Aerial object detection analysis: Challenges and preliminary results

Agnes Maria Polina[1*], *Hari* Suparwito[1], and *Rosalia Arum* Kumalasanti[1]

[1]Informatics Department, Sanata Dharma University, Yogyakarta, Indonesia

Abstract. Computer vision allows computers to retrieve information from images, videos, and other visual inputs. Unmanned Aerial Vehicle (UAV) technology is also used to assist computer vision in collecting image data from the air. This paper aims to perform tree object detection using UAVs by capturing images perpendicularly from above the object. Image data was collected from around Sleman Yogyakarta using DJI Pro 3 from 5 to 12 July 2023. A total of 162 images were used as a dataset. The YOLOv8n model was implemented to 162 images as the training and validation data. Next, 12 other images were used as testing data. The results showed that YOLOv8n could detect trees well from above. The confidence value of the testing dataset with the appropriate image capture is more than 80%. As a deep learning algorithm for object detection, the YOLO model can perform object detection quickly and accurately. The subsequent research will focus on analyzing the implementation of object detection using the YOLO algorithm to measure open green areas.

1 Introduction

Computer vision has become one of the fastest fields of knowledge and has many applications in various industries. Computer vision has great potential to change how humans interact with technology and the world around them. Computer vision is a field of artificial intelligence that allows computers to see and recognize objects around them, just like humans, using cameras and artificial intelligence (AI) technology. Computer vision will enable computers to retrieve information from images, videos, and other visual inputs. The role of computer vision in the development of technology has had a tremendous impact on almost all aspects of life [1].

The use of Unmanned Aerial Vehicle (UAV) for object detection is increasing. UAVs have evolved into relevant technologies, such as component miniaturization, the increase of computational capabilities, and the evolution of computer vision techniques have allowed an essential advance in the development of UAV technologies and applications. UAV applications have reached military and defence with visual navigation algorithms, detection and avoidance of obstacles and air decision-making [2].

The merging of these two technologies, computer vision and UAVs, is a new strength of current technology, especially for aerial image detection and recognition. With the ability to

* Corresponding author: a.m.polina@usd.ac.id

distinguish objects such as buildings, animals, plants, fruits, and vehicles, this kind of technology is promising. Some examples of using computer vision and machine learning algorithms for object detection and recognition can be mentioned here. Research in the field of 3-D city modelling from remote sensing data has succeeded in detecting manufactured objects, for example, buildings and roads, but ignores information about the vegetation components of a city. The Area Traffic Control System and You Only Look Once (YOLO) V3 detect vehicles with an accuracy of more than 70% to determine the volume of vehicles to monitor traffic jams on the highway [3].

Computer vision and Support Vector machine classification were used in analyzing fruit grading and sorting. Computer vision-based fruit grading systems can replace labor work for the inspection of fruit grading. In their research, SVM (Support Vector Machine), a machine learning technique, gave the highest accuracy, but ANFIS (Adaptive Neuro Fuzzy Interference System) showed the lowest accuracy rate. Still, it is easy to implement [4].

As we know, recently, climate change has had a significant impact on our environment. Previous research found that insect disasters caused by global climate change killed many trees, which inevitably became a role in forest fires. The state of the forests is a crucial predictor of forest fires. This study could diagnose dead trees from aerial photos using a mask-RNN approach that has been retrained with a transfer learning strategy. This method can detect dead and live trees based on aerial images. The best models have a mean average precision score (mAP) of 54%. This study can automatically generate and calculate the number of dead tree masks to label the dead trees in an image as an indicator of forest health, which might be related to the causative study of environmental changes and the prediction likelihood of forest fire [5].

We also share the same concern about climate change. Trees are one of the factors that can prevent or reduce the acceleration of climate change caused by industry and air pollution. In the first step, we proposed object detection analysis using UAV and implemented YOLOv8n as an algorithm. The main goal of our study is to detect objects in the form of trees and separate them from surrounding buildings or structures. The object would be captured from above perpendicularly.

2 Materials and methods

The first step in this study is the data collection. Image data was collected using a DJI mini 3 Pro (Fig.1) UAV drone from 5 to 12 July 2023. A total of 162 images were successfully captured and used as datasets. The images were captured in Sleman, Yogyakarta area.

One of the reasons for capturing image data directly and not using public data such as COCO [6], CIFAR [7] or others is that we wanted to apply custom data and presuppose an extraordinary environmental situation in the local area. We captured images from above perpendicularly.

https://doi.org/10.1051/e3sconf/202447502017

Fig. 1. DJI Mini 3 Pro

The second stage is data preprocessing. In data preprocessing, We did three things: transformation, selection, and image annotation. Modification is needed because the images generated by the drone have a different format from the image format to be processed. Here, we changed the image format from TIF format to JPG format.

Next, from the JPG formatted image, the selection is carried out to get a more similar and even picture of the tree's appearance from above. The last process is image annotation. In this process, the image will be marked or labelled by creating a rectangular box and determining that the image in the box is a tree. The image annotation process is done using the *labeling* tool.

After that, the image dataset was divided into 120 images as the training dataset, 30 as the validation dataset, and 12 as the testing dataset.

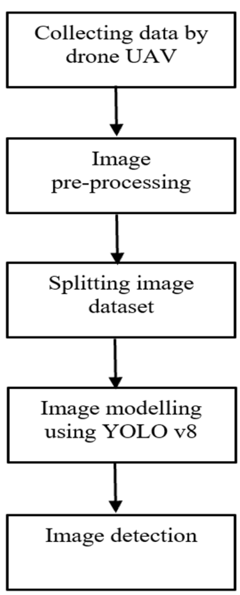

Fig. 2. The workflow diagram

Furthermore, YOLOv8n was implemented to detect and recognize images. Ultralytics, creators of YOLO, built YOLOv8, a new high-tech computer vision model that includes out-of-the-box support for object detection, classification, and segmentation tasks. YOLOv8 is available via a Python package and a command line interface [8]. YOLO is a deep-learning algorithm used for object detection in images. This algorithm allows fast and accurate object detection [9-11].

Why choose Ultralytics YOLO for the training model? Here are some compelling reasons to opt for YOLOv8's Train mode: Efficiency: Make the most out of the hardware, whether on a single GPU setup or scaling across multiple GPUs. Versatility: Train on custom datasets and readily available ones like COCO, VOC, and ImageNet. User-Friendly: Simple yet powerful CLI and Python interfaces for a straightforward training experience. Hyperparameter Flexibility: A broad range of customizable hyperparameters to fine-tune model performance [12].

Furthermore, several things can be considered in the use of YOLO, especially key features of Train Mode. The following are some notable features of YOLOv8's Train mode: Automatic Dataset Download: Standard datasets like COCO, VOC, and ImageNet are downloaded automatically on first use. Multi-GPU Support: To expedite the process, scale your training efforts seamlessly across multiple GPUs. Hyperparameter Configuration: The option to modify hyperparameters through YAML configuration files or CLI arguments. Visualization and Monitoring: Real-time tracking of training metrics and visualization of the learning process for better insights.

YOLO can also overcome the limitations of training datasets. As we know, collecting data will often lead to difficulties in obtaining large enough data because it consumes resources or time. YOLO is good enough to overcome the problem of the limited amount of training data [13].

What makes YOLO a superior model in image detection and recognition? YOLO (You Only Look Once) is an object detection network created by Joseph Redmon in 2016. YOLO works quite simply, just a Convolutional Neural Network that predicts several bounding boxes that will predict several bounding boxes and the class probability of each box (Fig. 3). YOLO receives an input image that is divided into a grid that is sent to a neural network to make the bounding box and class prediction. Each grid cell predicts B bounding boxes and the confidence score of each box. This confidence score reflects the model's confidence that the object inside the box is the predicted object. YOLO scores confidence as Pr(Object) * IOUtruth (Intersection of Union).

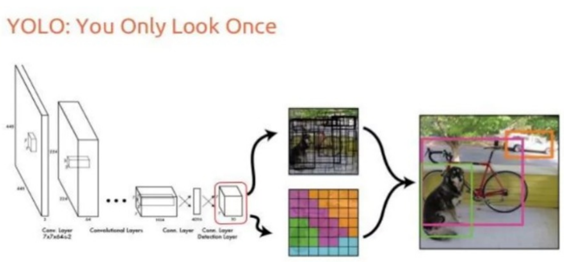

Fig. 3. The YOLO Architecture

Each bounding box has 4 predictions: x, y, w, and h. The coordinates (x, y) represent the box's center relative to the grid cell's boundary. Width (w) and Height (h) are relative predictions of the entire image. Each grid cell predicts conditional class probabilities C, Pr(Classi| object). These probabilities are connected to the grid cells with an object that will be expected [12].

YOLOv8 is a collection of convolutional neural network models constructed and trained with the PyTorch framework. Furthermore, the YOLOv8 package includes a single Python API that can interact with all of them using the same methods. That is why, to utilize it, we must have a Python environment, Jupyter Notebook is highly recommended. Even though applying a model with the default YOLOv8n weights is preferred, the Python module can also train a new model from scratch.

In this study, we trained the model using our dataset (custom dataset) to create a model and then use it for the testing dataset. The data sets in this research are collected using drone aerial detection techniques. Difficulty detecting objects from the air is a common problem in remote sensing. Some difficulties are often related to image resolution, lighting conditions, scale, shape variation, and vegetation. There are also ways to overcome some difficulties in detecting objects from the air, by using various image processing and data modelling techniques such as image segmentation, contrast enhancement, multispectral images, machine learning algorithms, and many more. Choosing a method suitable for the type of image and object to be detected is vital. A combination of various techniques is often necessary to overcome the difficulties of detecting objects from the air.

There are three experiments that we conducted in this study. The first is related to the degree of shooting, the color of the leaves and the tree's shape. The shooting distance, the distance from the drone camera to the tree object, is not considered. Instead, we believe the height of the building so that the drone does not crash into the building. This experiment is based on the supposition that our goal is to take aerial pictures at an angle perpendicular to the tree; also, in real life, we find that the color and shape of the tree are different. Then, there will be differences in the tree detection.

3 Results and discussion

YOLOv8n model algorithm for image detection was implemented on 12 sets of image testing data. The confidence values of the YOLOv8n model show two main differences in the images. Firstly, the YOLO model can accurately detect images, i.e., trees photographed from above (aerial photographs), and distinguish between trees and buildings. The high confidence value shows this. Secondly, the YOLO model can detect trees and can differentiate between trees and buildings, but the YOLO model provides a low confidence value. The low confidence value is obtained from images where the photo is taken from the side, and the tree has leaves that are not green (yellowish leaves).

Fig. 4. Photos of trees were taken perpendicularly from above.

 https://doi.org/10.1051/e3sconf/202447502017

Fig. 5. The tree photo could be separated from the surrounding building.

Figures 4 and 5 show that the YOLO image detection model can perform well. The image was captured from the top (aerial), and the tree leaves are green.

Fig. 6. The tree was photographed from the side.

Fig. 7. The trees were photographed from sides.

On the contrary, figures 6 and 7 were images captured from the side. The YOLO image detection model was less able to perform detection, but YOLO can still distinguish between trees and buildings.

Fig. 8. Three different photographs with differences in confidence.

Figure 8 shows that the YOLO image detection model can detect three trees with different confidence measures. The left image has a good result because the image was taken from above perpendicularly, and the leaves were green. The middle image has a good enough result since the tree image is mixed with the path image. The last image on the right has a lower confidence value because it is not the tree image, even though the color is green. Figure 9 shows that the YOLO image detection model is less able to detect trees with slightly yellowish leaf colors and the object did not capture perpendicularity. However, still YOLO could differentiate between trees and buildings.

Fig. 9. The color of the trees are yellowish, and the shooting angle is not perpendicular from above.

In other words, the idea conceived in this study to apply the YOLOv8n model to detect an object, a tree captured perpendicularly from above, and distinguish between 2 objects, a tree and a building, can be done well by YOLOv8n [12][14].

Although YOLO can detect trees well and distinguish them from buildings or objects around them, several things must be considered for good results. First is the image capture technique. In this study, aerial images are images taken from above. The detection results carried out by the YOLO model based on training data that has been trained can give different results. Taking pictures from above and perpendicular gives better results than pictures with a certain tilt angle. Even taking photos from the side provides inferior confidence results.

The second thing is the leaf's color and shape. As we know, most leaf colors are green, but the color of the leaves may be yellowish because the leaves have begun to age or other colors. as well as the shape of the leaves. So, it is essential to have training data that comes from a variety of leaf colors and leaf shapes. Diversity of data, color or form, is vital in recognition or object detection [15]. Previous research has also shown that the colour and position of the object are crucial in recognizing and detecting the object's confidence that has been trained. Hence, it is necessary to train diverse data so that whatever colour and position of the object can be detected in the testing data will provide a high confidence value [16].

Another possibility is the distance between the camera in the UAV and the object to be analyzed. Experiments are needed to determine the optimal distance between the object and the camera to recognize the object optimally. The previous study [1] also experienced the same thing. In their paper, they considered that the distance between the camera and the object needs to be considered in detail. In their study, they followed the contour of the height of the plain. Distance is very influential in determining the confidence level in object detection analysis. The distance variable is essential in animal object detection and recognition research: elephants, giraffes and zebras. The confidence value generated from the fully CNN-Retinanet model will be significantly influenced by the distance between the animal, the camera, and the human eye [17]. Comparing Figure 5 and Figure 9, we can see that Figure 5 has a higher confidence value of 0.89 than Figure 9, which only gives a value of 0.59. Although we did not measure in detail the height distance between the UAV camera and the tree, it can be seen that there is a difference in the height of the UAV camera between the two images. The distance between the UAV camera and the tree in Figure 9 is longer than the distance between the UAV camera and the tree in Figure 5.

This proves that the distance between the UAV camera and the object is very influential. If we know the distance in detail, it will strengthen previous research [17] that the distance between the camera and the object has an impact on object detection using YOLOv8n. Unfortunately, in this study, we did not analyze the distance between the object and the camera. At the time of data collection, we were only guided to fly the UAV slightly higher than the buildings to avoid collisions with the buildings. We flew the UAV drone at a height between 60-80 meters above the ground, considering that the buildings in the area have a height of about 50 meters.

4 Conclusions

Aerial object (tree) detection using the YOLOv8n model algorithm has been conducted. YOLOv8n can perform object detection well. This can be shown by YOLO's ability to distinguish between trees and buildings. Some photos cannot provide good confidence values due to insufficient training data and lack of variety, for example, between leaf colors and tree shapes. Overall, YOLO is very well used as an algorithm model for image detection, is easy to implement and produces optimal ability to distinguish objects.

Future work will focus on implementing object detection analysis using the YOLO algorithm to measure the open green area will be done perpendicularly, not from the side. Furthermore, we will enrich the train data with yellowish leaves.

We want to thank LPPM USD, as this research was only completed with their financial support.

References

1. B. Battseren, S. H. Mohamed, A. O. A. Diego, S. Saleh, and W. Hardt, Tree Detection and Localization Approach for UAV-based Forest Inspection, in Embedded Selforganising Systems, vol. 9, no. 3, pp. 73–81, (2022)

2. A. Al-Kaff, D. Martin, F. Garcia, A. de la Escalera, and J. M. Armingol, Survey of Computer Vision Algorithms and Applications for Unmanned Aerial Vehicles, in Expert Syst Appl, vol. 92, pp. 447–463, (2018)
3. M. S. Khatami, Deteksi Kendaraan Menggunakan Algoritma You Only Look Once (Yolo) V3, (2022)
4. L. Rajasekar, D. Sharmila, M. Rameshkumar, and B. S. Yuwaraj, Performance Analysis of Fruits Classification System Using Deep Learning Techniques, in Advances in Machine Learning and Computational Intelligence: Proceedings of ICMLCI, Springer, 2021, pp. 293–301, (2019)
5. C.-Y. Chiang, C. Barnes, P. Angelov, and R. Jiang, Deep Learning-Based Automated Forest Health Diagnosis From Aerial Images, in IEEE Access, vol. 8, pp. 144064–144076, (2020)
6. T.-Y. Lin et al., Microsoft coco: Common objects in context, in Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13, Springer, pp. 740–755, (2014)
7. A. Krizhevsky, CIFAR dataset. Accessed: Sep. 23, 2023. [Online]. Available: https://www.cs.toronto.edu/~kriz/cifar.html (2023)
8. S. Rath, Yolov8 Ultralytics: State-of-the-Art Yolo Models, (2023)
9. A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, Yolov4: Optimal Speed and Accuracy of Object Detection, arXiv preprint arXiv:2004.10934, (2020)
10. J. Redmon and A. Farhadi, Yolov3: An Incremental Improvement, arXiv preprint arXiv:1804.02767, (2018)
11. D.-J. Shin and J.-J. Kim, A Deep Learning Framework Performance Evaluation to Use Yolo in NVIDIA Jetson Platform, in Applied Sciences, vol. 12, no. 8, p. 3734, (2022)
12. J. Terven and D. Cordova-Esparza, A comprehensive review of YOLO: From YOLOv1 to YOLOv8 and Beyond, arXiv preprint arXiv:2304.00501, (2023)
13. G. Li, Z. Song, and Q. Fu, A New Method of Image Detection for Small datasets under the framework of YOLO network," in 2018 IEEE 3rd Advanced Information Technology, Electronic and Automation Control Conference (IAEAC), IEEE, pp. 1031–1035, (2018)
14. D. Ahmed, R. Sapkota, M. Churuvija, and M. Karkee, Machine Vision-Based Crop-Load Estimation Using YOLOv8, arXiv preprint arXiv:2304.13282, (2023)
15. M. Shaloo, G. Princz, and S. Erol, Real-Time Color Detection for Automated Production Lines Using CNN-Based Machine Learning, in International Symposium on Industrial Engineering and Automation, Springer, pp. 167–181, (2023)
16. X. Liu, D. Zhao, W. Jia, W. Ji, and Y. Sun, A Detection Method for Apple Fruits Based on Color and Shape Features, in IEEE Access, vol. 7, pp. 67923–67933, (2019)
17. J. A. J. Eikelboom et al., Improving the Precision and Accuracy of Animal Population Estimates with Aerial Image Object Detection, Methods Ecol Evol, vol. 10, no. 11, pp. 1875–1887,(2019)