# A Benchmark Sentinel-1 SAR Dataset for Airport Detection

Daochang Wang, *Student Member, IEEE*, Fan Zhang, *Senior Member, IEEE*, Fei Ma, *Member, IEEE*, Wei Hu, Yu Tang, and Yongsheng Zhou, *Member, IEEE*

***Abstract*—Airports are important targets in both military and civilian tasks. Synthetic aperture radar (SAR)-based airport detection has received increasing attention in recent years. However, due to the high cost of the SAR imaging and annotation process, there is no publicly available SAR dataset for airport detection, which leads to the fact that deep learning methods have not been applied to airport detection and no unified evaluation benchmark for traditional methods. To provide a benchmark for airport detection studies in SAR images, a large SAR airport dataset is presented in this article. It contains 624 SAR images from Sentinel 1B, covering 104 instances of airports with different scales, orientations, and shapes, which can realistically reflect the real world. Experiments with four deep learning-based methods and three traditional methods on this dataset demonstrate its effectiveness and challenge. It serves for the development of state-of-the-art airport detection algorithms or other related tasks. In addition, we found that airport runways in SAR images always have some parallel line segments. Inspired by this, the fusion of deep features and line segments is achieved by designing a line segment detector branch, which further improves the accuracy of airport detection.**

***Index Terms*—Airport detection, deep learning, linear segment, new airport dataset.**

## I. Introduction

AIRPORTS are highly valuable civilian and military facilities, where aircraft take off, land, and park. The detection of airports plays an important role in many practical applications, such as airport navigation, aerial reconnaissance, etc. In addition, accurate airport detection is also helpful for aircraft detection as aircraft are usually parked at the airport.

Compared with optical remote sensing, synthetic aperture radar (SAR) has the characteristics of all-weather and all-day imaging on the Earth's surface. Currently, airport detection in SAR images is already receiving extensive concern.

There are four major types of methods for airport detection in remote sensing images, including line-based, saliency-based, image segmentation-based, and deep learning-based. The difference in backscatter intensity between the airport runway and the surrounding environment results in significant line segments around the airport runway. The line-based methods utilize these line segments to detect the airport. Line segments are mainly detected by Hough transform [1], Radon transform, or line segment detector (LSD) [2]. After obtaining the line segments, airport detection is achieved by judging the length of the line segments [3]. There are also some improved methods. They obtain the airport support regions by line segments and then select the proper regions as the airports based on some prior information, such as size [4] and aspect ratio [5]. Saliency-based methods aim at predicting a saliency map, where the airport regions will have high saliency values. The saliency maps are commonly calculated based on line segments [6]–[9], frequency domain features [10], and superpixelwise geometric information [11]–[13]. Both methods rely on accurately identifying the line segments of the runway, however, identifying the complete line segments in the SAR image is a difficult task. Furthermore, there are several line segments in the SAR image, and identifying runways from these line segments also makes airport detection more difficult. The image segmentation-based approaches identify the airports according to the texture features [14]–[17], structural features [8], and intensity information [18]. Nevertheless, the complex feature extraction process and pixel-level analysis of these methods lead to high computational costs and poor practicability.

In recent years, with the development of deep learning theories [19], [20], deep learning-based methods have attracted a lot of attention in the field of remote sensing image processing. Wu et al. [21] proposed a Fourier-based rotation-invariant feature boosting (FRIFB) to achieve geographic object detection in optical images. Based on FRIFB, Wu et al. [22] proposed a spatial-frequency channel feature and constructed the optical remote sensing imagery detector (ORSIm detector). Graph convolutional network has also been applied to remote sensing image processing due to its powerful representation and analysis of irregular data [23]. Deep learning-based approaches have also achieved some success in the field of airport detection. Deep learning-based object detection models are utilized as a framework. Then some optimizations are performed on this basis, such as changing the anchor scale [24] or employing image segmentation modules [25] to achieve more accurate localization. Object detection models can be divided into two main categories: 1) two-stage and 2) single-stage. The two-stage detection models mainly consist of RCNN [26], fast RCNN [27], faster RCNN [28], and mask RCNN [29]. Nowadays, all research

Manuscript received 19 June 2022; revised 8 July 2022; accepted 12 July 2022. Date of publication 18 July 2022; date of current version 24 August 2022. This work was supported in part by the National Natural Science Foundation of China under Grant 61871413, Grant 62171016, and in part by the Fundamental Research Funds for the Central Universities under Grant XK2020-03. *(Corresponding author: Fei Ma.)*

The authors are with the College of Information Science and Technology, Beijing University of Chemical Technology, Beijing 100029, China (e-mail: 664479659@qq.com; zhangf@mail.buct.edu.cn; mafei@mail.buct.edu.cn; huwei@mail.buct.edu.cn; abctangyu@sina.com; zhyosh@mail.buct.edu.cn).

Digital Object Identifier 10.1109/JSTARS.2022.3192063

TABLE I
COMPARISON BETWEEN SAD AND OTHER OBJECT DETECTION DATASETS

| Datasets | Source | Number of Images | Image width | Instance | Public | Resolution |
|---|---|---|---|---|---|---|
| Literature [24], [25] | Gaofen-3 | 2479 | 500×500 | Airport | No | 1m |
| Literature [30] | Google Earth | 800 | 2048×2048 | Airport | No | 5m |
| Literature [31] | Google Earth | 343 | 885×1613 | Airport | No | 8m-16m |
| Literature [32] | Google Earth | 50 | 2048×2048 | Airport Aircraft | No | 0.23m |
| Literature [36] | Google Earth | 170 | 3000×3000 | Airport | No | 8m |
| Literature [37] | Google Earth | 600 | 600×600 | Airport | No | 19m-76m |
| Literature [39] | Gaofen-3 | 244 | 500×600 | Airport | No | 8m-60m |
| OpenSARUrban | Sentinel-1 | 33358 | 100×100 | Urban | Yes | 10m |
| SSDD | Sentinel-1 Radarsat-2 TerraSAR-X | 1160 | 300×300 600×600 | Ship | Yes | 1m-15m |
| GFB | Gaofen-3 | 4812 | 512×512 | Building | Yes | 1m |
| SAD | Sentinel-1 | 624 | 2048×2048 | Airport | Yes | 10m |

on airport detection is conducted based on two-stage detection models. For example, Yin et al. [30] used faster RCNN with the addition of hard sample mining to solve the problem of sample imbalance due to insufficient datasets. Chen et al. [31] also took faster RCNN as the basic detection network and modified the size of the convolution kernel and the anchor size to achieve more accurate airport localization. Zeng et al. [32] added a separate CNN after faster RCNN to further remove the false alarms. The main single-stage detection models include YOLO v3 [33], YOLO v4 [34], and YOLO v5. Up to now, few methods use single-stage detection models to detect airports. The reason is that single-stage object detection models localize and classify the object only once, which makes it difficult to achieve satisfactory detection results with small training datasets.

The deep learning-based methods have powerful feature extraction and feature representation capabilities. However, deep learning is a data-driven concept, and the performance of respective deep learning-based approaches strongly depends on the quality and quantity of given data [35]. A challenging and excellent dataset can accelerate the development of the field. In case of insufficient data, some methods use natural images for pretraining [36]–[38]. Other methods address the sample imbalance of data by hard sample mining [39]. However, the difference between remote images and natural images makes transfer learning unsatisfactory. And applying hard sample mining on a small dataset also does not help the model to fit adequately.

All of the above methods are applied to optical remote sensing images, but there are some interference factors in optical images, such as clouds, fog, light, etc., which will increase the difficulty of image preprocessing [40]. And the above airport datasets are not publicly available. In a contrast, SAR can achieve all-day and all-weather imaging of the Earth's surface. Therefore, an excellent SAR airport dataset can facilitate the development of airport detection.

With the rapid increase in the number of SAR images, many datasets have emerged in the SAR field, such as OpenSARUrban [41] for urban area interpretation, SSDD [42] for ship detection and GFB [43] for building semantic segmentation, which have played an important role in their respective interpretation tasks. However, there is no publicly available dataset for SAR airports. Intending to fill this gap and advance airport detection research in SAR images, in this article, we present a large-scale SAR airport dataset (SAD), which has been collected from Sentinel-1B images, and annotated by experts in aerial image interpretation. All the mentioned datasets are summarized in Table I.

Unlike natural images and optical images, SAR actively emits electromagnetic waves to the Earth's surface and processes the backscattered electromagnetic signal returned by the irradiated object to form an image, the intensity of the backscattered electromagnetic signal is determined by the material of the object itself. According to the roughness of the target surface, the backscattering can be classified as single scattering, double scattering, and volume scattering. The runway as a microrough surface, single scattering is the main scattering mechanism, the terminal building, oil tank, and other buildings in the airport will behave as double scattering, and the nearby forest will form volume scattering. The reason for the inaccurate detection of airports in SAR images is that the scattering of targets, such as lakes, roads, and shadows is similar to that of runways, causing errors in airport detection, which cause false detection of airport detection, as illustrated in Fig. 1.

In order to solve the problems that arise in airport detection, we propose an SAR airport detection method based on the fusion of line segment detector and deep network (FLDNet), which we hope will facilitate the development of future airport detection algorithms.

The main contributions of this article are summarized as follows.

1) A large-scale public dataset has been proposed for airport detection in SAR images. To the best of our knowledge, SAD is the first publicly available SAR airport dataset. It provides a benchmark resource for developing state-of-the-art airport detection algorithms or other relevant tasks.
2) We benchmarked state-of-the-art deep learning-based object detection algorithms and traditional airport detection algorithms on SAD, which can serve as a baseline for the development of future airport detection algorithms.

(a) (b) (c)

Fig. 1. Demonstration of airport and airport analogues in SAR images (red rectangle: airport area, yellow: airport analogues). (a) River areas. (b) Road areas. (c) Strip lakes.

3) To solve the problem of airport detection, FLDNet is proposed, and its effectiveness has been experimentally proven.

The rest of this article is organized as follows. Section II describes the details of the SAD. In Section III, a brief description of seven algorithms is given, which will be used for benchmark validation. The experimental results and analysis are reported in Section IV. An introduction to the structure and experiments of FLDNet is presented in Section V. Finally, Section VI concludes this article.

## II. SAD: SAR Airport Dateset

In this section, the SAD collection and annotation process is first described. Then we illustrate the challenges posed by the SAD for SAR image airport detection in terms of the size, aspect ratio, orientation, numbers of airport runways, and complex background of the airport targets.

In order to promote airport detection in the SAR field, we will publicly provide the original images with ground truth for the training set and testing set. Meanwhile, to facilitate the expansion and visualization of SAD, we converted all images in the dataset into JPG format and the annotation files into TXT format, and visualize the annotation results of all samples.

### A. Images Collection of SAD

Images in SAD are collected from Sentinel 1B. Sentinel 1B is equipped with a C-band SAR sensor. It can provide measurement data for land, forest, ocean, and glacier monitoring and mapping. In this study, we use Level-1 interferometric wide swath GRD products. The spatial resolution is $10 \times 10$ m and the incidence angle is about $20°$–$45°$. The polarization modes are VV and VH. And the number of looks is 5.

To increase the diversity of data, we collect images from multiple cities in multiple countries, which are carefully chosen by experts in aerial image interpretation. There are 624 airfield images of $2048 \times 2048$ pixels in SAD, which cover 104 airfield targets. The imaging time and polarization modes are varied for different images of the same airport. We mark the geographical location of some airports in Fig. 2. The airports in SAD cover a wide range of countries in Asia, Europe, Africa, North America, and South America, which makes the types of airports more diverse.

Fig. 2. Airport distribution in SAD.

### B. Annotation Method

Considering that the airport targets are sparsely distributed, we use horizontal bounding boxes for annotation. The usual representation of horizontal bounding boxes is $(c, x_c, y_c, h, w)$, where $c$ marks the category and $(x_c, y_c)$ denotes the positions of the bounding boxes' center in the image. $w$ and $h$ are the width and height of the bounding box, respectively. In the process of annotation, we refer to the location of the airport given in Google Earth. Some annotated images in SAD are shown in Fig. 3.

### C. Dataset Splits

SAD is divided into the training set with the amount of 420 images, and the test set with 204 images. It is worth mentioning

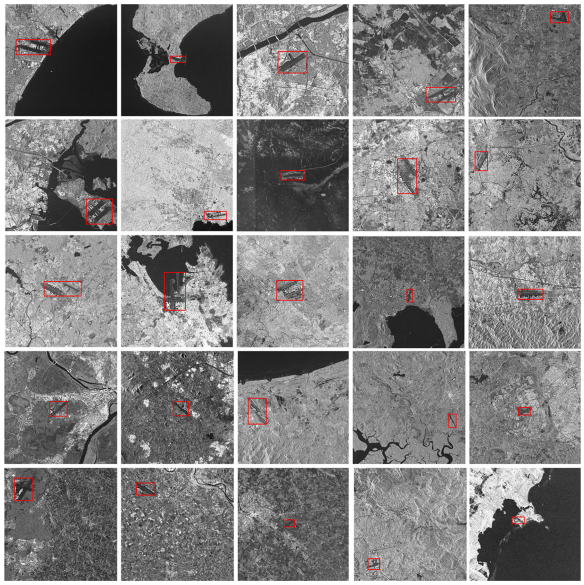


Fig. 3. Samples of annotated images in SAD.

TABLE II
STATISTICS OF AIRPORT AREA SIZE IN SAD

| Pixel Size | Small | Middle | Large |
|---|---|---|---|
| Proportion | 52.4% | 39.7% | 7.9% |

that different images of the same airport are uniformly classified as training or testing sets when we divide the dataset.

### D. Various Pixel Sizes of Airports

Following the convention in [44], we refer to the area of a horizontal bounding box, which we call pixel size for short, as a measurement for airport size. Fig. 4(a) illustrates the distribution of pixel sizes for the airport in our dataset. The pixel size of the smallest airport is 26 244 and the largest is 1 142 500, which occupies 27.239% of the whole image. We divide all instances in the dataset into three groups based on their pixel size: small (20 000 to 100 000), medium (100 000 to 300 000), and large for the range above 300 000. Table II illustrates the percentages of the three instance splits in SAD. It is clear that several small airports exist in SAD, and it is a difficult task to accurately detect all of them at a lower resolution (10 m). And the huge change in the number of pixels of different airports poses a challenge to the detection model.

### E. Various Numbers of Airport Runways

As an essential infrastructure in airports, runways play an important role in distinguishing airports from similar areas of airports. The difference in the number, length, and width of airport runways is also a major challenge for airport detection. Table III shows the statistics on the number of runways in SAD, which shows that most airports have one or two runways, but three or four runways still exist. The runway length ranges from 2400 to 4500 m and the width of the runway ranges from 24 to 80 m. In summary, the number, length, and width of runways in SAD are highly variable, which makes accurate detection of airports more challenging.

TABLE III
STATISTICS ON THE NUMBER OF RUNWAYS IN SAD

| Number of Runways | One | Two | Three | Four |
|---|---|---|---|---|
| Proportion | 42.3% | 43.3% | 4.8% | 9.6% |

TABLE IV
STATISTICS OF AIRPORT ORIENTATIONS IN SAD

| Degree | 0°-45° | 45°-90° | 90°-135° | 135°-180° |
|---|---|---|---|---|
| Proportion | 21.3% | 23.0% | 28.7% | 27.0% |

### F. Various Aspect Ratios of Airports

Aspect ratio is an important factor that affects the detection accuracy of anchor-based object detection models, such as faster RCNN, YOLO v3, etc. The aspect ratio of all targets in our dataset is counted to provide a reference for better model design. Fig. 4(b) illustrates the distribution of aspect ratios for airport targets in our dataset. We can see that airports vary greatly in aspect ratio. The aspect ratios of the airport targets in SAD are distributed between 0.2 and 5.2, with a relatively large cluster at around 0.8. Moreover, there are a large number of targets with a large aspect ratio in our dataset. Three airport targets with large aspect ratio variations are shown in Fig. 5. The wide distribution of aspect ratios poses a challenge for detection models to accurately locate complete airports.

### G. Various Orientations of Airports

As shown in Table IV, SAD achieves a good balance in the airports of different orientations, which requires the detection model to have good detection of targets in each direction. Moreover, our dataset is closer to real scenes, because it is common to see airports in all kinds of orientations in the real world. This article does not make a study of rotating anchor airport detection, but the uniform orientation distribution of SAD can pose a great challenge for rotating target detection.

### H. Complex Background

As described in Section I, one of the biggest difficulties to detect airports is the existence of airport-like regions in SAR images with the same scattering intensity as airports, i.e., lakes, roads, etc. SAD contains airports in several scenes, including seaside, mountains, cities, rivers, etc. Locating the airport accurately from several complex backgrounds is a challenging task.

## III. BENCHMARKING ALGORITHMS FOR SAD

To demonstrate that SAD is effective and challenging in the field of airport detection, and to provide some representative benchmark algorithms for this dataset, we carry out airport detection on our SAD. A comprehensive analysis of popular deep learning-based algorithms and several representative traditional

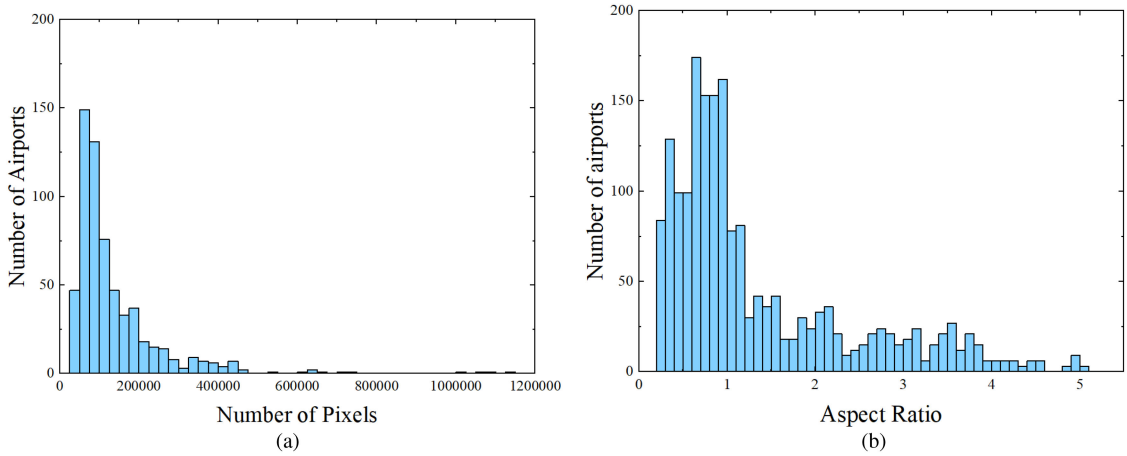



Fig. 4. Histogram statistics of the airport area in SAD. (a) Histogram of the number of pixels. (b) Histogram of aspect ratio.

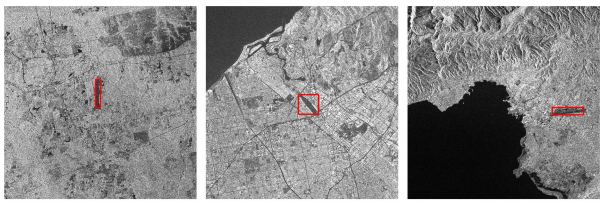



Fig. 5. Various aspect ratio of airports. (a) Aspect ratio = 5.2. (b) Aspect ratio = 0.8. (c) Aspect ratio = 0.2.

airport detection methods for SAR images in recent years has been conducted to achieve this task.

The reason for using deep learning algorithms in this study is the astonishing achievements they have already achieved. We select RetinaNet [45], YOLO v3 [33], YOLO X [46], and YOLO v5 as the deep learning algorithms to evaluate our dataset. We choose four deep learning algorithms, RetinaNet [45], YOLO v3 [33], YOLO X [46], and YOLO v5, to evaluate our dataset. To be specific, YOLO v3, RetinaNet, and YOLO v5 are anchor-based models, YOLO X represents detection models that perform detection in an anchor-free manner.

For the evaluation of traditional methods, three popular SAR image airport detection methods, including CSA-SOACM [18], LSG [7], and MSA [13] are used to evaluate the effectiveness and challenges of SAD.

To be fair, all the above methods use the optimal parameters of their respective papers. A description of each of the seven methods will be presented below.

### A. RetinaNet

RetinaNet, for the first time, matches the state-of-the-art accuracy of the complex two-stage detector, such as R-CNN, faster R-CNN, and mask R-CNN. RetinaNet identifies class imbalance during training as the main obstacle impeding the one-stage detector from achieving state-of-the-art accuracy and proposes a new loss function, i.e., focal loss, which eliminates this barrier. Airport detection tasks suffer from a quantitative imbalance between positive and negative samples, so RetinaNet is considered to be well suited as a benchmark method for SAD. In addition, RetinaNet has been widely used for various tasks after its proposal, including ship detection in SAR images [47], [48], multiscale objects detection [49], etc.

### B. YOLO v3

YOLO series, as the most important detection algorithm in the single-stage detector, can provide excellent detection results. YOLO v3, as the last detection network updated by the authors of the YOLO series, adds many small technologies compared to the previous versions, such as data augmentation, multiscale training, and batch normalization, among others. And DarkNet-53 become the new feature extractor network. In addition, YOLO v3 demonstrated for the first time that logical classifiers are more effective than Softmax. YOLO v3 is widely used for the detection of brick kilns [50] and deep-sea debris [51] in remote sensing images.

### C. YOLO v5

YOLO v5 is put forth by ultrlytics. Compared to previous versions of YOLO and other object detection models. YOLO v5 significantly improves the training speed while maintaining detection accuracy. CSPNet [52] is adopted as the backbone on account of its processing time. In order to deal with object scale changes, PANet [53] is used as a model neck to construct feature pyramids. It consists of YOLO v5s, YOLO v5m, YOLO v5l, and YOLO v5x network structures of different depths and widths. In the field of remote sensing, YOLO v5 is applied to crop circle detection [54] and vessel detection [55].

### D. YOLO X

YOLO X performs object detection in an anchor-free manner, meanwhile, a decoupled head and the leading label assignment strategy SimOTA is employed to accomplish state-of-the-art detection results. It chooses Darknet-53 and a spatial pyramid

pooling (SPP) layer as the backbone, but the anchor-free detection makes the number of predictions per position lower. Recently, YOLO X has been applied to ship detection [56].

### E. CSA-SOACM

CSA-SOACM utilizes vision-oriented saliency, knowledge-oriented saliency, and active contour models (ACM) [57] to achieve airport detection in SAR images. Specifically, superpixel segmentation and mathematical morphological filter are first employed to reduce the computational complexity of the subsequent steps. Then, a two-way complementary saliency analysis (CSA) scheme, which combines vision-oriented saliency and knowledge-oriented saliency, is proposed for the estimation of airport locations. With CSA, the airport support region can be obtained. Finally, the accurate airport area is obtained by relying on the powerful contour recognition ability of ACM. It is worth mentioning that SOACM only performs energy evolution in regions with high saliency, which will not only make the operation results more accurate but also greatly reduce the operation consumption time.

### F. LSG

LSG implements airport detection via line segment grouping and saliency analysis in large-scale SAR images. First, the line segment detection (LSD) algorithm is modified to use a ratio-based approach to replace the differential approach to obtain the gradient calculation results in SAR images, which leads to more accurate line segment detection results. During the line segment grouping step, airport support regions are obtained based on the *a priori* knowledge. The airports have an abundance of line segments that are typically parallel or vertical to each other and occur in clusters. Without complete detection of all line segments in the airport area, resulting in the possibility of dividing an airport into multiple airport support regions, selective nonmaximum suppression is used to merge overlapping airport support regions and nonoverlapping but close to each other airport support regions in the image. Finally, according to the knowledge that the grey value of runways in SAR images is relatively smaller than other regions, false alarm control is achieved by histogram statistics of airport support regions, and the final airport detection results are obtained.

### G. MSA

MSA is a multilayer abstraction saliency model for airport detection in SAR images. SLIC [58] superpixel generation method is first used to obtain superpixels, which are utilized along with the line segment detection results to obtain the airport support region. Superpixels are classified into three categories: 1) seed superpixel (intersecting directly with a line segment), 2) straddle superpixel (intersecting with a seed superpixel), and 3) useless superpixel (others). Taking the seed superpixel as the clustering center, by calculating the histogram intersection distance between different superpixels as the similarity measure, which achieves the aggregation of airport support regions. Next, three saliency cues for airport detection in SAR images are proposed. Local contrast saliency is proposed based on the fact that the gray value of the airport runway in the SAR image is relatively small. The spatially compact distribution of airports induces adobe deformation saliency. Considering the large scale of SAR images and the uniqueness of airports, the global uniqueness saliency is appropriate for airport detection. Finally, three saliency maps are integrated to obtain the final airport detection results.

## IV. Experiments and Analysis

In this section, we show and analyze the experimental results of the seven methods introduced in Section III. The experimental environment is first briefly introduced, and then the evaluation criteria of deep learning methods and traditional airport detection algorithms are unified. On this criterion, we conducted the comparison of the results of the complete SAD and the comparison of the results under low-shot, respectively, from which the results are analyzed.

### A. Implementation Details

All deep learning-based methods use Pytorch as the deep learning framework. The remaining three traditional airport detection methods are programmed with MATLAB. All experiments are implemented on Intel Core i5-11400 CPU at 2.60 GHz and 8-GB RAM, and NVIDIA GeForce GTX 1080 with 16-G memory.

Considering the purpose of the experiments is to validate the effectiveness and challenge of SAD, none of the deep learning-based methods are pretrained by SAR or natural images. It can also ensure the deep learning-based methods and traditional methods are compared in a fair way.

### B. Evaluation Criteria

In the field of deep learning-based object detection, valid metrics commonly employed to evaluate method performance include precision (P), recall (R), F1-score, average precision (AP), etc. Precision is the probability of the actual positive samples among all the samples that are predicted to be positive. The recall is the probability that the actual positive samples are predicted to be positive. F1-score can consider both accuracy and recall together, thus reflecting the performance of the algorithm in a balanced way. Average precision is proposed by VOC2010. Specifically, assuming that N anchors are detected, and there are M positive cases, we will get M recall (from 1/M to M/M). Then for each recall R, we can calculate the maximum precision. The final AP value is the average of M precisions. True positive (TP) indicates that the model correctly predicts the positive category. False positive (FP) indicates that the model incorrectly predicts the positive category. True negative (TN) indicates that the model correctly predicts the negative category. False negative (FN) indicates that the model incorrectly predicts the negative category. Precision, recall, and F1-score scores can be calculated by the following formula:

$$\text{Precision} = \frac{TP}{TP + FP} \tag{1}$$

TABLE V
SAD BENCHMARK EXPERIMENTAL RESULTS

| Models | Pixel Size | P | R | AP | F1-score | Train Time(min) | Test Time(ms) |
|---|---|---|---|---|---|---|---|
| RetinaNet | Small | 0.873 | 0.814 | 0.862 | 0.842 | 813.3 | 90.4 |
| | Middle | 0.912 | 0.884 | 0.931 | 0.898 | | |
| | Large | 0.924 | 0.932 | 0.937 | 0.933 | | |
| | All | **0.902** | 0.852 | **0.918** | **0.876** | | |
| YOLO v3 | Small | 0.867 | 0.789 | 0.837 | 0.826 | 489.4 | 28.5 |
| | Middle | 0.921 | 0.893 | 0.914 | 0.907 | | |
| | Large | 0.917 | 0.917 | 0.927 | 0.917 | | |
| | All | 0.897 | 0.858 | 0.903 | 0.874 | | |
| YOLO X | Small | 0.823 | 0.721 | 0.709 | 0.769 | 580.1 | 34.3 |
| | Middle | 0.889 | 0.807 | 0.822 | 0.846 | | |
| | Large | 0.943 | 0.884 | 0.841 | 0.913 | | |
| | All | 0.869 | 0.848 | 0.794 | 0.859 | | |
| YOLO v5 | Small | 0.854 | 0.844 | 0.862 | 0.849 | 536.4 | **8.5** |
| | Middle | 0.869 | 0.912 | 0.936 | 0.889 | | |
| | Large | 0.957 | 0.917 | 0.943 | 0.937 | | |
| | All | 0.865 | **0.865** | 0.907 | 0.873 | | |
| CSA-SOACM | Small | 0.544 | 0.544 | — | 0.544 | **0** | 1079 .0 |
| | Middle | 0.529 | 0.529 | | 0.529 | | |
| | Large | 0.417 | 0.417 | | 0.417 | | |
| | All | 0.529 | 0.529 | | 0.529 | | |
| LSG | Small | 0.217 | 0.578 | — | 0.316 | **0** | 578.5 |
| | Middle | 0.265 | 0.715 | | 0.387 | | |
| | Large | 0.091 | 0.250 | | 0.133 | | |
| | All | 0.235 | 0.627 | | 0.342 | | |
| MSA | Small | 0.073 | 0.801 | — | 0.134 | **0** | 21181.4 |
| | Middle | 0.092 | 0.843 | | 0.166 | | |
| | Large | 0.027 | 0.333 | | 0.049 | | |
| | All | 0.079 | 0.794 | | 0.134 | | |

$$\text{Recall} = \frac{TP}{TP + FN} \tag{2}$$

$$F1 - \text{score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision + Recall}}. \tag{3}$$

The deep learning-based approach uses a box to mark the predicted region. Traditional airport detection methods use a binary map to distinguish the airport from the background. In order to make the evaluation criteria consistent between traditional and deep learning methods. According to the definition in the literature [18], we label the airport detection results with the minimum horizontal bounding rectangle (MHBR).

The IoU between the predicted box or MHBR and the ground truth is calculated to verify the accuracy of the detection results

$$\text{IoU}\,(\text{pred}, gt) = \frac{\bigcap \left(B^{gt}, B^{\text{pred}}\right)}{\bigcup \left(B^{gt}, B^{\text{pred}}\right)} \tag{4}$$

where $B^{\text{pred}}$ and $B^{gt}$ denote the predicted box and ground truth, respectively. We set the IoU threshold for all methods to 0.5, i.e., we consider it a correct detection if IoU(pred, gt) is greater than 0.5.

With the above definition, the P, R, and F1-score of the traditional method can be calculated, and we do not use AP to compare the traditional method and the deep method simultaneously.

### C. Experimental Results for Baselines

To verify the effectiveness of the SAD and to establish a valid baseline for the SAD. The experiments are first conducted on the complete SAD using seven methods, and the results are presented in Table V. From Table V, it can be seen that the deep learning-based detection method is better than the traditional airport detection method, which is because the deep learning-based method can adjust the model parameters through the training set in the training stage. Compared to deep learning-based methods, traditional detection methods use the local contrast, line segments, and other characteristics of the airport for detection, which makes it unnecessary to train and saves a lot of training time. The focal loss helps RetinaNet balance positive and negative samples during training, making the model more fully trained, so its P, AP, and F1-score are the best among all methods. YOLO v5 builds a multilayer feature pyramid by adding PANet on the basic of FPN [59], which helps YOLO v5 to detect more small airports, so it can be seen that it has the highest R. It should be noted that since CSA-SOACM keeps only one detection result on each SAR image, it has the same P and R. In contrast, LSG and MSA do not limit the number of detections per image to detect more airports. And to ensure more valid results, we remove smaller regions that are unlikely to be airports from the detection results.

In addition, we divided the test set according to the description in Section II-D and tested them separately. From the experimental results in Table V, we find that the deep learning-based methods are less effective in detecting small airports because they usually contain only a small number of pixels, which makes it difficult to distinguish them from their surroundings, and the single runway of small airports also leads to a large variation in the aspect ratio of the airport target,

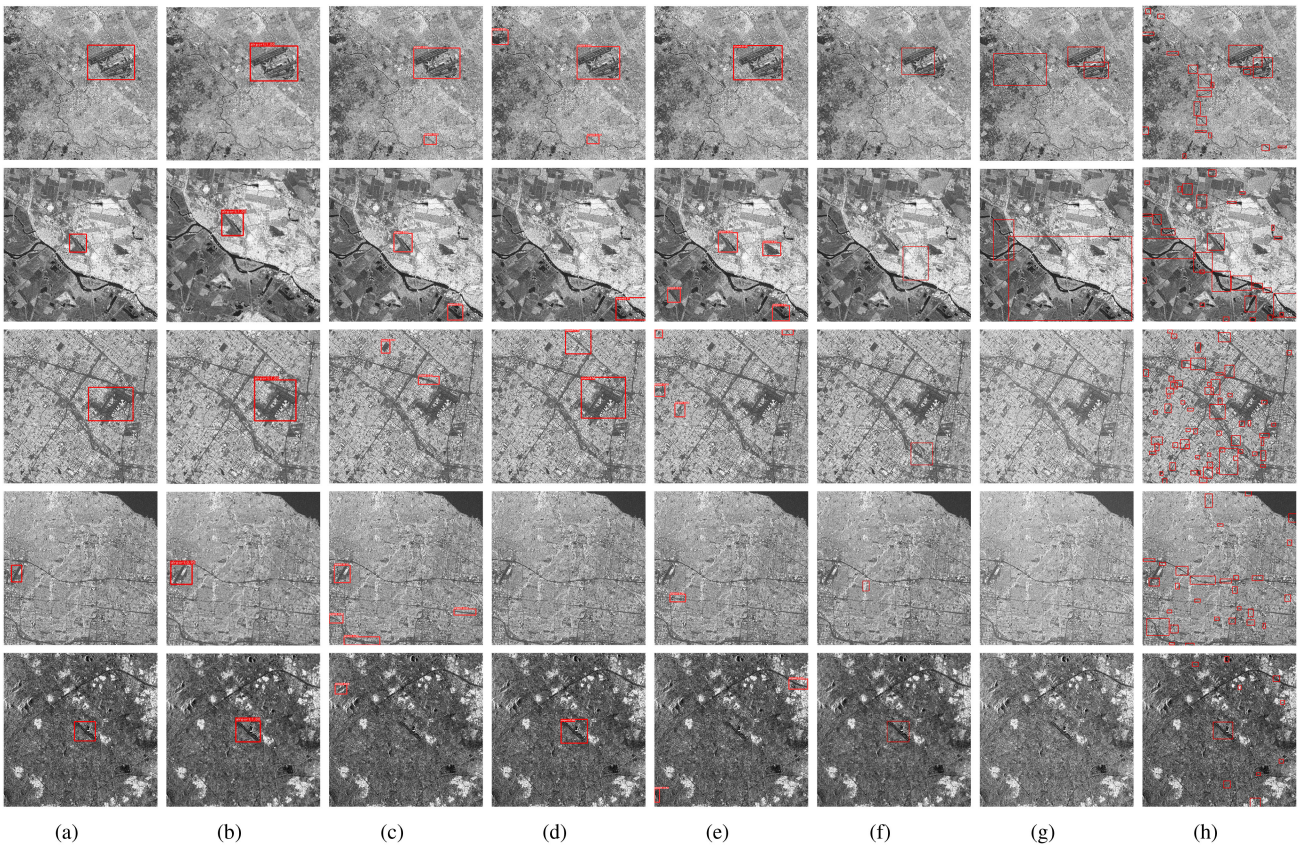



Fig. 6. Experimental results for baselines. (a) Ground Truth. (b) RetinaNet. (c) YOLO v3. (d) YOLO X. (e) YOLO v5. (f) CSA-SOACM. (g) LSG. (h) MSA.

which makes it difficult to locate it accurately. However, they show better detection results for large airports, which occupy a larger area in the image, resulting in less background and easier detection. Traditional airport detection methods, in contrast to deep learning-based methods, are more accurate for small airports. Since they are performing airport detection based on line segments and contrast, the lines of small airports are simpler (usually a pair of parallel lines). Nevertheless, they are much less effective in detecting large airports with complex structures and multiple pairs of parallel lines. Overall, both types of methods have good detection results for middle-sized airports, but small and large airports pose a greater challenge for them.

In order to see the detected result more directly, we selected five representative SAR images to show the detection results in Fig. 6. All images contain airport interference regions, such as rivers, roads, etc., which pose a considerable challenge for airport detection. Among the results of the deep learning-based approach, it can be seen that false detections are mainly concentrated in areas, such as rivers or shadows because those areas have low grey values similar to the airport runways. In addition, some airports (fifth row of Fig. 6) are located in the weakly scattered region, which makes the deep learning-based model miss it. CSA-SOACM uses the local contrast and line segment characteristics of airports to select airport support regions, but it misses most of the airports because the local contrast of some airports is not obvious and it retains only the most likely regions of airports in each SAR image. Compared with CSA-SOACM, LSG acquires airport support regions using only line segments, so more airport regions are lost. Also LSG uses line segments for region growth, which makes it appear to have large airport detection results. Based on LSG, MSA utilizes the histogram intersection distance between regions to merge airport support regions, which effectively reduces the appearance of large regions. However, regions, such as urban, rivers, and shadows also have high contrast, and the histogram intersection distance of the same type of scattered regions is low, which makes MSA have a lot of false detections in urban and river regions.

The boxes of all seven methods appear to fail to fit the target accurately, which is highly related to the varying aspect ratio, size, and orientations of the airport in SAD. For the deep learning-based approach, the large aspect ratio variation makes it difficult to fit all airports closely by setting the anchor in advance. YOLO v5 is the most effective of the four deep learning-based methods, thanks to its adaptive anchor calculation ability. In the traditional algorithm, when the airport is not vertically or horizontally distributed, it will not be marked accurately by the box. This is because all three methods calculate the gradient in vertical and horizontal directions to get the line segments in SAR images, and the accuracy of detection will be reduced if the line segments are not vertical or horizontal. Furthermore, the discontinuity of the line segments also leads to the division of an airport into multiple areas, which is most serious in the MSA.

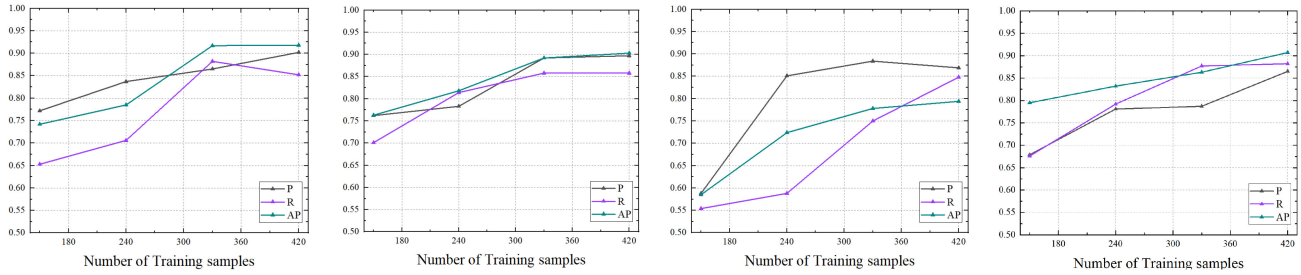



Fig. 7. Variation of metrics of all methods under the different number of training samples. (a) RetinaNet. (b) YOLO v3. (c) YOLO X. (d) YOLO v5.

### D. Low-Shot Experimental Results for Baselines

Deep learning is a data-driven concept, and the performance of respective deep learning-based approaches strongly depends on the quality and quantity of the given data. To test the effectiveness of SAD and whether it can pose a greater challenge for airport detection with reduced training samples, we perform low-shot experiments on SAD using RetinaNet, YOLO v3, YOLO X, and YOLO v5.

We randomly select 150, 240, and 320 training samples from the entire training set and the testing set is not modified. As the number of training samples decreases, the training set cannot completely cover the background and the features of the airports. Therefore, the generalization ability of the model becomes especially important. The results of the four detection models are presented in Fig. 7. The detection performance of all models decreases to different degrees as the training samples are reduced, which indicates that a small number of SAD training samples can pose a greater challenge for airport detection.

The four models maintain a qualified airport detection performance when the number of the training sample is equal to or greater than 330. It is should be noted that RetinaNet shows a drop in R when the training sample changes from 330 to 420, which is due to the use of focal loss. When the model learns sufficiently for simple targets, focal loss makes the model pay more attention to difficult targets, which makes the weight update of the model tend to detect airports in complex scenes. It makes the model miss detection of some simple airports. YOLO v5 still maintains a good detection performance when the training samples reduce to 240, and the other three methods have shown seriously missed detections. YoloX misses 40% of the airports and is unable to complete the task of airport detection properly. The R of all four methods drops below 0.7 when the sample is reduced to 150, and P also shows a considerable decrease. In summary, when the number of training samples is less than 240, all the methods have a large number of missed detections, which is an unacceptable error for the airport detection task.

## V. FLDNet

By reading the literature related to airport detection and experimenting on SAD, we realize that the existing airport detection methods have false detection in lake or river areas. To address these problems, we propose an SAR airport detection method based on the FLDNet, which further improves the detection accuracy. In this section, we introduce FLDNet and analyze the results and parameters.

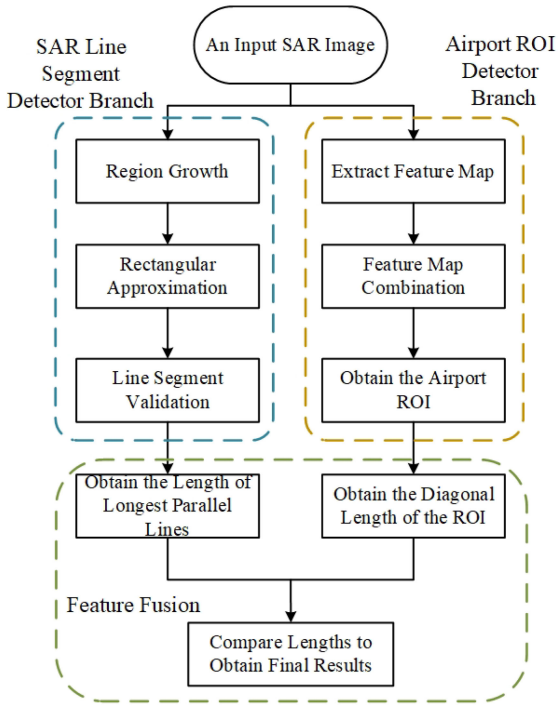



Fig. 8. Framework overview of FLDNet.

### A. Structure of FLDNet

FLDNet consists of two branches, i.e., the airport region of interest (ROI) detector branch and the line segment detector branch, as shown in Fig. 8. Line segment detection is performed on the complete SAR image due to the uncertainty about the area and number of ROIs. After acquiring the ROI, the line segments in the ROI are acquired in the result of line segment detection, achieving the fusion of line segment and deep features. We choose YOLO v5 as the airport ROI detector branch and establish the line segment detector branch by SARLSD [60]. YOLO v5 has been introduced in Section III, and we introduce the SARLSD and fusion process as follows.

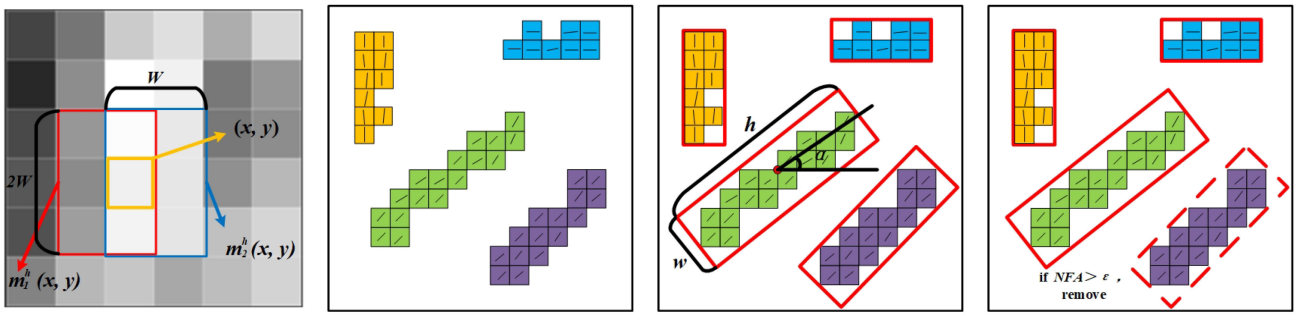



Fig. 9. SARLSD. (a) Horizontal direction gradient calculation for pixel $(x, y)$. (b) Region growth. (c) Rectangular approximation. (d) Line segment validation.

As shown in Fig. 9, SARLSD has four steps: 1) gradient calculation, 2) region growth, 3) rectangular approximation, and 4) line segment validation.

1) *Gradient calculation* is responsible for computing the gradient and the gradient direction of each pixel in the SAR image. The finite difference scheme is a commonly used gradient calculation method, however, applying it to SAR images results in false gradients due to the multiplicative nature of the speckle. To implement gradient calculation in SAR images, SARLSD references [61], a ratio-based calculation is used, which is implemented as follows. For a given pixel at the position $(x, y)$ in image $I$, we first calculate the weighted ratio $R^h(x, y)$ in the horizontal direction. As shown in Fig. 9(a), the red box and blue box with the size of $W \times 2W$ represent the pixels on the left region and right region of $(x, y)$. $m_1^h(x, y)$ and $m_2^h(x, y)$ denote the weighted values of the pixels in the red and blue boxes, respectively. They are defined as

$$m_1^h(x,y) = \sum_{x'=-W}^{W} \sum_{y'=1}^{W} I\left(x+x,' y+y'\right) \times e^{-\frac{|x'|+|y'|}{\beta}} \quad (5)$$

$$m_2^h(x,y) = \sum_{x'=-W}^{W} \sum_{y'=-W}^{-1} I\left(x+x,' y+y'\right) \times e^{-\frac{|x'|+|y'|}{\beta}} \quad (6)$$

where $\beta$ is set to control the dependencies of adjacent pixels. $|.|$ stands for absolute value. $W$ controls the size of the red and green boxes and is defined as

$$W = \lceil \log(10) \times \beta \rceil . \quad (7)$$

Once $m_1^h(x, y)$ and $m_2^h(x, y)$ are obtained, the weighted ratio $R^h(x, y)$ in the horizontal direction of $(x, y)$ is obtained by

$$R^h(x,y) = \frac{m_1^h(x,y)}{m_2^h(x,y)}. \quad (8)$$

The ratio $R^v(x, y)$ in the vertical direction can be calculated in the same way. Then the horizontal gradient component $G^h(x, y)$ and the vertical gradient component $G^v(x, y)$are defined as

$$G^h(x,y) = \log\left(R^h(x,y)\right) \quad (9)$$

$$G^v(x,y) = \log\left(R^v(x,y)\right). \quad (10)$$

The absolute value of the gradient $|GR(x, y)|$ and the gradient direction $\text{ang}(GR(x, y))$ of the pixel $(x, y)$ are defined as

$$|GR(x,y)| = \sqrt{G^h(x,y)^2 + G^v(x,y)^2} \quad (11)$$

$$\text{ang}\left(GR(x,y)\right) = \arctan\frac{G^v(x,y)}{G^h(x,y)}. \quad (12)$$

Repeating the above gradient calculation for each pixel in $I$, the complete gradient map of $I$ can be obtained.

2) *Region growth* is then used to obtain line segment candidates by aggregating pixels whose gradient direction difference is less than the angular tolerance $\tau$. Specifically, each region is grown starting from a pixel that is not assigned to any region, and then the neighboring pixels of this region are tested. The pixels are added to the region if the gradient direction difference between that pixel and the region is less than $\tau$. This process is repeated until no other pixel can be added to the region. It is worth pointing out that the pixels selected as initial regions always have larger gradient values, since they are more likely to belong to line segments. Moreover, the direction $\alpha$ of the region needs to be updated in each region growth iteration

$$\alpha = \arctan\left(\frac{\sum_{i=1}^{n} \sin\left(\text{ang}\left(GR\left(x_i, y_i\right)\right)\right)}{\sum_{i=1}^{n} \cos\left(\text{ang}\left(GR\left(x_i, y_i\right)\right)\right)}\right) \quad (13)$$

here $n$ is the number of pixels in the current region.
The region growth is repeated until all pixels of $I$ are assigned to any region. In Fig. 9(b), different line segment candidates are represented by different colors.

3) *Rectangular approximation* aims at employing the minimum bounding rectangle to approximately represent each candidate region, as shown in Fig. 9(c), to simplify subsequent calculations. The rectangle is described by four parameters: 1) center, 2) angle, 3) length, and 4) width. The center of the rectangle [red dot in Fig. 9(c)] is defined as the center of mass, and the mass of each pixel is the absolute value of its gradient. The angle of the rectangle [i.e., $\alpha$ in Fig. 9(c)] is defined as the angle between the principal axis

of inertia and the horizontal axis. The length and width are indicated in Fig. 9(c) as $w$ and $h$, respectively.

4) *Line segment validation* is to calculate the number of false alarms (NFA) to determine if the line segment candidate region is a true line segment. According to the Helmholtz principle [62], no meaningful structure should happen by chance in a random configuration. Therefore, the line segment can be verified by NFA. It is known that when all pixels of an image satisfies both uniform distribution and independent distribution in local directions, the line segment can be judged by NFA [63]. Let $H_0$ indicate a model meeting the uniform distribution and independent distribution in the local direction.

A key concept in the calculation of NFA is that of the aligned pixel, namely, the pixels in the rectangle whose direction is equal to the rectangle's direction, up to a tolerance $\tau$. We will reduce the width of the rectangle if the density of aligned pixels inside the rectangle is below the density threshold $D$.

For simplicity, let $r$ denote a rectangle in $I$ and the number of pixels in $r$ is $n(r)$, and $k(r)$ denote the number of aligned pixels. And suppose $I_0$ is a random image that obeys $H_0$. A rectangle of size $n(r)$ in $I_0$ contains $k_0(r)$ aligned pixels. Then the NFA can be written as

$$NFA(r) = N_R \times P_{H_0}\left(k_0(r) \geq k(r)\right) \tag{14}$$

where $N_R$ denotes the number of rectangles in $I$ and it can be approximated as $5 \times (M \times N)^{5/2}$ in an image of size $M \times N$ [60]. The $P_{H_0}(k_0(r) \geq k(r))$ can be calculated by

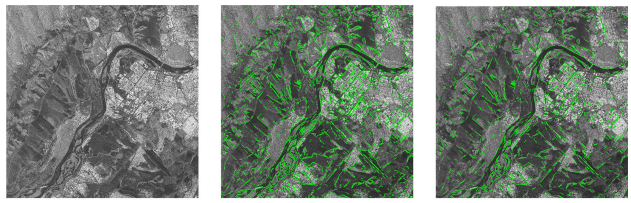
$$\begin{aligned} &P_{H_0}\left(k_0(r) \geq k(r)\right) \\ &= \sum_{x_1 + \cdots x_{n(r)} \geq k(r)} P\left(X_1 = x_1\right) \times \prod_{t=2}^{n(r)} P\left(X_t = x_t | X_{t-1} = x_{t-1}\right) \end{aligned} \tag{15}$$

which can be obtained using a dynamic programming algorithm and decreasing induction method [63]. Let $k = k(r)$, $n = n(r)$, $Y_i = \sum_{i=1}^{n} X_i$, where Xi indicates whether the $i$th pixel is aligned with the rectangle, i.e., $X_i = 1$ if $X_i$ is an aligned pixel and 0 otherwise, we can get

$$\begin{aligned} &P\left(Y_t \geq k\right) \\ &= P\left(Y_{t+1} \geq k | X_t = 0\right) \times P\left(X_t = 0\right) \\ &+ P\left(Y_{t+1} \geq k - 1 | X_t = 1\right) \times P\left(X_t = 1\right). \end{aligned} \tag{16}$$

Moreover, for x $\in \{0, 1\}$ and $k' > 1$

$$\begin{aligned} &P\left(Y_{t+1} \geq k' \mid X_t = x\right) \\ &= \sum_{y \in \{0,1\}} P\left(Y_{t+2} \geq k' - y, X_{t+1} = y \mid X_t = x\right) \\ &= P\left(Y_{t+2} \geq k' \mid X_{t+1} = 0\right) \times P\left(X_{t+1} = 0 \mid X_t = x\right) \\ &+ P(Y_{t+2} \geq k' - 1 \mid X_{t+1} = 1) \times P\left(X_{t+1} = 1 \mid X_t = x\right) \end{aligned} \tag{17}$$



(a) (b) (c)

Fig. 10. Postprocess for line segments. (a) SAR image. (b) Line segment. (c) Retaining line segments after postprocess.

where $P(Y_n \geq k' | X_{n-1} = x)$ can be given by the following equation:

$$P\left(Y_n \geq k' | X_{n-1} = x\right) = \begin{cases} 1 & \text{if } k' = 0 \\ P(1|x) & \text{if } k' = 1 \\ 0 & \text{otherwise} \end{cases}. \tag{18}$$

Calculate the NFA$(r)$ of all $r$ in $I$. Retain the rectangle $r$ if the NFA$(r)$ is less than the NFA threshold parameter $\epsilon$, otherwise the rectangle is removed. The final result after removing the error rectangles is shown in Fig. 9(d). NFA and $D$ are simultaneously applied to all rectangles in $I$, enabling SARLSD to obtain slender rectangles that represent line segments.

NFA and $D$ are simultaneously applied to all rectangles in $I$, enabling SARLSD to obtain slender rectangles that represent line segments.

The above four steps can detect the line segments in SAR images. But there are still too many redundant line segments for subsequent detection result fusion, as shown in Fig. 10(b). Hence, we further remove some too short line segments. According to the prior knowledge about the airport runway lengths (usually greater than 800 m) and SAR image resolution, the least pixel length of the line segments on the airport runway can be calculated. Those line segments shorter than this length will be removed from the line segment set. In Fig. 10(c), the postprocess can greatly decrease the number of line segments.

Airport ROI contains not only the airport area but also some airport-like areas, i.e., lakes, roads, and shadows. We find that the particular characteristic of the airport areas is that runways in SAR images always have long parallel line segments. While the false ROIs usually do not contain parallel lines, or the parallel lines are short in length. Hence, the parallel lines in ROIs are the key to performing the fusion of line segments and ROIs from deep network results.

*1) Parallel Line Segment Searching in ROIs:* The angles of the line segments are the key to determining whether they are parallel. In the origin SARLSD algorithm, the angle information of the line segment is not available.

Therefore, we make an improvement to SARLSD so that it can get the angle information of each line segment. Let $(x_1, y_1)$ and $(x_2, y_2)$ represent the coordinates of the two endpoints of a line segment, its angle $\eta$ can be obtained by

$$\eta = \arctan\left(\frac{y_2 - y_1}{x_2 - x_1}\right). \tag{19}$$

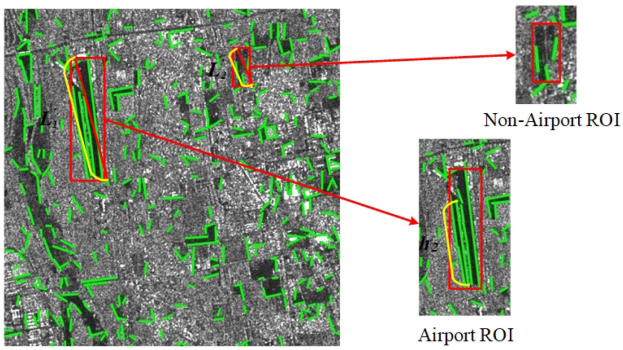



Fig. 11. Process of ROI refining is based on the longest parallel line segments in ROI. In the airport rectangle, the length of the diagonal is close to the length of the longest parallel lines inside it; on the contrary, the length of the parallel lines in the nonairport ROI tends to be shorter.

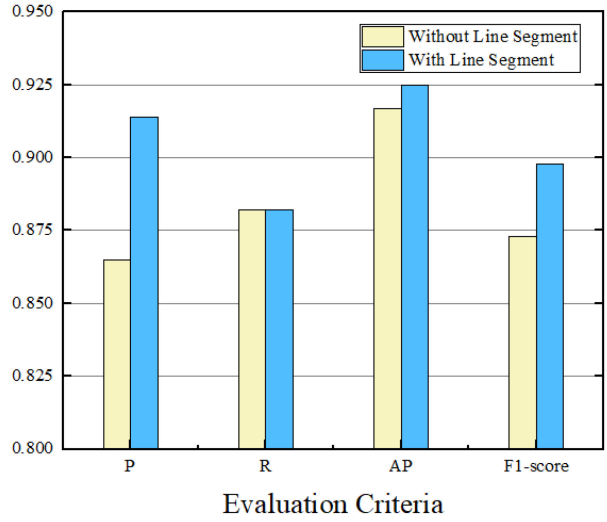



Fig. 12. Comparison of results with and without line segment ($\lambda = 0.1$).

Note that we set $\eta$ to $\pi/2$ when $x_1$ and $x_2$ are equal. Two line segments are judged to be parallel if their angle difference is less than the angle threshold $\mu$.

In our method, the distance between two line segments is not used as a criterion for judging whether they are parallel or not. Because, a line segment, requires calculating the distance between it and all the other line segments, which is a huge computational burden.

*2) Airport ROI Refining Based on Parallel Line Segments:* As an airport often have multiple runways, an airport ROI may contain multiple parallels. But the main runway is the longest and most representative of the airport. Therefore, the longest parallel (corresponding to the main runway of the airport) is first selected when determining whether the ROI is an airport.

Fig. 11 illustrates the process of ROI refining. In the airport ROI, the longest internal parallel is close to the length of the main diagonal. On the contrary, the length of the parallel lines in the nonairport ROI tends to be far shorter than that of the diagonal. Hence, the false ROI can be further removed by comparing the length of the ROI's diagonal and that of the longest parallel line in the ROI. The lengths of the parallels have been given in refined SARLSD. The length of the diagonal can be obtained by

$$L = \lambda \times \sqrt{(X_2 - X_1)^2 + (Y_2 - Y_1)^2} \tag{20}$$

where $(X_1, Y_1)$, $(X_2, Y_2)$ are the horizontal and vertical coordinates values of the upper-left and lower-right corners of the ROI, respectively. $\lambda$ is the scaling factor that determines the scaling of the diagonal length of the ROI. If the lengths of the longest parallel lines exceed the threshold $L$, the ROI will be retained. Otherwise, the ROI will be removed from the results.

In addition, it is worth mentioning that the line segment detector branch and the airport ROI detection network branch do not interfere with each other, so the airport ROI detector branch can be replaced with any deep learning-based object detection network.

### B. Experimental Analysis of FLDNet

The reason for the superior performance of FLDNet over other detection models is the introduction of line segment detector branches. To illustrate that the introduction of line segment detector branches can improve the accuracy rate of airport detection, we compare YOLO v5 and FLDNet. The results in Fig. 12 show that the line segment detector can greatly improve P from 86.5% to 91.4%. AP increases from 91.7% to 92.5% and the F1 score also increases from 87% to 90%.

Five representative images of the airport and the corresponding detection results are given in Fig. 13. It can be seen that the biggest difficulty in airport detection is the presence of airport-like regions, such as rivers, roads, lakes, and shadows. As can be seen in Fig. 13(c), the airport detection results obtained using YOLO v5 are susceptible to interference from these regions. The weak scattering and regular shape of the airport runway, as described in Section I, cause parallel lines to appear in the airport, which is also verified in Fig. 13(d). Combining the line segments with the deep detection results, we can achieve the removal of the false detection regions and get better results, and obtain the results as in Fig. 13(e).

FLDNet improves the airport detection accuracy, but also increases the test time. For the single image test, YOLO v5 consumes 8.5 ms, which can be obtain from Table V. In comparison, FLDNet requires 204.3 ms for testing. The increase in test time is caused by the presence of line segment detector and the fusion of line segment and deep features. The former consumes more time, about 183.9 ms, and feature fusion takes 11.9 ms. Moreover, through experiments, we find that the number of ROIs does not affect the time cost of feature fusion, attributing to the simplicity of feature fusion.

### C. Detection Performance With Different Parameter Settings

Several free parameters exist for FLDNet, and different parameters can have an impact on the final detection results. For airport ROI detector branch, a detailed explanation can be found

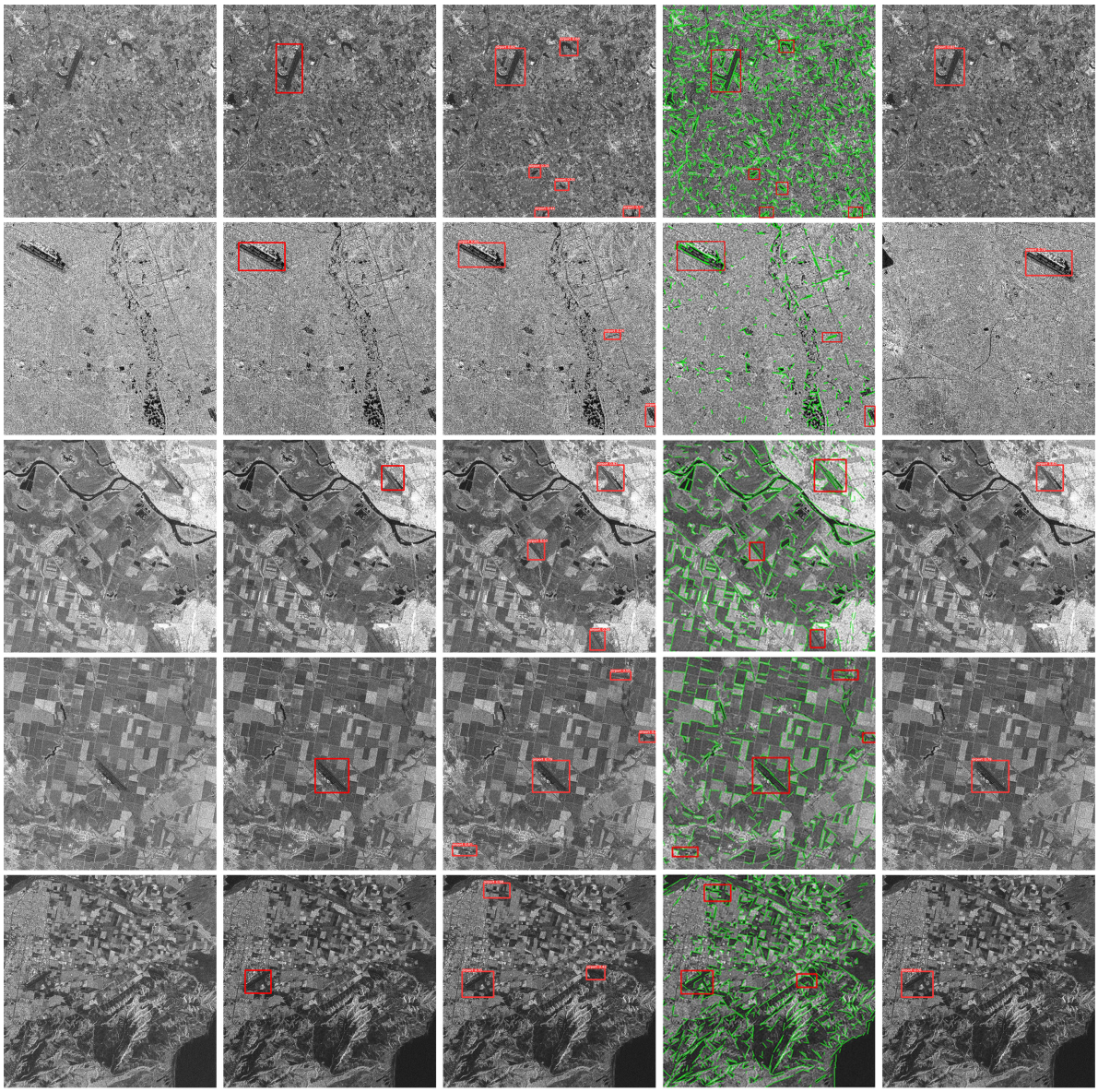



Fig. 13. Detection results of FLDNet. (a) Original SAR image. (b) SAR image with ground truth. (c) Detection results without line segments. (d) The result of line detection [red marked boxes are manually marked to highlight the detection result area in (c)]. (e) Detection results after adding line segments.

in [49], which will not be studied in this article. Four parameters exist in line segment detector branch, we found that the NFA threshold $\epsilon$ has little effect on the detection result and is usually set to 1. The ratio calculation weight $\beta$ is used to control the size of the pixel block used to calculate the gradient, with $\beta$ being small, the gradient calculation will become inaccurate, to make the detection results more accurate, $\beta$ needs to be set to 4 or 5. $\tau$ is used to control the angle range when clustering pixels. More regions of line segments are obtained as $\tau$ increases. $D$ represents the density of aligned pixels within a rectangle, which aims to avoid situations where two straight edges are present in the region with an angle between them smaller than the tolerance $\tau$. Low values of $D$ lead to inconsistent detections, while too large values of $D$ will have the effect of overcutting the line segments into smaller sizes. A detailed analysis of the four parameter settings for the line segment detection can be found in [55].

During the fusion of line segments and airport ROIs, two parameters need to be set, i.e., the angle threshold $\mu$ and the scaling factor $\lambda$. $\mu$ is used to determine whether line segments are parallel, and when $\mu$ is larger, more parallel lines appear, and thus, more ROIs are preserved. On the contrary, more ROIs are judged to be falsely detected ROIs, which are removed when $\mu$ becomes small. We explored the variation of P, R, and AP between $\mu$ from 0 to 7 and show it in Fig. 14(a). When $\mu$ is at 0 to 3, more ROIs are removed, the removed ROIs contain both airport and nonairport, so R is compromised, and the change in P varies according to the ratio of airports to nonairports

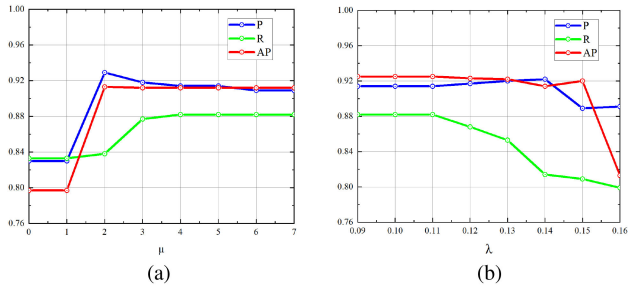



Fig. 14. Effect of different parameters on the detection results. (a) Angle threshold $\eta$. (b) Scaling factor $\lambda$.

in the removed ROIs. After $\mu$ is greater than 4, parallel lines are detected in all airport ROIs, so that P is increased without harming R. It can be seen that the optimal values of P, R, and AP are achieved when $\mu = 4$ or 5.

$\lambda$ is the parameter that controls the diagonal length $L$. As $\lambda$ increases, $L$ becomes larger, which makes it difficult for the parallel line length $h$ to be larger than $L$. And more airport ROIs will be judged as nonairports and removed. When $\lambda$ is reduced, more airports may be detected, but more false detection areas are inevitable. Therefore, we investigated the effect of varying $\lambda$ from 0.09 to 0.16 on P, R, and AP. The results are presented in Fig. 14(b). As can be seen that P and R are more sensitive to the change of $\lambda$, and the AP curve is relatively robust. R shows a decreasing trend when the $\lambda$ is from 0.11 to 0.14. On the contrary, P is unchanged when $\lambda$ is between 0.11 and 0.14. FLDNet performs poorly when $\lambda$ is greater than 0.14. With $\lambda$ in the interval from 0.09 to 0.11, FLDNet can achieve a rise in P without damaging R. Hence, we set $\lambda$ as the median of this interval in our experiments, i.e., $\lambda = 0.1$.

## VI. Conclusion

In this article, we first expose an SAD. It provides a benchmark resource for developing airport detection algorithms or other related tasks. Four state-of-the-art deep learning-based algorithms and three traditional airport detection algorithms demonstrate that SAD is a challenging SAR dataset for airport detection and can serve as a baseline for future algorithm development. In addition, an SAR airport detection method based on the line segment detector and deep network is proposed in this article. By fusing the airport ROIs obtained from the airport ROI detection network and the parallel line segments obtained from SARLSD, the detection accuracy is further improved and the number of false alarms has been effectively reduced especially in complex SAR images.

## References

[1] J. Illingworth and J. Kittler, "A survey of the hough transform," *Comput. Vision, Graph., Image Process.*, vol. 44, no. 1, pp. 87–116, 1988.

[2] R. Grompone, V. Gioi, J. Jakubowicz, J. Morel, and G. Randall, "LSD: A fast line segment detector with a false detection control," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 4, pp. 722–732, Apr. 2010.

[3] G. Tang, Z. Xiao, Q. Liu, and H. Liu, "A novel airport detection method via line segment classification and texture classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 12, no. 12, pp. 2408–2412, Dec. 2015.

[4] J. Tu, F. Gao, J. Sun, A. Hussain, and H. Zhou, "Airport detection in SAR images via salient line segment detector and edge-oriented region growing," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 14, pp. 314–326, Nov. 2020.

[5] Ü. Budak, U. Halıcı, A. Şengür, M. Karabatak, and Y. Xiao, "Efficient airport detection using line segment detector and fisher vector representation," *IEEE Geosci. Remote Sens. Lett.*, vol. 13, no. 8, pp. 1079–1083, Aug. 2016.

[6] X. Yao, J. Han, L. Guo, S. Bu, and Z. Liu, "A coarse-to-fine model for airport detection from remote sensing images using target-oriented visual saliency and CRF," *Neurocomputing*, vol. 164, pp. 162–172, 2015.

[7] N. Liu, Z. Cui, Z. Cao, Y. Pi, and S. Dang, "Airport detection in large-scale SAR images via line segment grouping and saliency analysis," *IEEE Geosci. Remote Sens. Lett.*, vol. 15, no. 3, pp. 434–438, Mar. 2018.

[8] Z. Li, Z. Liu, and W. Shi, "Semiautomatic airport runway extraction using a line-finder-aided level set evolution," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 7, no. 12, pp. 4738–4749, Dec. 2014.

[9] D. Zhu, B. Wang, and L. Zhang, "Airport target detection in remote sensing images: A new method based on two-way saliency," *IEEE Geosci. Remote Sens. Lett.*, vol. 12, no. 5, pp. 1096–1100, May 2015.

[10] L. Zhang and Y. Zhang, "Airport detection and aircraft recognition based on two-layer saliency model in high spatial resolution remote-sensing images," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 10, no. 4, pp. 1511–1524, Apr. 2017.

[11] F. Ma, F. Zhang, D. Xiang, Q. Yin, and Y. Zhou, "Fast task-specific region merging for SAR image segmentation," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5222316.

[12] F. Ma, F. Zhang, Q. Yin, D. Xiang, and Y. Zhou, "Fast SAR image segmentation with deep task-specific superpixel sampling and soft graph convolution," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art no. 5214116.

[13] N. Liu, Z. Cao, Z. Cui, Y. Pi, and S. Dang, "Multi-layer abstraction saliency for airport detection in SAR images," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 12, pp. 9820–9831, Dec. 2019.

[14] C. Tao, Y. Tan, H. Cai, and J. Tian, "Airport detection from large IKONOS images using clustered SIFT keypoints and region information," *IEEE Geosci. Remote Sens. Lett.*, vol. 8, no. 1, pp. 128–132, Jan. 2011.

[15] Ö. Aytekin, U. Zöngür, and U. Halici, "Texture-based airport runway detection," *IEEE Geosci. Remote Sens. Lett.*, vol. 10, no. 3, pp. 471–475, May 2013.

[16] W. Lv, K. Dai, L. Wu, X. Yang, and W. Xu, "Runway detection in SAR images based on fusion sparse representation and semantic spatial matching," *IEEE Access*, vol. 6, pp. 27984–27992, 2018.

[17] Z. Zhang, C. Zou, P. Han, and X. Lu, "A runway detection method based on classification using optimized polarimetric features and hog features for polsar images," *IEEE Access*, vol. 8, pp. 49160–49168, 2020.

[18] Q. Zhang, L. Zhang, W. Shi, and Y. Liu, "Airport extraction via complementary saliency analysis and saliency-oriented active contour model," *IEEE Geosci. Remote Sens. Lett.*, vol. 15, no. 7, pp. 1085–1089, Jul. 2018.

[19] H. Li, W. Hu, W. Li, J. Li, Q. Du, and A. Plaza, "A $^3$ CLNN: Spatial, spectral and multiscale attention convLSTM neural network for multisource remote sensing data classification," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 2, pp. 747–761, Feb. 2022.

[20] D. Hong et al., "More diverse means better: Multimodal deep learning meets remote-sensing imagery classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 5, pp. 4340–4354, May 2021.

[21] X. Wu, D. Hong, J. Chanussot, Y. Xu, R. Tao, and Y. Wang, "Fourier-based rotation-invariant feature boosting: An efficient framework for geospatial object detection," *IEEE Geosci. Remote Sens. Lett.*, vol. 17, no. 2, pp. 302–306, Feb. 2020.

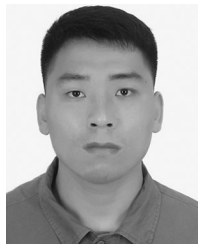
[22] X. Wu, D. Hong, J. Tian, J. Chanussot, W. Li, and R. Tao, "ORSim detector: A novel object detection framework in optical remote sensing imagery using spatial-frequency channel features," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 7, pp. 5146–5158, Jul. 2019.

[23] D. Hong, L. Gao, J. Yao, B. Zhang, A. Plaza, and J. Chanussot, "Graph convolutional networks for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 7, pp. 5966–5978, Jul. 2021.

[24] L. Chen et al., "A new framework for automatic airports extraction from SAR images using multi-level dual attention mechanism," *Remote Sens.*, vol. 12, no. 3, 2020, Art. no. 560.

[25] S. Tan, L. Chen, Z. Pan, J. Xing, Z. Li, and Z. Yuan, "Geospatial contextual attention mechanism for automatic and fast airport detection in SAR imagery," *IEEE Access*, vol. 8, pp. 173627–173640, 2020.

[26] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 580–587.
[27] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 1440–1448.
[28] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. 28th Int. Conf. Adv. Neural Inf. Process. Syst.*, 2015, pp. 91–99.
[29] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 2961–2969.
[30] S. Yin, H. Li, and L. Teng, "Airport detection based on improved faster RCNN in large scale remote sensing images," *Sens. Imag.*, vol. 21, no. 1, 2020, Art. no. 49.
[31] F. Chen, R. Ren, Tim Van de Voorde, W. Xu, G. Zhou, and Y. Zhou, "Fast automatic airport detection in remote sensing images using convolutional neural networks," *Remote Sens.*, vol. 10, no. 3, 2018, Art. no. 443.
[32] F. Zeng et al., "A hierarchical airport detection method using spatial analysis and deep learning," *Remote Sens.*, vol. 11, no. 19, 2019, Art. no. 2204.
[33] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," Apr. 2018. [Online]. Available: https://arxiv.org/abs/1804.02767
[34] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "YOLOv4: Optimal speed and accuracy of object detection," 2020. [Online]. Available: http://arxiv.org/abs/2004.10934
[35] X. Sun et al., "Fair1m: A benchmark dataset for fine-grained object recognition in high-resolution remote sensing imagery," *ISPRS J. Photogrammetry Remote Sens.*, vol. 184, pp. 116–130, 2022.
[36] P. Zhang, X. Niu, Y. Dou, and F. Xia, "Airport detection on optical satellite images using deep convolutional neural networks," *IEEE Geosci. Remote Sens. Lett.*, vol. 14, no. 8, pp. 1183–1187, Aug. 2017.
[37] S. Li, Y. Xu, M. Zhu, S. Ma, and H. Tang, "Remote sensing airport detection based on end-to-end deep transferable convolutional neural networks," *IEEE Geosci. Remote Sens. Lett.*, vol. 16, no. 10, pp. 1640–1644, Oct. 2019.
[38] Z. Huang, C. O. Dumitru, Z. B. P. Lei, and M. Datcu, "Classification of large-scale high-resolution SAR images with deep transfer learning," *IEEE Geosci. Remote Sens. Lett.*, vol. 18, no. 1, pp. 107–111, Jan. 2021.
[39] B. Cai, Z. Jiang, H. Zhang, D. Zhao, and Y. Yao, "Airport detection using end-to-end convolutional neural network with hard example mining," *Remote Sens.*, vol. 9, no. 11, 2017, Art. no. 1198.
[40] X. Wen, Z. Pan, Y. Hu, and J. Liu, "An effective network integrating residual learning and channel attention mechanism for thin cloud removal," *IEEE Geosci. Remote Sens. Lett.*, vol. 19, Mar. 2022, Art no. 6507605.
[41] L. Huang et al., "OpenSARship: A dataset dedicated to sentinel-1 ship interpretation," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 11, no. 1, pp. 195–208, Jan. 2018.
[42] T. Zhang et al., "SAR ship detection dataset (SSDD): Official release and comprehensive data analysis," *Remote Sens.*, vol. 13, no. 18, 2021, Art. no. 3690.
[43] J. Xia, N. Yokoya, B. Adriano, L. Zhang, G. Li, and Z. Wang, "A benchmark high-resolution GaoFen-3 SAR dataset for building semantic segmentation," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 14, pp. 5950–5963, May 2021.
[44] S. Wei, X. Zeng, Q. Qu, M. Wang, H. Su, and J. Shi, "HRSID: A high-resolution SAR images dataset for ship detection and instance segmentation," *IEEE Access*, vol. 8, pp. 120234–120254, 2020.
[45] T. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proc. IEEE Int.Conf. Comput. Vis.*, 2017, pp. 2980–2988.
[46] Z. Ge, S. Liu, F. Wang, Z. Li, and J. Sun, "Yolox: Exceeding yolo series in 2021," 2021. [Online]. Available: https://arxiv.org/abs/2107.08430
[47] Z. Sun et al., "An anchor-free detection method for ship targets in high-resolution SAR images," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 14, pp. 7799–7816, Jul. 2021.
[48] T. Miao et al., "An improved lightweight retinaNet for ship detection in SAR images," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 15, pp. 4667–4679, Jun. 2022.
[49] P. Wang, X. Sun, W. Diao, and K. Fu, "FMSSD: Feature-merged single-shot detection for multiscale objects in large-scale remote sensing imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 5, pp. 3377–3390, May 2020.
[50] U. Nazir, U. K. Mian, M. U. Sohail, M. Taj, and M. Uppal, "Kiln-Net: A gated neural network for detection of brick kilns in South Asia," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 13, pp. 3251–3262, Jun. 2020.
[51] B. Xue et al., "An efficient deep-sea debris detection method using deep neural networks," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 14, pp. 12348–12360, Nov. 2021.
[52] C. Wang, H. M. Liao, Y. Wu, P. Chen, J. Hsieh, and I. Yeh, "CSPNet: A new backbone that can enhance learning capability of CNN," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops*, 2020, pp. 390–391.
[53] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, "Path aggregation network for instance segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 8759–8768.
[54] M. L. Mekhalfi, C. Nicoló, Y. Bazi, M. M. A.l Rahhal, N. A. Alsharif, and E. Al Maghayreh, "Contrasting YOLOv5, transformer, and efficientdet detectors for crop circle detection in desert," *IEEE Geosci. Remote Sens. Lett.*, vol. 19, Jun. 2022, Art. no. 3003205.
[55] F. Gao, Y. Huo, J. Wang, A. Hussain, and H. Zhou, "Anchor-free SAR ship instance segmentation with centroid-distance based loss," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 14, pp. 11352–11371, 2021.
[56] Q. Du, T. Celik, Q. Wang, and H.-C. Li, "Fully convolutional lightweight pyramid network for vehicle detection in aerial images," *IEEE Geosci. Remote Sens. Lett.*, vol. 19, 2022, Art. no. 3506905.
[57] B. Al-Diri, A. Hunter, and D. Steel, "An active contour model for segmenting and measuring retinal vessels," *IEEE Trans. Med. Imag.*, vol. 28, no. 9, pp. 1488–1497, Sep. 2009.
[58] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk, "SLIC superpixels compared to state-of-the-art superpixel methods," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 11, pp. 2274–2282, Nov. 2012.
[59] T. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 2117–2125.
[60] C. Liu, R. Abergel, Y. Gousseau, and F. Tupin, "Lsdsar, A Markovian a contrario framework for line segment detection in SAR images," *Pattern Recognit.*, vol. 98, 2020, Art. no. 107034.
[61] R. Touzi, A. Lopes, and P. Bousquet, "A statistical and geometrical edge detector for SAR images," *IEEE Trans. Geosci. Remote Sens.*, vol. 26, no. 6, pp. 764–773, Nov. 1988.
[62] A. Desolneux, L. Moisan, and J. Morel, *From Gestalt Theory to Image Analysis: A Probabilistic Approach*, vol. 34. Berlin, Germany: Springer, 2007.
[63] A. Myaskouvskey, Y. Gousseau, and M. Lindenbaum, "Beyond independence: An extension of the a contrario decision procedure," *Int. J. Comput. Vis.*, vol. 101, no. 1, pp. 22–44, 2013.



**Daochang Wang** (Student Member, IEEE) is currently working toward the master's degree majoring in electronic information (computer technology) with the Beijing University of Chemical Technology, Beijing, China.

Her research interests include synthetic aperture radar image processing and target detection.

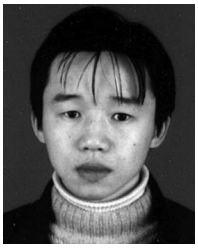
**Fan Zhang** (Senior Member, IEEE) received the B.E. degree in communication engineering from the Civil Aviation University of China, Tianjin, China, in 2002, the M.S. degree in signal and information processing from Beihang University, Beijing, China, in 2005, and the Ph.D. degree in signal and information processing from the Institute of Electronics, Chinese Academy of Sciences, Beijing, China, in 2008.

He is currently an Professor in Electronic and Information Engineering with the Beijing University of Chemical Technology, Beijing, China. His research interests include remote sensing image processing, high performance computing, and artificial intelligence.

Dr. Zhang is an Associate Editor for IEEE ACCESS and a Reviewer for the IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING, the IEEE JOURNAL OF SELECTED TOPICS IN APPLIED EARTH OBSERVATIONS, *Remote Sensing*, the IEEE GEOSCIENCE AND REMOTE SENSING LETTERS, and the *Journal of Radars*.

**Fei Ma** (Member, IEEE) received the B.S., M.S., and Ph.D. degrees in electronic and information engineering from the Beijing University of Aeronautics and Astronautics, Beijing, China, in 2013, 2016, and 2020, respectively.

From 2017 to 2018, he was a Research Fellow with the Department of Electrical Engineering, McGill University, Montreal, QB, Canada. Since 2020, he has been an Associate Professor with the College of Information Science and Technology, Beijing University of Chemical Technology, Beijing, China. His research interests include synthetic aperture radar image processing, machine learning, artificial intelligence and target detection.



**Wei Hu** received the B.S. and M.S. degrees in computer science from the Dalian University of Science and Technology, Dalian, China, in 1999 and 2002, respectively, and the Ph.D. degree in computer science from Tsinghua University, Beijing, China, in 2006.

He is currently an Associate Professor of Computer Science with the Beijing University of Chemical Technology, Beijing, China. His research interests include computer graphics, computational photography, and scientific visualization.

Dr. Hu has been a Reviewer for the IEEE TRANSACTIONS ON VISUALIZATION AND COMPUTER GRAPHICS, the *Euro Graphics*, and the *Pacific Graphics*.

**Yu Tang** Authors' photographs and biographies not available at the time of publication.

**Yongsheng Zhou** (Member, IEEE) received the B.E. degree in communication engineering from the Beijing Information Science and Technology University, Beijing, China, in 2005, and the Ph.D. degree in signal and information processing from the Institute of Electronics, Chinese Academy of Sciences, Beijing, China, in 2010.

He is currently a Professor of Electronic and Information Engineering with the College of Information Science and Technology, Beijing University of Chemical Technology, Beijing, China. His research interests include target detection and recognition from microwave remotely sensed image, digital signal and image processing, etc.