

Comparison between convolutional neural network and K-nearest neighbours object detection for autonomous drone

Annisa Istiqomah Arrahmah, Rissa Rahmania, Dany Eka Saputra

Department of Computer Science, School of Computer Science, Bina Nusantara University Bandung Campus, Jakarta, Indonesia

Article Info

Article history:

Received Mar 11, 2022

Revised May 28, 2022

Accepted Jun 11, 2022

Keywords:

Autonomous drones
Convolutional neural network
K-nearest neighbours
Object detection
Pinhole model

ABSTRACT

In autonomous drones, the drone's ability to move depends on the drone's capacity to know its position, either in relative or absolute position. The Pinhole model is one of the methods to calculate a drone's relative position based on the triangle similarity concept using a single camera. This method utilizes bounding box information generated from an object detection algorithm. Thus, accuracy of the generated bounding box is crucial, and selection of object detection algorithm is necessary. This paper compares and evaluates machine learning and deep learning object detection methods to determine which method is suitable for distance measurement using a single camera for autonomous drone's controller based on pinhole model. A novel K-nearest neighbours-based (KNN-based) object detection is constructed to represent the machine learning method while you only look once version 5 (YOLOv5) convolutional neural network (CNN) architecture is selected to represent the deep learning method. A dataset consists of two different classes, with a total of 1520 images, collected from the unmanned aerial vehicle (UAV) camera for training and evaluation purposes. Confusion matrix and intersection over union (IoU)/generalized intersection of union (GIoU) matrix are used to evaluate the performance of both methods. The result of this paper shows the performance of each system and concludes the suitable type of object detection algorithm for the autonomous UAV purpose.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



Corresponding Author:

Annisa Istiqomah Arrahmah

Department of Computer Science, School of Computer Science

Nusantara University Bandung Campus

St. Pasir Kaliki No.25-27, Ciroyom, Kec. Andir, Bandung, West Java 40181, Jakarta, Indonesia

Email: annisa.arahmah@binus.ac.id

1. INTRODUCTION

Nowadays, unmanned aerial vehicle (UAV), commonly known as drone technology, gets special attention not only in industries but also among researchers. To date, drone technology has been mainly used prior to military purposes [1]. But recently, drones have been widely used for private purposes such as monitoring agricultural crops and mining [2], recognizing traffic flow [3], investigation of disaster damage [4], and delivering goods to difficult areas [5], even for cinematic purposes [6]. There are two device units commonly used by human operators to control the drone: the ground unit and the airborne unit. A human operator controls and receives feedback from the airborne unit by using a ground unit. An airborne unit is a flying unit that is utilized with global positioning system (GPS), a camera and another light device used for a specific operation. An airborne unit uses low computing devices compared to ground units due to weight limitations. In autonomous drones, human operators are replaced by artificial intelligence to control the airborne unit. Practically, autonomous drones can be categorized as flying robots.

Research on autonomous drones generally focuses on the maneuverability of the drone's airborne unit. A drone's ability to maneuver depends on the drone's basic ability to know its position, either in absolute position (based on earth's latitude and longitude coordinates) or relative position (based on certain reference points). A typical drone uses GPS and inertial measurement unit (IMU) device to determine its position [7], [8]. In several environments, GPS signal cannot be received by drone while IMU depends on initial reference and is affected by errors from IMU device. Other approaches can be done by utilizing the drone's camera and computer vision to detect obstacles and calculate its position relative to the obstacle [9], [10]. One method to calculate obstacle position using a single camera is by using the pinhole model [11], [12]. This method utilizes bounding box information generated from the object detection algorithm and calculate its position relative to the obstacle using triangle similarity concept [13]. This method depends on the accuracy of bounding box information created from the object detection algorithm; thus, selection of this algorithm becomes crucial. Object detection algorithm can be done either by using machine learning or deep learning algorithm [14]. Both methods have their own advantages and disadvantages.

There are several studies that have been focused on reviewing and/or comparing object detection algorithms, especially comparing machine learning and deep learning algorithms. Zou *et al.* [14], provide a review of more than 400 object detection papers yielding deep analysis of its technical evolution and recent state of the art detection methods. The methods are divided into two periods: traditional object detection period and deep learning-based detection period. Argawal *et al.* [15], compare object detection algorithm specific to deep learning algorithm with various detection applications. Lu *et al.* [9], specifically review vision-based method using deep learning for UAVs collision avoidance. Gauman and Leibe [16] analyze the generic object detection process based on the machine learning method. This research compares classification-based object detection and part-based model detection. Aposporis [17] reviews and summarizes object detection methods that can be used for UAVs based on previous research. He categorizes them into two main methods, machine learning-based and convolutional neural network-based (CNN-based). All the previous papers show the most effective object detection method among the methods being compared. However, there is no work that is found in evaluating and comparing machine learning and deep learning object detection method with the same dataset specifically for UAV purpose.

In this paper, the implementation of object detection based on machine learning and deep learning algorithm for distance measurement using pinhole model in UAV autonomous controller are compared and reviewed. K-nearest neighbours-based (KNN-based) object detection is selected to represent machine learning based object detection while CNN is selected to represent deep learning-based object detection. A novel KNN object detection algorithm based on color histogram feature extraction and you only look once version 5 (YOLOv5) architecture is selected for both systems. A dataset consisting of two different classes is collected from UAV cameras for training and evaluating both systems. Result evaluation is done by using confusion matrix for classification accuracy and generalized intersection of union (GIoU)/intersection over union (IoU) loss matrices for localization loss function.

This paper is divided into the following sections: section 1 presents the background of the research and the previous work related to this paper. Section 2 provides the related works for this research. Section 3 discusses the method used in this research. Section 4 shows the testing result and discussion. The conclusion is given in section 5.

2. RELATED WORKS

There are several machine learning-based classifier algorithms that can be used for object detection. One of the common algorithms is K-nearest neighbors classifier. There are several studies that have been conducted for object detection purposes using KNN algorithm. Putra *et al.* [18], utilizes histogram of oriented gradient (HOG) feature extraction and KNN classification to detect vehicles in highway. The bounding box size used in this research is fixed. Schmitt McCoy [19], utilizes speeded-up robust features (SURF) feature extraction and KNN classification to detect more than one object. A multi-level grid is used to detect the desired object. Localization process is done by using voting method on each level; thus, the detection process is a time-consuming process. Putri *et al.* [20], uses scale invariant feature transform (SIFT) and KNN classifier to detect and track the object for motor control on humanoid robot. The object interest is placed in a discarded background thus, random sample consensus (RANSAC) algorithm is used for the localization process rather than sliding window method. The purpose of the object detection process in this research is to detect objects for UAV/drone positioning. Thus, the object detection process is done in a real time process with a cluttered background and a time-consuming process needs to be avoided.

Basic approach on object detection using machine learning treats category detection as an image classification process [16]. A feature representation and a trained classifier is used to distinguish the class of interest from anything else using those features. The decision can be made using the decision value of the

classifier to determine the presence of the interest object in the tested image. If the desired object is embedded amid the clutter or background image, a window search can be inserted. Then, all possible sub-windows of the image are tested with the trained classifier. Thus, on each window, a result is obtained whether the window contains the desired object or not. To comply with all the required conditions for the drone's positioning, research is conducted using the object detection method. The objective of the algorithm is to distinguish between two different types of objects and to localize the object then draw a bounding box around the object using classifier method.

CNN, a class of deep neural networks, has been widely used for object detection applications. To date, many researchers have developed primitive CNN into a wide variety of architecture for object detection purposes. There are two types of CNN object detection scheme architecture namely one-stage and two-stage detector [21], [22].

The objection of this research is to detect and differentiate two different objects for UAV/drone positioning using pinhole model in real-time. Thus, one stage object detector YOLOv5 architecture, developed by Pham *et al.* is chosen [23]. This architecture is the newest YOLO architecture with outstanding performance compared to all previous versions [24]. This architecture is built in Python programming language, which makes installation and integration on embedded devices easier. As with any other one stage object detector, in YOLOv5 architecture, the localization and classification process are implemented at the same time (dense detection) [24]. By using YOLOv5 object detector for UAV positioning purpose, this research is conducted into three phases: data preprocessing, model training and inference.

3. METHOD

The research is conducted according to a basic machine learning process, which consists of data preparation, training/modelling, and testing Figure 1. The three processes are conducted for KNN and CNN algorithm. The data collected from the Testing phase are analyzed to determine each algorithm's performance. The details of each phase are explained in the next section.

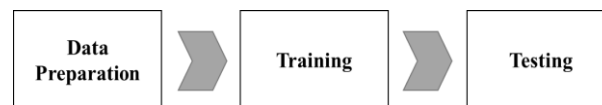


Figure 1. The research method

2.1. Data preparation phase

During the data preparation process, two different kinds of datasets are made for the KNN object detection and CNN object detection. For the KNN object detection, a self-made dataset is chosen for the training process. The dataset consists of two kinds of boxes with different sizes (15 cm and 30 cm) and patterns. Example of the dataset can be seen in Figure 2. Figure 2(a) shows the 15 cm label dataset while Figure 2(b) shows the 30 cm label dataset. Each image is captured with a different angle from an indoor area with a cluttered background. A total of 1520 data is collected with 1:1 ratio between 15 cm box and 30 cm box. All the dataset is divided into three parts: 70% for training, 20% for validation, and 10% for testing. Two types of datasets are prepared for the object localization and object classification process. In the first dataset, each image's file name is labelled with 'small' and 'big' based on the size and the pattern of the objects. This dataset is used for the object classification process. For the object localization process, a new dataset is produced by using the previous dataset. Each image from the previous dataset is split into 100 pieces, called window. Each window is labelled as 'background' shown in Figure 3(a) and 'object' shown in Figure 3(b). A window is labelled as an object if it contains part of both boxes. A window is labelled as background if there is no part of the box on the window or less than 10% of the box's part is contained on the window. A total of 21.196 data is gathered with 1:1 ratio. The second dataset is used to classify the input image to determine which window contains the object; thus, a bounding box can be created around the selected windows.

For the CNN object detection, the dataset being used is the same as the previous dataset used in KNN object detection. It consists of 1520 images containing two unique objects of varied sizes (15 cm and 30 cm) taken in different angles, background, and distance Figure 2. Object labelling is conducted for each image in the dataset. The purpose of labelling is to get a bounding box that matches the truth location of the desired object in the image, called ground truth. This process aims to get information containing the location of the detected object in the image in the form of pixel coordinate of the region of interest (ROI) box top left

and bottom right of the interested object. The labelling process using Roboflow is shown in Figure 4 and the format for the ground truth bounding box in YOLO are class, x_{center} , y_{center} , width and height. The final dataset is then divided into 70% for training, 20% for validation and 10% for testing.



Figure 2. Example of the box dataset (a) box with 15 cm size and (b) box with 30 cm size

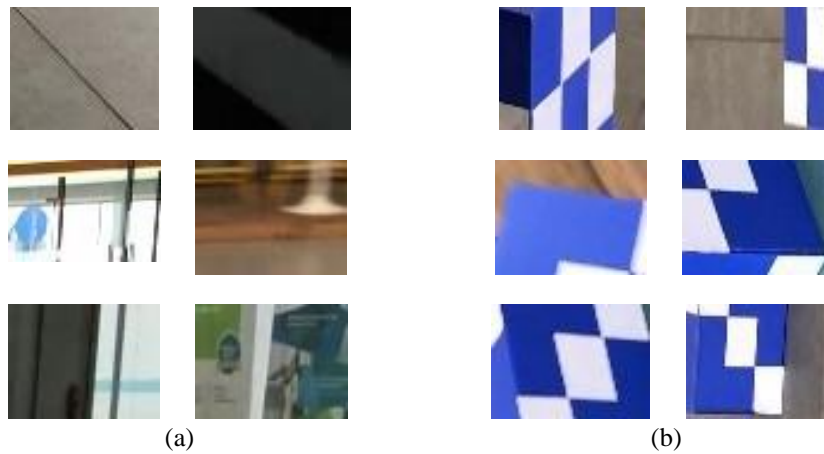


Figure 3. Example of the background dataset (a) background and (b) object



Figure 4. Bounding box and labelling result in YOLO format

2.2. Training phase

In the KNN object detection training process, two models are generated for classification and localization purposes. A feature extraction method is selected to extract features between classified datasets.

Color histogram feature extraction technique is utilized to extract global features of the image in the dataset [25]. The extraction method is used both for classification and localization process. Minkowski distance metric is used for each training. Value of neighbor (k) for each training is selected experimentally with the highest training accuracy result. The first training generates model with k=3. This model uses the first dataset for classification purposes. The second training generates model with k=331. This model uses the second dataset for localization purposes.

In the CNN model training and evaluation stage, architecture, and original model configuration of the YOLOv5 is being tuned following the purpose of the research. The system consists of two classes, namely 0 for 30cm box and 1 for 15 cm box. The YOLOv5 structure is chosen with the depth and the width of the neural network channel are 0.33 and 0.5 respectively. The anchor boxes parameter shown in Figure 5 is auto learned based on the training data. The batch size and epoch used in the research are 16 and 300, as the validation result shows the highest accuracy value.

```

# anchors
anchors:
  - [10,13, 16,30, 33,23] # P3/8
  - [30,61, 62,45, 59,119] # P4/16
  - [116,90, 156,198, 373,326] # P5/32

# YOLOv5 backbone
backbone:
  # [from, number, module, args]
  [[-1, 1, Focus, [64, 3]], # 0-P1/2
  [-1, 1, Conv, [128, 3, 2]], # 1-P2/4
  [-1, 3, C3, [128]],
  [-1, 1, Conv, [256, 3, 2]], # 3-P3/8
  [-1, 9, C3, [256]],
  [-1, 1, Conv, [512, 3, 2]], # 5-P4/16
  [-1, 9, C3, [512]],
  [-1, 1, Conv, [1024, 3, 2]], # 7-P5/32
  [-1, 1, SPP, [1024, [5, 9, 13]]],
  [-1, 3, C3, [1024, False]], # 9
  ]

# YOLOv5 head
head:
  [[-1, 1, Conv, [512, 1, 1]],
  [-1, 1, nn.Upsample, [None, 2, 'nearest']],
  [[-1, 6], 1, Concat, [1]], # cat backbone P4
  [-1, 3, C3, [512, False]], # 13

  [-1, 1, Conv, [256, 1, 1]],
  [-1, 1, nn.Upsample, [None, 2, 'nearest']],
  [[-1, 4], 1, Concat, [1]], # cat backbone P3
  [-1, 3, C3, [256, False]], # 17 (P3/8-small)

  [-1, 1, Conv, [256, 3, 2]],
  [[-1, 14], 1, Concat, [1]], # cat head P4
  [-1, 3, C3, [512, False]], # 20 (P4/16-medium)

  [-1, 1, Conv, [512, 3, 2]],
  [[-1, 10], 1, Concat, [1]], # cat head P5
  [-1, 3, C3, [1024, False]], # 23 (P5/32-large)

  [[17, 20, 23], 1, Detect, [nc, anchors]], # Detect(P3, P4, P5)

```

Figure 5. YOLOv5 parameters value: anchors, backbone, and head

2.3. Testing and experiment

For the testing phase, an experiment is conducted. The experiment ran each algorithm based on live video feed from the drone. The target object is placed in a fixed position. Then, the drone is flown from several positions relative to the target object. The drone is positioned so that the camera always captures the target object. Next, the captured images from the video are inserted to the CNN and KNN object detection for the inference processing.

For KNN algorithm, object detection via classification is done by utilizing two models generated from the classification training process. Detailed flowchart of the object detection process for one frame image can be seen in Figure 6. The input video from the drone's camera is converted into frames. Each frame is being resized into 640 x 480. Then, a color histogram feature extraction is performed to the image. The first model of KNN classifier is used to classify the image. This phase generates the class label of each image depending on the box detected in the image. The next step is splitting the image into several grids/windows. On each window, feature extraction is executed and a second model of KNN classifier is used to classify between two classes, namely the background and non-background or object. If a window belongs to object class, then this window is included in the bounding box area of interest. The bounding box size information is updated according to the classification result of all the windows images. Thus, the object location in the image is obtained based on the pixel location of the windows that are categorized as object class.

For the inference process of CNN object detection, a trained weight obtained from the previous step is used to identify two boxes on any image obtained from the video frames recorded from the UAV camera. If the presence of the desired box is detected, a bounding box is drawn around the boxes and the probability of the object between 15 cm or 30 cm box is displayed. The architecture rebuilt with the trained weight is used to predict the object in the image. During the prediction process, mean average precision (mAP), precision and recall of each bounding box are computed using non-max suppression [26].

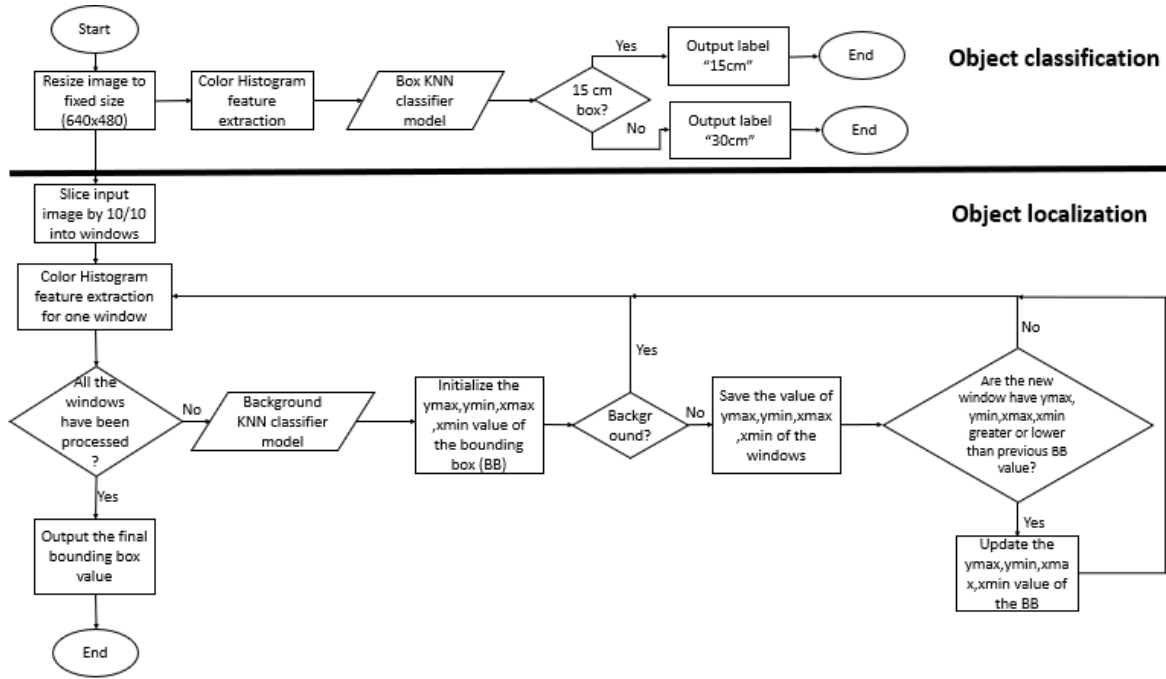


Figure 6. KNN object detection inference process

4. RESULTS AND DISCUSSION

An evaluation is done to compare the result between the two mentioned methods. A total of 164 new images captured from drones are used to test both systems. Confusion matrix is generated during the test on both algorithms to compare the accuracy of the classification process. Ground truth is generated from the testing dataset and GIoU/IoU is used to evaluate the object detection inference. Example of KNN object detection can be seen in Figure 7(a) while the result of CNN object detection can be seen in Figure 7(b).

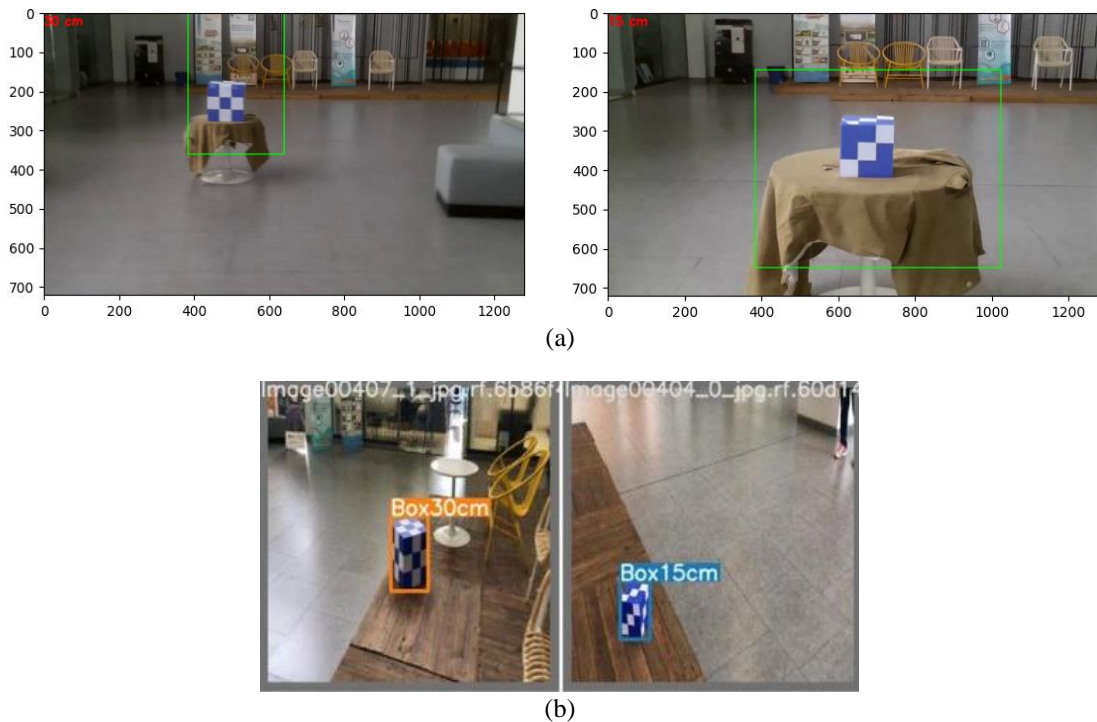


Figure 7. Result of (a) object detection using KNN and (b) object detection using CNN

4.1. Accuracy as confusion matrix

Classification performance between two classes of both object detection algorithms is evaluated using confusion matrix. Confusion matrix for KNN-based classification and YOLOv5 classification is shown in Figures 8 and 9, respectively. In the KNN object detection, the box 15 cm class is denoted as 1 while the box 30 cm class is denoted as 0 and there is no background class. Thus, the generated confusion matrix purely evaluates the classification result only. Based on Figure 8, the true positive (TP) value is 0.83, true negative (TN) value is 0.8, false positive (FP) value 0.2, and false negative (FN) value is 0.17. Different from the previous algorithm, because the YOLOv5 algorithm is a dense detection, the confusion matrix generated from the YOLOv5 algorithm includes both localization and classification result. Based on Figure 9, the TP value of box 15 cm class is 0.99, the TP value of box 30 cm class is 0.99. The background FN value of each class is 0.01 and there is no prediction error between box classes. The background FP predicted as box 15 cm is 0.71 while predicted as box 30 cm is 0.29, means that sometimes even if there is no box in the pictures, a bounding box is still created. Based on the classification result, TP rate for CNN based object detection (99%) is higher than KNN based classification (80-83%).

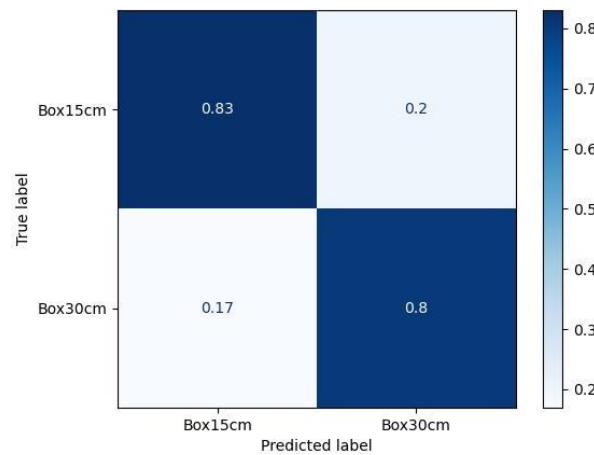


Figure 8. Classification performance of KNN-based object detection

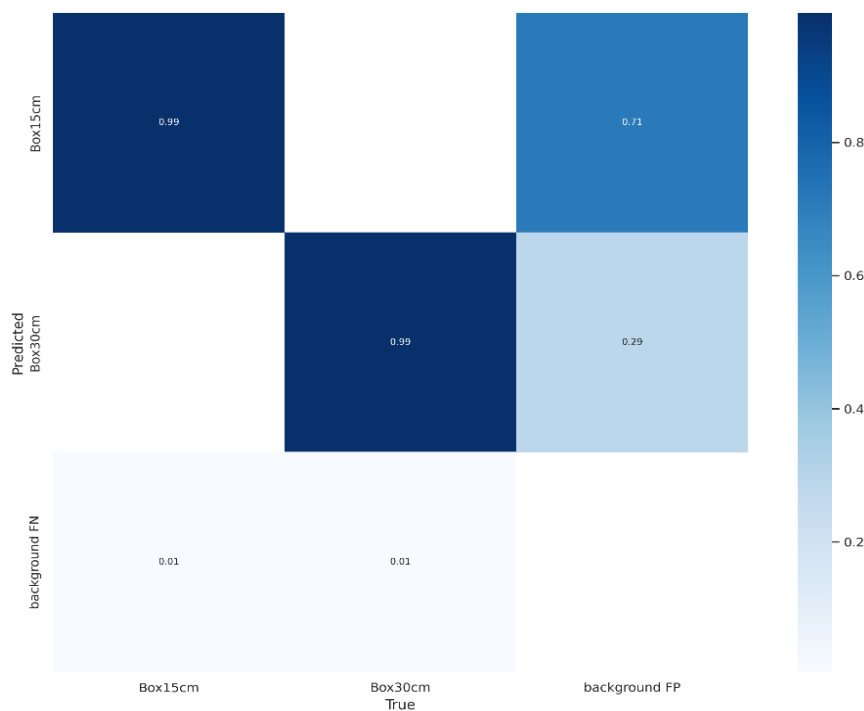


Figure 9. Classification performance of CNN-based (YOLOv5) object detection

4.2. GIoU/IoU bounding box regression

The localization result of both algorithms can also be expressed using GIoU/IoU evaluation metric. IoU metric evaluation is used for the KNN-based object detection, while GIoU metric evaluation is used for YOLOv5 object detection. First, ground truth bounding boxes from the testing set are generated. In KNN-based object detection algorithm, predicted bounding boxes from the model are generated. Intersection of Union is determined using (1). IoU evaluation metric is selected because machine learning based object detection has varying parameters beyond the classification algorithm itself. Several images are selected from the test set to represent the IoU value of the algorithm. Figure 10(a) shows the IoU evaluation on ten images of the test dataset. The result shows that the average IoU value is 0.1156, indicating that the bounding box generated from the KNN algorithm narrowly overlaps with the ground truth. Thus, KNN is a poor algorithm for object detection applications because the IoU value is small.

$$IoU = \frac{\text{Area of overlap}}{\text{Area of Union}} \quad (1)$$

In YOLOv5 object detection, tensorflow's GIoULoss loss metric (commonly used for deep learning object detection algorithm) is used to calculate the intersection of union. The implementation is based on bounding box regression loss calculation introduced by Rezatofighi *et al.* [27]. While the IoU loss metric focuses only on the overlap area, the GIoU loss metric shows the optimal loss metric by comparing each new prediction closeness with the ground truth. Figure 10(b) shows the GIoU loss result on validation dataset based on (2). Based on the result, the GIoU loss is close to zero, which also means that the GIoU value is close to one. Therefore, the YOLOv5 is a much better algorithm compared to KNN based algorithm.

$$L_{GIoU} = 1 - GIoU \quad (2)$$

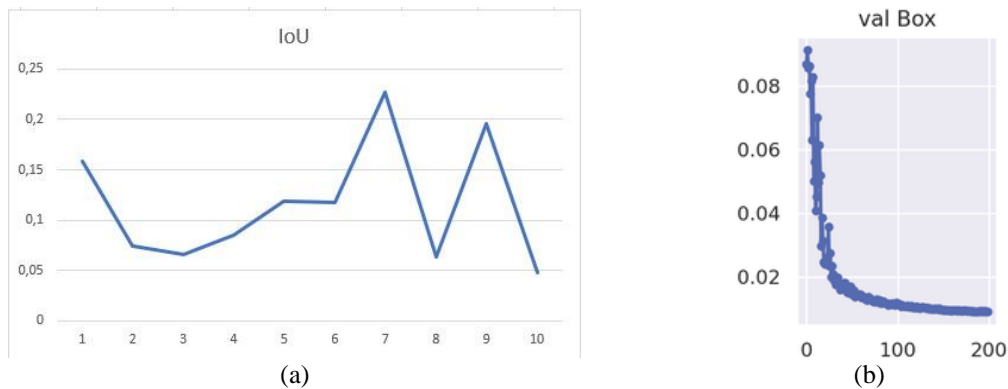


Figure 10. Bounding box regression based on (a) IoU for KNN-based object detection and (b) GIoU loss for CNN-based object detection

5. CONCLUSION

In this paper, comparison between two object detection algorithms to determine distance for UAV autonomous controller using pinhole model is evaluated. Dataset with a total of 1,520 images is collected from UAV camera with two different classes (30 cm and 15 cm boxes) for training and evaluation purposes. The first algorithm is based on novel KNN-based object detection. In this system, the classification and localization process are done separately. Both processes are implemented using KNN classifier with color histogram feature extraction. The second algorithm is based on CNN one stage detector, i.e., YOLOv5 architecture. This system is a dense detector where the classification and localization process are implemented at the same time. The evaluation of both systems is done using confusion matrix and IoU/GIoU loss metrics. The confusion matrix result shows that the CNN classification is slightly better than KNN classifier while classifying between two classes (30 cm and 15 cm boxes). TP value of both classes in KNN classifier are 0.8 and 0.83 respectively. TP value of both classes in CNN classifier are 0.99. The IoU/GIoU result shows that CNN-based object detection is much better than KNN-based object detection. The IoU for KNN-based object detection is close to zero (the average value is 0.1156) while the GIoU for CNN-based object detection is close to one. While object detection is used to determine obstacle distance from the drone's position using triangle similarity concept in pinhole model, the accuracy of the bounding box size

information is crucial. In conclusion, because the GIOU loss and confusion matrices shows that CNN is better than KNN, thus, the YOLOv5 object detection is a better choice for object detection especially for autonomous UAV purpose.

ACKNOWLEDGEMENTS

The authors would like to thank RTTO BINUS University for their support in this research. This research is funded by the International Research Grant from BINUS University, with contract number 017/VR.RTT/III/2021.





REFERENCES

- [1] A. C. Doctor and J. I. Walsh, "The coercive logic of militant drone use," *The US Army War College Quarterly: Parameters*, vol. 51, no. 2, pp. 73–84, 2021, doi: 10.55540/0031-1723.3069.
- [2] D. C. Tsouros, S. Bibi, and P. G. Sarigiannidis, "A review on UAV-based applications for precision agriculture," *Information*, vol. 10, no. 11, p. 349, Nov. 2019, doi: 10.3390/info10110349.
- [3] F. Outay, H. A. Mengash, and M. Adnan, "Applications of unmanned aerial vehicle (UAV) in road safety, traffic and highway infrastructure management: Recent advances and challenges," *Transportation Research Part A: Policy and Practice*, vol. 141, pp. 116–129, Nov. 2020, doi: 10.1016/j.tra.2020.09.018.
- [4] S. S. Kim, T. H. Kim, and J. S. Sim, "Applicability assessment of uav mapping for disaster damage investigation in Korea," in *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 42, no. 3/W8, pp. 209–214, Aug. 2019, doi: 10.5194/isprs-archives-XLII-3-W8-209-2019.
- [5] M. Eichleay, E. Evens, K. Stankevitz, and C. Parker, "Using the Unmanned Aerial Vehicle Delivery Decision Tool to Consider Transporting Medical Supplies via Drone," vol. 7, no. 4, pp. 500–506, Dec. 2019, doi: 10.9745/GHSP-D-19-00119. [Online]. Available: <https://www.ghspjournal.org/content/7/4/500.short>.
- [6] S. Lee and Y. Choi, "Reviews of unmanned aerial vehicle (drone) technology trends and its applications in the mining industry," *Geosystem Engineering*, vol. 19, no. 4, pp. 197–204, 2016, doi: 10.1080/12269328.2016.1162115.
- [7] H. D. K. Motlagh, F. Lotfi, H. D. Taghirad, and S. B. Germi, "Position Estimation for Drones based on Visual SLAM and IMU in GPS-denied Environment," *2019 7th International Conference on Robotics and Mechatronics (ICRoM)*, 2019, pp. 120–124, doi: 10.1109/ICRoM48714.2019.9071826.
- [8] M. Kan, S. Okamoto, and J. H. Lee, "Development of drone capable of autonomous flight using GPS," *Lecture Notes in Engineering and Computer Science*, vol. 2, Mar. 2018.
- [9] Y. Lu, Z. Xue, G. S. Xia, and L. Zhang, "A survey on vision-based UAV navigation," *Geo-Spatial Information Science*, vol. 21, no. 1, pp. 21–32, 2018, doi: 10.1080/10095020.2017.1420509.
- [10] F. Bonin-Font, A. Ortiz, and G. Oliver, "Visual navigation for mobile robots: A survey," *Journal of Intelligent and Robotic Systems: Theory and Applications*, vol. 53, no. 3, pp. 263–296, 2008, doi: 10.1007/s10846-008-9235-4.
- [11] D. E. Saputra, A. S. M. Senjaya, J. Ivander, and A. W. Chandra, "Experiment on Distance Measurement Using Single Camera," 2021, pp. 80–85, doi: 10.1109/icoiaact53268.2021.9564010.
- [12] C. Chen *et al.*, "Obtaining world coordinate information of UAV in gnss denied environments," *Sensors*, vol. 20, no. 8, p. 2241, Apr. 2020, doi: 10.3390/s20082241.
- [13] R. K. Megalingam, V. Shriram, B. Likhith, G. Rajesh, and S. Ghanta, "Monocular distance estimation using pinhole camera approximation to avoid vehicle crash and back-over accidents," *2016 10th International Conference on Intelligent Systems and Control (ISCO)*, 2016, pp. 1–5, doi: 10.1109/ISCO.2016.7727017.
- [14] Z. Zou, Z. Shi, Y. Guo, and J. Ye, "Object Detection in 20 Years: A Survey," *arXiv preprint arXiv:1905.05055*, pp. 1–39, 2019, doi: 10.48550/arXiv.1905.05055.
- [15] S. Agarwal, J. O. D. Terrail, and F. Jurie, "Recent advances in object detection in the age of deep convolutional neural networks," *arXiv preprint arXiv:1809.03193*, 2018, doi: 10.48550/arXiv.1809.03193.
- [16] K. Grauman and B. Leibe, "Visual object recognition," *Synthesis lectures on artificial intelligence and machine learning*, vol. 5, no. 2, pp. 1–181, 2011, doi: 10.1007/978-3-319-77223-3_10.
- [17] P. Aposporis, "Object detection methods for improving UAV autonomy and remote sensing applications," *Proceedings of the 2020 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, ASONAM*, 2020, pp. 845–853, doi: 10.1109/ASONAM49781.2020.9381377.
- [18] F. A. I. A. Putra, F. Utamingrum, and W. F. Mahmudy, "HOG feature extraction and knn classification for detecting vehicle in the highway," *IJCCS (Indonesian Journal of Computing and Cybernetics Systems)*, vol. 14, no. 3, pp. 231–242, 2020, doi: 10.22146/ijccs.54050.
- [19] D. Schmitt and N. McCoy, "Object classification and localization using SURF descriptors," *CS*, vol. 229, pp. 1–5, 2011.
- [20] D. I. H. Putri, Martin, Riyanto, and C. Machbub, "Object detection and tracking using SIFT-KNN classifier and Yaw-Pitch servo motor control on humanoid robot," *2018 International Conference on Signals and Systems, ICSigSys 2018-Proceedings*, 2018, pp. 47–52, doi: 10.1109/ICSIGSYS.2018.8373566.
- [21] M. Carranza-García, J. Torres-Mateo, P. Lara-Benítez, and J. García-Gutiérrez, "On the performance of one-stage and two-stage object detectors in autonomous vehicles using camera data," *Remote Sensing*, vol. 13, no. 1, pp. 1–23, 2021, doi: 10.3390/rs13010089.
- [22] L. Du, R. Zhang, and X. Wang, "Overview of two-stage object detection algorithms," *Journal of Physics: Conference Series*, vol. 1544, no. 1, p. 012033, May 2020, doi: 10.1088/1742-6596/1544/1/012033.
- [23] M. T. Pham, L. Courtrai, C. Friguet, S. Lefèvre, and A. Baussard, "YOLO-fine: One-stage detector of small objects under various backgrounds in remote sensing images," *Remote Sensing*, vol. 12, no. 15, p. 2501, Aug. 2020, doi: 10.3390/RS12152501.
- [24] D. Thuan, "Evolution of Yolo algorithm and Yolov5: the state-of-the-art object detection algorithm," *Oulu University of Applied Science*, 2021.
- [25] M. Auleria, A. I. Arrahmah, and D. E. Saputra, "A review on K-N nearest neighbour based classification for object recognition," *2021 International Conference on Data Science and Its Applications (ICoDSA)*, 2021, pp. 274–280, doi: 10.1109/ICoDSA53588.2021.9617466.





- [26] J. Hosang, R. Benenson, and B. Schiele, "Learning non-maximum suppression," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 4507–4515.
- [27] H. Rezatofghi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid, and S. Savarese, "Generalized intersection over union: a metric and a loss for bounding box regression," *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 658–666, doi: 10.1109/CVPR.2019.00075.

BIOGRAPHIES OF AUTHORS







Annisa Istiqomah Arrahmah     received Bachelor of Science in Electrical Engineering from Bandung Institute of Technology, Indonesia in 2015. She received Master of Engineering and Master of Science in 2016 and 2017 from a dual-degree master program in interdisciplinary program of information systems, Pukyong National University, Busan, South Korea and School of Electrical Engineering and Informatics, Bandung Institute of Technology, Bandung, Indonesia. She is currently a researcher in School of Computer Science, Bina Nusantara University, Bandung, Indonesia. Her research interests include bio cryptography, object detection and IoT application. She can be contacted at email: annisa.arahmah@binus.ac.id.



Rissa Rahmania     received a bachelor's degree in Electrical Engineering majoring in Telecommunication Engineering and a master's degree in Electrical Engineering majoring in Telecommunication Engineering from the School of Electrical Engineering, Telkom University, Bandung, Indonesia, in 2015 and 2017, respectively. She is currently a researcher in the school of computer science, Bina Nusantara University, Bandung, Indonesia. Her research interests include signal/image processing, deep learning, and object detection applications. She can be contacted at email: rissa.rahmania@binus.ac.id.



Dany Eka Saputra     currently serves as a Lecturer at Computer Science Department, BINUS @Bandung, BINUS University. He has a bachelor's degree in Aeronautics and Astronautics from Institut Teknologi Bandung in 2007. He also received his Master and Doctoral Degree in the same university in Electrical Engineering and Informatics, in 2012 and 2019 respectively. He has research interest and experience in blockchain, electronic cash, IoT, and autonomous drones. His current research heavily focused on autonomous drone navigation capability. He can be contacted at email: dany.eka@binus.ac.id.