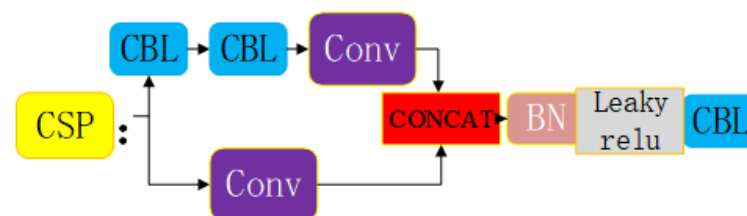Additionally, the other CSP 2 structure is applied in the Neck, as shown in Figure 5.

Figure 5. The structure of CSP 2.

The benefits of adopting CSPnet include enhancing the learning ability of CNN, which adopts a lightweight structure to reduce memory cost while maintaining detection accuracy.

3.1.3. Neck

An FPN and PAN structure is used to combine the characteristics of the target on different layers, as shown in Figure 6. The FPN structure delivers strong semantic features

from top to bottom, whereas the PAN structure delivers strong positioning features from the bottom upwards. It is beneficial for improving the accuracy of target identification.
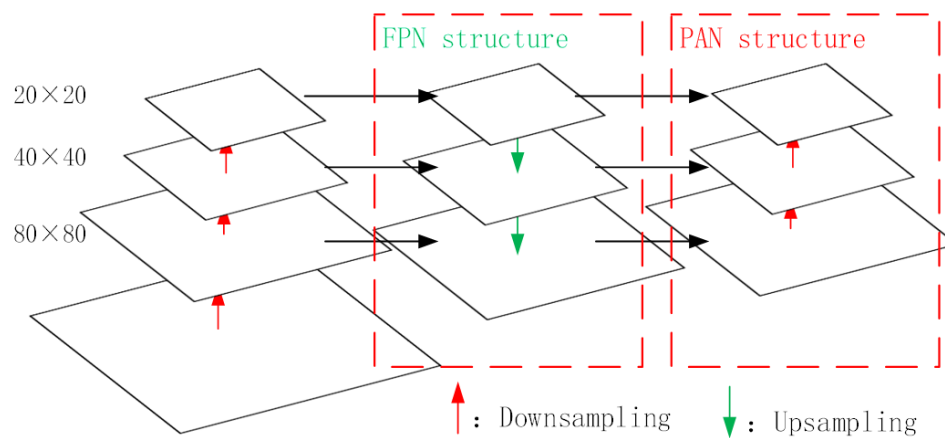

Figure 6. The structure of FPN and PAN.

3.1.4. Prediction

In Yolov5, CIOUs_Loss is used to create the loss function of the bounding box. In the post-processing procedure during target detection, screening for many target boxes usually requires NMS operation. The weighted NMS in Yolov5 can improve the recognition of the target to be detected.

In the official description, there are four versions of the target detection networks, namely Yolov5s, Yolov5m, Yolov5l, and Yolov5x, as shown in Table 2.

Table 2. The parameters of different Yolov5 networks.

	Yolov5s	Yolov5m	Yolov5l	Yolov5x
Depth multiple	0.33	0.67	1.00	1.33
Width multiple	0.50	0.75	1.00	1.25

The four structures are used to control the depth and width of the network through the above two parameters, where depth multiple controls the depth of the network and width multiple controls the width of the network. All four networks have the same frame structure, with only the depths and widths differing.

3.2. Small Object Detection in Remote Sensing

We analyzed the principle of Yolov5 image recognition, taking the commonly used $640 \times 640 \times$ pixel network input as an example (Yolov5s). After several iterations of down-sampling, the largest feature image is 80×80, and each field of vision is only 8×8. If an image of more than 5000 pixels is input, the smallest object pixel width or height the network can identify is around 62.5. However, the size of many objects on the satellite remote sensing images are smaller than this pixel width limit, making them difficult to identify. If the input image size of the network is forcibly increased, it is easy to exceed the GPU's memory, and this also greatly increases the detection time. More effective methods are required to run the algorithm under the limitation of the device.

Based on the above analysis, we made the following improvements to enhance Yolov5's capabilities for detecting small targets:

1. Modification of the model: we added a small detection anchor box, reduced the multiple of the sampling layer as much as possible within the GPU's memory limit, and added a small target feature extraction layer. On this basis, we used a large number of real satellite images for training.

2. Image segmentation: we developed an image segmentation program that is similar to manually cutting large images into several small pieces, and merges the model's outputs after recognition.

As shown in Figure 7, our proposed model is modified on the basis of Yolov5m. An initial detection anchor box has been added, and the corresponding novel structure has been modified to make an additional sampling and fusion layer.
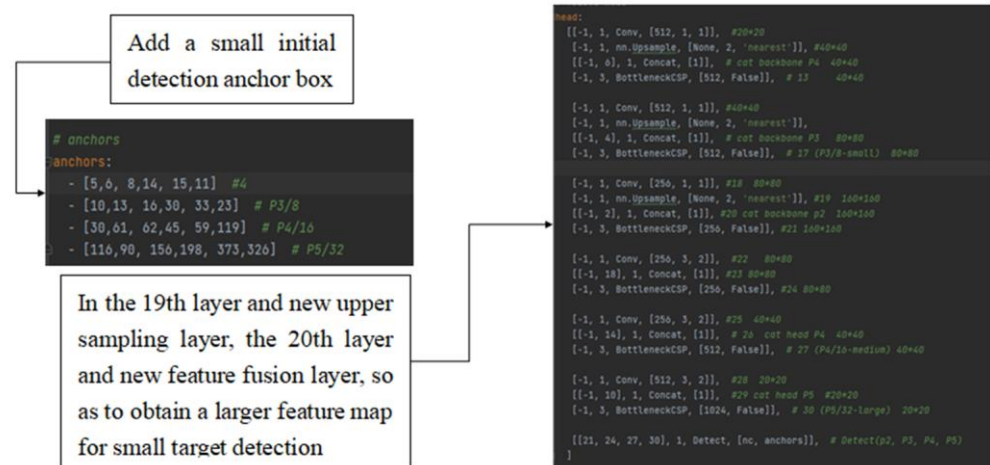

Figure 7. The modification of our proposed Yolov5 network for satellite image detection.

The image segmentation algorithm is implemented as follows: first, we divide the original large-sized image into multiple small slices. The size of a slice should be 640 pixels (as the neural network input is 640×640), with a 20% overlap reserved between each small slice, to prevent small targets from being missed. Each small image is resized to 640×640 as a prediction input. We obtain the detection results from the network and concatenate the output of multiple slices into the detection results of the original large-sized image. Figure 8 shows a simple example of the image segmentation detection method.
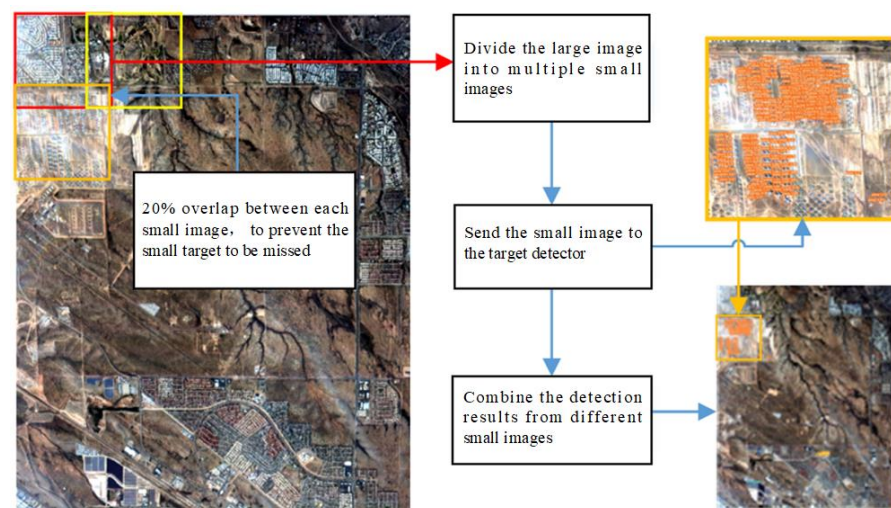

Figure 8. An example of the image segmentation detection method.

4. Image Compression

The basic structure of JPEG2000 compression is shown in Figure 9. First, the original image data are preprocessed, and then the results are transformed by a discrete wavelet. Before the formation of the output bit stream, the transformation coefficient should be

quantized and the embedded block should be coded using EBCOT with optimized truncation. The decoder uses the inverse process of the encoder. After the compressed image data flow is stored or transmitted, the data stream is first unpacked and the entropy decoded, and then the reverse quantization and inverse wavelet transform are carried out [19]. The result of the reverse transform is processed and synthesized in the post process to obtain the reconstructed image data.
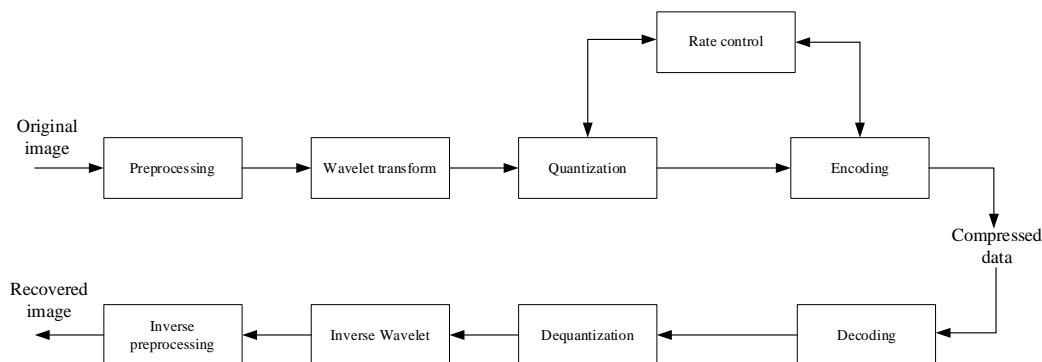
Figure 9. The processing of JPEG200 encoding and decoding.

The general steps of the JPEG2000 coding process are as follows:

(1). The original image is decomposed into each color component. As there is a certain correlation between components, the redundancy between data can be reduced and the compression efficiency can be improved by decomposing the related components.

(2). The image and each component image are decomposed into rectangular image slices, which are the basic processing units of the original image and the reconstructed image.

(3). Wavelet transform is applied to each image slice to generate multi-level coefficient images, which can be reconstructed into images with different resolutions.

(4). The decomposed wavelet coefficients are quantized to form the code block of the rectangular array.

(5). Entropy coding is carried out for the coefficient bit plane in the code block.

(6). The region of interest can be encoded at a higher quality compared with the image background region.

(7). Fault tolerance is increased by adding corresponding identifiers to the bit stream.

(8). A header structure is added to the first section of each stream to describe the attributes of the source image. According to the header structure, the positioning operation, extraction operation, decoding operation and image reconstruction can be completed, so that it has the characteristics of reproduction accuracy and an operation-of-interest region, and can use an optional file format to describe the image and its various components.

The decoding algorithm for JPEG2000 is as follows. First, read the JPEG2000 image format file header information and file the header logo from the JPEG2000 code stream in order to extract the decoding parameters, such as image size, image block number, wavelet transform mode, and other information, and then unpack to obtain the code word. In the decoding data, each bit plane contains three processes: the cleaning process, the amplitude refining process, and the importance propagation process, which are combined with inverse arithmetic encoding to decode together. Each process will output the decoded bit stream, and carry out inverse quantization, inverse discrete wavelet transforms, and inverse multicomponent transform according to the sequence of decoding in each process, and finally output the decoded data. The decoder can extract the required part from the compressed code stream for decoding, such as extracting only a certain component or parsing only to a certain quality layer. The JPEG2000 standard gives the decoder a lot of flexibility because of its multiple decoding characteristics [20].

JPE2000 compression has the following features.

(1). High compressibility

JPEG2000 can achieve relatively higher compression multiples than other methods while ensuring higher fidelity of the reconstructed images.

(2). Compatible with lossy compression and lossless compression

It is compatible with lossy compression and lossless compression implementations. This can avoid format problems caused by inconsistent lossy and lossless compression algorithms and facilitate processing and conversion. Particularly in the field of remote sensing information, lossless compression and lossy compression are often used for comprehensive processing, so as to make operational processing more convenient.

(3). Measurability

Measurability refers to gradual transmission by image quality, resolution, component, or spatial locality. Specifically, it can realize the transmission of the overall outline, and then gradually insert pixels into it to improve the image resolution. This makes it easy to quickly scan the image without considering the details of the image.

(4). Support area-of-interest coding

We can freely choose the region of interest, arbitrarily access and process this region, and realize the priority display of this region. It is possible to compress the selected area of interest and other areas in different proportions to achieve a low pressure ratio and even lossless compression in some areas to ensure the appropriate roles of image details and other information.

(5). Good error code robustness

Robust coding technique refers to preserving some redundant information in the process of encoding the original image information. This redundant information is used to resist errors in transmission [19].

5. Simulation Results

5.1. Detection Performance Based on the Original Image

In this section, we used the proposed Yolov5m network to test the detection performance on real satellite images. The detection accuracies of the small objects on the satellite images were different under different settings. First, we used the DOTA1.0 data set as the training set to obtain a good target detection network [21], and we used the original Yolov5s as the detection algorithm. Then, we used this model to detect the images taken by satellite. The recognition results were not very good as the original images were of poor quality without any preprocessing. The images in Figures 10 and 11 are the detection results of real satellite images.

Few planes and ships could be detected when we used the model trained by Yolov5m; the model needed to be revised to better detect small targets. Therefore, we used our new model trained by the modified Yolov5, along with the image segmentation detection procedure. The outputs are shown in Figures 12 and 13; the detection results were much better compared with the previous outputs, with both detection accuracy and recall rate of small targets significantly improved.

The detection performance of the proposed model on the DOTA set was also evaluated, with some results shown in Figures 14 and 15. As the image quality in the DOTA set was good, the detection performance remained excellent.
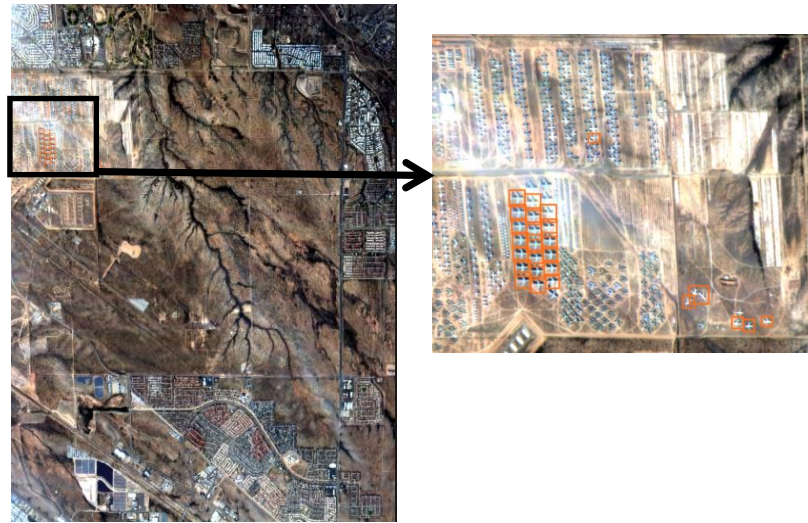
Figure 10. The detection performance for aircraft with the original Yolov5 in the original satellite image.
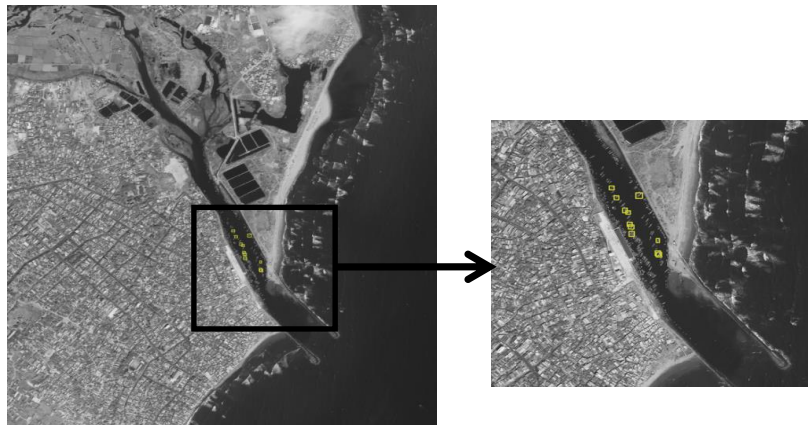

Figure 11. The detection performance for ships with the original Yolov5 in the original satellite image.
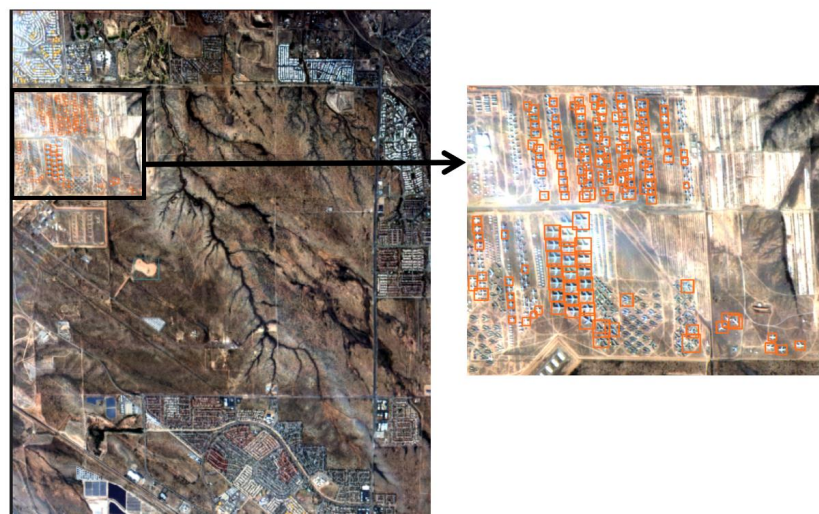

Figure 12. The detection performance for aircraft with modified Yolov5 in the original satellite image.
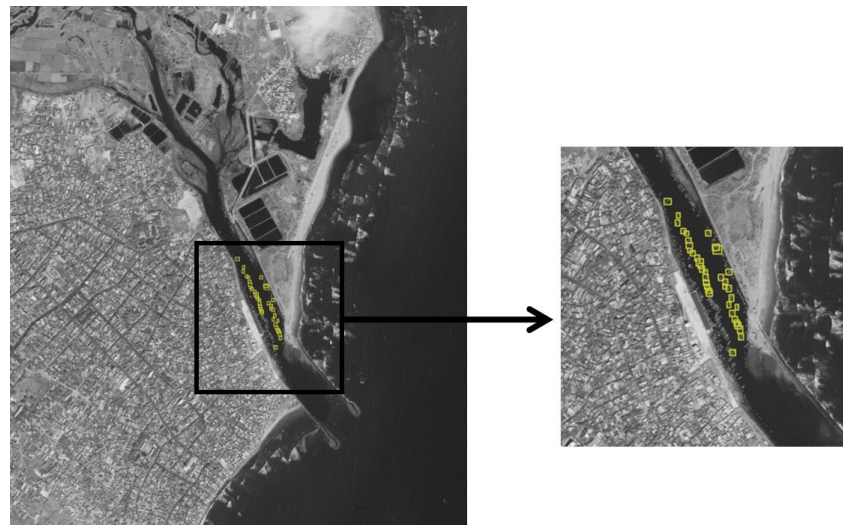
Figure 13. The detection performance for ships with modified Yolov5 in the original satellite image.

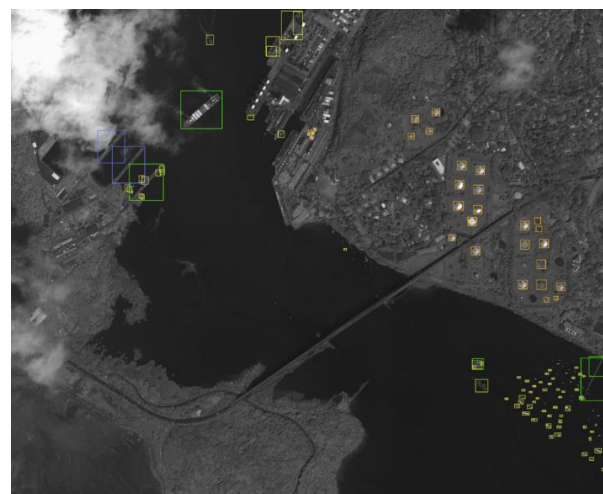**Figure 14.** The detection performance for aircraft in the DOTA set.

Figure 15. The detection performance of scene of river in the DOTA set, where yellow boxes represent detected ships, green boxes represent detected harbors, orange boxes represent detected storage tanks and blue boxes represent detected bridges.

5.2. Compression Performance Analysis

An image of the Shanghai area was used as the analysis set. In the compression, the small image size was 1536×256 pixels, which was the smallest input unit for the ADV212 in our design. There were 9744 small images in total in the analysis set, and except for three pictures whose compression ratios were lower than 4:1, the rest were consistent with 4:1. When the compression ratio was 10:1, there were 4558 images with a fixed compression ratio of 9.931:1 and the remaining images were the same, with ratios of 4:1 and 2:1. The real average compression ratio was 11.46:1 when the ADV212 ratio was settled as 2:1 and 4:1 and the real average compression ratio was 12.64:1 when the ADV212 ratio was settled as 10:1.

When the compression ratio of ADV212 was settled as 2:1, the maximum real compression ratio was 100.17:1. There were 17 images whose compression ratio was larger than 50:1 and 37 images whose compression ratio was larger than 40:1. The image with the maximum compression ratio was the sea surface at the estuary of the Yangtze River, and the actual compression ratio was 100:1. Most areas in the image were smooth with low contrast as shown in Figure 16.
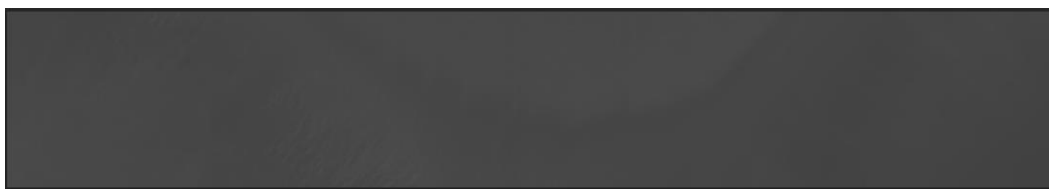

Figure 16. The image with the maximum compression ratio (smooth Yangtze River).

The image with the minimum compression ratio was the sea surface at the estuary of the Yangtze River, and the actual compression ratio was 3.59:1. Most areas of the image were rough with high contrast as shown in Figure 17.
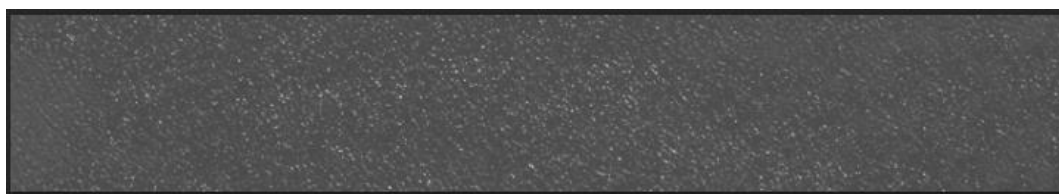

Figure 17. The image with the minimum compression ratio (dynamic Yangtze River).

Table 3 shows the main indices of image compression, and we chose the buildings image in Figure 18 as an example to test the JPEG2000 performance with different compression rates. Note that the compression rate in ADV212 was not the final compression rate; ADV212 merely guaranteed that the SINR of the compressed image was larger than a particular threshold. This means that the compression rate set in ADV212 was the lower bound of the real compression rate. The least significant bit in the image was 12 bits. From Table 4, we can see that PSNR was larger than 50 dB when the compression ratios were 2:1 and 4:1, which means the compressed image was nearly the same as the original image. When the compression ratio was 10:1, the PSNR was reduced to 46.7 dB as the size was reduced. In addition, SSIM and MUSSIM, which are used to value the similarity of different images, were both larger than 0.99 with different ratios. Thus, ADV212 can deal with spaceborne image compression.

Table 3. The main indices of the image compression.

Abbreviation	Full Name
MSE	Mean Square Error
RMSE	Root Mean Square Error
PSNR	Peak Signal-to-Noise Ratio
SSIM	Structural Similarity
UQI	Universal Quality Index
MSSSIM	Multi-scale Structural Similarity
SCC	Spatial Correlation Coefficient
RASE	Relative Average Spectral Error
VIFP	Visual Information Fidelity

Figure 18. The original image of buildings.

Table 4. The compression indices' comparison with different compression rates.

Compression Rate Set in ADV212	2:1	4:1	10:1
Size after compression	82 KB	82 KB	59 KB
MSE	38,508.815755208336	38,508.815755208336	91,708.72526041667
RMSE	196.2366320420536	196.2366320420536	302.8344849260346
PSNR	50.473864443709296	50.473864443709296	46.70535950680697
SSIM	0.9966228304307309	0.9966228304307309	0.9934589782137567
UQI	0.9999321996609332	0.9999321996609332	0.9998540667907824
MSSSIM	(0.9993673349222898 + 0j)	(0.9993673349222898 + 0j)	(0.9986344999440212 + 0j)
ERGAS	291.1348285608546	291.1348285608546	426.4397995898409
SCC	0.7612884162903569	0.7612884162903569	0.6717694464656159
RASE	72.78370714021365	72.78370714021365	106.60994989746023
VIFP	0.3912624911023905	0.3912624911023905	0.3430493507692442

5.3. Effect of Compression on Target Detection

Figures 19 and 20 show the target detection results based on the original image and the compressed image, respectively. The original image shows the Shanghai Pudong International Airport. We can see that most of the aircraft in the original image can be detected correctly. The detection results with a compressed image with a compression ratio of 12 were nearly the same as those for the original image, except for few false detections and missed detections. We marked the false detections and missing detections in Figure 20. Therefore, it was better to compress the original image before target detection, as this could reduce the detection complexity and achieve a similar detection performance.
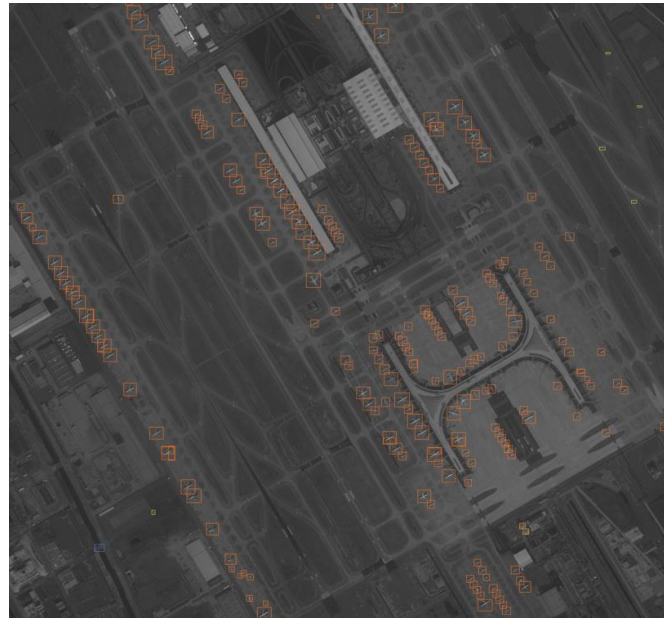
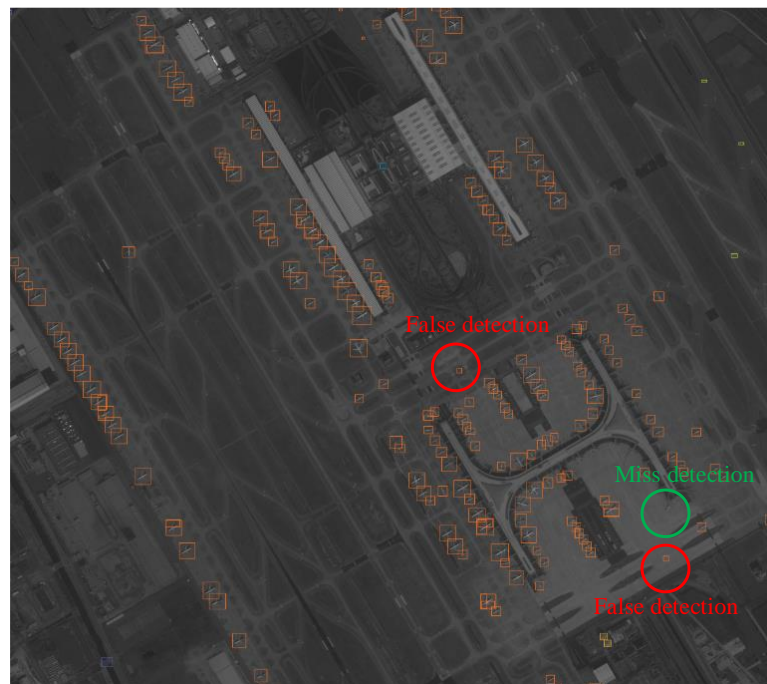**Figure 19.** Detection results based on the original image.

Figure 20. Detection results based on the compressed image.

6. Conclusions

In this paper, we proposed a spaceborne target detection and compression storage system that can detect intended targets and send data to the ground. It can improve transmission efficiency, especially for high-resolution satellite image transmission systems. We proposed an FGPA and ASIC method to realize image compression and used the Jetson Xavier NX to realize target detection. We modified the Yolov5 algorithm to improve detection performance for small targets. In order to restore the real application scenario, raw satellite images were used for tests, instead of an open-source dataset that has been pre-processed at high quality. The experimental results showed that Yolov5 performs well in real satellite image detection with appropriate revisions and is feasible for implementation

on a satellite with few hardware additions. In addition, image compression has little effect on target detection, enabling the detection of targets on compressed images, which can reduce the data rate requirement of the interface and increase efficiency. In the future, we will try to use the neural network to detect a large target on a satellite image with low-resolution, such as an airport.

References

1. Wei, J.; Mi, L.; Hu, Y.; Ling, J.; Li, Y.; Chen, Z. Effects of Lossy Compression on Remote Sensing Image Classification Based on Convolutional Sparse Coding. *IEEE Geosci. Remote Sens. Lett.* **2022**, *19*, 1–5. [CrossRef]
2. Chen, Z.; Hu, Y.; Zhang, Y. Effects of Compression on Remote Sensing Image Classification Based on Fractal Analysis. *IEEE Trans. Geosci. Remote Sens.* **2019**, *57*, 4577–4590. [CrossRef]
3. Li, J.; Fu, Y.; Li, G.; Liu, Z. Remote Sensing Image Compression in Visible/Near-Infrared Range Using Heterogeneous Compressive Sensing. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2018**, *11*, 4932–4938. [CrossRef]
4. Chong, Y.; Zhai, L.; Pan, S. High-Order Markov Random Field as Attention Network for High-Resolution Remote-Sensing Image Compression. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 1–14. [CrossRef]
5. Öztürk, E.; Mesut, A. Performance Evaluation of JPEG Standards, WebP and PNG in Terms of Compression Ratio and Time for Lossless Encoding. In Proceedings of the 2021 6th International Conference on Computer Science and Engineering (UBMK), Ankara, Turkey, 15–17 September 2021; pp. 15–20.
6. Naveen Kumar, S.; Vamshi Bharadwaj, M.V.; Subbarayappa, S. Performance Comparison of Jpeg, Jpeg XT, Jpeg LS, Jpeg 2000, Jpeg XR, HEVC, EVC and VVC for Images. In Proceedings of the 2021 6th International Conference for Convergence in Technology (I2CT), Maharashtra, India, 2–4 April 2021; pp. 1–8.
7. Jiang, J.; Xu, H.; Zhang, S.; Fang, Y.; Kang, L. FSNet: A Target Detection Algorithm Based on a Fusion Shared Network. *IEEE Access* **2019**, *7*, 169417–169425. [CrossRef]
8. Wang, L.; Li, B.; Fu, R. Feature Fusion and Coordinate Attention for Small Target Detection. In Proceedings of the 2022 7th International Conference on Intelligent Computing and Signal Processing (ICSP), Xi'an, China, 15–17 April 2022; pp. 629–633.
9. Wang, H.; Yu, Y.; Cai, Y.; Chen, X.; Chen, L.; Li, Y. Soft-Weighted-Average Ensemble Vehicle Detection Method Based on Single-Stage and Two-Stage Deep Learning Models. *IEEE Trans. Intell. Veh.* **2021**, *6*, 100–109. [CrossRef]
10. Yang, W.; BO, D.; Tong, L.S. TS-YOLO: An efficient YOLO Network for Multi-scale Object Detection. In Proceedings of the 2022 IEEE 6th Information Technology and Mechatronics Engineering Conference (ITOEC), Chongqing, China, 4–6 March 2022; pp. 656–660.
11. Wang, Y.; Niu, P.; Guo, X.; Yang, G.; Chen, J. Single Shot Multibox Detector with Deconvolutional Region Magnification Procedure. *IEEE Access* **2021**, *9*, 47767–47776. [CrossRef]
12. Liao, M.; Shi, B.; Bai, X. TextBoxes++: A Single-Shot Oriented Scene Text Detector. *IEEE Trans. Image Process.* **2018**, *27*, 3676–3690. [CrossRef] [PubMed]
13. Liao, M.; Zhu, Z.; Shi, B. Rotation-sensitive regression for oriented scene text detection. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 5909–5918.
14. Liu, B.; Zhao, W.; Sun, Q. Study of object detection based on Faster R-CNN. In Proceedings of the 2017 Chinese Automation Congress (CAC), Jinan, China, 20–22 October 2017; pp. 6233–6236.
15. Ma, J.; Shao, W.; Ye, H.; Wang, L.; Wang, H.; Zheng, Y.; Xue, X. Arbitrary-Oriented Scene Text Detection via Rotation Proposals. *IEEE Trans. Multimed.* **2018**, *20*, 3111–3122. [CrossRef]
16. Zhang, Z.; Guo, W.; Zhu, S.; Yu, W. Toward Arbitrary-Oriented Ship Detection with Rotated Region Proposal and Discrimination Networks. *IEEE Geosci. Remote Sens. Lett.* **2018**, *15*, 1745–1749. [CrossRef]
17. Ding, J.; Xue, N.; Long, Y.; Xia, G.-S.; Lu, Q. Learning RoI Transformer for Oriented Object Detection in Aerial Images. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 2844–2853.
18. Jablonski, B.; Makowski, D.; Perek, P. Evaluation of NVIDIA Xavier NX Platform for Real-Time Image Processing for Fusion Diagnostics. In Proceedings of the 2021 28th International Conference on Mixed Design of Integrated Circuits and System, Lodz, Poland, 24–26 June 2021; pp. 63–68.

19. Wu, B.F.; Lin, C.-F. A high-performance and memory-efficient pipeline architecture for the 5/3 and 9/7 discrete wavelet transform of JPEG2000 codec. *IEEE Trans. Circuits Syst. Video Technol.* **2005**, *15*, 1615–1628.
20. Zhe, C.; Zongtao, D.; Zhaoyun, S.; Dongmei, W. ADV212-based High Resolution Still Image Compression System Design. In Proceedings of the 2010 Second International Conference on Information Technology and Computer Science, Kiev, Ukraine, 24–25 June 2010; pp. 170–173.
21. DOTA. Available online: https://captain-whu.github.io/DOTA/index.html (accessed on 20 February 2023).