Journal of Physics: Conference Series

PAPER • OPEN ACCESS

Aircraft type recognition based on YOLOv8

To cite this article: Chunyue Lei *et al* 2024 *J. Phys.: Conf. Ser.* **2787** 012047

Journal of Physics: Conference Series **2787** (2024) 012047 doi:10.1088/1742-6596/2787/1/012047

Aircraft type recognition based on YOLOv8

Chunyue Lei [1], Jingsheng Zeng [1], Yuming Xia [1] and Fengqian Pang [1,2]

[1]School of Information Science and Technology, North China University of Technology, Beijing 100114, China

[2]Corresponding author's e-mail: fqpang@ncut.edu.cn

Abstract: The field of aircraft type recognition has a broad range of applications. This technology serves as a valuable tool for airlines and airports in the scheduling and maintenance of aircraft. Furthermore, it provides crucial support to aviation regulatory authorities in their efforts to ensure safety in aviation. However, the task of recognizing aircraft types in real-world scenarios presents a significant challenge due to the subtle differences between various aircraft. Moreover, practical applications need real-time detection of aircraft type. To address these issues, this paper introduces the YOLOv8 object detector and investigates the performance of five different versions of YOLOv8 in the context of aircraft type recognition, to identify the most suitable model for this task. Experimental results on the Military Aircraft Detection Dataset demonstrate that the YOLOv8l model outperforms the others, achieving the highest performance with a COCO AP score of 84.2.

1. Introduction

Aircraft Type Recognition (ATR) is a critical task in the aviation field [1-4]. By recognizing the type of aircraft in real-time, airlines and airports can better manage and dispatch aircraft. For aircraft regulatory authorities, real-time detection of aircraft types ensures airspace safety. However, manual recognition of aircraft types not only tends to be error-prone but also falls short in terms of detection speed [1-4]. With the rapid development of deep learning in recent years, researchers have introduced deep learning methods into the task of ATR. In real-world application scenarios, due to factors such as shooting angles and distances when detection equipment captures aircraft targets, recognition presents certain challenges. Moreover, the identification of aircraft types is a fine-grained feature, which increases the difficulty of the task.

Recently, Zhang et al. [2] combined Conditional Generative Adversarial Networks with a Region of Interest (ROI)-weighted loss function to precisely detect the keypoints of aircraft. In [3], Zhao et al. integrated the advantages of the CNN, AutoEncoder (AE), and Extreme Learning Machine (ELM) for ATR in remote sensing images. Huo et al. [4] utilized Vision Transformers (ViT) to recognize granularity categories of aircraft accurately and robustly. Li et al. [1] employed Long Short-Term Memory (LSTM) to model High-Resolution Range Profile (HRRP) features for the recognition and rejection of aircraft type.

In January 2023, Ultralytics released the YOLOv8 framework, which was known as the latest state-of-the-art (SOTA) detection model. YOLOv8 has proven its effectiveness in numerous applications, such as detection or ripeness identification for agricultural products [5-6], fire detection for smart cities [7], and real-time detection for unmanned aerial vehicles [8]. In this paper, we also introduced the YOLOv8 object detector to identify aircraft categories as a paradigm of fine-grained detection.

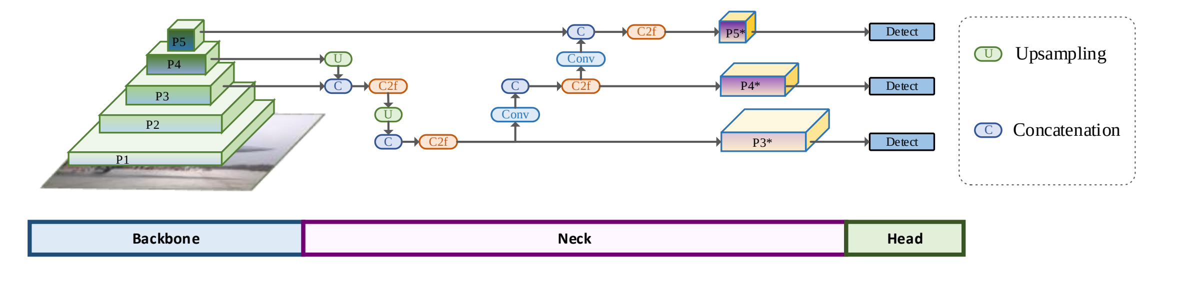

Figure 1. The YOLO v8 framework for plane-type recognition. It consists of three parts from left to right: Backbone, Neck, and Head. The Backbone extracts the feature maps with various scales. Then, the output multi-scale feature maps are fused by the Neck to cope with the small object. In the end, the outputs of the Neck are fed into the Head to detect the plane type.

Moreover, we further compared and investigated the performance of various YOLOv8 versions to provide an optimal one for fine-grained aircraft detection.

2. Method

YOLOv8 [9] is known as the latest state-of-the-art (SOTA) YOLO model, which was released by Ultralytics. It is proposed based on YOLOv5 and combines the advantages of other YOLO frameworks [10], including YOLOv3, YOLOv4, YOLOv7, and so on. There are five versions of YOLOv8 with different scaled model capacities: nano (n), small (s), medium (m), large (l), and extra-large (x). We denoted these YOLOv8 versions as YOLOv8n/s/m/l/x. In this section, we first introduced the framework of YOLOv8. Then the details of the network components were described in Section 2.2. Finally, Section 2.3 provided the Anchor-free strategy and losses used in the model.

2.1. Framework

The architecture of YOLOv8 is composed of three main components, backbone, neck, and head, as shown in Figure 1. The CSPDarkNet53 is selected as its backbone to extract rich features, which outputs the feature maps with various scales, as shown **P1~5** on the left of Figure 1. The Neck part is comprised of concatenation, upsampling, convolutional, and C2f layers, which present a U-shape structure. It is responsible for fusing the multi-scale features and attaching the ability to detect small objects. The head of YOLOv8 utilized the popular decoupled head, where classification and localization branches are separated. The decoupled head combined with the corresponding losses (details in Section 2.3) to alleviate the problem of the semantic gap between classification and regression tasks.

2.2. Network components

Backbone. YOLOv8 employs a pretrained CSPDarkNet53 as its backbone. From left to right in Figure 2, the backbone derives different scaled feature maps whose spatial resolution reduces and channel dimension arises. **P1~5** in Figure 2 corresponds to that in Figure 1.

Dense short-cut connections are introduced among the modules in the backbone, which allows the gradient to propagate deeper as a solution of gradient vanishing.

- **Convolutional layer (*Conv*)**: As shown in Figure 2, all the *Conv* adopt the same parameters. Especially, the kernel size was set to 3×3 with a stride to 2 and padding to 1, abbreviated as 3|2|1.
- **Conv-BN-SiLU block (*ConvBNS*)**: *ConvBNS* block is a combination of a convolutional layer, a Batch Normalization (*BN*) layer, and Sigmoid a Linear Unit (*SiLU*) layer. It serves as a kind of basic module for YOLOv8.

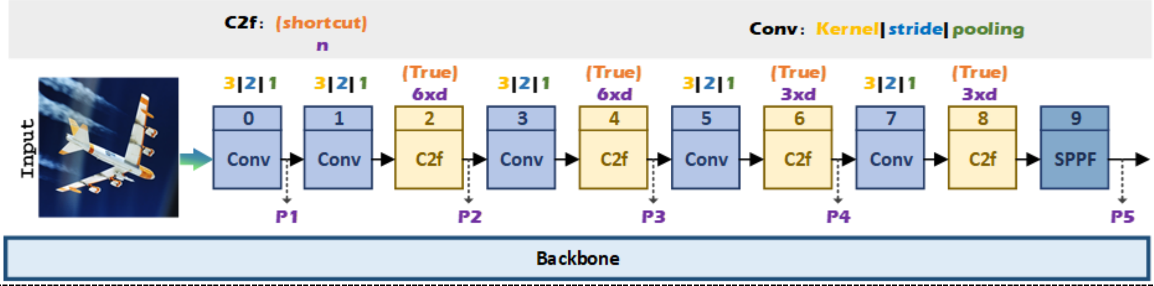

Figure 2. The details of the backbone. It is established by alternately stacking the convolutional layers (*Conv*) and *C2f* modules, and a Spatial Pyramid Pooling Fast module (*SPPF*) is placed at the end of the backbone.

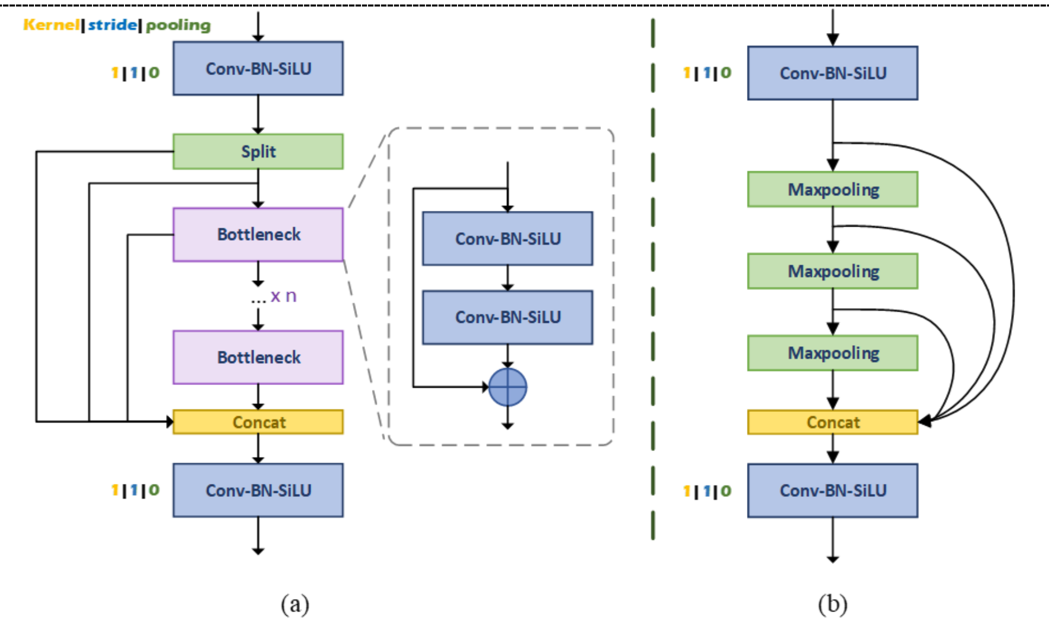

Figure 3. The modules in the backbone. (a) The structure of the C2f module containing two ConvBNS and n Bottleneck modules; (b) The structure of the SPPF module (two Conv are placed at the beginning and the end, as well as several maxpooling layers and a concatenation layer in the middle).

- **C2f module (*C2f*)**: The *C2f* was a key point in YOLOv8's Backbone and Neck. It can rich the model's gradient information to prevent gradient vanishing due to the long-distance transmission. Meanwhile, *C2f* also reduces the computational cost of the overall model. The *C2f* module is shown in Figure 3 (a), which mainly consists of a *split* layer, *n Bottleneck* module, and two *ConvBNS* at the beginning and the end. The *Bottleneck* had two *ConvBNS* with residual connections on the right of Figure 3 (a).
- **Spatial Pyramid Pooling Fast (*SPPF*)**: The *SPPF* at the end of the Backbone further extracts the crucial feature for the input plane image. The structure of *SPPF* is shown in Figure 3 (b), and it comprises three maxpooling layers and two *ConvBNS* in the fore-and-aft position. The results of the maxpooling layers are concatenated and fed into the last *ConvBNS*.

Neck. The Neck fuses the multi-scale feature maps produced by the backbone and then feeds them into the Head to detect the plane type. As shown in Figure 1, the Neck is depicted as a three-step U-shape network, in which each step corresponds to a scale of feature maps. There are three kinds of scale inputs (**P3**, **P4**, and **P5**) fed into the Neck so that the richer representation (**P3***, **P4***, and **P5***) is obtained.

Head. Following YOLOX, YOLOv8 also utilizes the decoupled head to accelerate the convergence of the model and enhance the accuracy of prediction. This decoupled head is split into two branches in accordance with the classification and localization tasks. The classification and localization branches have the same architecture as the three *Convs*. The first two *Convs* are set kernel size to 3×3, stride to 1, and padding to 1. The last *Conv* is set to kernel size to 1×1, stride to 1, and padding to 0. However, the number of output channels in the last *Conv*s is different for the two branches. They are set to the number of classes and 4 for the classification and localization branches, respectively.

2.3. Anchor-free strategy and losses

YOLOv8 employs the Task-Aligned Assigner (TAA) to select the positive samples when training the model. The core advantage of TAA lies in that it considers not only the samples' confidence score but also the Intersection over Union (IoU) between the regression bounding box and ground truth. As described in Section 2.2, the head has two separate classification and regression branches. The classification branch adopts Binary Cross Entropy (BCE) loss as the supervision signal, while the regression branch uses the Complete IoU (CIoU) and Distribution Focal (DF) losses.

3. Experiment

3.1. Dataset

In this paper, we employed the Military Aircraft Detection Dataset (MADD) as the training dataset for the model. MADD contains a total of 12, 398 images of 46 types of aircraft. The images are derived from a wide range of sources, including newspapers, natural images, and various other sources. The number of images for each type of aircraft in the dataset is imbalanced. We randomly selected 15/15 images from each type of aircraft (a total of 690/690 images) to form the validation/test sets during the experiment setup, while the rest were used as the training set.

3.2. Experimental setup

In the following experiments, the important hyper-parameters are listed as follows. The batch size is configured at 16, and the learning rate is set to 0.01. During the training process, the first three epochs are used for warmup, and we train the model for a total of 100 epochs. We evaluate the validation set after training one epoch. The best model is selected according to the mean Average Precision (mAP) on the validation set, which is then applied to the test set. We inherit the default data augmentation strategy in Ultralytics (The GitHub repository of Ultralytics: https://github.com/ultralytics/ultralytics). In this paper, we utilize COCO evaluation metrics from the COCO challenge, including AP, mAP75, and mAP50.

Table 1. Detection results (%) of different YOLOv8 versions.

Model	AP	mAP75	mAP50
YOLOv8n	67.3	70.4	73.1
YOLOv8s	77.3	81.2	83.1
YOLOv8m	82.4	85.5	87.4
YOLOv8l	**84.2**	**87.1**	**89.0**
YOLOv8x	84.1	86.8	88.8

Journal of Physics: Conference Series **2787** (2024) 012047 doi:10.1088/1742-6596/2787/1/012047

3.3. Experimental result

We perform a series of experiments on MADD for five YOLOv8 versions, containing YOLOv8n, YOLOv8s, YOLOv8m, YOLOv8l, and YOLOv8x. The experimental results are summarized in Table 1. As can be seen from Table 1, the AP of both YOLOv8n and YOLOv8s is less than 80%, and their performance is significantly lower than that of YOLOv8m/l/x. Comparing the last three rows in Table 1, the performance of YOLOv8l and YOLOv8x in terms of AP is comparable and significantly higher than YOLOv8m. By comparing the AP, mAP75, and mAP50 of the last two rows in Table 1, it can be concluded that YOLOv8l achieves the highest performance (84.2% AP, 87.1% mAP75, and 89.0% mAP50). Therefore, YOLOv8l is chosen as the best model for the aircraft type recognition task.

4. Conclusion

In this paper, we introduced the YOLOv8 framework for the aircraft-type recognition task. With the aid of the YOLOv8 framework, we achieved accurate classification and localization for aircraft even if there are subtle differences between various aircraft types. Furthermore, we conducted five versions of YOLOv8 on MADD to identify the most suitable model. Experiment results demonstrate that YOLOv8l and YOLOv8x achieved an AP that is at least 1.7% higher than that of YOLOv8n/s/m. YOLOv8l was deemed the optimal model owing to its marginally higher AP (by 0.1%) and fewer parameters compared to YOLOv8x.

References

[1] Li Y., Guo L., Wang Y., Yang H., Hu S., & Hu J. (2023). A Method of Radar HRRP Aircraft Type Identification and Rejection Based on LSTM. 2023 IEEE 2nd International Conference on Electrical Engineering, Big Data, and Algorithms (EEBDA), 1, 115-1, 118.

[2] Zhang Y., Sun H., Zuo J., Wang H., Xu G., & Sun X. (2018). Aircraft Type Recognition in Remote Sensing Images Based on Feature Learning with Conditional Generative Adversarial Networks. Remote Sensing, 10 (7), Article 7.

[3] Zhao B., Tang W., Pan Y., Han Y., & Wang W. (2021). Aircraft Type Recognition in Remote Sensing Images: Bilinear Discriminative Extreme Learning Machine Framework. Electronics, 10 (17), Article 17.

[4] Huo Y., Peng Y., & Lyu M. (2023). Automated Aircraft Recognition via Vision Transformers. 2023 IEEE Aerospace Conference, 1-8.

[5] Yang G., Wang J., Nie Z., Yang H., & Yu S. (2023). A Lightweight YOLOv8 Tomato Detection Algorithm Combining Feature Enhancement and Attention. Agronomy, 13 (7), Article 7.

[6] Xiao B., Nguyen M., & Yan W. Q. (2023). Fruit ripeness identification using the YOLOv8 model. Multimedia Tools and Applications.

[7] Talaat F. M. & ZainEldin H. (2023). An improved fire detection approach based on YOLO-v8 for smart cities. Neural Computing and Applications, 35 (28), 20, 939-20, 954.

[8] Wang G., Chen Y., An P., Hong H., Hu J., & Huang T. (2023). UAV-YOLOv8: A Small-Object-Detection Model Based on Improved YOLOv8 for UAV Aerial Photography Scenarios. Sensors, 23 (16), Article 16.

[9] Jocher G., Chaurasia A., & Qiu J. (2023). YOLO by Ultralytics (Version 8.0.0). https://github.com/ultralytics/ultralytics.

[10] Terven J. & Cordova Esparza D. (2023). A Comprehensive Review of YOLO: From YOLOv1 and Beyond (arXiv: 2304.00501).