Validation of Global Airport Spatial Locations From Open Databases Using Deep Learning for Runway Detection

Chen Ji, Liang Cheng, Ning Li, Fanxuan Zeng, and Manchun Li

***Abstract*—Airports are important transportation hubs, but their locations searched on open databases are not reliable. Manual verification of these locations is time-consuming and labor-intensive, so that a rapid and automated validation of their spatial location is necessary. In this study, three datasets of global airports were collected and fused into one dataset through coordinate and name matching. The fused dataset contained 46 290 airport (with runway) records. Then, we downloaded the remote sensing images of these airports from Google Earth. To determine whether there were airports in these images, we proposed a process framework. In this framework, we used a two-scale runway detector based on YOLOv3 to initially detect the airport runway, then used a re-classifier based on ResNet-101 to improve the accuracy of the initial detection results and gave a comprehensive result score. The precision of this process framework and the airport recall rate on the test dataset reached 95.4% and 95.6%, respectively. The framework was applied to airport locations around the world. When the threshold of the result score was set to 0.65, 29 259 airport records passed the verification. In addition, we manually verified the application results. The accuracy of the process framework reached 91%, while its speed was 15 times faster than that of the manual verification. The results showed that the entire process framework can quickly and reliably help verify the spatial locations of airports worldwide and provide processing ideas for the validation of the spatial locations of other remote sensing objects.**

***Index Terms*—Airport, deep learning, open database, remote sensing, runway detection, spatial location.**

I. Introduction

AIRPORTS are important transportation hubs, and their spatial locations are related to people's lives and social economy [1], [2], which is why it is important to study the spatial distribution of airports around the world. Usually, the location of commercial airports can be easily found on web maps, but these only represent a part of the total number of airports worldwide, and other airports cannot be found on such maps. Fortunately, some open-source geographical name and airport thematic databases contain a large amount of airport location data, which, however, contains errors. For example, the database provides a coordinate without an airport being located, or a coordinate with a position deviation. In general, these data have great value, but they cannot be used directly. To get reliable information from these datasets, the validation of the airport coordinates is essential. In recent years, high-resolution remote sensing techniques have developed rapidly [3], [4], and many map suppliers (Google Earth, Microsoft Bing Map, MapBox, etc.) have released remote sensing image maps with global coverage, providing a material basis for verifying airport spatial location data through remote sensing images. However, there are thousands of airports around the world, and they are continuously changing. It is time-consuming and labor-intensive to verify the location of airports by manually viewing remote sensing images, which is not a good solution. The field of computer vision has been focusing on image interpretation, and many methods of pattern recognition have been proposed. Especially since the birth of AlexNet in 2012 [5], deep learning algorithms have been revived, and various fast and accurate object detection algorithms have been continuously proposed. Some scholars have applied these algorithms to remote sensing images and achieved good results [6], [7]. Applying the object detection algorithm to the remote sensing image enables the computer to determine whether there is an airport in the image and frame the location of the airport relative to the image. Compared to a manual workflow, this automated method to verify the spatial locations of the airport is not only more efficient, but also entails lower costs. Overall, it is very important to collect the airport location data as completely as possible and automatically detect airports in remote sensing images.

For the collection of airport location data, thanks to the development of modern Internet and information technology, such data can be obtained from open source geographical names and thematic databases on the Internet. The geographical names database includes GeoNames (GNO),[1] established by a

Manuscript received May 28, 2020; revised September 1, 2020 and October 12, 2020; accepted October 24, 2020. Date of publication October 30, 2020; date of current version January 6, 2021. This work was supported in part by the National Key Research and Development Plan under Grant 2017YFB0504205 and in part by National Science Foundation of China under Grant 41622109, Grant 41371017, and Grant 41701374. *(Corresponding author: Liang Cheng.)*

The authors are with the Jiangsu Provincial Key Laboratory of Geographic Information Science and Technology, Key Laboratory for Land Satellite Remote Sensing Applications of Ministry of Natural Resources, School of Geography and Ocean Science, Nanjing University, Nanjing, Jiangsu 210023, China, with the Collaborative Innovation Center of South China Sea Studies, Nanjing, Jiangsu 210023, China, with the Jiangsu Center for Collaborative Innovation in Geographical Information Resource Development and Application, Nanjing, Jiangsu 210023, China, and also with Jiangsu Center for Collaborative Innovation in Novel Software Technology and Industrialization, Nanjing, Jiangsu 210023, China (e-mail: jichen@smail.nju.edu.cn; lcheng@nju.edu.cn; mg1827061@smail.nju.edu.cn; dg1827001@smail.nju.edu.cn; limanchun@nju.edu.cn).

Digital Object Identifier 10.1109/JSTARS.2020.3034609

[1][Online]. Available: http://www.geonames.org

European organization, USGS Geographic Names Information System (GNIS),[2] established by the US government, and GEOnet Names Server (GNS),[3] established by the US military. The airport thematic database includes OurAirports (OAP).[4] These databases contain data such as airport names and spatial locations, but they have different data structures and classifications, and there are repetitions in the recorded content. How to integrate the geographical names data from multiple sources has always attracted the attention of scholars. Usually, the fusion of geographical name data is realized by the method of rule matching [8], [9], [10]. In recent years, some scholars have proposed a deep learning method, in which the computer itself proposes matching rules to realize the fusion of geographical names data [11]. Although it is not difficult to obtain the spatial location data of the airport, the accuracy of the data is debatable. Some researchers used OSM and other map data to verify geographical name data [12], [13], but such network maps themselves also have data accuracy problems. As an important way to observe the earth, remote sensing images can accurately reflect the position of objects on the ground. Therefore, the better way to verify the spatial location of geographic objects is the confirmation through remote sensing images. For airports with runways, the validation of the spatial location is to detect the airport in the remote sensing image at the corresponding coordinate.

The detection of airports in remote sensing images has attracted the attention of researcher in recent years. The airport is a large-scale object in the remote sensing image. There are obvious commonalities and differences between different airports. Airports are usually composed of runways, aprons, airplanes, vehicles, and buildings. However, in different airports, these facilities are combined in different ways, resulting in huge differences in the appearance of airports. The background around the airport is also very complex. The surrounding roads, fields, residential areas, and other features are mixed with the airport, which increases the difficulty of airport detection. Generally speaking, the airport is detected by identifying local features of the airport. In this way, detection methods can be divided into two categories, one is to detect the airport by recognizing aircrafts [14]–[19], and the other is to detect the airport by recognizing runways. However, there are no aircrafts in remote sensing images of many airports, so it is more versatile to detect airports by recognizing runways.

The detection of airport runways can be divided into traditional methods and deep learning methods. The basic idea of the traditional method is to first extract the features in the image, and then use a classifier to confirm the features belonging to the runway, so as to locate the position of the airport. Feature extraction methods include visual saliency [20]–[25], line segment features [26], or multi-spectral features [27]–[31]. Classifiers generally use machine learning methods, such as SVM [26] and CRF [23]. With the application of convolutional neural networks in computer vision in 2012, deep learning methods have gradually become the mainstream of object detection in remote sensing [32], [33]. The object detection method based on deep learning is divided into a two-stage and a one-stage method. The two-stage method, represented by Fast-RCNN [34], Faster-RCNN [35], and Mask-RCNN [36], has a better accuracy, while the one-stage method, represented by YOLO [37]–[39] and SSD [40], is less time consuming. Although the two-stage method [14], [41]–[44] has also obtained good results in runway detection, it is still relatively slow for the detection in large-scale remote sensing images, which is not conducive to a global application. The one-stage method [45]–[48]is faster and suitable for global applications, but its ability to detect small objects needs to be improved. To solve this problem, some multi-scale methods for remote sensing images have been proposed [49]–[51].

In addition to selecting a good algorithm, the use of deep learning methods for object detection in remote sensing also requires the preparation of high-quality samples to train the model. The first step in sample preparation is sample labeling, which is usually done with rectangular frames. However, due to the directionality of the airport runway, the samples marked with a regular rectangular frame contain a large number of non-runway features, which is not conducive to model convergence. Therefore, for the sample labeling of airport runways, it is more suitable to use polygons [52], [53]. The use of polygons for sample labeling can ensure that the sample contains as little irrelevant information as possible after augmenting samples. The second step in sample preparation is data augmentation. The purpose of data augmentation is to expand samples with limited labels, so that the trained model has a better generalization ability. Histogram equalization, brightness change, cropping, rotation, mirroring, and other operations are commonly used methods for sample augmentation [54]. In particular, the same object may appear on remote sensing images in different directions. For object detection algorithms based on deep learning, these objects are completely different. In order to make the trained deep learning model have stronger directional generalization ability, some scholars have proposed rotation-invariant methods [55], [56]. In addition, the sample augmentation can also improve the generalization ability of the model, so the rotation method of augmentation is very important for the object detection of remote sensing images.

In summary, the spatial location of the airport from the open source database is of great value, but it is not reliable. It is important to verify these locations using object detection algorithms. However, due to the complexity of the airport and its background, it is very challenging to automatically detect the airport through remote sensing images. Although many studies have proposed lots of methods for airports detection, no study has applied such methods to verify the spatial location of airports around the world. The purpose of this study, thus, is to detect the presence of airport runways in remote sensing images to verify the spatial locations of airports (with runways). The main contributions of this work can be summarized as follows.

1) The authenticity of datasets on the spatial location of airports was verified. Integrated multi-source data containing airport locations and remote sensing images were downloaded at corresponding coordinates, and automated methods were used to detect whether these images contain airports with runways or not.

[2][Online]. Available: https://www.usgs.gov
[3][Online]. Available: http://geonames.nga.mil
[4][Online]. Available: https://ourairports.com

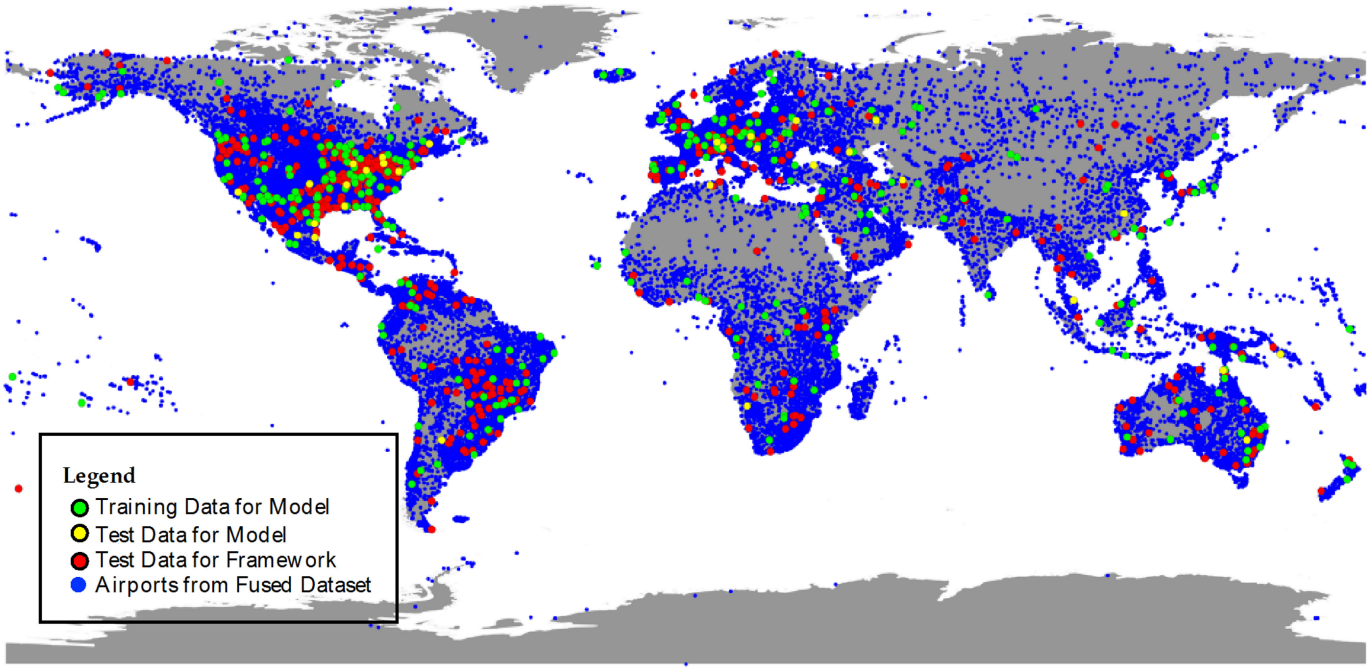


Fig. 1. Distribution of global airports and experimental data from fused dataset.

2) We proposed a runway detection process framework based on the combination of a two-scale runway detector and a re-classifier. Compared with ordinary runway detection approaches, this process framework has a higher accuracy and recall rate and can be extended to the detection of other remote sensing objects.
3) We provided the fused dataset of airports, the airport labeled samples and the code of the proposed framework on GitHub[5] for other researchers to download for their own studies.

II. Data Sources and Framework

A. Data sources

In this study, four open access databases containing airports were integrated to form a complete dataset. This dataset contained 5 8751 airports around the world, including 46 290 airports with runways. 500 locations in this dataset were randomly selected as the validation data and verified manually (Section IV-B). The spatial distribution of the airports is shown in Fig. 1, including global airports with runways (blue), the training data for the deep learning model (green), the test data for the deep learning model (yellow), and the test data for the proposed framework (red).

As shown in Table I, the location data of the airports come from OurAirports.com (OAP), GeoNames.org (GNO), GEOnet Names Server (GNS), and the USGS Geographic Names Information System (GNIS). OAP is a dataset providing global airport information, which contains four kinds of airports, including airports with runways, heliports, seaplane bases and balloonports. GNO provides global geographic names data, among which the categories related to airports are airports, airbases, heliports, and seaplane bases. GNS provides geographic names data except for the United States, while GNIS only provides data for the United States.

TABLE I
Data Sources of This Article

Type	Source	Coverage	Count
airport location	OurAirports	global	55225
	GeoNames	global	43656
	GEOnet Names Server	global (excluding the US)	41847
	USGS Geographic Names Information System	US	
	this paper	global	58751(total) 46290(with runway)
airport image	Google Earth (level-17)	37 * 37 tiles the longitude span is about 0.1°	46290

The images of the airports were taken from Google Earth (level-17), which were downloaded as tiles and spliced into a complete image in the size of 37 * 37. Each tile is a picture with a size of 256 * 256 pixels. In this size, the longitude span of the image is about 0.1 degree to avoid that the airport cannot be covered by the image due to data error.

B. Framework

The whole framework of airport locations validation proposed in this article is shown in Fig. 2. The framework is roughly

[5][Online]. Available: github.com/jichen1226/GlobalAirportsValidation

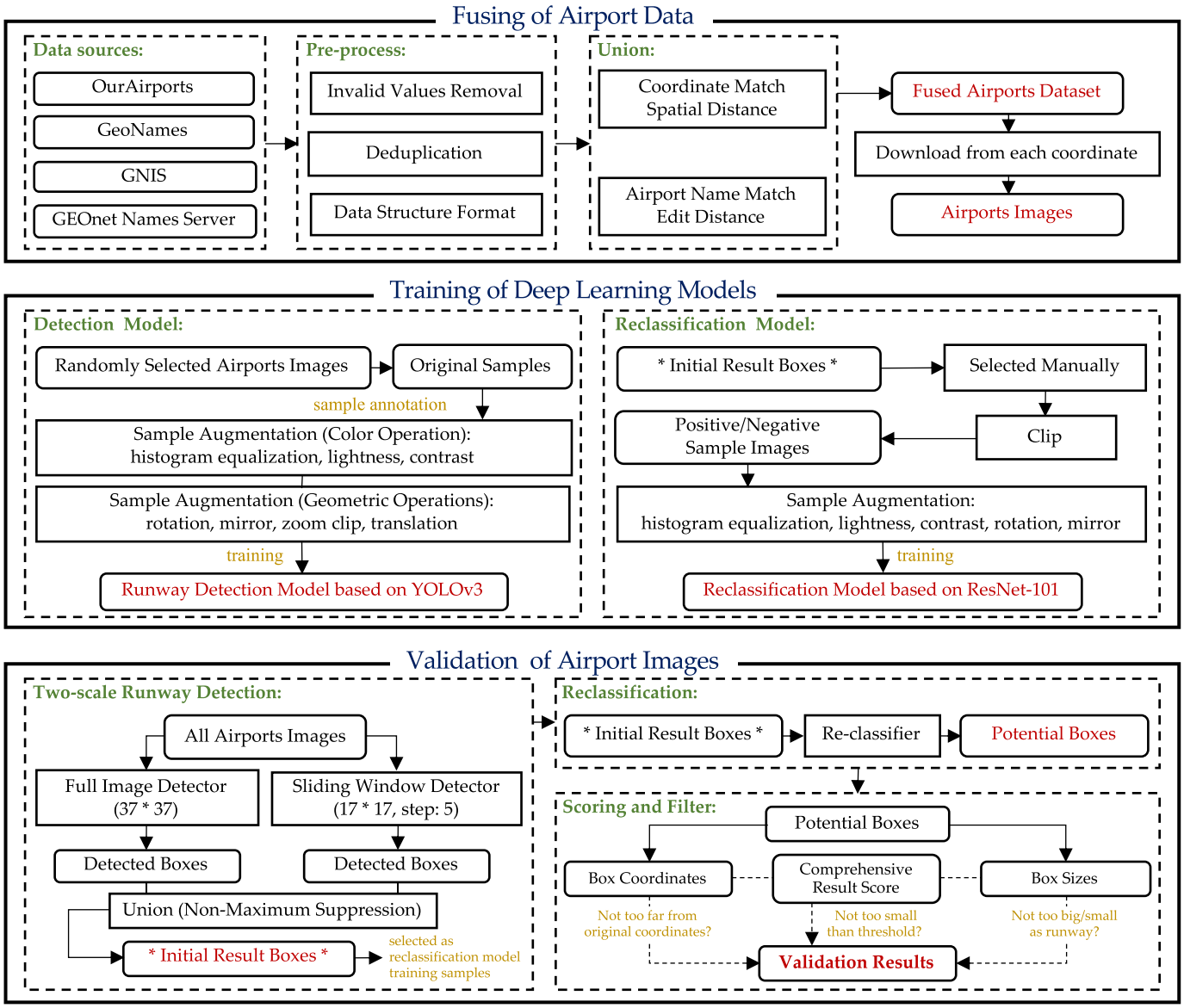


Fig. 2. Framework of validating global airport spatial locations.

divided into three parts: (1) Fusing of airport data: the three datasets of global airport location termed OAP, GNO, and GNS + GNIS were fused to get a more complete dataset. (2) Training of deep learning models: Two deep learning models were trained. The first one was the runway detection model based on YOLOv3 to detect runways in images, and the second one was the reclassification model based on ResNet-101 to reclassify the detected boxes for improving the accuracy. (3) Validation of airport images: The two trained models were combined and applied to images of all locations from fused dataset to obtain potential boxes. Based on these boxes, the prior knowledge of airports and the comprehensive scores of the two models' confidence were used to filter the final results. These final result boxes were regarded as real runways. If there were runways in one image, it was considered that there was an airport with runways here, and the location was verified.

1) Fusing of Airport Data: The purpose of data fusion was to fuse the global airport datasets from OAP, GNO, and GNS + GNIS into one set. To achieve this, the basic idea was to match the two sets of data, GNO and GNS + GNIS, with OAP data as the benchmark. When the data could be matched, the OAP dataset already contained these airport records, and GNO or GNS + GNIS were only used as information supplement; when the data could not be matched, the OAP dataset did not contain these airports, and the airport records were added from the other datasets.

Due to the different classification criteria of each dataset, pre-processing work was required on each set before starting the matching. The pre-processing included redefining the airport type, data deduplication, and formatting the data structure. The data structure and annotations of the fused data after pre-processing are shown in Table II. Data matching was based

TABLE II
STRUCTURE AND ANNOTATION OF FUSED DATA

Column	Annotation	Column	Annotation
id	unique identifier for airports	lat	latitude of the airport from original data
name	airport name	lng	longitude of the airport from original data
type	type of airports, including airports with runways, heliports, seaplane bases and balloonports	source	data source of records, including OAP, GNO, GNS, GNIS

on the assumption that if there were airport records with similar spatial locations and similar names in different datasets, these records were considered to refer to the same airport. In coordinate match, the difference between each location record in the OAP dataset and the records in other data sets is calculated. If the differences in latitude and longitude were less than the threshold (0.04°), the coordinate match was successful. If the data failed to match the coordinates, the text similarity was calculated based on the airport's name. The text similarity was calculated by the edit distance, as in (1). Before the calculation of the text similarity, keywords related to place names (such as Shanghai) and airport types (such as Airport) were removed from the airport name; then, the words were reordered, and finally the edit distance of the new string was calculated. If the name similarity was greater than 0.8, coordinate matching was performed again (with greater threshold: 0.08°). After the above processing, unmatched airport data were finally obtained and added to the OAP dataset to form a more complete airport dataset.

$$lev_{a,b}(i,j) = \begin{cases} \max(i,j) & \text{if } \min(i,j) = 0, \\ \min \begin{cases} lev_{a,b}(i-1,j)+1 \\ lev_{a,b}(i,j-1)+1 \\ lev_{a,b}(i-1,j-1)+1 \end{cases} & \text{otherwise.} \end{cases} \tag{1}$$

2) Training of Deep Learning Models: In this framework, two deep learning models were trained. The first was the runway detection model, which uses the neural network structure of YOLOv3. YOLOv3 was one of the most advanced object detectors at the time of writing this article. It has a good balance in speed and accuracy [39]. For global validation, speed is the key to validation, so that this study used YOLOv3 as the airport runway detector. The second model was the reclassification model. This model used ResNet-101 as the neural network structure and employed a part of the runway detection results as training samples. All sample images used in the training were from Google Earth and were labeled and augmented. Fig. 3 shows some examples of sample labeling and data augmentation.

Object detection in the field of computer vision is divided into three stages: image classification, object localization, and semantic segmentation. For the different stages, the sample annotation methods are also different. Runway detection belongs to the detection stage of object localization, in which the sample is commonly labeled with a regular rectangle. However, airport runways are mostly oblique in remote sensing image. If these are labeled with a regular rectangle, the label box contains too much information about surrounding features, which is not conducive to the training of the model. Therefore, the runway sample annotation in this study was labeled with polygons to ensure that the runway samples after the rotation method of data augmentation contained as little surrounding ground feature information as possible.

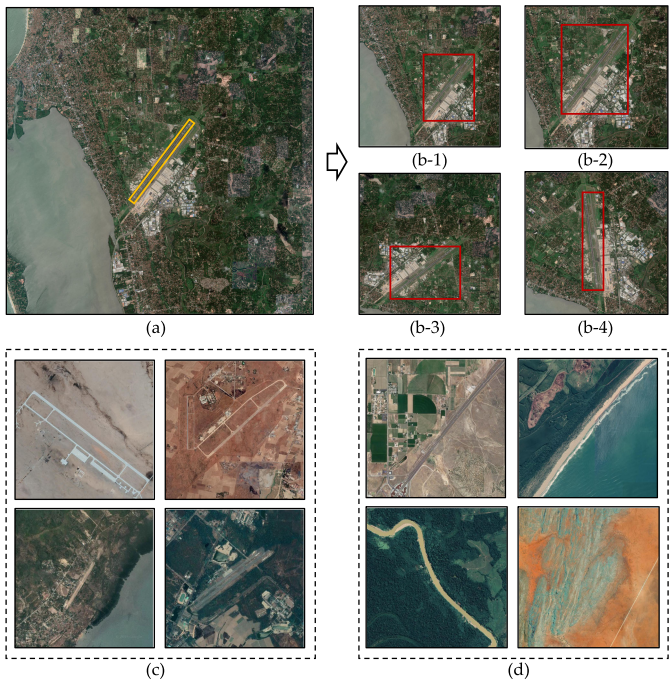


Fig. 3. Examples of training samples: (a) Raw labeled image; (b) Samples after data augmentation for runway detector; (c-d) Positive and negative samples for re-classifier.

Limited by the number of labeled airport runway samples, and to train a model with stronger generalization ability, it is necessary to augment the sample data. Data augmentation is divided into color operations, geometric operations, and labeled box operations [57]. For remote sensing images, the proportion of labeled boxes is small, so that we did not use labeled box operations. As shown in Table III, methods such as histogram equalization, lightness, contrast, rotation, mirror, zoom clip, and translation were used to augment samples. The color operation made the model adapt to images in different regions, the rotation and mirror methods made the model adapt to runways in different directions, and the zoom clip method made the model adapt to runways of different sizes. After data augmentation, the final samples were 128 times larger than the labeled samples.

Before training, we customized some hyperparameters and the training strategy. YOLOv3 uses anchor boxes to predict bounding boxes. To obtain anchor boxes, the usual approach is to use k-means algorithm clustering from the ground truth boxes in the training dataset. The k-means algorithm needs to specify the

TABLE III
THE METHODS OF SAMPLE DATA AUGMENTATION USED IN THIS STUDY

Type	Method	Param	Multiple
Color operations	histogram equalization	-	2
	lightness and contrast	with other methods	-
Geometric operations	rotation	45% + 1~5%	8
	mirror	horizontal	2
	zoom clip	80%, 60%, 40%, 20%	4
	translation	with zoom clip	-
Total	-	-	128

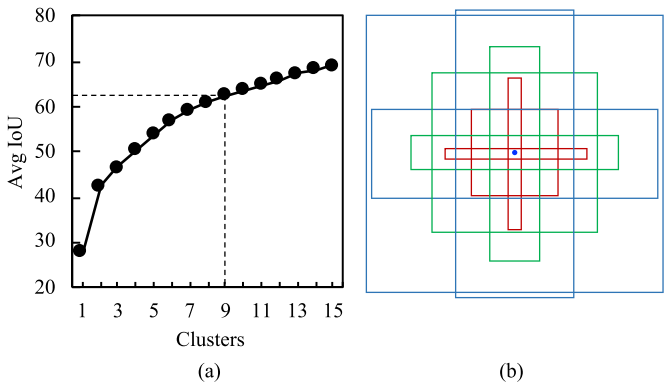


Fig. 4. Selection of anchor box: (a) The average IOU with various choices for the number of clusters. (b) The shapes of anchor boxes at 9 anchor boxes. In YOLOv3, the blue, green and red anchor boxes are used to predict at different scales of 13*13, 26*26, and 52*52, respectively.

number of clusters. As shown in Fig. 4(a), we have calculated different numbers of cluster and calculated the average value of the maximum IoU (Intersection over Union) between the bounding box and each anchor box. Obviously, as the number of clusters increases, the average IoU between the anchor box and the bounding box will stabilize. More clusters will allow greater overlap between the anchor box and the bounding box. However, as the number of anchor boxes increases, the number of convolution filters in the prediction filter will increase linearly. This will result in a larger network size and increase training time. Therefore, we stuck with 9 anchor boxes, to stay true with the original YOLOv3 implementation. By using 9 anchor boxes, the average IoU reached 62%. When the input size was set to 416, the sizes of anchor boxes were (7,104 112,9 63,63 26,153 157,36 118,122 65,210 211,87 204,217) in our training. Their shapes are shown in Fig. 4(b). The training of the runway detection model was based on the pre-trained model released on the website of YOLOv3 and we used the Adam algorithm as the optimizer for training. There were 100 epochs in the whole training process. The learning rate of the first 50 epochs was set to 0.001, and the layers of darknet body were frozen, and the learning rate of the last 50 epochs was set to 0.0001, which unfroze all layers. The training of the reclassification model was based on the pre-training model of Resnet-101. The entire training had a total of 100 epochs, using Adam algorithm as the optimizer, and the learning rate was set to 0.0001.

3) Validation of Airport Images: As mentioned at the beginning of Section II-B, the first step of airport validation was the runway detection. To adapt to different scales of the airport, two runway detectors of different scales were applied to the target image. The first detector detected medium and large airport runways at the entire scale of the image (37 * 37 tiles); the second detector detected small and medium airport runways in the image with a sliding window of 17 * 17 tiles (with a step size of 5). The reason why these two window sizes were selected for detection are discussed in Section IV-A. After detection, the results of the two detectors were used as candidate boxes for non-maximum suppression (NMS), and the detected boxes were merged into the initial result boxes.

Initial results were obtained through the runway detection, but there were some non-runway objects (such as rivers, highways, strip clouds, etc.) in the results. To get more reliable results, the second step of our airport validation was the reclassification of the initial results. The re-classifier determined whether all potential boxes obtained in the first step were airport runways. Fig. 5 shows the strategy for runway detection and reclassification. The third step was to filter the result box based on the prior knowledge of airports. If the box was too small (smaller than the size of one tile) or too large (larger than the size of 25*25tiles) as a runway, this box was discarded. If the center coordinates of the box were too far (more than 0.04°) from the location to be verified, this box was removed. Finally, based on the confidence scores of the runway detector and the re-classifier, a scoring formula for evaluating the final result was employed:

$$score_{result} = \begin{cases} score_{detector} \text{ if } score_{detector} > 0.9, \\ \frac{2 * score_{detector} * score_{reclassifier}}{score_{detector} + score_{reclassifier}} \quad \text{otherwise.} \end{cases} \tag{2}$$

When the result score was greater than 0.65 (empirical value), the box was judged to be an airport runway, and the image containing the result box was verified. At the same time, longitude and latitude in the center of the result box were compared with the records in the fusion dataset to correct the offset coordinates in the dataset.

III. RESULTS

A. Performance of the Proposed Framework

This article proposed a data fusion method based on coordinate and name matching. After the data fusion of three open source datasets, there were 58 751 airports, including 46 290 airports with runways. We manually selected 300 airports for labeling and data augmentation, and 38 400 samples were used to train the runway detection model. In addition, 30 airports were

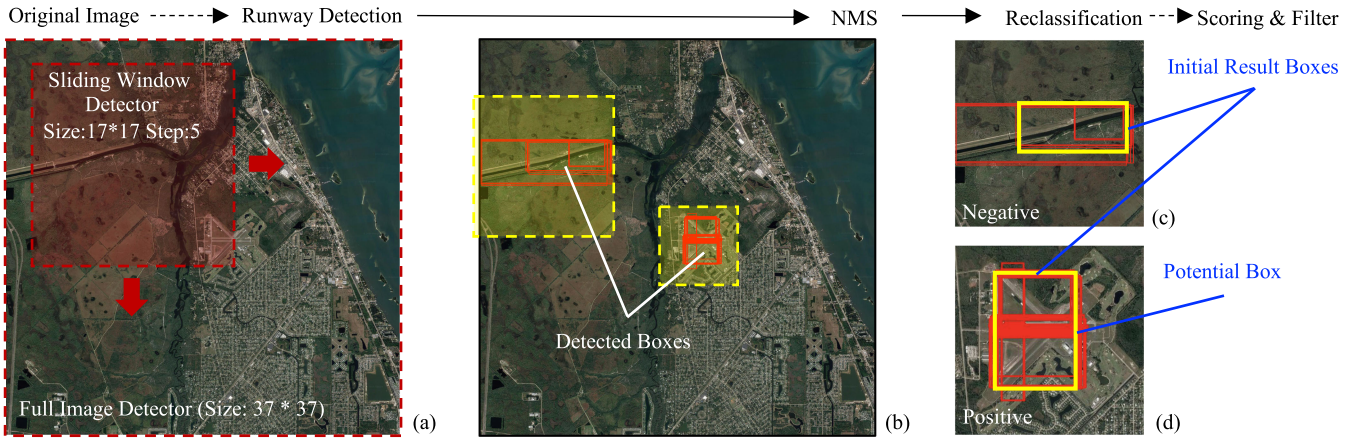


Fig. 5. Strategy for runway detection and reclassification from the Google Earth image (37 * 37 tiles) at airport coordinates: (a) The red dotted boxes are 2 runway detector areas of different scales. (b) The red solid boxes are detected results by runway detection. (c)–(d) The yellow solid boxes are obtained by merging detected boxes after NMS, and the box in (c) is judged as negative result while the box in (d) be judged as positive result by the re-classifier.

selected as the test data for the runway detection model. The images of these 30 airports were also labeled and augmented, generating 3840 test samples. After training of the runway detection model, the detection strategy proposed in Section II-B was applied to all images of airports around the world to obtain initial results. From the initial results, we selected 1222 potential regions (positive: 811, negative: 411) as training data for the reclassification model. These data were augmented by 45-degree rotation and histogram equalization to generate 19552 (positive:12976, negative: 6576) samples. The data augmentation method was the same as that for the training data. Fig. 6 shows the performance of model in different stages during training. When the epoch reached 100, the loss of the runway detection model was 5.2, the test precision was 92.8%, and the recall was 85.1%. The loss of the reclassification model was 6.7, the test precision was 85.9%. By using a NVIDIA 2080Ti graphics card, the whole training process for detection and reclassification model took about 5 days.

The trained models were used as the detector and re-classifier of the framework proposed in this article, and the framework performed well. The test results are shown in Table IV. The precision, recall, and speed of this framework were better than those proposed by previous scholars (Method1 [43], Method2 [58], Method3 [44]). This was due to the rapid development of deep learning algorithms, which improved the precision. The data augmentation method and recognition strategy used in this article improved the recall rate. If only the runway detection model (based on YOLO) was used, the precision of the result was 92.8%, the runway recall rate was 85.1%, and the airport recall rate was 91.7%. When using the framework proposed in this article, the precision was improved by 2.6% to 95.4%, the runway recall rate was increased by 5.3% to 90.4%, and the airport recall rate was increased by 4.1% to 95.8%. The multi-scale runway detector in the framework improved the model's ability to identify small airports, thereby increasing the resulting recall rate. The improvement of the precision of the detection results was attributed to the addition of the re-classifier.

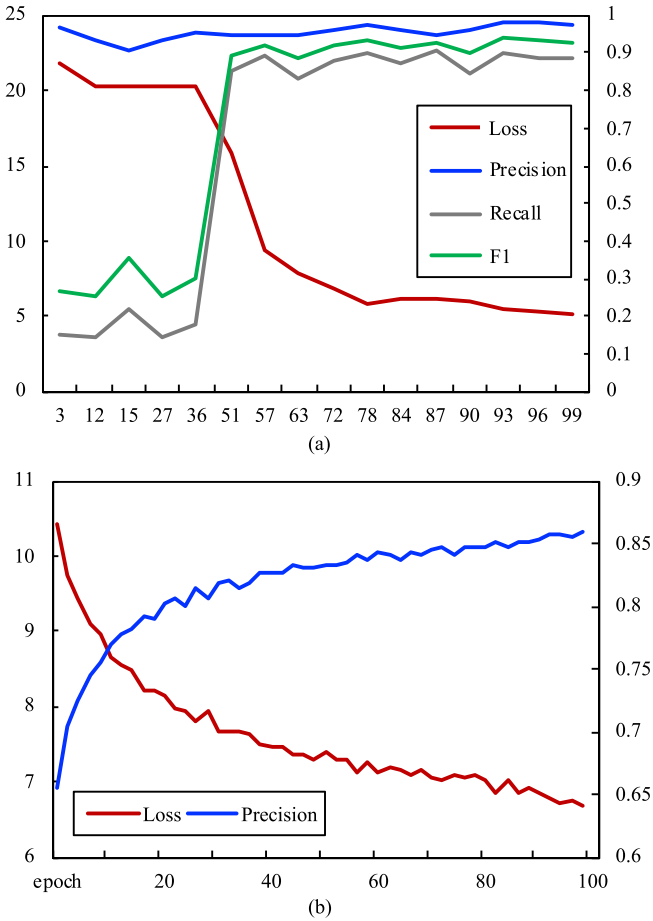


Fig. 6. Loss, precision, recall and f1 of the trained model with different epoch in test: (a) The runway detection model based on YOLOv3. (b) The reclassification model based on ResNet-101.

B. Application of the Proposed Framework

The framework proposed in this article was applied to remote sensing images of 46290 airport locations worldwide, and the

TABLE IV
PERFORMANCE OF THE RUNWAY DETECTION MODEL AND THE FRAMEWORK PURPOSED IN THE ARTICLE

Model	Precision	Recall (Runway)	Recall (Airport)	Speed (s/image)
Methods1	80.0%	-	83.7%	29.8
Methods2	93.1%	-	95%	24.2
Methods3	94.3%	-	95.4%	26.4
Single-scale runway detector (Faster-RCNN)	94.1%	84.4%	89.8%	7.8
Single-scale runway detector (YOLO)	92.8%	85.1%	91.7%	**2.4**
Two-scale runway detector + Re-classifier	**95.4%**	**90.4%**	**95.8%**	8.3

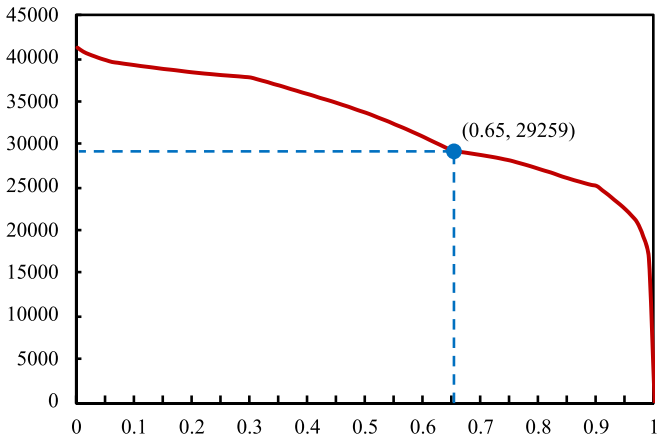


Fig. 7. Results of the application of the framework to airport locations around the world. In this chart, the x-axis represents the threshold of the result score, and the y-axis represents the number of results.

results were scored according to (2) in Section II-B-3). The remote sensing image of each location was derived from the level-17 images of Google Earth, with a resolution of about 1.2 m. An entire image was composed of 37 * 37 tiles, where each tile was a picture composed of 256 * 256 pixels, so that the image size was 9472 * 9472 pixels (about 18 MB). To complete the validation of all images, the framework took 102.9 hours, with an average of 8 seconds per image. The framework detected initial result boxes in 41296 images. Based on the experience of visual sampling, we recommend using a score of 0.65 as the threshold for passing the validation (Fig. 7). Under this threshold, 29 259 images passed the validation, that is, there were airports with runways in those images. The reliability of these results is discussed in Section IV-B.

IV. DISCUSSION

A. *Selection of Image and Sliding Window Size*

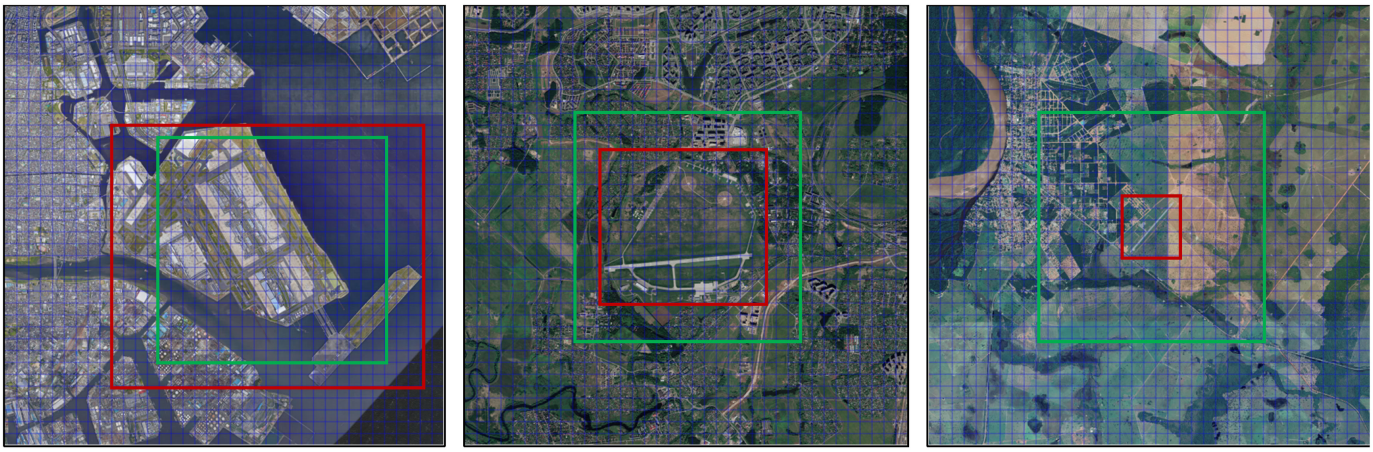
Applying the runway detection model to a suitable size of images could detect runways more accurately. Conversely, some small runways in the images could not be detected, or some objects with long linear features (such as highways and rivers) were incorrectly detected as runways. However, there are huge differences in the size of different airports. As shown in Fig. 8, large airports may have multiple runways with a length of several km, while small airports have runways with a length of only a few hundred m. Because a fixed-scale detector cannot complete the detection and validation of airport runways in different locations, we used the sliding window method to achieve a multi-scale recognition of the images, so as to accurately complete the detection of airport runways of different sizes.

Based on Google Earth's level-17 images, our study counted pixels of the runway length, the minimum number of tiles required to contain the runway, and the airport. The results are shown in Table V, and the first column (Airport Scale) in the table refers to the ratio of the height/width of the airport to the height/width of the image. When the image was composed of 37 * 37 tiles, 95% of the airports accounted for 10 to 80% of it. In our framework, by the zoom method of data augmentation, the training sample images were clipped by the proportion of the airport to the entire image (20–80%), so that the trained model has a better generalization ability for multi-scale detection. When the detection window was fit to the entire image (37 * 37 tiles), airports with a scale of 20 to 80% could be well detected, and when the detection window was half the image size (17 * 17 tiles), airports with a scale of 10 to 40% could be well detected. Therefore, the use of these two window sizes allows the detector to more accurately process images containing airports of various sizes.

B. *Reliability of Validation Results*

To prove the reliability of the framework for the validation of the spatial locations of airports worldwide, we sampled some airports from the global airport location dataset and checked the results of the framework manually. The sampling considered the size and spatial distribution of the airports. Based on the spatial distribution of the whole samples, we ensured that large, medium, and small airports were sampled. The results of the manual verification are shown in Table VI. The total number of samples was 500. Among them, 292 airports were found in 281 sample images by manual verification. The framework proposed in this article correctly identified 455 sample images and the accuracy reached 91.0%. Among the 281 sample images containing airports, 266 images were correctly identified with an accuracy of 94.7%. Among the last 219 images without airports, 198 images were correctly judged, and the judgment accuracy was 86.3%.

We further analyzed the reasons for the misidentifications. There were 15 images containing airports, but they were wrongly judged by the framework. In one image of them, there was no runway but aircrafts. In two images of them, the airports



(a) (b) (c)

Fig. 8. Examples of airports on Google Earth Image (37 * 37 tiles): (a) Large airport in Japan. (b) Medium airport in Russia. (c) Small airport in Brazil. The blue box represents the range of one tile, the red box represents the extent of the airport, and the green box represents the range of the sliding window (17 * 17 tiles).

TABLE V
STATISTICS OF AIRPORT SCALE IN SAMPLE IMAGES

Airport Scale (in the image size of 37*37 tiles)	Count	Pixel range of the runway length	Tile range of runway on x-axis/y-axis	Tile range of airport on x-axis/y-axis
< 10%	10	435 ~ 888	1~3/1~3	1~3/1~3
10 ~ 20%	127	490 ~ 1901	1~7/1~7	1~7/1~7
20 ~ 30%	60	816~ 2818	1~9 / 1~11	1~11/2~11
30 ~ 40%	28	778 ~ 3758	2~13/2~12	7~14/4~14
40 ~ 60%	51	825 ~ 5327	1~19/1~20	6~22/3~20
60 ~ 80%	19	1192 ~ 7168	1~22/1~22	14~29/10~29
> 80%	5	3382 ~ 5749	1~21/9~19	22~36/19~33
All	300	435 ~ 7168	1~22/1~22	1~36/1~33

TABLE VI
SAMPLING STATISTICS OF THE TEST RESULTS BY THE VALIDATION FRAMEWORK APPLICATION

Continent	Images	Airports	Runways	Detected number	Positive number	Correct Validation	Accuracy
Africa	43	20	23	24	22	40	93.0%
Asia	38	29	37	41	37	37	97.4%
Europe	80	52	56	63	55	72	90.0%
North America	199	109	123	149	125	179	89.9%
Oceania	31	20	23	23	19	26	83.9%
South America	88	51	52	57	47	80	90.9%
In Islands	21	11	11	13	12	21	100.0%
Total	500	292	325	370	317	455	91.0%

were abandoned, and the runways were broken. In six images of them, the detector found airports, but they were excluded after comprehensive scoring. In the last six images, the detector did not find the airport runways. From the perspective of an intercontinental distribution, the accuracy of the results in Asia and on islands was highest, while the accuracy of the results in Oceania was relatively low. The reason why the accuracy of the results in Oceania, North America, and South America was lower than in other regions was that there was a large number of ultra-small airports in these regions. These airport runways were located in farmland areas and appeared like empty spaces. This made it difficult for the detector to distinguish these runways from surrounding fields. Although the results of the framework had errors and omissions, the overall accuracy reached 91.0%, and it only took 1.1 hours for a complete evaluation of the dataset. In contrast, the manual verification took 16 hours. For all airports around the world, manual verification is estimated to take 1543 hours, while the framework proposed in this study is estimated to take only 102.9 hours. As the data structure continually changes and hardware equipment is regularly updated, this gap will enlarge.

V. Conclusion

In this article, we proposed a process framework that validated the spatial location of airports from a fused dataset. The fused airport dataset was extracted from four open access databases and contained 46 290 airport (with runway) records. The process framework consisted of two deep learning models. The first was a detection model based on YOLOv3 for detecting airport runways on two-scales. The second was a reclassification model based on ResNet-101 for correcting the results that were misidentified by the detection model. The process framework performed well on the test dataset, with a precision of 95.8% and a recall rate of 95.8%. In addition, 1% manual sampling verification was performed on airports from the fused dataset to test the reliability of our framework. These tests showed that the accuracy of the framework was up to 91.0% and the speed was 15 times that of the manual verification. Finally, the process framework was applied to a global airport fusion dataset, of which 29 259 airport records were verified, indicating that there were airports with runways at the respective location in Google Earth.

Our research still has some limitations: (1) The airport location data only came from structured databases, unfused with unstructured text from the Internet and POIs of web maps to further expand the fused dataset. (2) The detector did not combine aircraft and other object detection model, which could have further improved the reliability of the results. (3) The re-classifier did not consider the differences in surface coverage around the airport. Improvements to address these shortcomings will be reflected in further research. In the future, we will detect and discover airports that are not included in existing datasets from the remote sensing images of globe coverage, rather than just detecting airports in images at known coordinates.

References

[1] D. Liu, L. He, and L. Carin, "Airport detection in large aerial optical imagery," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Proc.*, May 2004, vol. 5, p. no. V–761, doi: 10.1109/ICASSP.2004.1327222.

[2] Y. Wang and L. Pan, "Automatic airport recognition based on saliency detection and semantic information," *IJGI*, vol. 5, no. 7, Jul. 2016, Art. no. 115.

[3] G. Cheng *et al.*, "Object detection in remote sensing imagery using a discriminatively trained mixture model," *ISPRS J. Photogrammetry Remote Sens.*, vol. 85, pp. 32–43, Nov. 2013.

[4] Y. Chen, W. Li, C. Sakaridis, D. Dai, and L. Van Gool, "Domain adaptive faster R-CNN for object detection in the wild," 2018, pp. 3339–3348, Accessed: May 07, 2020. [Online]. Available: http://openaccess.thecvf.com/content_cvpr_2018/html/Chen_Domain_Adaptive_Faster_CVPR_2018_paper.html

[5] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Proc. Syst., vol.* 25, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds., 2012, pp. 1097–1105.

[6] K. Li, G. Wan, G. Cheng, L. Meng, and J. Han, "Object detection in optical remote sensing images: A survey and a new benchmark," *ISPRS J. Photogrammetry Remote Sens.*, vol. 159, pp. 296–307, Jan. 2020.

[7] G. Cheng and J. Han, "A survey on object detection in optical remote sensing images," *ISPRS J. Photogrammetry Remote Sens.*, vol. 117, pp. 11–28, Jul. 2016.

[8] G. Cheng, X. Lu, X. Ge, H. Yu, Y. Wang, and X. Ge, "Data fusion method for digital gazetteer," in *Proc. 18th Int. Conf. Geoinformatics*, Jun. 2010, pp. 1–4, doi: 10.1109/GEOINFORMATICS.2010.5567523.

[9] E. Acheson, J. Villette, M. Volpi, and R. S. Purves, "Gazetteer matching for natural features in Switzerland," in *Proc. 11th Workshop Geographic Inf. Retrieval*, NY, USA, 2017, pp. 1–2.

[10] P. D. Smart, C. B. Jones, and F. A. Twaroch, "Multi-source toponym data integration and mediation for a meta-gazetteer service," in *Geographic Information Science*, Berlin, Heidelberg, 2010, pp. 234–248, doi: 10.1007/978-3-642-15300-6_17.

[11] E. Acheson, M. Volpi, and R. S. Purves, "Machine learning for cross-gazetteer matching of natural features," *Int. J. Geographical Inf. Sci.*, Apr. 2019, Accessed: May 07, 2020. [Online]. Available: https://www.tandfonline.com/doi/abs/10.1080/13658816.2019.1599123

[12] D. Ahlers, "Assessment of the accuracy of geonames gazetteer data," in *Proc. 7th Workshop Geographic Inf. Retrieval*, Orlando, FL, Nov. 2013, pp. 74–81, doi: 10.1145/2533888.2533938.

[13] B. Jaggi and L. Tang, "Geographic location of the firm and credit rating accuracy," *J. Accounting, Auditing Finance*, vol. 32, no. 2, pp. 155–181, Apr. 2017.

[14] F. Zeng *et al.*, "A hierarchical airport detection method using spatial analysis and deep learning," *Remote Sens.*, vol. 11, no. 19, Jan. 2019, Art. no. 2204.

[15] Q. Luo and Z. Shi, "Airplane detection in remote sensing images based on object proposal," in *Proc. IEEE Int. Geosci. Remote Sens. Symp. (IGARSS)*, Jul. 2016, pp. 1388–1391.

[16] M. Zhu, Y. Xu, S. Ma, S. Li, H. Ma, and Y. Han, "Effective airplane detection in remote sensing images based on multilayer feature fusion and improved nonmaximal suppression algorithm," *Remote Sens.*, vol. 11, no. 9, Jan. 2019, Art. no. 1062.

[17] Z. Chen, T. Zhang, and C. Ouyang, "End-to-end airplane detection using transfer learning in remote sensing images," *Remote Sens.*, vol. 10, no. 1, Jan. 2018, Art. no. 139.

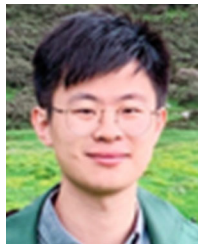
[18] Y. Xu, M. Zhu, P. Xin, S. Li, M. Qi, and S. Ma, "Rapid airplane detection in remote sensing images based on multilayer feature fusion in fully convolutional neural networks," *Sensors*, vol. 18, no. 7, Jul. 2018, Art. no. 2335.

[19] S. Bo and Y. Jing, "Region-based airplane detection in remotely sensed imagery," in *Proc. 3rd Int. Congr. Image Signal Proc.*, Oct. 2010, vol. 4, pp. 1923–1926.

[20] D. Zhao, J. Li, Z. Shi, Z. Jiang, and C. Meng, "Subjective saliency model driven by multi-cues stimulus for airport detection," *IEEE Access*, vol. 7, pp. 32118–32127, 2019.

[21] 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. Top. Appl. Earth Observ. Remote Sens.*, vol. 10, no. 4, pp. 1511–1524, Apr. 2017.

[22] T. Zhu, Y. Li, Q. Ye, H. Huo, and T. Fang, "Integrating saliency and resNet for airport detection in large-size remote sensing images," in *Proc 2nd Int. Conf. Image, Vis. Comput. (ICIVC)*, Chengdu, Jun. 2017, pp. 20–25, doi: 10.1109/ICIVC.2017.7984451.
[23] 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, Sep. 2015.
[24] 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.
[25] 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.
[26] U. Budak, U. Halici, A. Sengur, 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.
[27] Q. Guo, H. Wang, and F. Xu, "Aircraft detection in high-resolution SAR images using scattering feature information," in *Proc. 6th Asia-Pacific Conf. Synthetic Aperture Radar (APSAR)*, Nov. 2019, pp. 1–5.
[28] Q. Guo, H. Wang, L. Kang, Z. Li, and F. Xu, "Aircraft target detection from spaceborne SAR image," in *Proc. IGARSS 2019–2019 IEEE Int. Geosci. Remote Sens. Symp.*, Jul. 2019, pp. 1168–1171.
[29] 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.
[30] X.-R. Sun, W.-B. Sun, and D.-J. Li, "Detection and analysis for synthetic aperture radar image of airport," *DEStech Trans. Eng. Technol. Res.*, vol. 0, no. icicr, 2019.
[31] 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.
[32] L. Ma, Y. Liu, X. Zhang, Y. Ye, G. Yin, and B. A. Johnson, "Deep learning in remote sensing applications: A meta-analysis and review," *ISPRS J. Photogrammetry and Remote Sens.*, vol. 152, pp. 166–177, Jun. 2019, doi: 10.1016/j.isprsjprs.2019.04.015.
[33] L. Zhang, L. Zhang, and B. Du, "Deep learning for remote sensing data: A technical tutorial on the state-of-the-art," *IEEE Geosci. Remote Sens. Mag.*, vol. 4, no. 2, pp. 22–40, Jun. 2016.
[34] R. Girshick, "Fast R-CNN," 2015, *arXiv:1504.08083 [cs]*
[35] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," 2020, *arXiv:1506.01497 [cs]*.
[36] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," 2018, *arXiv:1703.06870 [cs]*.
[37] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," 2016, *arXiv:1506.02640 [cs]*.
[38] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," 2020, *arXiv:1612.08242 [cs]*.
[39] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," Apr. 2018, Accessed: May 07, 2020. [Online]. Available: https://arxiv.org/abs/1804.02767v1
[40] W. Liu *et al.*, "SSD: Single shot multibox detector," *arXiv:1512.02325 [cs]*, vol. 9905, pp. 21–37, 2016.
[41] 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.
[42] Y. Wang, C. Wang, H. Zhang, Y. Dong, and S. Wei, "Automatic ship detection based on retinanet using multi-resolution Gaofen-3 imagery," *Remote Sens.*, vol. 11, no. 5, Mar. 2019, Art. no. 531.
[43] F. Chen, R. Ren, T. 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, Mar. 2018, Art. no. 443.
[44] 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.*, pp. 1–5, 2019.
[45] Y. Xie, J. Cai, R. Bhojwani, S. Shekhar, and J. Knight, "A locally-constrained YOLO framework for detecting small and densely-distributed building footprints," *Int. J. Geographical Inf. Sci.*, pp. 1–25, Jun. 2019.
[46] Y.-L. Chang, A. Anagaw, L. Chang, Y. Wang, C.-Y. Hsiao, and W.-H. Lee, "Ship detection based on YOLOv2 for SAR imagery," *Remote Sens.*, vol. 11, no. 7, Apr. 2019, Art. no. 786.
[47] Z. Wu, X. Chen, Y. Gao, and Y. Li, "RAPID target detection in high resolution remote sensing images using yolo modEL," *Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci.*, vol. XLII–3, pp. 1915–1920, Apr. 2018.
[48] F. Xia and H. Li, "Fast detection of airports on remote sensing images with single shot multibox detector," *J. Phys., Conf. Ser.*, vol. 960, Jan. 2018, Art. no. 012024.
[49] Z. Xiao, Y. Gong, Y. Long, D. Li, X. Wang, and H. Liu, "Airport detection based on a multiscale fusion feature for optical remote sensing images," *IEEE Geosci. Remote Sens. Lett.*, vol. 14, no. 9, pp. 1469–1473, Sep. 2017.
[50] A. Van Etten, "You only look twice: Rapid multi-scale object detection in satellite imagery," 2018, *arXiv:1805.09512*.
[51] A. Van Etten, "Satellite imagery multiscale rapid detection with windowed networks," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Waikoloa Village, HI, USA, Jan. 2019, pp. 735–743.
[52] X. Yang *et al.*, "Automatic ship detection in remote sensing images from google earth of complex scenes based on multiscale rotation dense feature pyramid networks," *Remote Sens.*, vol. 10, no. 1, Jan. 2018, Art. no. 132.
[53] F. Bi, J. Hou, L. Chen, Z. Yang, and Y. Wang, "Ship detection for optical remote sensing images based on visual attention enhanced network," *Sensors*, vol. 19, no. 10, May 2019, Art. no. 2271.
[54] W. Li, C. Chen, M. Zhang, H. Li, and Q. Du, "Data augmentation for hyperspectral image classification with deep CNN," *IEEE Geosci. Remote Sens. Lett.*, vol. 16, no. 4, pp. 593–597, Apr. 2019.
[55] G. Cheng, P. Zhou, and J. Han, "Learning rotation-invariant convolutional neural networks for object detection in VHR optical remote sensing images," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 12, pp. 7405–7415, Dec. 2016.
[56] G. Cheng, J. Han, P. Zhou, and L. Guo, "Multi-class geospatial object detection and geographic image classification based on collection of part detectors," *ISPRS J. Photogrammetry Remote Sens.*, vol. 98, pp. 119–132, Dec. 2014.
[57] B. Zoph, E. D. Cubuk, G. Ghiasi, T.-Y. Lin, J. Shlens, and Q. V. Le, "Learning data augmentation strategies for object detection," 2019, *arXiv:1906.11172 [cs]*.
[58] Y. Xu, M. Zhu, S. Li, H. Feng, S. Ma, and J. Che, "End-to-end airport detection in remote sensing images combining cascade region proposal networks and multi-threshold detection networks," *Remote Sens.*, vol. 10, no. 10, Sep. 2018, Art. no. 1516.



Chen Ji received the B.Eng. degree in surveying and mapping from Nanjing Normal University, Nanjing, China, in 2013, and the M.E. degree in geographic information system from Beijing Normal University, Beijing, China, in 2016. He is currently working toward the Ph.D. degree in geography from Nanjing University, Nanjing, China.

His research interests include deep learning, GIS, and remote sensing applications.

Liang Cheng received the Ph.D. degree in photogrammetry and remote sensing from Wuhan University, Wuhan, China, in 2008.

He is currently a Professor with the Jiangsu Provincial Key Laboratory of Geographic Information Science and Technology, Nanjing University, Nanjing, China. His current research interests include integration of multisource spatial data and remote sensing image processing.

Ning Li received the B.Sc. degree in geographic information science from Chang'an University, Xi'an, China, in 2018. He is currently pursuing the M.E. degree in remote sensing applications with the Department of Geographic and Oceanographic Sciences, Nanjing University, Nanjing, China.

His research interests include GIS and remote sensing applications.

Fanxuan Zeng received the B.Eng. degree and the M.E. degree in surveying and mapping from China University of Mining & Technology, Xuzhou, China, in 2015 and 2018, respectively. He is currently pursuing the Ph.D. degree in geography from Nanjing University, Nanjing, China.

His research interests include remote sensing image processing and remote sensing applications.

Manchun Li received the Ph.D. degree in cartography from Nanjing University, Nanjing, China, in 1992.

He is currently a Professor with the Jiangsu Provincial Key Laboratory of Geographic Information Science and Technology, Nanjing University. His research interests include GIS and remote sensing applications.