Global Journal of Engineering and Technology [GJET].
ISSN: 2583-3359 (Online)
Frequency: Monthly
Published By GSAR Publishers
Journal Homepage Link- https://gsarpublishers.com/journal-gjet-home/

DETECTING AIRPLANES IN UAV IMAGES USING DEEP LEARNING MODEL

BY

Burak ZAHAL[1], Hüseyin CANBOLAT[2]
[1,2]Department of Defense Technology, Ankara Yıldırım Beyazıt University

Article History

Received: 15/12/2024
Accepted: 19/12/2024
Published: 21/12/2024

Vol – 3 Issue – 12

PP: - 35-38

Abstract:
Computer and artificial intelligence technology has been researched and developed since the 1950s. In the 1980s, research efforts in the field of deep learning, a subfield of machine learning, intensified. Currently, deep learning has provided solutions to many problems. Object detection technology has made significant progress with the use of algorithms such as You Only Look Once (YOLO), Open CV, etc., which enable instant detection of multiple objects in an image. Recently, UAVs have become critically important for countries. The use of UAVs in important studies is increasing day by day due to their areas of use and tasks. Real-time operation and the ability to take images of the desired quality according to camera capabilities are a data source for deep learning studies. This study aims to detect civil and military aircraft from UAV images using different versions of YOLO deep learning algorithms. For this purpose, a dataset was first created to train the YOLO deep learning model. Python-based programming language was used as software for training the model, and different libraries were used. Images containing different types and categories of aircraft were first taught to the created model. Then, for verification purposes, images were given to the model with the relevant codes and the ability to detect the aircraft object was measured. During the tests, the parameters required for the selection of the appropriate model were changed and their effects on the system were examined. Considering all these tests, it is estimated that the use of UAVs will increase and, in parallel, object detection with images obtained from UAVs will have critical importance in the fields of military, agriculture, health and autonomous technology.

***Index Terms**- AI, UAV, artificial intelligence, deep learning, object detection, YOLO*

1. INTRODUCTION

Object detection from UAV (Unmanned Aerial Vehicle) images using deep learning models has gained significant attention in fields such as defense, agriculture, disaster management, and environmental monitoring. UAVs can rapidly survey large areas and collect data from various altitudes, providing valuable insights. However, object detection in UAV images comes with specific challenges:

- Complex Backgrounds: Environments like forests, cities, and oceans can make object differentiation difficult.
- Variable Resolution: Objects may appear very small due to high-altitude image capture.
- Viewpoint Variation: Changing flight angles and perspectives complicate detection.
- Motion and Noise: UAV movement and sensor noise can affect image quality.

Object detection is a computer vision task that involves identifying and localizing objects in images or videos. It plays a crucial role in various applications, including surveillance, autonomous vehicles, and robotics. Object detection algorithms are generally classified into two main types: single-shot detectors and two-stage detectors. One of the earliest deep learning-based breakthroughs in object detection was the R-CNN (Regions with CNN features) model, developed by Ross Girshick and his team at Microsoft Research in 2014. This model combined region proposal methods with convolutional neural networks (CNNs) to detect and localize objects in images effectively. Object detection algorithms can also be categorized based on how many times the input image is processed by the network. This distinction highlights differences in efficiency and accuracy between various detection models [1].

YOLO (You Only Look Once), developed by Joseph Redmon and his team, formulates object detection as a regression task that predicts bounding boxes and associated class probabilities. Unlike traditional methods, YOLO processes the entire image at once during inference, allowing it to leverage the global context for more accurate predictions. Its high speed makes it ideal for real-time applications.

2. DEEP LEARNING

Deep learning is a machine learning technique within the field of artificial intelligence that surpasses many traditional algorithms in tasks like speech and image recognition. It has become a highly active research area in the machine learning and pattern recognition communities, achieving significant success across various applications, including speech recognition, computer vision, natural language processing, and numerous industry products. Neural networks are the foundational technology behind deep learning, enabling the design of intelligent systems [2].

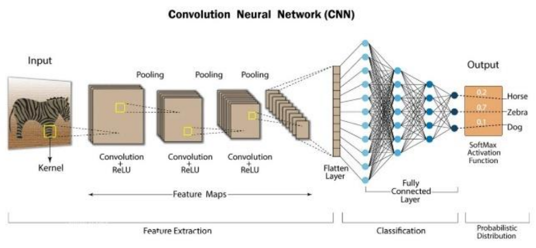

Figure 2.1 CNN architecture

Three major types of deep learning models are:

1. Convolutional Neural Networks (CNNs): Primarily used for image-related tasks such as classification, object detection, and image segmentation.
2. Recurrent Neural Networks (RNNs): Effective in processing sequential data, commonly applied in tasks like language modeling, speech recognition, and time-series forecasting.
3. Transformer Models: Widely used in natural language processing and computer vision, enabling applications like machine translation, text summarization, and image captioning.

As deep learning continues to receive significant attention, its capabilities are rapidly advancing. It allows systems to perform complex tasks with greater accuracy and automation. Application areas like computer vision, image processing, automated driving, and signal processing are evolving, with new sub-applications emerging as methods and technologies improve. For example, computer vision includes tasks such as image classification, object detection, and semantic segmentation. As research progresses, deep learning applications will expand, and previously unexplored areas will benefit from improved accuracy and efficiency [3].

Deep learning powers numerous applications. For instance, it is a key technology behind autonomous vehicles, helping them recognize traffic signs and distinguish pedestrians from streetlights. It also enables voice control in consumer electronics such as smartphones, tablets, smart TVs, and hands-free speakers.

3. YOLO (You Only Look Once)

You Only Look Once (YOLO) is a widely recognized object detection algorithm known for its high accuracy and fast processing speed. Its neural network architecture is one of the best in the field, contributing significantly to its popularity. Due to its effectiveness, searching for object detection algorithms on Google often brings up YOLO as the top result [4]. The YOLO algorithm uses a single Convolutional Neural Network (CNN) that divides the input image into a grid of S×S cells. Each grid cell is responsible for predicting a fixed number of bounding boxes. Along with each bounding box, the cell also predicts class probabilities, indicating the likelihood of specific objects (such as people, cars, or dogs) being present.

The main concept behind YOLO is that each grid cell can detect multiple objects within its region. For each detected object, the algorithm predicts its class and location, including the center coordinates and the bounding box's width and height. This unified method allows for fast and efficient object detection.

One of YOLO's greatest advantages is its rapid inference speed, making it ideal for real-time image processing. This makes it particularly suitable for applications such as video surveillance, autonomous vehicles, and augmented reality. Additionally, YOLO's simple architecture and low training data requirements make it easy to implement and adapt to various tasks. Despite its strengths, YOLO has some limitations. It may struggle with detecting small objects and can face challenges in performing fine-grained object classification. Nonetheless, YOLO remains a powerful tool for object detection, enabling new possibilities for researchers and practitioners alike. As computer vision continues to advance, it will be exciting to see how YOLO and other object detection algorithms evolve, overcoming their current limitations and expanding their capabilities. In general, the YOLO algorithm was used in the study because YOLO is the easiest and most useful deep learning model in the field of object detection [5].

4. TEST PHASE AND FINDING

Creating a dataset for YOLO involves collecting relevant images or videos and organizing them into folders like train, valid, and test. Afterward, data labeling is done using tools such as LabelImg or Roboflow, where bounding boxes are created around objects of interest. Each labeled image generates a text file with object class IDs and normalized bounding box coordinates in YOLO format. Next, the Darknet framework is set up by cloning its repository and compiling the code with GPU and OpenCV support if needed. The model configuration files, including yolov4.cfg, obj.data, obj.names, and text files listing image paths, are prepared. After adjusting parameters such as batch size, subdivisions, and filter values in the configuration file, model training begins using a pre-trained weight file like “yolov4.conv.137.” The training process periodically saves weights and calculates performance metrics such as mean Average Precision (mAP). Once the training is complete, the model is validated using evaluation commands to assess its precision, recall, and mAP scores. Finally, the trained model is tested on images or videos for real-time object detection.

In the testing phase of our study, Google Colab, a free cloud-based Jupyter Notebook environment provided by Google since 2017, was initially planned for use. Colab supports machine learning and deep learning model training on CPUs, GPUs, and TPUs. It enables access to high-performance computing environments, facilitating data-intensive tasks. Data is processed from a cloud storage space and re-uploaded after the study using appropriate code implementations.

During the training process, all YOLO versions were configured and tested on Google Colab. The platform used Python 3.10.12 with a Tesla T4 GPU, offering approximately 15GB of system memory. However, during YOLOv3 training, about 10GB (2/3 of the memory) was utilized, causing system overload, slow performance, and hardware overheating.

To achieve more accurate results aligned with the study objectives, advanced YOLO versions such as YOLOv5s, YOLOv5x, YOLOv8x, and YOLOv10x were selected. Since these versions required more memory and processing power, the environment was shifted to a computer running the Ubuntu operating system. Python 3.8.10 and PyTorch 2.4.1 were installed to support the algorithms. This high-performance platform provided faster and more stable training, ensuring error-free results.

To achieve accurate predictions and fast results during testing, reference values were set with an image size of 640 pixels, a batch size of 10, and 50 iterations per epoch.

In YOLO, key evaluation metrics such as metrics/precision, metrics/recall, mAP (Mean Average Precision), and mAP@0.5:0.95 are used to assess the model's object detection performance. Each metric provides a unique perspective on the model's effectiveness:

- *metrics/precision*: Indicates how accurate the model's predictions are by measuring the proportion of correct positive detections.
- *metrics/recall*: Reflects the model's ability to detect all relevant objects, showing how many actual objects were correctly identified.
- *mAP@0.5*: Evaluates detection accuracy across all classes using a fixed Intersection over Union (IoU) threshold of 0.5.
- *mAP@0.5:0.95*: Provides a more comprehensive evaluation by averaging the model's performance across multiple IoU thresholds, offering a deeper insight into detection quality.

These metrics collectively help determine the overall effectiveness of the YOLO model in object detection tasks. In our study, object detection was conducted using the YOLO deep learning model. Training was performed across different YOLO versions using the same parameters for comparative evaluation. Parameter adjustments within the same version were also explored. Training times varied based on dataset size, image quality, and model version efficiency, ranging from 23 minutes to 9 hours. The shortest training time occurred with 50 validation images in YOLOv3, while the longest was with 750 images in YOLOv10x.

- The first test was done with YOLOv3 by changing the number of validation images. The reference value was 82 and the validation image number was reduced to 50 and the test was repeated.
- Our next test was the model training done with YOLOv5s and YOLOv5x. In this training, the epoch value was increased from 50 to 100 and the effect of the epoch value on object detection was observed.
- In the third stage of the test, the batch number was selected as 5 and 10 with YOLOv8x.
- The last test preferred for the study was done with YOLOv10x. In this test, the image pixel value was increased from 640 to 750 and the comparison was made.

Table 4.1 Test Results

Version\Parameter	metrics/mAP_0.5:0.95	metrics/precision	metrics/recall
yolov3(82 Image)	0.50715	0.94794	0.67021
yolov3(50 Image)	0.59017	0.88941	0.8427
yolov5s (50 Epoch)	0.46454	0.89217	0.71536
yolov5s (100 Epoch)	0.51157	0.89031	0.72959
yolov5x (50 Epoch)	0.5968	0.948	0.75111
yolov8x (10 Batch)	0.48633	0.83032	0.67416
yolov8x (10 Batch)	0.48874	0.88031	0.64794
yolov10x (640 px)	0.48729	0.82607	0.59925
yolov10x (750 px)	0.52208	0.8958	0.64794

5. CONCLUSION

This study focuses on object detection using UAV images, which are crucial for national security. Due to restrictions on capturing images of sensitive locations like airports and military airfields, the dataset size was limited. The study evaluated different YOLO model versions (YOLOv3, YOLOv5, YOLOv8, and YOLOv10) using the same dataset. The tests also examined the effectiveness of reference parameters like epoch, precision, recall, and mAP@0.5:0.95, along with dataset size, image resolution, and hardware quality. The best performance was achieved with YOLOv5x using 50 epochs.

Test results revealed variations in model performance, with YOLOv5x achieving the highest mAP@0.5:0.95 value of 0.596, while YOLOv8 underperformed. The precision value for YOLOv5x was 0.948, indicating high accuracy in object detection. YOLOv3 showed a higher recall (0.894) with fewer validation images compared to YOLOv5x (0.85), suggesting that simpler versions are better at detecting objects in less complex images. The study emphasized the importance of selecting correct parameters like dataset size, epoch, image dimensions, and batch size to avoid poor performance, high costs, and long training times.

REFERENCES

1. Kundu R., “YOLO: Algorithm for Object Detection Explained [+Examples]”, January 2023, Accessed 10 December 2024, <https://www.v7labs.com/blog/yolo-object-detection>.
2. Mishra C., Gupta D. L., “Deep Machine Learning and Neural Networks: An Overview”, IAES International Journal of Artificial Intelligence (IJ-AI) Vol. 6, No. 2, June 2017, pp. 66~73, <DOI:10.11591/ijai.v6.i2.pp66-73>.
3. “What Is Deep Learning?”, Accessed 10 December 2024, <https://www.mathworks.com/discovery/deep-learning.html>.
4. Zvornicanin E., “What Is YOLO Algorithm?”, March 2018, Accessed 12 December 2024, <https://www.baeldung.com/cs/yolo-algorithm>.
5. Dey I., “What Is YOLO Algorithm?”, 12 July 2023, Accessed 12 December 2024, <https://medium.com/@ishudey11032002/what-is-yolo-algorithm-ef5a3326510b>.
6. Karpuram, “YOLO Algorithm for Custom Object Detection”, 25 June 2024, Accessed 12 December 2024, <https://www.analyticsvidhya.com/blog/2022/06/yolo-algorithm-for-custom-object-detection/>.