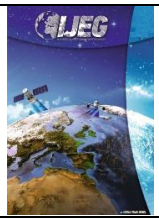
International Journal of Engineering and Geosciences
https://dergipark.org.tr/en/pub/ijeg
e-ISSN 2548-0960



Divide and conquer object detection (DACOD) method for runway detection in remote sensing images

Atakan Körez[*1]

[1]*Gazi University, Technology Faculty, Computer Engineering, Ankara, Turkey*

Keywords
Remote sensing
Runway detection
Convolutional neural networks (CNN)
Oriented object detection

ABSTRACT

In recent years, parallel to the developments in satellite technology, obtaining and processing remote sensing images has become quite common. While airports are the first points to be targeted by enemy forces in times of war, they are very critical points in times of peace due to their significance for transportation, trade, and economy networks. The runways are the most distinctive feature of airports. There are many studies on detecting the runways in remote sensing images (RSIs). However, existing methods for detecting the runway objects that have an excessive width in high-resolution (4137 x 4552 pixels and above) RSIs may be insufficient. In this study, a Divide and Conquer Object Detection (DACOD) method is proposed for the runway objects that have an excessive width in high-resolution RSIs. In the proposed method, images are divided into images of 1024 x 1024 pixels, and the runway objects in these images are detected as oriented. Then, the detection results are merged by using the angles and the final runway detection results are obtained. The experimental results demonstrate that the proposed model yields good results (%81.5 mAP). This is an 11% mAP increase when compared to the best results in The State of The Art (SOTA) object detection models using the same dataset.

1. INTRODUCTION

Obtaining and using optical remote sensing images (RSIs) has become quite common as a result of the increase in the number of satellites sent to space with the development of technology and the development of the remote viewing equipment used in satellites. Today, RSIs are used in many areas from agricultural activities to urban planning, and from disaster management to military applications (Cheng and Han, 2016).

Airports have both strategic and economic importance. Besides being the first target of the enemy forces in times of war and peace, they are critical places in terms of their locations as crossroads of transportation, trade, and economic networks. Runways are the most distinctive feature of airports. Each airport has at least one runway. Based on this fact, the most salient elements used in airport detection in RSIs are runways.

There are many studies on detecting the runways in RSIs. We can divide these studies into 2 groups according to the methods they use. The first group consists of studies based on the determination of long and wide lines parallel to each other, which are the determining geometric features of the runways (Wu et aI., 2014; Zhang et al., 2020; Li et al., 2014; Lv et al., 2018; Akbar et al., 2019). The other group consists of the object detection studies based on the classification process made according to the textural differences of the objects that make up RSIs (Tao et al., 2010; Aytekin et al., 2013; Zongur et al., 2009; Tang et al., 2015). For this reason, in the aforementioned studies, the images are divided into small pieces of varying sizes from 32 x 32 pixels to 512 x 512 pixels and runway detection is made within these pieces.

Deep learning, a sub-branch of machine learning, has gained popularity in recent years with the development of parallel programming capabilities of graphics cards and the emergence of datasets containing large amounts of data (Bengio et aI., 2016). Convolutional neural networks (CNN), which are the best known of the deep learning techniques, have been favored especially after

* **Corresponding Author**
*(atakan.korez@gazi.edu.tr) ORCID ID 0000-0003-3704-267X

Cite this article
Korez A (2022). Divide and conquer object detection (DACOD) method for runway detection in remote sensing images. International Journal of Engineering and Geosciences, 7(2), 154-160

Research Article / DOI: 10.26833/ijeg.937061
Received: 13/05/2021; Accepted: 10/06/2021

the great success of the AlexNet (Krizhevsky et al., 2012) model in the International Large Scale Visual Recognition Challenge (ILSVRC) competition held in 2012. So, there emerged many studies that detect objects in optical RSIs using CNN (Yu et al., 2020; Wu et aI., 2020; Song et al., 2021; Ju et al., 2019; Wang et al., 2019). The RSIs used in these studies usually have a high resolution such as 4000 x 4000 pixels. Because the convolution process is an expensive operation and there are many convolution processes in the early layers of CNNs, large images are divided into small pieces (usually 1024 x 1024) to avoid the computational burden. For this reason, the data sets used in these studies (such as DOTA (Xia et aI., 2018) and HRSC2016 (Liu et aI., 2017) were created to contain small object classes such as aircraft, ships, cars, buses, etc. that can remain in 1024 x 1024 parts.

Today, there is not any data set and object detection model within the scope of detecting runway objects in high-resolution and large-size (80 MB and above) RSIs using CNN in the object detection area. In this study; the data set, containing high-resolution RSIs of various airports, is created by using Google Earth and Bing Maps, and then a model that detects the runway objects that have excessive width in this data set is presented. The specific features of the data set used are given in Table 1.

When Table 1 is examined; while the average image size of the data set used is 4137 x 4552 pixels, the average size of the runway objects contained in the images is 3221 x 87 pixels. Such large images create memory insufficiency and computational burden. Accordingly, the largest image sizes that current graphics cards can process at once are calculated using State of The Art (SOTA) object detection models and are detailed in Table 2. The graphics cards used in the tests are respectively 8 GB Nvidia GTX1080 and 16 GB Nvidia Tesla V100, and the backbone of SOTA object detection models is ResNet50 (He et al., 2016).

As can be seen in Table 2, the maximum image size that the GTX 1080 graphics card with 8 GB memory can process at one time is 2300 x 2300 pixels, while the maximum image size that the V100 graphics card with 16 GB memory can process at one time is 3600 x 3600 pixels. An Out of Memory error is obtained when an operation is attempted with a larger image for both graphics cards. The data set used in the study includes 376 images larger than 2300 x 2300 pixels and 318 images larger than 3600 x 3600 pixels. For this reason, it is inevitable to obtain an Out of Memory error in any train operation to be performed without any image pre-processing.

Table 1. Data set specific features

Specific Features	
Total Image Count	398 images
Minimum Image Size (width x height)	448 x 576 pixels
Maximum Image Size (width x height)	15904 x 11328 pixels
Average Image Size (width x height)	4137 x 4552 pixels
Maximum Runway Object Width	8216 pixels
Minimum Runway Object Height	64 pixels
Grater Runway Width Count*	60 images
Average Runway Object Size (width x height)	3221 x 87 pixels

* Number of runway objects with a width greater than the width of the image it contains

So, in this study, in order to avoid memory insufficiency (Out of Memory error) and computational burden, the images are first divided into 1024 x 1024 pixels pieces. Then, runway detection is made in these images with a method that detects objects as oriented. In the last stage; the runway detections in more than one 1024 x 1024 image piece are determined as a single runway by using the rotation angles according to the origin (Figure 1). Contributions and main objectives of proposed model are as follows;

- An object detection model is introduced that merges multiple object detection results based on rotation angles,
- Using proposed model to avoid the computational burden and overcome memory problems, it is possible to detect runway objects that have excessive width by dividing them into small pieces in high-resolution remote sensing images,
- It is the first study in this field that detects the runway objects that an excessive width in high-resolution RSIs using CNN.

In the second section of this study, the proposed model is explained in detail. The third section contains the results of the experiments performed according to the different threshold and angle combinations of the proposed model. Also in this section, to demonstrate the effectiveness of the proposed model, there is a comparison of the proposed model with SOTA object detection models. The fourth and last part is the conclusion of this study.

Table 2. Relationship between input image size and graphic cards

SOTA Model	Maximum Input Size		Flops (GFlops)		Params (Millions)	
	GTX1080	Tesla V100	GTX1080	Tesla V100	GTX1080	Tesla V100
Faster RCNN	2250 x 2250	3600 x 3600	971.35	2455.3	41.12	41.12
Mask RCNN	2200 x 2200	3600 x 3600	973.08	2501.61	43.75	43.75
Cascade RCNN	2200 x 2200	3500 x 3500	954.58	2351.96	68.93	68.93
RetinaNet	2300 x 2300	3600 x 3600	1058.36	2588.09	36.1	36.1
SSD512	2000 x 2000	3100 x 3100	1435.6	3219.51	36.04	36.04

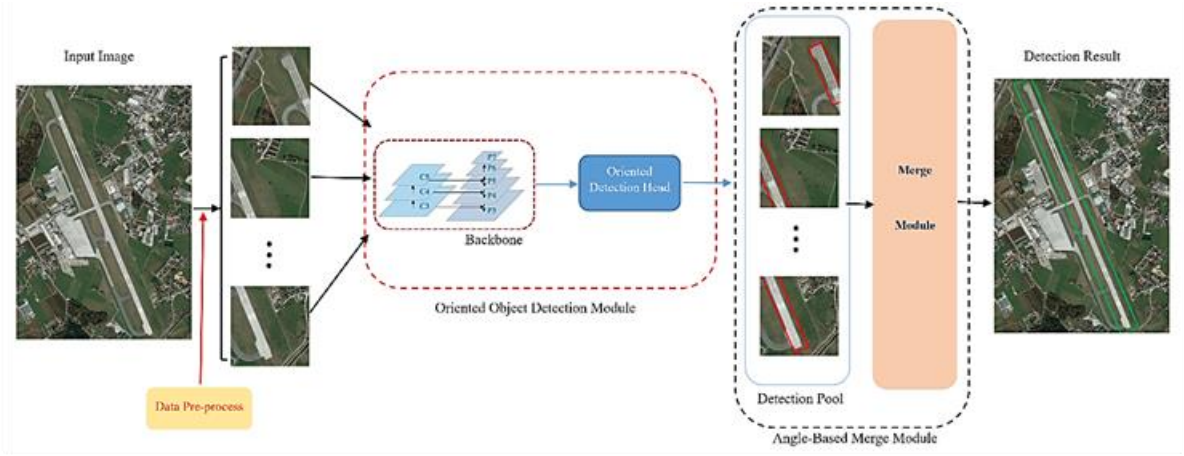


Figure 1. Overall architecture of proposed model

2. METHOD

The proposed model consists of two main components. These are; the Oriented Object Detection module using the RoI Transformer structure proposed in the study Ding (Ding et al., 2019), and the Angular-based Merge module, which angular merges the runway object detections predicted by the oriented object detection module. In this section, the mentioned modules are explained in detail.

2.1. Oriented Object Detection Module

In the detection of objects in RSI, making use of the structural properties of the objects directly affects the success. So, this module includes a backbone consisting of a Feature Pyramid Network (Tsung-Yi et al., 2017) and a ResNet50, also includes a RoI Transformer-based Oriented Detection Head. This module is designed according to training with labels suitable for oriented bounding box structure instead of standard horizontal bounding box structure. As a result of this training process, generating oriented bounding box predictions. Thus, detection results suitable for angular calculation can be transferred to the next module, Angular-based Merge.

2.2. Angular-based Merge Module

It is the module where the object detections coming from the detection module are merged according to the rotation angles and a single and complete runway detection is obtained. This module consists of 2 parts. These are the detection pool and merge module.

The detection results produced by the oriented object detection module are first collected in the detection pool. Object detections in the detection pool are delivered to the Merge module sequentially. The pool is not emptied until all of the images to be used while detecting the runway object are processed by the merge module. At the end of the final runway detection result generated by the merge module, the pool is automatically emptied. In order not to occupy more space in the GPU memory, only oriented detection results are kept in the Detection Pool.

The Merge module processes the detection results in the detection pool sequentially. The algorithm of this module is given in Algorithm 1. This algorithm consists of 3 stages.

1) Detection results in the detection pool may contain more than one bounding box. With the help of the non-maximum suppression technique, these bounding boxes are reduced to a single bounding box. The threshold value (N_t) in this process is set as 0.9, as will be mentioned in the experiments section. Obtained detection results are put into a new list (***DL***),

2) In this stage, oriented angle values of all detection results in DL are calculated at first. Then these detection results are grouped according to the angle values. In the second stage, all operations are performed over bounding boxes containing object detection. In the angle calculation process specified by (1) in Algorithm 1, the binary coordinate values (x, y) that form the bounding box are numbered starting from 0 clockwise (Figure 2),

Then, in accordance with equation 1, the angle value is calculated using the binary coordinate values of the corner points 0 and 3 of the bounding box. If the bounding box whose angle is calculated is the first, the binary coordinate value of the bounding box and the oriented angle value are added into a new group list (***GL***) and then the next bounding box is processed. The angle value of the next bounding box is calculated and compared with the angle value of the first elements of the existing group lists. The process referred to as comparison is to take the absolute value of the difference between the angles of both bounding boxes. If the result of the operation is less than 10º (in the experiments section, why this value is selected is explained), these two bounding boxes are assumed to belong to the same runway object, and the processed bounding box is added

as a new element to the relevant group list. If the angle value of the bounding box subjected to the calculation is not related to any ***GL*** (the difference is greater than 10º), this bounding box is considered to belong to a new runway object. Therefore, a new ***GL*** is created and the binary coordinate values and the oriented angle value of this bounding box are added as the first element to this new list. The comparison is done until all elements of the ***DL*** are finished.

$$\emptyset = tan^{-1}\left(\frac{y_4 - y_1}{x_4 - x_1}\right) \quad (1)$$

In Equation 1, while (*x1, y1*) binary coordinate value refers to the corner numbered 0, (*x4, y4*) binary coordinate values represent the corner numbered 3 clockwise.

3) At this stage, a single runway object is obtained by combining the bounding box values in each ***GL***. Additionally, since the dimensions of the runway objects in the images that make up the data set used within the scope of the study are larger than 1024, the number of elements of the ***GL*** to be merged should be at least 2. In the merge process specified with (2) in Algorithm 1; the binary coordinate values of the corner points 0 and 3 of the first bounding box of ***GL*** are added to a list. Then, the binary coordinate value of corner points 1 and 2 of the last element of the ***GL*** list is also added to this list, resulting in a complete bounding box. The process is completed by assigning 0.9 value selected as the threshold to the score value field of this bounding box. Bounding boxes obtained as a result of the merge process no (2) are added to the final result list (***R***). ***R*** is considered as the prediction result of proposed model

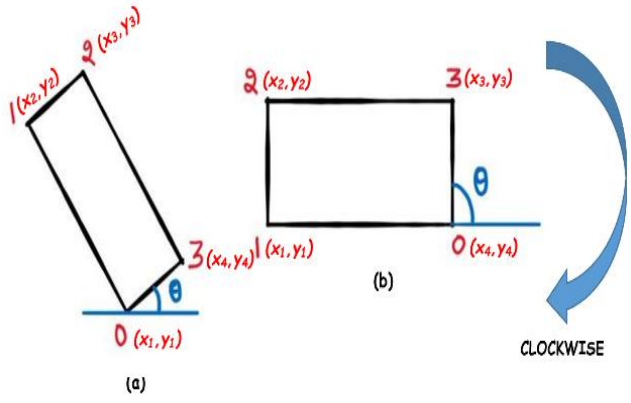


Figure 2. Bounding box numbering process. (a) Angle value less than 90º. (b) Angle value equal to 90º.

Algorithm 1 Merge Module

Input : $S=\{s_1,..,s_N\}$, $D=\{d_1,..,d_N\}$, N_t
S is list of inital detection boxes,
D is the list of detection boxes,
N_t *is the NMS threshold, where* $0 \leq N_t \leq 1$
Output : $R=\{r_1,..,r_N\}$
R is the full image runway detection results
Step1 : Reducing multiple object detections using NMS and adding them to the detection list(*DL*).
while $D \neq$ empty **do**
  if iou(d_i, s_i) $\geq N_t$ **then**
    $DL \leftarrow d_i$
end
Step2 : Grouping object detections according to oriented angles (*GL*).
j = 1, k = 1
for i in len(*DL*) **do**
  calculate angle (*DL[i]*) **(1)**
  if i = 1 **then**
    $GL_j[1] \leftarrow DL[i]$
    continue
  else
    while k ≤ j **do**
      if $|\text{angle}(GL_k[1]) - angle(DL[i])| \leq 5$ **then**
        $GL_k \leftarrow DL[i]$
        break
      else k++
    end
    if k > j **then**
      j ++
      $GL_j[1] \leftarrow DL[i]$
    end
  end
end
Step 3: Merge grouped object detections.
while j > 0 **do**
  if len(GL_j)> *1* **then**
    merge GL_j *elements* **(2)**
    $R \leftarrow GL_j$
  end
  j --
end

3. EXPERIMENTS, RESULTS AND DISCUSSION

3.1. Data Set and Evaluation Metric

Today, there are many datasets consist of remote sensing images such as DOTA and HRSC2016. However, none of these datasets contain a runway object. In this study, the data set we used is created by the Presidency of Defense Industries of The Republic of Turkey. This dataset includes high-resolution RSIs of various airports obtained from Google Earth and Bing Maps. We divide these images into small blocks with a size of 1024 × 1024 pixels, and there is a 200-pixels overlap between the adjacent small blocks. Of the 6012 images obtained after dividing, 4208 of them are used as trains and the remaining 1804 images are used for the test. Stochastic gradient descent (SGD) is chosen as the optimizer of the proposed model. The training of proposed model is completed after 24 epochs. The learning rate is initially chosen as 0.0025, it is reduced by 10% on the 16th and 22nd epochs. To evaluate the performance of proposed model, we used mean average precision (mAP) as a

benchmark that reflects the overall performance of object detection algorithms. The mAP is defined as,

$$\text{mAP} = \int_0^1 P(R)dR \quad (2)$$

where P indicates the precision rate; R suggests the recall rate.

3.2. Experiments with Different Configurations

Experiments with different configurations are carried out to measure the performance of the proposed model. The configurations we use are the nms threshold value, which has an important place in the success of the model we propose, and the angular difference value we use when comparing bounding boxes. Experimental results are given in Table 3.

Table 3. Performance comparison on threshold value and angular difference value

NMS Threshold Value	Angle Difference Value ≤ 5º	≤ 10º	≤ 15º	≤ 20º	≤ 25º
0.5	0.722*	0.728	0.743	0.732	0.717
0.6	0.743	0.750	0.754	0.738	0.730
0.7	0.771	0.768	0.773	0.746	0.739
0.8	0.787	0.801	0.798	0.761	0.752
0.9	0.794	**0.815**	0.803	0.802	0.782

* All values are *m*AP

When Table 1 is examined; it is seen that the best result is obtained with the 0.9 threshold value and configuration where the angular difference is less than 10º (0.815 mAP). The reason for this can be explained as follows,

- The long side of the runway objects in the images is often long enough to correspond to more than one 1024x1024 image piece. Moreover, their width is not as small as those of objects such as planes, cars, and ships that are frequently found in RSIs. So, they are very easy to detect. For this reason, keeping the threshold value high directly increases the success as it eliminates false detections.

- In addition to the main runways, there are smaller auxiliary runways at the airports. Since the auxiliary runways and main runways are very close to each other, when the angular difference value increases, the proposed model identifies the auxiliary runways as the main runway. According to the results of the experiments, the highest success is obtained when the angular difference is less than 10º.

3.3. Proposed Model Compared with SOTA Object Detection Models

To objectively evaluate the performance of the proposed model; SOTA object detection models are trained on the data set used in this study. According to the test results made after the training processes, the object detection performance comparison of the SOTA object detection models (Ren et al., 2014; He et al., 2017; Zhaowei et al., 2019; Lin et al., 2017; Wei et al., 2016). and the proposed model is given in Table 4. All images in the data set are downscaled to 1024 x 1024 in order not to get an Out of Memory error. The training of all models is completed after 24 epochs. The learning rate is initially chosen as 0.0025, it is reduced by 10% on the 16th and 22nd epochs.

When Table 4 is examined, it is seen that the proposed model shows the best performance by far. The proposed model performs 11% better than the RoI Transformer model it uses as an object detection module. This is because; while the downscale preprocessing process reduces the accuracy value, the proposed model can detect the objects with the help of the Angular-based Merge module without the downscale preprocessing.

As can be seen from the object detection performances; the downscale preprocessing process causes the SOTA models (especially RetinaNet) to perform poorly. However, downscale preprocessing is also a must to get rid of the Out of Memory error, which is explained in detail in Table 2. Although the inference time of the proposed model is slow, it is at a level that can be ignored according to the object detection performance it achieves (It even has a good speed compared to SSD512, RetinaNet, and Mask RCNN models). So, considering the data set used within the scope of the study, it is understood that the proposed model has a reasonable speed. Object detection results on the test images of the proposed model can be seen in Figure 3.

Table 4. Object detection performance comparison of the proposed model and SOTA models.

Model	Inference Time(FPS)	*m*AP
Faster RCNN	4.6	0.539
Mask RCNN	1.5	0.500
Cascade RCNN	3.9	0.623
RetinaNet	1.6	0.103
SSD512	1.3	0.520
RoITransformer	4.2	0.703
Proposed Model	1.7	0.815

(a) (b) (c) (d)

Figure 3. Visualization of detection results. In the images with more than one runway object, each runway's detection is shown in a different color.

4. CONCLUSION

In this study, a simple and effective object detection method is proposed to detect runway objects that have an excessive width in high-resolution remote sensing images. The proposed model aims to detect runway objects in high-resolution images based on the divide and conquer philosophy. To avoid the computational burden and overcome memory problems (especially Out of Memory error), in the proposed method, remote sensing images are first divided into 1024x1024 small images. Then, runway objects are detected as oriented, and obtained detection results are merged taking into account the angular differences. So, the predicted output of the model is obtained. Experiments show that the proposed model achieves 81.5% mAP results. Also, SOTA object detection models are trained on the data set used within the scope of the study, and the object detection performances are compared with the proposed model. According to the comparison results, the proposed model achieves an 11% mAP increase compared to the best-performing SOTA model. In the next study, improvements will be made to enable the proposed model to perform instance segmentation with object detection jointly.

ACKNOWLEDMENT

I would like to thank the Presidency of Defence Industries of The Republic of Turkey for providing the dataset we use in this study.

Conflicts of interest

The authors declare no conflicts of interest.

REFERENCES

Akbar J, Shahzad M, Malik MI, Ul-Hasan A & Shafai F (2019). Runway Detection and Localization in Aerial Images using Deep Learning. Digital Image Computing: Techniques and Applications (DICTA), 1 - 8, Perth,Australia.

Aytekin Ö, Zöngür U & Halici U (2013). Texture-Based Airport Runway Detection. IEEE Geoscience and Remote Sensing Letters, 10-(3), 471-475.

Bengio Y, Goodfellow I & Courville A (2016). Deep Learning. MIT Press. ISBN: 978-0262035613.

Cheng G & Han J (2016). A Survey on Object Detection in Optical Remote Sensing Images. ISPRS Journal of Photogrammetry and Remote Sensing ,117, 11-28.

Ding J, Xue N, Long Y, Xia G & Lu Q (2019). Learning RoI Transformer for Oriented Object Detection in Aerial Images. IEEE Conference Computer Vision and Pattern Recognition (CVPR), 2844 -2853, Seul, Korea.

He K, Zhang X, Ren S & Sun J (2016). Deep Residual Learning for Image Recognition. IEEE Conference Computer Vision and Pattern Recognition (CVPR), 770 -778, Nevada, USA.

He K, Gkioxari G, Dollar P & Girshick G (2017). Mask R-CNN. IEEE International Conference on Computer Vision (ICCV), 324-333, Venice, Italy.

Ju M, Luo J, Zhang P, He M & Luo H (2019). A Simple and Efficient Network for Small Target Detection. IEEE Access, 7, 85771-85781.

Li Z, Liu Z & Shi W (2014). Semiautomatic Airport Runway Extraction Using a Line-Finder-Aided Level Set Evolution. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 7(12), 4738-4749.

Lin T Y, Goyal P, Girshick R, He K & Dollar P (2017). Focal loss for dense object detection. IEEE Transactions on Pattern Analysis and Machine Intelligence, 99, 2999-3007.

Liu Z, Yuan L, Weng L & Yang Y A (2017). High Resolution Optical Satellite Image Dataset for Ship Recognition and Some New Baselines. 6th International Conference on Pattern Recognition Applications and Methods, 324 – 333, Porto, Portugal.

Lv W, Dai K, Wu L, Yang X & Xu W (2018). Runway Detection in SAR Images Based on Fusion Sparse Representation and Semantic Spatial Matching. IEEE Access, 6, 27984-27992.

Krizhevsky A, Sutskever I & Hinton G E (2012). ImageNet classification with deep convolutional neural networks. Advances in Neural Information Processing Systems, 25, 1097–1105.

Ren S, He K, Girshick R & Sun J (2015). Faster R-CNN: Towards real-time object detection with region proposal networks. IEEE Transactions on Pattern Analysis and Machine Intelligence, 39-(6), 91–99.

Song Q, Yang F, Yang L, Liu C, Hu M & Xia L (2021). Learning Point-Guided Localization for Detection in Remote Sensing Images. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 14, 1084-1094.

Tang G, Xiao Z, Liu Q & Liu H (2015). A Novel Airport Detection Method via Line Segment Classification and Texture Classification. IEEE Geoscience and Remote Sensing Letters, 12 (12), 2408-2412.

Tao C, Tan Y, Cai H & Tian J (2010). "Airport Detection from Large IKONOS Images Using Clustered SIFT Keypoints and Region Information. IEEE Geoscience and Remote Sensing Letters, 8-(1), 128-132.

Tsung-Yi L, Dollar P, He K, Hariharan B & Belongie S (2017). Feature Pyramid Networks. IEEE Conference Computer Vision and Pattern Recognition (CVPR), 936-944, Hi, USA.

Wang Y, Zhang Y, Zhao L, Sun X & Guo Z (2019). SARD: Towards Scale-Aware Rotated Object Detection in Aerial Imagery. IEEE Access, 7, 173855-173865.

Wei L, Anguelov D, Erhan D, Szegedy C, Reed S, Fu S Y & Berg A (2016). SSD: Single Shot MultiBox Detector. The 14th European Conference on Computer Vision (ECCV), 21 -37, Amsterdam, Holland.

Wu W, Xia R, Xiang W, Hui B, Chang Z, Liu Y & Zhang Y (2014). Recognition of Airport Runways in FLIR Images Based on Knowledge. IEEE Geoscience and Remote Sensing Letters, 11, 1534-1538.

Wu Y, Zhang K, Wang J, Wang Y, Wang Q & Li Q (2020). CDD-Net: A Context-Driven Detection Network for Multiclass Object Detection. IEEE Geoscience and Remote Sensing Letters, 3, 1-4.

Xia GS, Bai X, Ding J, Zhu Z, Belongie S, Luo J, Datcu M, Pelillo M & Zhangi L (2018). DOTA: A large-scale dataset for object detection in aerial images. IEEE Conference Computer Vision and Pattern Recognition (CVPR), 3974-3983, Utah, USA.

Yu Y, Yang X, Li J & Gao X (2020). A Cascade Rotated Anchor-Aided Detector for Ship Detection in Remote Sensing Images. IEEE Transactions on Geoscience and Remote Sensing, 2, 1-14.

Zhang Z, Zou C, Han P & Lu X (2020). A Runway Detection Method Based on Classification Using Optimized Polarimetric Features and HOG Features for PolSAR Images. IEEE Access, 8, 49160-49168.

Zhaowei C & Vasconcelos N (2019). Cascade R-CNN: High Quality Object Detection and Instance Segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence, 43-(5), 1483 – 1498.

Zöngür U, Halici U, Aytekin O & Ulusoy I (2009). Airport runway detection in satellite images by Adaboost learning. SPIE Image and Signal Processing for Remote Sensing XV, 1-12, Berlin, Germany.