



A benchmark dataset for deep learning-based airplane detection: HRPlanes

Tolga Bakirman ^{*1} , Elif Sertel ² 

¹Yildiz Technical University, Geomatics Engineering Department, Türkiye

²Istanbul Technical University, Geomatics Engineering Department, Türkiye

Keywords

Airplane detection
Deep learning
YOLO
Faster R-CNN
Google Earth

Research Article

DOI: 10.26833/ijeg.1107890

Received:23.04.2022

Revised:09.11.2022

Accepted:01.12.2022

Published:08.05.2023

Abstract

Airplane detection from satellite imagery is a challenging task due to the complex backgrounds in the images and differences in data acquisition conditions caused by the sensor geometry and atmospheric effects. Deep learning methods provide reliable and accurate solutions for automatic detection of airplanes; however, huge amount of training data is required to obtain promising results. In this study, we create a novel airplane detection dataset called High Resolution Planes (HRPlanes) by using images from Google Earth (GE) and labeling the bounding box of each plane on the images. HRPlanes include GE images of several different airports across the world to represent a variety of landscape, seasonal and satellite geometry conditions obtained from different satellites. We evaluated our dataset with two widely used object detection methods namely YOLOv4 and Faster R-CNN. Our preliminary results show that the proposed dataset can be a valuable data source and benchmark data set for future applications. Moreover, proposed architectures and results of this study could be used for transfer learning of different datasets and models for airplane detection.



1. Introduction

The rapid technological advancements in remote sensing systems have significantly improved [1] the availability of very high-resolution remote sensing imagery [2] to be used for the detection of geospatial objects such as airplanes, ships, buildings, etc. [3]. Airplane detection is essential in various fields such as airport surveillance [4], transportation activity analysis [5], defence and military applications and satellite imagery is a significant data source for this purpose with the advantages of covering large areas very quickly and periodically [6].

Airplane detection studies from earlier years are generally based on template matching and machine learning. For example, Liu et al. [7] and Xu and Duan [8] have utilized deformable templates for airplane detection. Although this method is flexible and outperforms rigid shape matching, it still needs various types of information for template design [9]. Compared to template matching, machine learning methods have

been used more widely for this purpose. Various feature extraction methods and classifiers are investigated in the literature. Sun et al. [10] utilized a spatial sparse coding bag of words (BOW) model combined with a linear support vector machine. This model uses sliding windows to extract features and employs a spatial mapping strategy to encode geometric information. Zhang et al. [11] proposed a rotation invariant histogram of oriented gradient (HOG) features for the detection of complex objects in high resolution imagery. They also improve their method further by using a generic discriminative part-based model later on [12]. Lei et al. [13] proposed a novel color-enhanced rotation-invariant Hough Forest to train a Pose-Estimation-based Rotation-invariant Texton Forest. Liu and Shi [14] investigated the airplane feature possessing rotation invariant that combined with sparse coding and radial gradient transform. Machine learning methods require manually extracted features and thus, their performance is heavily depending on selecting accurate hand-crafted features

* Corresponding Author

(bakirman@yildiz.edu.tr) ORCID ID 0000-0001-7828-9666
(sertele@itu.edu.tr) ORCID ID 0000-0003-4854-494X

Cite this article

Bakirman, T., & Sertel, E. (2023). A benchmark dataset for deep learning-based airplane detection: HRPlanes. International Journal of Engineering and Geosciences, 8(3), 212-223

[15]. Deep learning approaches offer end-to-end solutions using automatic feature extraction.

Recent studies illustrate that deep learning-based airplane detection methods not only outperform conventional object detection algorithms but also provide feasible solutions. Chen et al. [16] combined classification and localization processes for better optimization using transfer learning. Xu et al. [17] proposed a multilayer feature fusion process that fuses the shallow and deep layer features in fully convolutional neural networks (FCN). Zhu et al. [18] utilized the L2 norm normalization, feature connection, scale scaling, and feature dimension reduction for more efficient fusion of low- and high-level features. Alganci et al. [6] assessed different deep learning approaches namely Faster Regional Convolutional Neural Network (Faster R-CNN), Single Shot Multi-box Detector (SSD) and You Only Look Once Version 3 (YOLOv3) for airplane detection from very high resolution satellite imagery. Wu et al. [19] proposed Weakly Supervised Learning in AlexNet which requires only image-level labelled data contrary to other object detection methods. Zhou et al. [20] introduced Multiscale Detection Network to detect small-scale aircrafts in a multiscale detection architecture manner. Ji et al. [21] developed a Faster R-CNN based model that combines multi-angle features driven and majority voting strategy. Shi et al. [22] introduced Deconvolution operation with a Position Attention mechanism (DPANet) that captures the external structural feature representation of aircraft during the feature map generation process. Wu et al. [23] proposed a self-calibrated Mask R-CNN model that performs object recognition and segmentation in parallel. Zeng et al. [24] utilize a top-down approach for aircraft detection in large scenes. Once the airport area is extracted with U-Net, Faster-RCNN with a feature enhancement module is applied for target detection. Chen et al. [25] have proposed two-stage aircraft detection network. The first stage creates region proposals using a circular intensity filter and the second stage detects targets by using a combination of rotation-invariant histogram of oriented gradient and vector of locally aggregated descriptors.

In order to train deep neural networks, huge amounts of images and corresponding labels are needed. Researchers have proposed datasets including airplanes for this purpose. Xia et al. [26] introduced the DOTA dataset with 15 classes including airplanes using imagery from Google Earth, Jilin-1 and Gaofen-2 satellites. This dataset has been then expanded, improved, and renamed as the iSAID dataset [27]. Lam et al. [28] utilized multi-source imagery to generate the xView dataset which has passenger/cargo planes and 59 other classes. More recently, Shermeyer et al. [29] took the advantage of synthetic data to create RarePlanes dataset. The dataset consists of 253 WorldView-3 real and 50,000 synthetic imageries, and corresponding 14,700 hand-annotated and 630,000 simulated plane labels, respectively.

In this study, we create a huge novel dataset solely for airplanes from very high-resolution Google Earth imagery using only hand-annotated labels. Our dataset, HRPlanes, includes images obtained from the biggest airports across the world to represent a variety of landscape, seasonal, and data acquisition geometry

conditions. We also evaluate our dataset using state-of-the-art deep neural networks namely YOLOv4 and Faster R-CNN to analyze the performance of two different object detection algorithms.

2. Dataset

The imagery required for the dataset has been obtained from Google Earth. We have downloaded 4800 x 2703 sized 3092 RGB images from the biggest airports of the world such as Paris-Charles de Gaulle, John F. Kennedy, Frankfurt, Istanbul, Madrid, Dallas, Las Vegas and Amsterdam Airports, and aircraft boneyards like Davis-Monthan Air Force Base.

Dataset images were annotated manually by creating bounding boxes for each airplane using HyperLabel software [30]. Quality control of each label was conducted by visual inspection of independent analysts who were not included in the labelling procedure. A total of 18,477 airplanes have been labelled. A sample image and corresponding minimum boxes for airplanes can be seen in Figure 1.

The dataset has been approximately split as 70% (2166 images), 20% (615 images) and 10% (311 images) for training, validation and testing, respectively.

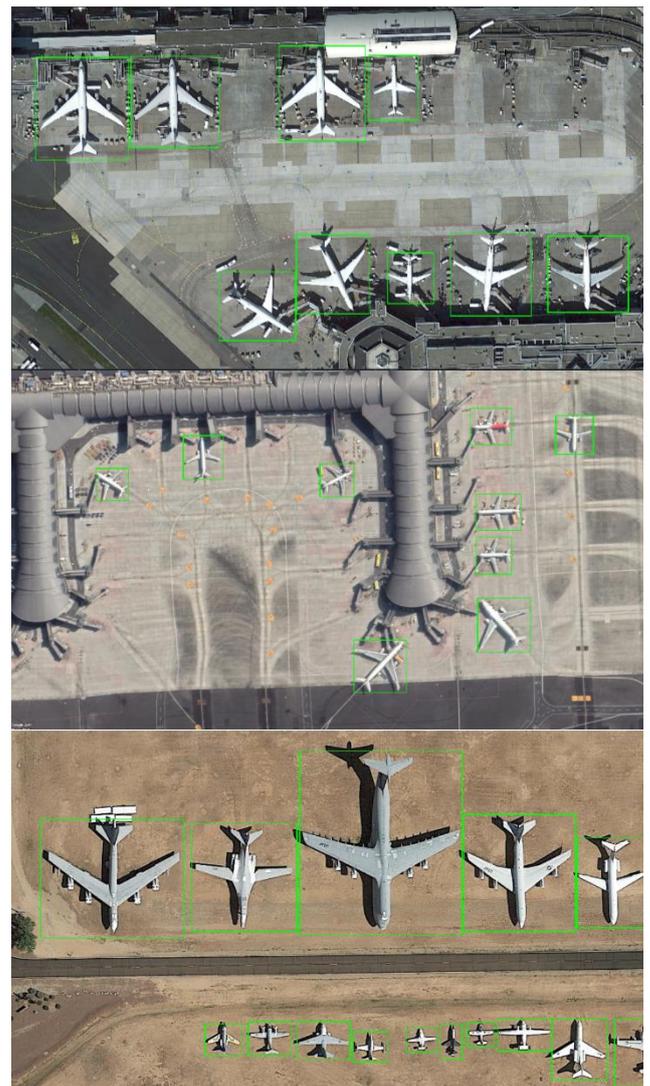


Figure 1. Sample images from the HRPlanes dataset

3. Methods

The proposed dataset has been evaluated using state-of-the-art YOLOv4 and Faster R-CNN methods. Both methods have been widely utilized in various object detection applications in the literature.

3.1. YOLOv4

The first version of YOLO is proposed by [31], which is highlighted to be extremely fast. The network which runs on Darknet framework is a global model since it uses features from the whole image to predict bounding boxes. Therefore, the model is able to learn general representations of the objects. The input image is divided by an $S \times S$ grid. The grid that covers the center of an object is responsible for detecting the object. The bounding boxes are predicted with a confidence score which is calculated by: $\text{Pr}(\text{object}) \times \text{IoU}_{GT}$. YOLO has a GoogLeNet [32] based architecture which has 24 convolutional and 2 fully connected layers. The network does not utilize the inception module, instead it uses a 1×1 reduction layer with a 3×3 convolutional layer. While the linear activation function is used in the final layer, leaky rectified linear activation is exploited in all other layers. Even though YOLO is fast and sensitive to false positives, it still has problems with localization and recall.

YOLOv2 [33] aims to improve the shortcomings of the first version of YOLO by simplifying the network with better representations. The first modification is adding batch normalization which also allows to removing dropouts from the model. YOLOv2's classification network is trained on 448×448 resolution, instead of 224×224 . The network is then shrunk to 416×416 in order to obtain locations in the feature map as an odd number. The fully connected layers are removed in YOLOv2 and anchor boxes are added using dimension clusters to predict bounding boxes. Anchor boxes require manually determined box dimensions. To overcome this problem, k-means clustering is used to determine the bounding box priors. Even though the use of anchor boxes results in slight decrease in accuracy, an intermediate increase is obtained in the recall. The most significant improvement in YOLOv2 is the new backbone. Instead of using GoogLeNet based architecture, Darknet-19 is proposed which consists of 19 convolutional and 5 maximum pooling layers.

YOLOv3 [34] uses logistic regression to calculate the objectness score for each predicted bounding box. The bounding boxes are predicted at three different scales by extracting features in all scales using a modified feature pyramid network [35]. In order to extract more semantic information, features from the earlier layers of the network are concatenated with up-sampled features from layers at the later stages. The feature extraction network Darknet-19 is more extended with 3×3 and 1×1 convolutional layers and skip connections. The improved version has 53 convolutional layers and is referred as Darknet-53. Although YOLOv3 performs better with small objects compared to older YOLO versions, this is not the case for medium and large objects.

YOLOv4 [36] aims to reach the optimum balance between resolution, layers and parameters in order to obtain accurate results rapidly. The Darknet-53 backbone network is improved with Cross Stage Partial (CSP) module [37] which is called CSPDarknet-53 which contains 29 convolutional layers 3×3 , a 725×725 receptive field and 27.6 M parameters. With high number of convolutional layers, a larger receptive field allows the network to obtain a better detector. In this scenario, the feature map of the base layer is partitioned and merged using a cross-stage hierarchy which provides more gradient flow. Additionally, Spatial pyramid pooling (SPP) [38] is integrated into the backbone of the network (CSPDarknet53 with Mish activation) in order to increase the receptive field even further. This helps to extract the main characteristic features without slowing the network. Path aggregation network (PANet) with Leaky activation is used for feature extraction instead of the FPN used in YOLOv3. Additionally, some CNN components have been integrated into the backbone and detector in order to improve the network further such as dropblock regularization, cross mini-batch normalization, CutMix and Mosaic data augmentation [36]. Whilst the authors select optimal hyper-parameters while applying genetic algorithms, they also provide some additional improvements to the network such as Self-Adversarial Training (SAT), modified spatial attention module, modified PAN, and Cross mini-Batch Normalization (CmBN). SAT is a data augmentation technique that consists of two forward-backward stages. In this way, the original image is altered to create the deception. Then, the network is trained using the modified images. SAM is modified by switching from spatial-wise attention to pointwise attention and the shortcut connection of PAN is replaced with concatenation. CmBN is a modified version of Cross-Iteration Batch Normalization that gathers information in mini-batches within a single batch.

3.2. Faster R-CNN

Regions with CNN features (R-CNN) [39] combines region proposal with high-capacity CNNs that allows for bottom-up region proposals for better localization and performance. R-CNN consists of three parts. The first part creates region proposals using selective search which are not dependent on the class. The first part of the network creates around 2000 region proposals. The second part performs feature extraction using Caffe [40] with shared parameters for all classes. Since proposed regions can be in any size, they are dilated and warped to 227×227 . Caffe network extracts a fixed size low-dimensional feature vector for each proposed region using five convolutional and two fully connected layers. The final part of the network scores each extracted feature vector utilizing class-specific linear SVMs. All regions with scores are then analyzed with non-maximum suppression for each class to obtain the best region proposal with the highest IoU.

Feature extraction for each region proposal rather than a whole image increases computational cost and storage space. In order to overcome drawbacks, improve speed and increase the accuracy of the multi-stage

structured R-CNN method, a single-stage Fast-RCNN method that jointly learns the classification of proposal with refined localization is proposed [41]. The Fast R-CNN network processes the whole image to create a feature map. A region of interest (RoI) pooling layer similar to SPPnet [38] takes the feature map and extracts a fixed-size feature vector using maximum pooling applied on each channel. Softmax probability estimates and bounding box positions are then produced by feeding each feature vector into a series of fully connected layers. During the training of Fast R-CNN, stochastic gradient descent mini-batches are sampled hierarchically. Computation and memory costs are shared for RoIs created in the same images during forward-backward passes. Additionally, the network jointly optimizes the classifier and the regressor in a single stage.

Faster R-CNN [42] is based on a region proposal network (RPN). The detector in the network works with the rectangular object proposals by the RPN. Object proposals are then used by Fast R-CNN for detection. RPN also shares created features with Fast R-CNN thus, it does not increase computational cost. RPN also utilizes attention mechanisms that direct the Fast R-CNN detection network where to look. The anchors are created for each location as translation invariant and multi-scale. A multi-task loss is calculated by considering log loss and regression loss. Log loss is the classification loss between classes. For a single class airplane detection network, the classification loss calculates loss over airplane versus not plane. The regression loss is calculated once the anchor contains the desired object. Note that an object will have a defined maximum number of anchors. However, the anchor with the minimum loss will be attained as detection [42].

3.3. Accuracy Metrics

The results are assessed using Microsoft COCO [43] evaluation metrics, which consist of various Average Precision (AP) and Average Recall (AR) values. In this study, we have used the first 3 evaluation metrics of Microsoft COCO to assess test results. These are namely mean average precision (mAP), mAP at 50% Intersection over Union (IoU), and mAP at 75% IoU. AP is a summarization metric derived from the precision-recall curve. It is calculated by the weighted mean of precision values for different recall threshold values varying from 0 to 1 (Equation 1):

$$AP = \sum_n (R_n - R_{n-1}) \times P_n \quad (1)$$

True Positives (TPs) are determined by IoU thresholds. For example, an IoU threshold of 50% means that the predicted bounding box will be counted as TP once it has equal to or greater than the IoU value of 50% with the ground truth, and AP is calculated based on this assumption which is the AP at 50% IoU threshold and referred as PASCAL VOC metric [44-45]. It is the same case for the AP at the 75% IoU threshold, which is stricter. The mAP is calculated by averaging APs calculated for 10 IoU thresholds from 50% to 95% with a

0.05 increment for all classes. Since we have only one class in this study, mAP and AP values are identical.

4. Results and Discussion

Experiments were conducted in an Intel Core i9-9900K 3.6 GHz CPU and an NVIDIA GeForce RTX 2080 Ti GPU. The training process for YOLOv4 was carried out in the Darknet framework. We conducted several initial experiments to find out the best hyperparameter configuration. Our results demonstrated that increasing batch size and subdivision affect the performance positively. This is similar for input image size as well. Higher image sizes provide better performance; however, computing load also increased for bigger image sizes, which is directly limited by the GPU. After manual hyper-tuning experiments, the best configuration for our hardware was found as; the input image of 416 x 416 pixels for the network size, 64 as the batch size, and 32 for the subdivision value. The learning rate, decay and momentum were input as 0.001, 0.0005 and 0.949, respectively. Complete-IoU loss was used as the loss function. YOLOv4 allows to apply various user-defined augmentation techniques namely mosaic, cut-mix, mix-up and blur. We have applied these augmentation methods however our results show that using only mosaic augmentation improved the results considerably.

Faster R-CNN network was trained on TensorFlow Object Detection API. The input image size was 1024 x 600. Momentum optimizer were utilized with a 0.0001 learning rate. We have used the random horizontal flip method for data augmentation. Both networks were trained using pre-trained weights from the MS COCO dataset. As explained for YOLOv4 experiments, Faster R-CNN hyperparameters for training have been determined empirically. YOLOv4 training took approximately 12 hours to complete while the Faster R-CNN training process was around 10 hours.

Deep neural networks have been evaluated using the same test dataset for two different models. The evaluation results are given in Table 1. The evaluation results show that both networks perform well up to 75% IoU threshold; the mAP value of YOLOv4 is 73.02%; whereas Faster R-CNN provided slightly better performance with 76.40%. Although YOLOv4 produces very high mAP at 50% IoU value of 99.15%, this value reduces with increasing IoU values and decreases to 91.82% for mAP at 75% IoU. YOLOv4 seems superior considering 50% and 75% IoU threshold. The decrease rate of AP with increasing IoU is higher for YOLOv4 compared to Faster R-CNN. This indicates that YOLOv4 cannot perform efficiently in higher IoU threshold levels higher than 80% in our dataset.

Table 1. Evaluation results based on average precision

	YOLOv4	Faster R-CNN
mAP	73.02%	76.40%
mAP@IoU=50%	99.15%	96.80%
mAP@IoU=75%	91.82%	90.00%

We present some figures to illustrate the obtained results with two different models; where purple boxes

represent YOLOv4 and green boxes represent Faster R-CNN results. Since images are collected from different satellites, the real-world coverage of the images is different with respect to spatial resolution while represented with 4800 x 2703 pixels. For very high-resolution images, the airplanes can be seen as very big objects within the scene with respect to image patch size. We will be using the term large scale for these examples such as Figure 2a, 2b, 2c, and 2d. For high resolution images, the image patch is covering a bigger area in which we will be using the term small scale such as Figure 2e and 2f. The assessment of Figure 2 including samples from Amsterdam Schiphol Airport shows that

airplanes represented with bigger object boxes could be clearly identified with both YOLOv4 and Faster R-CNN architectures as can be seen in Figure 2a, 2b, 2c and 2d. However, boundaries of bounding boxes for YOLOv4 seem better at this scale specifically in Figure 2d compared to Faster R-CNN of Figure 2c, in which the tails of some airplanes are not included within the bounding boxes. We have a small-scale image example in Figure 2e and 2f, in which there are also airplanes of different sizes available. Faster R-CNN (Figure 2e) produces better results compared to YOLOv4 specifically for small airplanes.



Figure 2. Prediction results of Faster R-CNN (green) and YOLOv4 (purple) from Amsterdam Schiphol Airport

Figure 3 shows prediction results for a small-scale imagery from Istanbul Sabiha Gokcen Airport. On the contrary to Figure 2f, YOLOv4 seems to be performing better in this example (Figure 3b). Both architectures have also detected the small propeller aircraft which is located in the upper left of the image (Figure 3a and 3b). Figure 3c and 3d present predictions for commercial planes on large scale in the same airport. In this example, YOLOv4 seems to create better bounding boxes (Figure 3c) than Faster R-CNN (Figure 3d). This may be resulted due to boarding bridges near the planes.

Our HRPlanes dataset consists of imageries from airports around different parts of the world. Some prediction results from Chengdu Shuangliu International Airport are shown in Figure 4. Both methods performed sufficiently with similar bounding boxes under clear (Figure 4a and 4b) and hazy (Figure 4c and 4d) atmospheric conditions.

Military plane samples from Davis Monthan Air Force Base are shown in Figure 5. Both prediction results seem similar for fully visible airplanes in terms of both detection and bounding boxes as can be seen on the

center part of the image patch in Figures 5a and 5b; whereas, only YOLOv4 is able to detect tails of aircrafts on the southern part of the image patch (Figure 5b). Faster R-CNN could not able to capture parts of the aircrafts in this example. In another military planes example (Figure 5c and 5d), both architectures performed well and successfully detected seven military planes within the image patch. However, in some rare

cases Faster R-CNN does not seem to create accurate bounding boxes for military planes and generate bounding boxes representing almost half of the airplanes (Figure 5e). Bounding boxes of four out of six airplanes are not completely generated. YOLOv4 could detect all six airplanes and generate bounding boxes for all of them for the image patch (Figure 5f).



Figure 3. Prediction results of Faster R-CNN (green) and YOLOv4 (purple) from Istanbul Sabiha Gokcen Airport

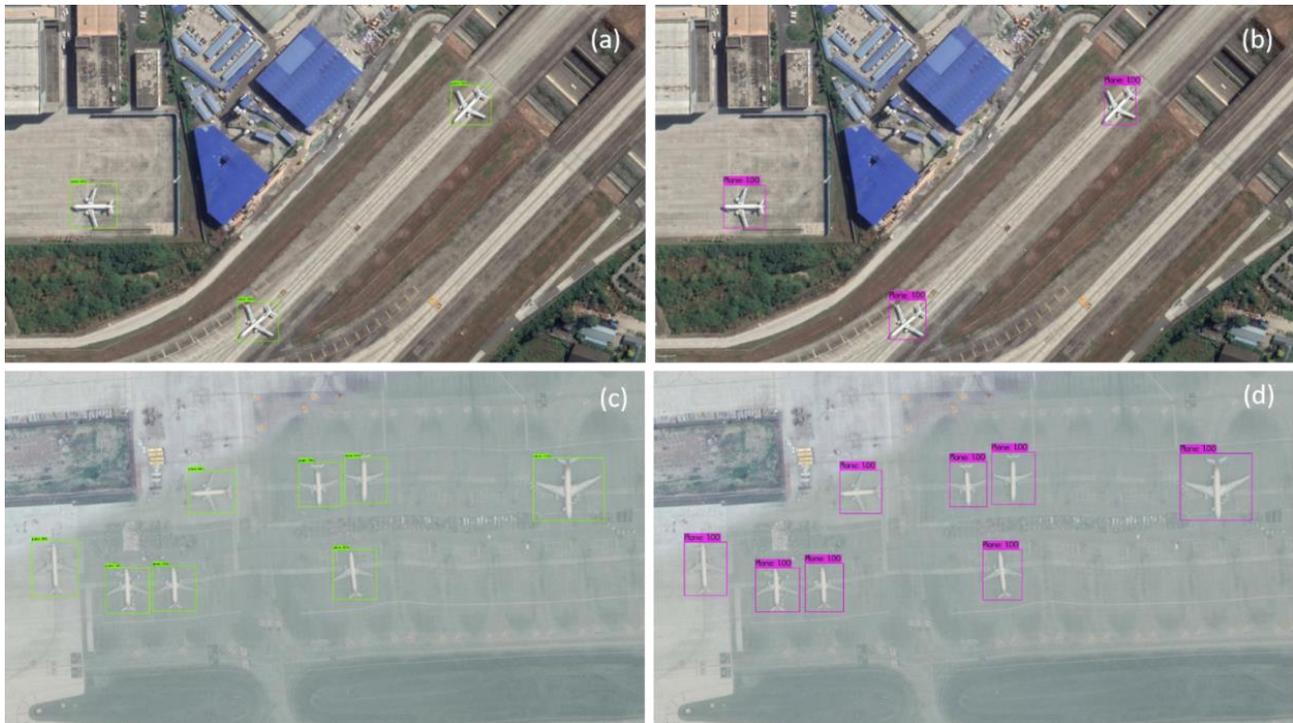


Figure 4. Prediction results of YOLOv4 (purple) and Faster R-CNN (green) from Chengdu Shuangliu International

Some small-scale examples which mostly consist of small propeller aircrafts are given in Figure 6. Faster R-CNN seems to have significantly performed better in

these imageries considering YOLOv4 have created various False Positives (FP) in different parts of the image (Figure 6b, 6d, and 6f). Some FP detections have

lower confidence levels, which could be eliminated by increasing the confidence level to 0.50 such as cases in [Figure 6b](#) to improve the results of YOLOv4.

We show a plane graveyard example in [Figure 7](#). Since the background is not complex and the planes are well

aligned, both methods have performed efficiently in both examples. Even in small scale images, YOLOv4 was able to detect plane even though only their nose cones are visible which can be seen in upper right corner of [Figure 7b](#).

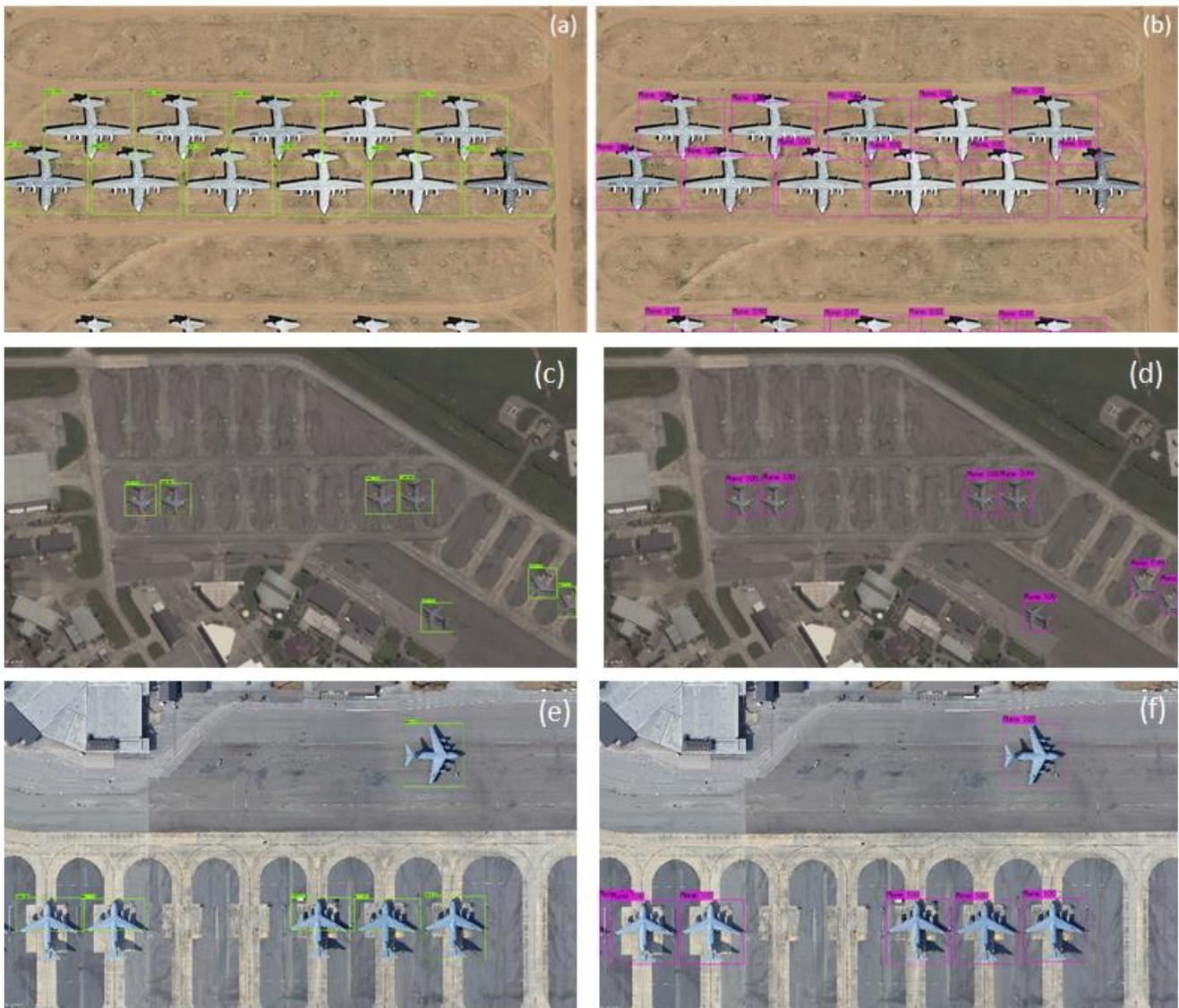


Figure 5. Prediction results of Faster R-CNN (green) and YOLOv4 (purple) from Davis Monthan for military planes

According to the prediction results, partly absent airplanes are generally detected by YOLOv4 ([Figure 5b](#) & [Figure 7b](#)). Only in some rare cases, Faster R-CNN is also able to detect them ([Figure 8a](#)). Additionally, it can be said that both methods can successfully identify airplanes in crowded scenes thanks to non-max suppression technique, even though Faster-RCNN has skipped one plane in [Figure 8c](#). Both networks were even able to detect moving planes, which has a motion blur effect in the image ([Figure 8e](#) and [8f](#)).

Compared to similar studies in the literature, our results show promising results. Chen et al. [16] have also created a plane detection dataset from Google Earth images and obtained 96.23% mAP at 50% IoU with their proposed network. Zhou et al. [20] have also used a

similar dataset that consists of Google Earth and Tianditu images. They have achieved a mAP at 50% IoU value of 90.66% using the proposed deep and wider module. Ji et al. [21] trained a modified Faster R-CNN network with two datasets (RSOD and DIOR) that consist of Google Earth imagery and the calculated mAP at 50% IoU values are 94.82% and 95.25%. Shi et al. [22] have utilized DOTA and DIOR datasets with the developed DPANet and obtained mAP values of 89.95% and 66.21%, respectively. Wu et al. [23] obtained a mAP at 50% IoU value of 96.80% using an improved version of DOTA with their improved Mask R-CNN network. As can be seen from the accuracy values obtained in the other recent studies in the literature, our results are also encouraging for plane detection from high resolution imagery.

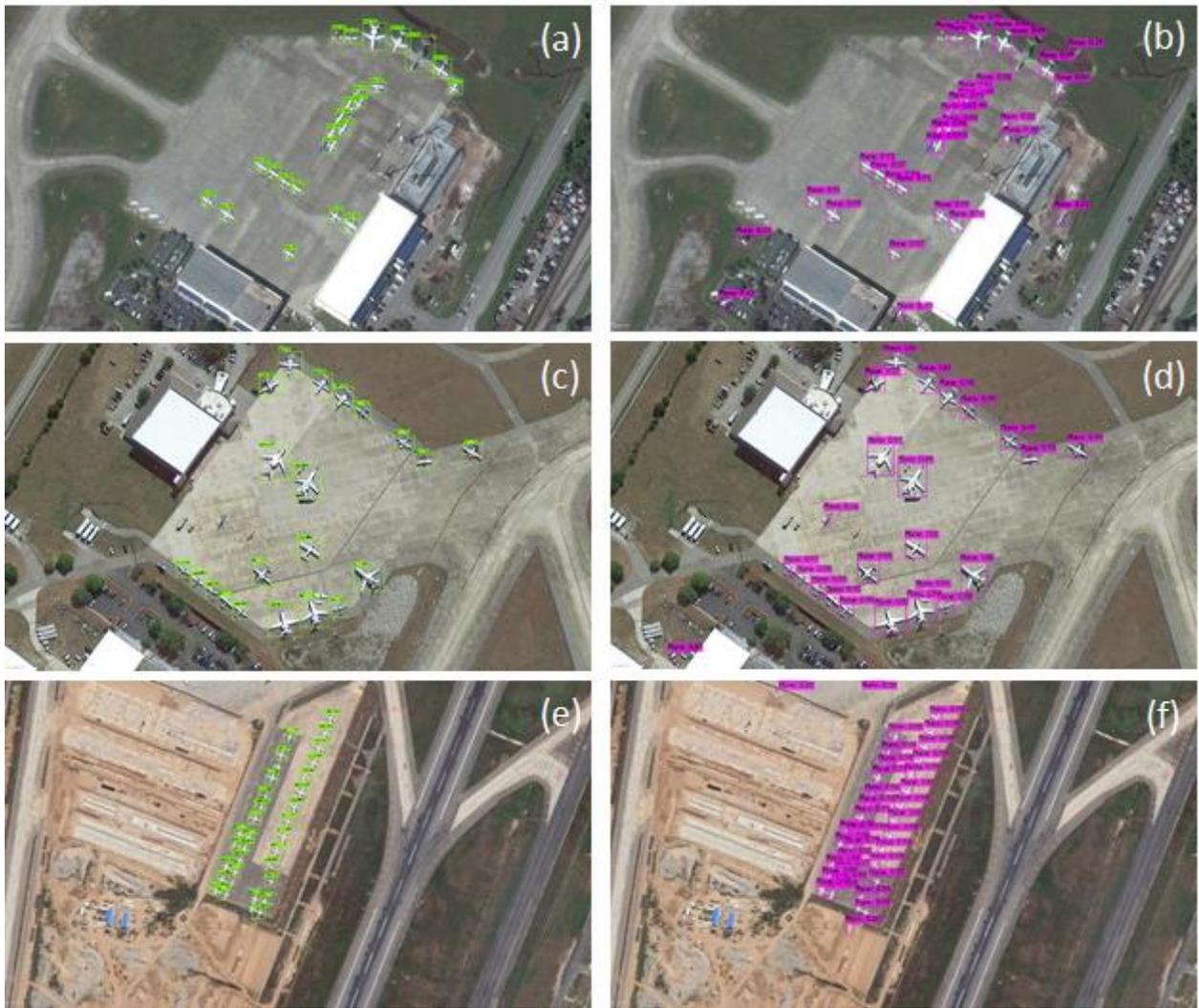


Figure 6. Prediction results of Faster R-CNN (green) and YOLOv4 (purple) from a small-scale image with small propeller aircrafts



Figure 7. Prediction results of Faster R-CNN (green) and YOLOv4 (purple) from a plane graveyard

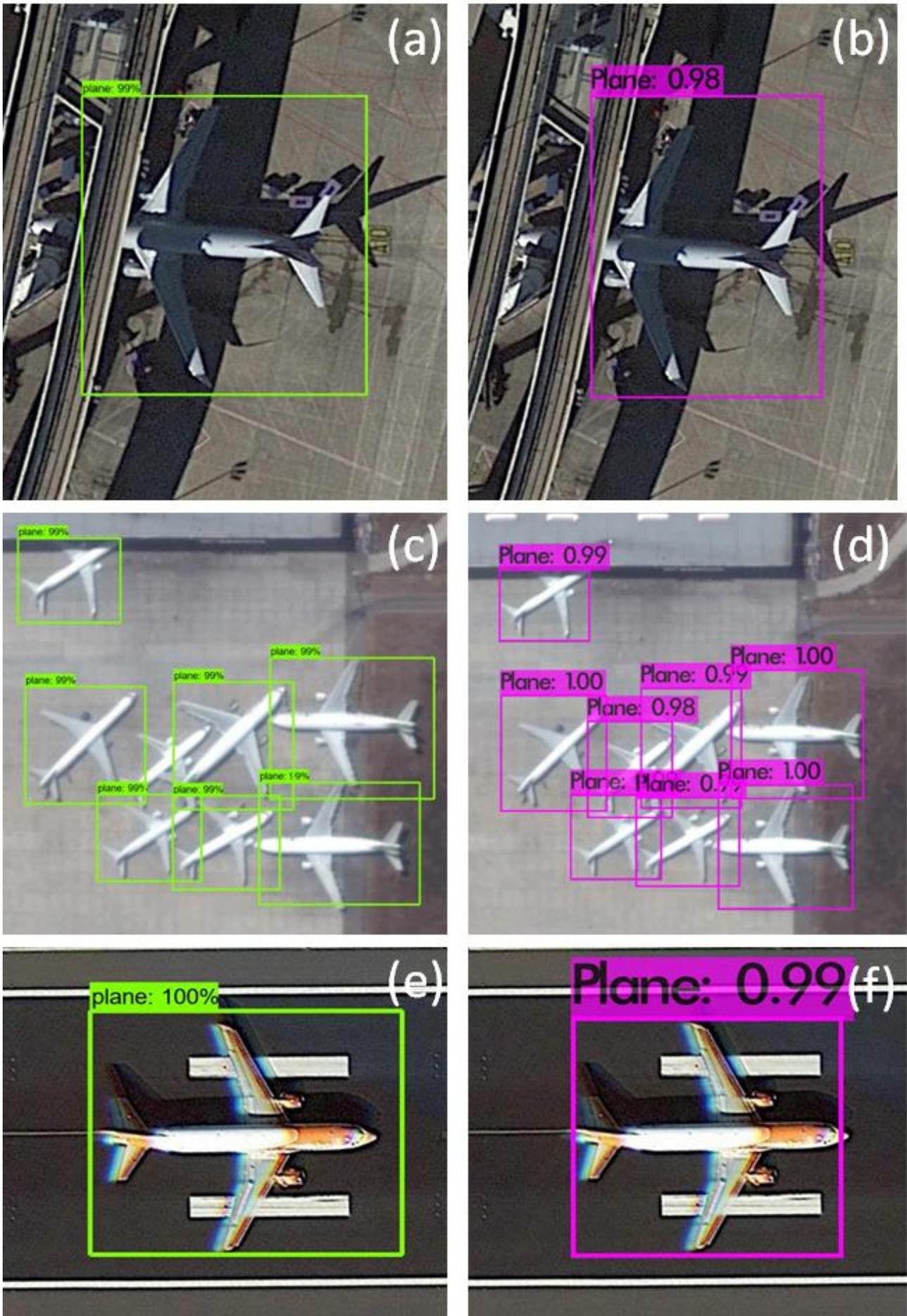


Figure 8. Visibility, density and motion blur examples

5. Conclusion

We created a novel airplane detection dataset, called HRPlanes, that includes image patches of commercial and military airplanes. We generated this new dataset to be a benchmark for deep learning-based object detection methods specifically for airplane cases. We evaluated the performance of YOLOv4 and Faster R-CNN on the created dataset with various experiments with different hyperparameters. Both models have provided satisfactory results for mAP at 75% IoU threshold above 90% which is quite high. Our results show that the dataset provides highly accurate information to train deep neural networks efficiently. Our proposed models and hyperparameter setting could be used for various remote sensing-based airplane detection tasks and our model weights could be directly used for the inference of similar datasets and transfer learning of different datasets. Moreover, our test dataset could be used by different researchers to evaluate the new model proposals. After further analysis and quality checks, it is planned to share the whole dataset publicly. Train and validation sets of the dataset are available upon request from the corresponding author. Currently, the weights of both networks and test dataset are available on: <https://github.com/TolgaBkm/HRPlanes>

Acknowledgement

The authors would like to thank all researchers from Istanbul Technical University, Center for Satellite Communications and Remote Sensing for their assistance during data processing. We are also grateful to Google Earth for providing high resolution satellite imagery.

Author contributions

Tolga Bakirman: Conceptualization, Data curation Methodology, Software, Validation, Writing-Original draft preparation **Elif Sertel:** Conceptualization, Data curation, Methodology, Software, Validation, Visualization, Investigation, Writing-Reviewing and Editing.

Conflicts of interest

The authors declare no conflicts of interest.

References

- Kaynarca, M., Demir, N., & San, B. T. (2020). Yeraltı Suyu Kaynaklarının Uzaktan Algılama ve CBS Teknikleri Kullanarak Modellenmesine Yönelik bir Yaklaşım: Kırkgöz Havzası (Antalya). *Geomatik*, 5(3), 241-245.
- Efe, E., & Alganci, U. (2023). Çok zamanlı Sentinel 2 uydu görüntüleri ve makine öğrenmesi tabanlı algoritmalar ile arazi örtüsü değişiminin belirlenmesi. *Geomatik*, 8(1), 27-34.
- Li, K., Wan, G., Cheng, G., Meng, L., & Han, J. (2020). Object detection in optical remote sensing images: A survey and a new benchmark. *ISPRS journal of photogrammetry and remote sensing*, 159, 296-307.
- Körez, 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.
- Biyik, M. Y., Atik, M. E., & Duran, Z. (2023). Deep learning-based vehicle detection from orthophoto and spatial accuracy analysis. *International Journal of Engineering and Geosciences*, 8(2), 138-145.
- Alganci, U., Soydas, M., & Sertel, E. (2020). Comparative research on deep learning approaches for airplane detection from very high-resolution satellite images. *Remote sensing*, 12(3), 458.
- Liu, G., Sun, X., Fu, K., & Wang, H. (2012). Aircraft recognition in high-resolution satellite images using coarse-to-fine shape prior. *IEEE Geoscience and Remote Sensing Letters*, 10(3), 573-577.
- Xu, C., & Duan, H. (2010). Artificial bee colony (ABC) optimized edge potential function (EPF) approach to target recognition for low-altitude aircraft. *Pattern Recognition Letters*, 31(13), 1759-1772.
- 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.
- Sun, H., Sun, X., Wang, H., Li, Y., & Li, X. (2011). Automatic target detection in high-resolution remote sensing images using spatial sparse coding bag-of-words model. *IEEE Geoscience and Remote Sensing Letters*, 9(1), 109-113.
- Zhang, W., Sun, X., Fu, K., Wang, C., & Wang, H. (2013). Object detection in high-resolution remote sensing images using rotation invariant parts based model. *IEEE Geoscience and Remote Sensing Letters*, 11(1), 74-78.
- Zhang, W., Sun, X., Wang, H., & Fu, K. (2015). A generic discriminative part-based model for geospatial object detection in optical remote sensing images. *ISPRS journal of photogrammetry and remote sensing*, 99, 30-44.
- Lei, Z., Fang, T., Huo, H., & Li, D. (2011). Rotation-invariant object detection of remotely sensed images based on texton forest and hough voting. *IEEE Transactions on Geoscience and Remote Sensing*, 50(4), 1206-1217.
- Liu, L., & Shi, Z. (2014). Airplane detection based on rotation invariant and sparse coding in remote sensing images. *Optik*, 125(18), 5327-5333.
- Ball, J. E., Anderson, D. T., & Chan, C. S. (2017). Comprehensive survey of deep learning in remote sensing: theories, tools, and challenges for the community. *Journal of applied remote sensing*, 11(4), 042609
- Chen, Z., Zhang, T., & Ouyang, C. (2018). End-to-end airplane detection using transfer learning in remote sensing images. *Remote Sensing*, 10(1), 139.

17. Xu, Y., Zhu, M., Xin, P., Li, S., Qi, M., & Ma, S. (2018). Rapid airplane detection in remote sensing images based on multilayer feature fusion in fully convolutional neural networks. *Sensors, 18*(7), 2335.
18. Zhu, M., Xu, Y., Ma, S., Li, S., Ma, H., & Han, Y. (2019). Effective airplane detection in remote sensing images based on multilayer feature fusion and improved nonmaximal suppression algorithm. *Remote Sensing, 11*(9), 1062.
19. Wu, Z. Z., Weise, T., Wang, Y., & Wang, Y. (2020). Convolutional neural network based weakly supervised learning for aircraft detection from remote sensing image. *IEEE Access, 8*, 158097-158106.
20. Zhou, L., Yan, H., Shan, Y., Zheng, C., Liu, Y., Zuo, X., & Qiao, B. (2021). Aircraft detection for remote sensing images based on deep convolutional neural networks. *Journal of Electrical and Computer Engineering, 2021*, 1-16.
21. Ji, F., Ming, D., Zeng, B., Yu, J., Qing, Y., Du, T., & Zhang, X. (2021). Aircraft detection in high spatial resolution remote sensing images combining multi-angle features driven and majority voting CNN. *Remote Sensing, 13*(11), 2207.
22. Shi, L., Tang, Z., Wang, T., Xu, X., Liu, J., & Zhang, J. (2021). Aircraft detection in remote sensing images based on deconvolution and position attention. *International Journal of Remote Sensing, 42*(11), 4241-4260.
23. Wu, Q., Feng, D., Cao, C., Zeng, X., Feng, Z., Wu, J., & Huang, Z. (2021). Improved mask R-CNN for aircraft detection in remote sensing images. *Sensors, 21*(8), 2618.
24. Zeng, B., Ming, D., Ji, F., Yu, J., Xu, L., Zhang, L., & Lian, X. (2022). Top-Down aircraft detection in large-scale scenes based on multi-source data and FEF-R-CNN. *International Journal of Remote Sensing, 43*(3), 1108-1130.
25. Chen, X., Liu, J., Xu, F., Xie, Z., Zuo, Y., & Cao, L. (2022). A Novel Method of Aircraft Detection under Complex Background Based on Circular Intensity Filter and Rotation Invariant Feature. *Sensors, 22*(1), 319.
26. Xia, G. S., Bai, X., Ding, J., Zhu, Z., Belongie, S., Luo, J., ... & Zhang, L. (2018). DOTA: A large-scale dataset for object detection in aerial images. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 3974-3983).
27. Waqas Zamir, S., Arora, A., Gupta, A., Khan, S., Sun, G., Shahbaz Khan, F., ... & Bai, X. (2019). isaid: A large-scale dataset for instance segmentation in aerial images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops* (pp. 28-37).
28. Lam, D., Kuzma, R., McGee, K., Dooley, S., Laielli, M., Klaric, M., ... & McCord, B. (2018). xview: Objects in context in overhead imagery. *arXiv preprint arXiv:1802.07856*.
29. Shermeyer, J., Hossler, T., Van Etten, A., Hogan, D., Lewis, R., & Kim, D. (2021). Rareplanes: Synthetic data takes flight. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision* (pp. 207-217).
30. HyperLabel. <https://docs.hyperlabel.com/>
31. Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 779-788).
32. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., ... & Rabinovich, A. (2015). Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1-9).
33. Redmon, J., & Farhadi, A. (2017). YOLO9000: better, faster, stronger. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 7263-7271).
34. Redmon, J., & Farhadi, A. (2018). Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*.
35. Lin, T. Y., Dollár, P., Girshick, R., He, K., Hariharan, B., & Belongie, S. (2017). Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2117-2125).
36. Bochkovskiy, A., Wang, C. Y., & Liao, H. Y. M. (2020). Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*.
37. Wang, C. Y., Liao, H. Y. M., Wu, Y. H., Chen, P. Y., Hsieh, J. W., & Yeh, I. H. (2020). CSPNet: A new backbone that can enhance learning capability of CNN. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops* (pp. 390-391).
38. He, K., Zhang, X., Ren, S., & Sun, J. (2015). Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE transactions on pattern analysis and machine intelligence, 37*(9), 1904-1916.
39. Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 580-587).
40. Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., ... & Darrell, T. (2014, November). Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the 22nd ACM international conference on Multimedia* (pp. 675-678).
41. Girshick, R. (2015). Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision* (pp. 1440-1448).
42. Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems, 28*.
43. Lin, T. Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., ... & Zitnick, C. L. (2014). Microsoft coco: Common objects in context. In *Computer Vision-ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13* (pp. 740-755). Springer International Publishing.
44. Everingham, M., Van Gool, L., Williams, C. K., Winn, J., & Zisserman, A. (2010). The pascal visual object classes (voc) challenge. *International journal of computer vision, 88*, 303-338.

45. Yaban, B., Alganci, U., & Sertel, E. (2022). Aircraft detection in very high-resolution satellite images using YOLO-based deep learning methods. *Intercontinental Geoinformation Days*, 4, 270-273.



© Author(s) 2023. This work is distributed under <https://creativecommons.org/licenses/by-sa/4.0/>