4th Intercontinental Geoinformation Days

igd.mersin.edu.tr

Aircraft detection in very high-resolution satellite images using YOLO-based deep learning methods

Berkay Yaban[1], Ugur Alganci *[2], Elif Sertel [2]

[1] Istanbul Technical University, Graduate School, İstanbul, Türkiye
[2] Istanbul Technical University, Civil Engineering Faculty, Geomatics Engineering Department, İstanbul, Türkiye

Keywords
Remote sensing
Deep Learning
Convolutional Neural Network

Abstract

With the recent developments in remote sensing technology, satellite images with high spatial and temporal resolution have been becoming widely available. Very high resolution (VHR) satellite images are very appropriate data sources for geospatial object detection using deep learning algorithms. Airplane detection from satellite images is one of the significant application areas to support airspace inspection, airline traffic control, and defense applications. In this study, we compared various variants of YOLOv5 (You Only Look Once) models and the Scaled-YOLOv4 model for aircraft detection from satellite images. We implemented different hyperparameters, optimization algorithms, and data augmentation methods. Finally, based on the results of numerous experiments, we evaluated the advantages and disadvantages of both methods. Our analysis illustrated that the best mAP@0.50:0.95 value of 0.865 belongs to the YOLOv5x model with 16 batch sizes. Whereas, in terms of computational efficiency, the Scaled-YOLOv4 model has the shortest duration in the training.

1. Introduction

Aircraft detection from satellite images is an important topic since obtained information is used for traffic control, airport activity monitoring, environmental impact assessments, and defense applications. Satellite systems with their capabilities of covering large areas, including high spatial details, fast data collection, and processing times are important sources of information for the geospatial object detection such as planes, ships, storage areas, and buildings (Alganci et al., 2020; Bakirman et al., 2022; Cheng and Han, 2016; Psiroukis et al. 2021).

Manual digitization of geospatial objects from satellite images is highly dependent on the experience of the operator and it is time-consuming. Therefore, it is essential to develop accurate automatic approaches for geospatial object detection. Deep learning-based approaches have become widespread in 2012 and later, especially after the successful conclusion of the ImageNet competition.

Different disciplines and applications have benefited from DL methods. In the Remote Sensing (RS) domain, DL methods are also used for the detection of different geospatial objects, land cover/use segmentation, and pan-sharpening. For the object detection tasks, You Only Look Once (YOLO) models are common since accurate and fast results could be obtained using YOLO models (Redmon et al., 2016; Li et al., 2017; Krizhevsky et al., 2012; Wang et al., 2021).

In recent years, with the developments in graphics cards and the production of GPU-based solutions, deep learning-based methods have become more common. In addition, the Google Colab platform has made a significant contribution to deep learning studies with its cloud-based computing environment and efficiency to implement different DL frameworks and libraries.

In this study, we aimed to automatically detect airplanes from very high-resolution satellite images using the High Resolution Planes (HRPlanes) data set and a new test data set generated from satellite images of different airports and air bases obtained from the Google Earth platform.

We implemented different experimental designs for YOLOv5 variations and Scaled-YOLOV4 models, and these compared two YOLO models. For experimental designs, we tried various hyperparameter values, optimization functions, and data augmentation methods. We compared the results of our experiments based on Mean Average Precision (mAP) values.

* **Corresponding Author**

*(yaban20@itu.edu.tr) ORCID ID 0000-0002-5648-8506
(alganci@itu.edu.tr) ORCID ID 0000-0002-5693-3614
(sertele@itu.edu.tr) ORCID ID 0000-0003-4854-494X

Cite this study

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

2. Method

2.1. Data and Environment

We used the HRPlanes dataset that includes high-resolution satellite images from various airports across the world (Bakirman & Sertel, 2022). The sizes of the images are 4800 x 2703 pixels. HRPlanes dataset was divided into 1686 training and 404 validation images. In addition to this dataset, we also collected 100 images from Google Earth and used them as the independent test set.

We implemented our experiments in the Google Colab Pro development platform, in which we could able to use an Nvidia P100 graphics card. We used YOLOv5x and YOLOv5l variants of YOLO5 and the Scaled- YOLOv4 models (Mahendrakar et al., 2021; Wang et al., 2021).

2.2. Data Augmentation

We used Hue, Saturation, Value (HSV), and mosaic data augmentation methods to synthetically increase the dataset. HSV specifies colors based on hue, saturation, and brightness values rather than Red, Green, and Blue (RGB) values. HSV provides better results for the object with a specific color. The mosaic method, on the other hand, combines 4 training images into one image with certain ratios. Thus, the trained model can learn the identification of objects at a smaller scale than normal (Hao & Zhili, 2020).

2.3. YOLOv5 Algorithm

In region-based algorithms, possible positions of individual objects are fed into the network. For this reason, the processing load increases, and it takes a long time to get results from the model. The most important feature of the YOLO algorithm is that it works fast because it passes through the neural network at once by dividing the image into grids (Jocher et al., 2022). The purpose of griding is to detect the object and enclose it in a bounding box. If both grids detect an object, it uses a non-maximum suppression method to eliminate clutter. With this method, the bounding box with the smaller probability value is removed (Krizhevsky et al., 2012)

The difference between YOLOv5 from older YOLO algorithms is that it uses the Pytorch framework (Jocher et al., 2022). In architecture, the backbone is the feature extraction layer. An interlayer called BottleNeck is used to obtain more information while estimating objects (Figure 1).

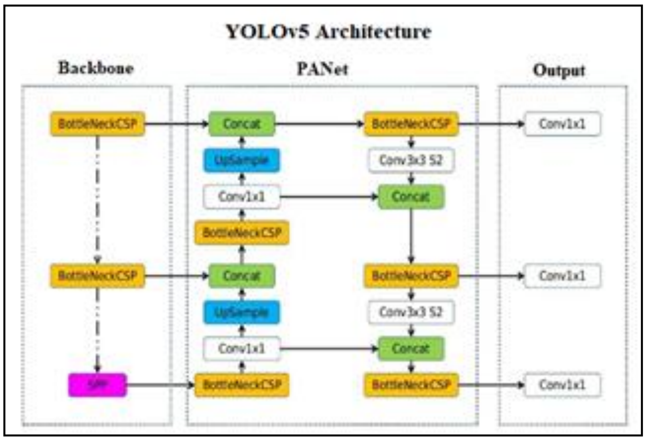

Figure 1. YOLOv5 Architecture [2]

2.4. Scaled-YOLOv4 Algorithm

Scaled-YOLOv4, which is a different version of YOLOv4, aims to improve the training time by scaling the model. To ensure this, both the depth, width, and resolution values of the model and the structure of the network are scaled using the Cross Stage Partial (CSP) pproach (Mahendrakar et al., 2021). CSP divides the input into two different paths and convolutions to one path (Jocher et al., 2022; Wang et al., 2020). In the output part; combining these two paths provides the result.

2.5. Experiment Design

In this study, we conducted and evaluated 11 experiments with different settings including various model, network, optimizer batch size, and data augmentation combinations. Yolov5 and Scaled-Yolov4 tests were performed using the Pytorch library. 1686 training images and 404 validation images were used in each test. The network sizes were 640x640, 960x960, and 1280x1280 in YOLOv5 tests and 416x416 in Scaled-YOLOv4 tests. In order to make comparisons between the tests, the iteration number was kept constant at 100, but different hyperparameters and augmentations were used. While 16 and 8 values are used as batch size, mosaic and HSV (hue, saturation, value) are used as data augmentation. These configurations are summarized in Table 1.

Table 1. Experimental setup

No	Model	Network	Optimizer	Batch Size	Augmentation
Exp-1	YOLOv5m	1280x1280	SGD	8	Image HSV-Saturation(0,7)- Hue(0,015)-Value(0,4)
Exp-2	YOLOv5m	1280x1280	Adam	8	Image HSV-Saturation(0,7)- Hue(0,015)-Value(0,4)
Exp-3	YOLOv5x	960x960	SGD	16	Image HSV-Saturation(0,7)- Hue(0,015)-Value(0,4)-Mosaic
Exp-4	YOLOv5x	640x640	SGD	16	Image HSV-Saturation(0,7)-Hue (0,015)-Value(0,4)
Exp-5	YOLOv5l	960x960	SGD	8	-
Exp-6	YOLOv5l	640x640	Adam	8	Mosaic
Exp-7	YOLOv5l	640x640	SGD	16	Image HSV-Saturation(0,7)-Hue(0,015)-Value(0,4)
Exp-8	YOLOv5l	960x960	SGD	8	Image HSV-Saturation(0,7)-Hue(0,015)-Value(0,4)-Mosaic
Exp-9	Scaled-YOLOv4	416x416	SGD	8	Image HSV-Saturation(0,7)-Hue(0,015)-Value(0,4)
Exp-10	Scaled-YOLOv4	416x416	SGD	16	Image HSV-Saturation(0,7)-Hue(0,015)-Value(0,4), Mosaic
Exp-11	Scaled-YOLOv4	416x416	SGD	16	-

2.6. Evaluation Metrics

The results of different experiments were evaluated with mean average precision(mAP) Precision alues at Union (IoU) threshold value of 0.50 (mAP@0.50), and avarege of AP values from IoU of 0.5 to 0.95 (mAP@0.50:0.95), Precision and Recall metrics (Henderson and Ferrari, 2016).

3. Results and Discussion

Our results showed that among the YOLOv5 models, the X and M models achieved higher mAP values. Through them, the YOLOv5x with 960x960 network size (Exp-3) yielded the best outcome (Table 2). In addition, this implementation provided higher mAP values in initial steps, which pointed out a faster learning capability with less iterations.

However, increasing the network size in the L model (Exp –8) resulted in a mAP value that was comparable to the YOLOv5x (960) model, which, while halving the training time.

When the optimization functions are compared, models trained with Adam produced lower mAP values than the models trained with Stochastic Gradient Descent (SGD). It also took longer in terms of training time. Thus, we recommend use of SGD in similar experiments.

Increments in the batch size improved the detection accuracy, however it requires more computational power, which resulted in increased computation time in our experiment setup.

When the detection results are evaluated visually, it can be commented that, both models provided satisfactory detections, even with challenging background and atmospheric conditions. More over both models are able to detect airplanes with different sizes (Figure 3).

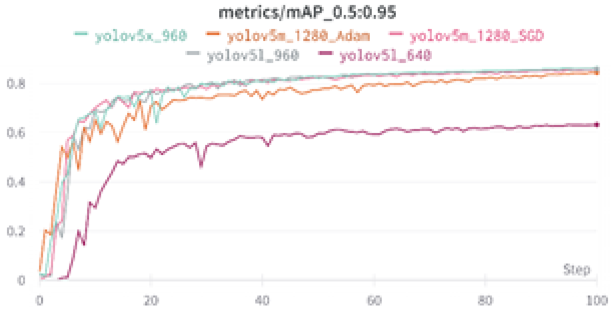

Figure 2. mAP graphic for YOLOv5 implementations.

Table 2. Evaluation metric results

No	Precision	Recall	mAP@0.50	mAP@0.50:0.95	Time
Exp-1	0.979	0.976	0.987	0.863	5h 9min 10sec
Exp-2	0.988	0.977	0.993	0.860	9h 12min 5 sec
Exp-3	**0.994**	**0.978**	**0.994**	**0.865**	16h 4min 23sec
Exp-4	0.993	0.979	0.994	0.799	12h 14min 42sec
Exp-5	0.941	0.968	0.979	0.857	8h 6min 34sec
Exp-6	0.978	0.968	0.982	0.789	7h 59min 12sec
Exp-7	0.980	0.977	0.983	0.805	5h 6min 12sec
Exp-8	0.990	0.983	0.992	0.864	8h 2min 17sec
Exp-9	0.843	0.973	0.972	0.754	1h 9min 19sec
Exp-10	0.879	0.976	0.98	0.796	1h 39min 10sec
Exp-11	0.877	0.975	0.979	0.782	1h 49min 41sec

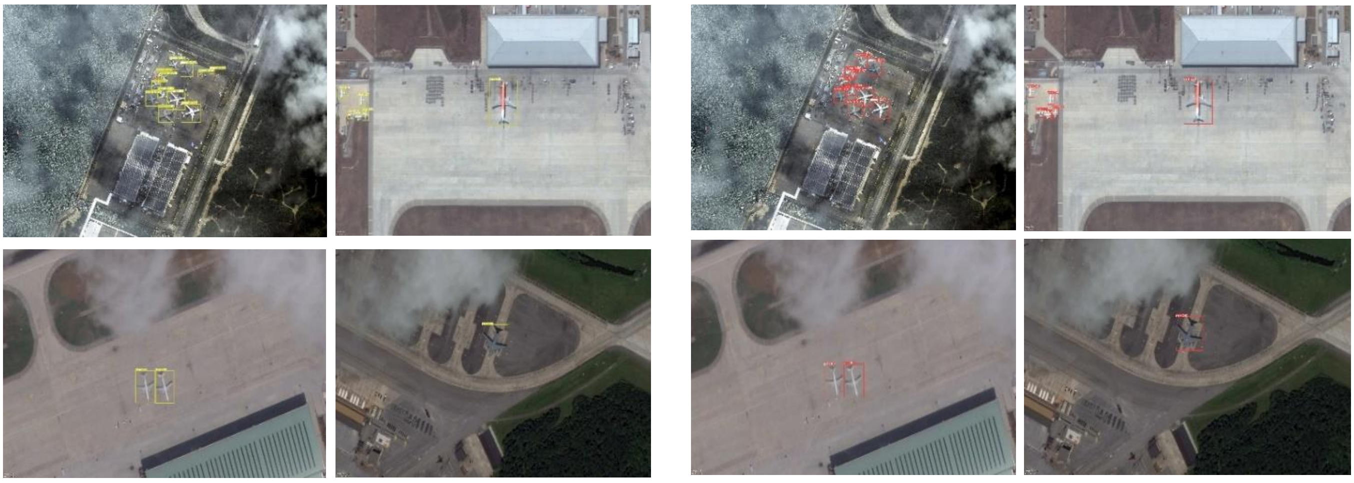
(a) (b)

Figure 3. Detection previews from a) Scaled-YOLOv4 and b) YOLOv5 Architecture

4. Conclusion

Within the scope of this study, YOLOv5 and Scaled-YOLOv4 deep learning models were evaluated with a total of 11 tests with different hyperparameters, augmentations and network sizes. In this context, YOLOv5 models produced the highest mAP values.

In particular, increasing the number of batches (batch size) provided an increase of 0.042 at the value of mAP@0.50:0.95 for the same model. In addition, increasing the network size provided an increase the value of mAP@0.50:0.95 for the all models. Preferring smaller scale models of YOLOv5 and using more powerful graphics cards can enable model training with higher number of batch sizes, thus may result in higher success rates indirectly.

In addition, for systems that do not need very high accuracy, the Scaled-YOLOv4 model can be used to save time. It completed the model training about 4 to 6 times faster than the YOLOv5 models.

References

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.

Bakirman, T, and Sertel. E. (2022). HRPlanes: High Resolution Airplane Dataset for Deep Learning. arXiv preprint arXiv:2204.10959.

Bakirman, T., Komurcu, I. & Sertel, E. (2022). Comparative analysis of deep learning-based building extraction methods with the new VHR Istanbul dataset, Experts Systems with Applications, 202, 117346.

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.

Hao, W., & Zhili, S. (2020, November). Improved Mosaic: Algorithms for more Complex Images. In Journal of Physics: Conference Series (Vol. 1684, No. 1, p. 012094). IOP Publishing.

Henderson, P., & Ferrari, V. (2016, November). End-to-end training of object class detectors for mean average precision. In Asian conference on computer vision (pp. 198-213). Springer, Cham.

Jocher, G., Stoken, A., Borovec, J., NanoCode012, ChristopherSTAN, Changyu, L., Laughing, tkianai, yxNONG, Hogan, A., lorenzomammana, AlexWang1900, et al (2021). “ultralytics/yolov5: v4.0 - nn.SiLU() activations, Weights & Biases logging, PyTorch Hub integration,” Jan. 2021. [Online]. Available: https://doi.org/10.5281/zenodo.4418161

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems, 25.

Li, X., Wang, S., Jiang, B., & Chan, X. (2017, December). Airplane detection using convolutional neural networks in a coarse-to-fine manner. In 2017 IEEE 2nd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC) (pp. 235-239). IEEE.

Mahendrakar, T., White, R. T., Wilde, M., Kish, B., & Silver, I. (2021). Realtime Satellite Component Recognition with yolov5. In Small Satellite Conference.

Psiroukis, V., Malounas, I., Mylonas, N., Grivakis, K. E., Fountas, S., & Hadjigeorgiou, I. (2021). Monitoring of free-range rabbits using aerial thermal imaging. Smart Agricultural Technology, 1, 100002.

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).

Wang, C. Y., Bochkovskiy, A., & Liao, H. Y. M. (2021). Scaled-yolov4: Scaling cross stage partial network. In Proceedings of the IEEE/cvf conference on computer vision and pattern recognition (pp. 13029-13038).

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).