

The Detection of a Single Wet Rice Field Bund on Unmanned Aerial Vehicle Image Using a Convolutional Neural Network

I Made Gede Sunarya¹, I Md Dendi Maysanjaya², Ida Bagus Made Yudha Wirawan³, Gede Budi Setiawan⁴

sunarya@undiksha.ac.id¹, dendi.ms@undiksha.ac.id², bagus.yudha@undiksha.ac.id³

Virtual, Vision, Image and Pattern Research Group, Department of Informatics Engineering, Faculty of Engineering and Vocational, Universitas Pendidikan Ganesha, Jl. Udayana No. 11, Singaraja, Buleleng, Bali, Indonesia - 81116, Indonesia ^{1,2,3}

Abstract. This research is a preliminary study in the development of UAV in intelligent agriculture. This study aimed to detect Single Wet Rice Field Bund UAV Image Using CNN. The data used in this research were 170 images divided into Training (150), Validation (10), and Testing (10). The proposed stages were image acquisition, pre-processing, image labeling, training, and testing. The inputs were Wet Rice Field Bund UAV images, and the output was a bounding box as the result of rice bund detection. The training was carried out using Google Collaboratory and GPU features using CNN YOLO V5. In the training process, the best Recall, Precision, mAP0.5, mAP0.5:0.95 values were respectively 1, 1, 0.9952, 0.6358 with a total processing time of 0.858 hours. At the testing stage, Bounding boxes were detected at confidence thresholds of 0.1, 0.2, and at confidence 0.3, 0.4, 0.5 were able to detect a bounding box of 80%.

Keywords: automatic detections, bund rice field, UAV images, Deep Learning, CNN.

1 Introduction

The rapid development of information and computer technology affects people's perspective on agriculture technology. Some aspects that used to be done manually and took a long time are now being pushed to be faster and done automatically and digitally. An example is the way of spatial data acquisition in agriculture, which is currently starting to use unmanned aerial vehicles (UAV). Mapping technology with UAV is an alternative in addition to other mapping technologies such as aerial photography, both large and small scale, manned, and satellite-based mapping. This technology is very promising to be applied and developed under the topographical and geographical characteristics of Indonesia, especially for large areas such as rice fields.

UAV is an unmanned aircraft flown using a remote controller, smartphone, or computer. UAVs can be equipped with high-resolution cameras that allow users to monitor a particular location

from a height in real-time. By using UAV, data can be obtained at a relatively low cost, relatively fast time, and safely in various weather conditions. UAV is an unmanned system (Unmanned System), which is an electro-mechanical based system that can carry out programmed missions, with the characteristics of (i) uncrewed aircraft, (ii) operating in fully or partially independent mode, (iii) This system is designed to be used repeatedly. There are several types of UAV, namely fixed-wing and multi-rotor. Fixed-wing drones provide an advantage in range and flight time over multi-rotor. However, Fixed Wing requires a significant open location for take-off and landing. Fixed Wing UAV is suitable for wide-scale mapping surveys such as aerial photography and other surveys. The Multi-Rotor UAV is suitable for mapping that does not have a large open area for take-off and landing. Surveys with multiple rotors provide an advantage in terms of the level of object detail that can be obtained. Multi-Rotor uses several motors as its propulsion, requiring more power sources resulting in reduced range and flight time.

The development of UAV is widely used in mapping implementation [1][2][3][4][5]. In this study, the use of UAVs used to support agriculture was still controlled manually and automatically using waypoints to determine the movement of the UAV. The use of manual or automatic waypoints required determining the limits of the movement of the UAV so that it did not cross the treatment limits of the agricultural area. The automated movement of the UAV by considering the boundaries of the agricultural area in the UAV image has not yet been developed. When UAV is capable of automatically determining the boundaries of the agricultural area, it can move automatically, which is limited to agricultural areas. In determining the boundaries of agricultural areas using UAV imagery, it is necessary to study how the detection and segmentation of vegetation in agricultural areas can be determined so that the boundaries of agricultural areas can be determined. One of the agricultural areas in the wet rice field is a rice field filled with water. One method of detecting and segmenting objects in digital images can use the Deep Learning method. One of the deep learning that is widely used in object segmentation is Convolution Neural Network (CNN) [6][7]. One of the CNN architectures used to detect objects in the form of bounding boxes is You Look Only Once (YOLO). Some researchers use YOLO to detect objects in UAV images [8] [9][10][11][12].

This research is a preliminary study in the development of UAVs in intelligent agriculture. Based on the problems and research done previously, the researcher proposed the detection of a single wet rice field bund on an unmanned aerial vehicle image using a convolutional neural network.

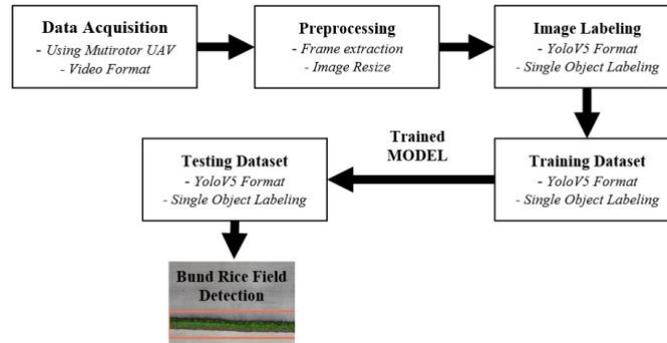


Fig. 1. Research methodology.

2 Methodology

This research method consisted of five stages: image acquisition, pre-processing, image labeling, training, and testing. The inputs in this research were wet rice field bund UAV images and the output was bounding box as the result of rice bund detection. The proposed research chart is shown in **Figure 1**.

2.1 Data acquisition

The image acquisition process is a process to collect research data. The research data was in the form of Wet Rice Field Bund UAV video format. The data were gathered using a multirotor UAV with the DJI Mavic Mini type. The data were taken at an angle of 90 degrees with a height between 25m – 50m from the surface of the rice fields. The resulting data was in video format with a resolution of 1920 x1080 pixels. The data were taken in sunny weather with a time range of 10 am – 12 am. The screenshot of the image on the video from the image acquisition process is shown in **Figure 2**.



Fig. 2. Screenshot image from data acquisition video.

2.2 Pre-processing

The pre-processing stage is the stage used to process the video data from the data acquisition into data used in the following process. In the pre-processing stage, there were two stages, namely frame extraction and image resizing. Frame extraction was done to extract video data into image frames. The resulting images frame had a resolution of 1920 x 1080 pixels. The next stage was Image Resize which aims to change the image size to 416 x 416 pixels. Image Resize was done to reduce the size of the image so as to reduce the computation time. Some pre-processed images can be shown in **Figure 3**.



Fig. 3. Images result of the pre-processing stage.



Fig. 4. Labeling stage and the labeling information.

2.3 Image labeling

The Image Labeling stage is the stage for labeling the rice field objects contained in the wet rice field bund UAV images. The labeling process was carried out on the rice field images dataset. The labeled images contained the coordinates of the ground-truth bounding box. The label information format in YoloV5 was in the form of a text file that contained the coordinates of the bounding box. The research team carried out the labeling process by looking at the position of the rice fields on the UAV images. This stage used the LabelImg application. **Figure 4** shows the stages of labeling and label information data.

2.4 Training dataset

The dataset that has been labeled in the previous stage was trained to produce a model used in the Testing process. The model form was a model that already had a pattern whose results were in the form of weights. These weights were used in the Testing process. The number of images used for training was 150 training images, and the validation data was ten validation images. The training used tools on Google Collaboratory with Graphics Processing Unit (GPU) and Convolutional Neural Network with YOLO (You Look Only One) version 5. YOLOv5 architecture is shown in **Figure 5**.

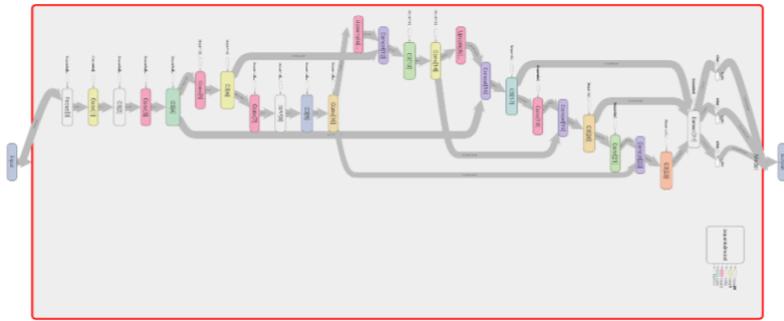


Fig. 5. YOLOv5 architecture [13].

The testing was done by calculating the value of Recall, Precision, F1-Measure, mAP [14]. In the testing stage, it is necessary to calculate the True Positive (FP), True Negative (TN), False Positive (FP), False Negative (FN) values. A Recall is defined as the ratio of the total number of correctly classified positive samples divided by the total number of positive samples. High Recall indicated that the class was recognized correctly. The Recall was calculated by equation (1).

$$Recall = \frac{TP}{TP + FN} \quad (1)$$

The precision value was obtained by dividing the total number of correctly classified positive samples by the total number of positive samples predicted as in equation (2). High Precision indicates positive labeled examples are indeed positive.

$$Precision = \frac{TP}{TP + FP} \quad (2)$$

F1 Score was used to see the balance between Recall and Precision. F1 Score is shown in equations (3).

$$F1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (3)$$

Mean Average Precision (mAP) is the average of the Average Precision values. The calculation of Average Precision (AP) is shown in equation (4) (5).

$$AP = \sum (r_{n+1} - r_n) p_{interp}(r_{n+1}) \quad (4)$$

$$p_{interp}(r_{n+1}) = \max_{\tilde{r} \geq r_{n+1}} p(\tilde{r}) \quad (5)$$

2.5 Testing Dataset

A testing dataset was used to determine the capability of the architectural model generated in the training process. At this stage, 10 images were used to be a testing dataset. At this stage, variations of the confusion threshold value were used with values of 0.1, 0.2, 0.3, 0.4, 0.5. The images used in the testing process are shown in **Figure 6**. The images used were images containing rice fields bund.

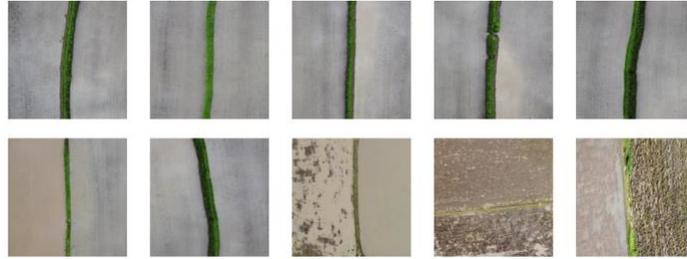


Fig. 6. Images of testing dataset.

3 Result and discussion

At this stage, the research results obtained at the testing stage were presented. The method used in this research was Convolutional Neural Network with YOLO architecture version 5. The data were acquired using a multicopter UAV in the first stage, and produced data in video format. In the pre-processing stage, the frame was extracted to obtain images with a resolution of 1920 x 1080 pixels. To reduce processing, the image was resized to 416 x 416 pixels. After pre-

processing the image, then image labeling was done. The resulting image was an image containing a single bund rice field. The label given was in the form of a bounding box on the part of the area that contains rice fields. The label used was the Pematang name. An image contained only one label. The label was the coordinates of the bounding box stored in a text file. The results of the labeling bounding box process on several images can be seen in **Figure 7**.

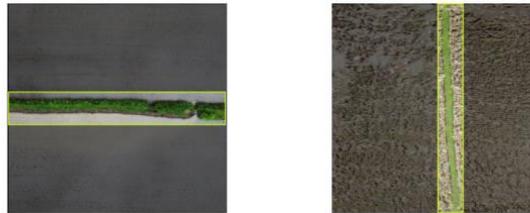


Fig. 7. Images result of labeling stage.

The training stage aimed to generate weights in the model used in the testing process. The training process was carried out using CNN YOLO version 5, run using Google Collaboratory. The training process used the Graphical Processing Unit (GPU) on the Google Collaboratory. The training process on the Training dataset with 150 images data, 500 epochs, 16 batch sizes were carried out in 0.858 hours. The results of the training process are shown in **Figure 8**.

```

Epoch      gpu_mem  box      obj      cls  total  targets  img_size
0/499      1.83G    0.1139  0.01793  0    0.1309  20      416: 100% 10/10 [00:18:00:00, 1.07s/it]
Class      Images  Targets  P      R      mAP@.5  mAP@.5-.95: 100% 1/1 [00:01:00:00, 1.77s/it]
all        20      0        0      0      0        0

Epoch      gpu_mem  box      obj      cls  total  targets  img_size
1/499      1.9G     0.1072  0.01733  0    0.1245  19      416: 100% 10/10 [00:05:00:00, 1.81it/s]
Class      Images  Targets  P      R      mAP@.5  mAP@.5-.95: 100% 1/1 [00:00:00:00, 1.96it/s]
all        20      0        0.000216  0.05  1.24e-05  1.24e-06

.....

Epoch      gpu_mem  box      obj      cls  total  targets  img_size
497/499    1.9G    0.02502 0.000205  0    0.03323  16      416: 100% 10/10 [00:05:00:00, 1.90it/s]
Class      Images  Targets  P      R      mAP@.5  mAP@.5-.95: 100% 1/1 [00:00:00:00, 2.92it/s]
all        20      1        1        1        0.995  0.991

Epoch      gpu_mem  box      obj      cls  total  targets  img_size
498/499    1.9G    0.02455 0.000717  0    0.03327  14      416: 100% 10/10 [00:05:00:00, 1.88it/s]
Class      Images  Targets  P      R      mAP@.5  mAP@.5-.95: 100% 1/1 [00:00:00:00, 2.94it/s]
all        20      1        1        1        0.995  0.990

Epoch      gpu_mem  box      obj      cls  total  targets  img_size
499/499    1.9G    0.02268 0.000244  0    0.03393  13      416: 100% 10/10 [00:05:00:00, 1.90it/s]
Class      Images  Targets  P      R      mAP@.5  mAP@.5-.95: 100% 1/1 [00:00:00:00, 2.46it/s]
all        20      1        1        1        0.995  0.609

Optimizer stripped from runs/train/yolov5s_results/weights/last.pt, 14.79B
500 epochs completed in 0.858 hours.

```

Fig. 8. Result of 500 epoch training stage.

The graph of Recall, Precision and mAP in training using 500 epochs is shown in **Figure 9**.

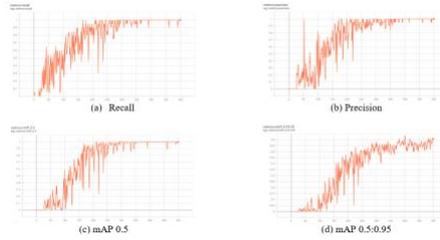


Fig. 9 Recall, precision and mAP graphich for 500 epoch.

Based on the graph, it can be seen that the results obtained have followed the trend of values in a better direction. Visually, in the 500th epoch, graphics Recall, Precision and mAP have shown a stable pattern. The results of the best Recall, Precision, mAP0.5, mAP 0.5:0.95 values respectively were 1, 1, 0.9952, 0.6358. Graphic comparison of Recall, Precision and F1 on the training dataset with confidence is shown in **Figure 10**.



Fig. 10 Recall, precision, F1 training dataset compare by confidence.

The results obtained from the detection process were in the form of bounding boxes in areas containing rice fields. The results of the bounding box detection in the training images are shown in **Figure 11**.



Fig. 11. Bounding box detection result of the training dataset.

After the training process was carried out, a model with updated weights was generated. This model was used in the Testing process. Graphic comparison of Recall, Precision, and F1 on the validation dataset with confidence was shown in **Figure 12**. The results of bounding box detection on the validation images are shown in **Figure 13**.

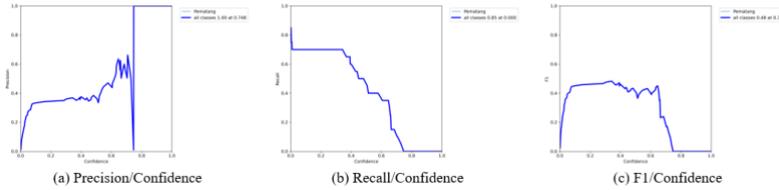


Fig. 12 Recall, precision, F1 validation dataset compare by confidence.

Based on **Figure 10** and **Figure 12**, the table of Recall, Precision, F1 values in the training, and testing dataset is shown in Table 1.



Fig. 13. Bounding box detection result of the validation dataset.

Table 1. Value of recall, precision, F1 with confidence.

	Recall		Precision		F1 Score	
	Value	Confidence	Value	Confidence	Value	Confidence
Training dataset	1	0.00	1.00	0.479	1	0.478
Validation dataset	0.85	0.00	1.00	0.748	0.48	0.342

Table 2. Bounding box detection with confidence threshold.

Images Number	Confidence Threshold					Total
	0.1	0.2	0.3	0.4	0.5	
1	1	1	1	1	1	10
2	1	1	1	1	1	10
3	1	1	1	1	1	10
4	1	1	1	1	1	10
5	1	1	1	1	1	10
6	1	1	1	1	1	10
7	1	1	1	1	1	10
8	1	1	0	0	0	7
9	1	1	0	0	0	7
10	1	1	1	1	1	10
Percentage (%)	100	100	80	80	80	

1 = detected, 0 = undetected

In the testing stage, the model from the training phase was used to detect rice fields in the training dataset. The detection results were varied with confidence-threshold values of 0.1, 0.2, 0.3, 0.4, 0.5. The detection success rate on ten images in the Training dataset is shown in Table 2. The results of bounding box detection in the testing dataset that were detected and those that were not detected are shown in **Figure 14**.



Fig. 14. Bounding box result of the testing dataset.

Based on Table 2, it can be seen that images numbers 8 and 9 were not detected when the confidence threshold was 0.3, 0.4, and 0.5. The detection results of the detected and undetected bounding box testing images can be seen in **Figure 14**.

In **Figure 14**, the detected images have a relatively different color from the background. In the images that have been detected visually, it looks like it had a different background with the color of the rice fields. The rice fields were relatively green and the background had a variety of colors. The images that were not successfully detected visually at a confidence threshold of 0.3, 0.4, 0.5. The color of the rice fields had a color similar to the background so that it had a low confidence value.

4 Conclusions

This study proposed the detection of a single wet rice field bund on unmanned aerial vehicle images using a convolutional neural network. The method of CNN architecture YOLO version 5 was proposed with steps of data acquisition, pre-processing, images labeling, training dataset, testing dataset. The data used in this research were training dataset (150 images), validation dataset (10), testing dataset (10). The training was carried out using Google Collaboratory and GPU features. In the training process, the best Recall, Precision, mAP0.5, mAP 0.5:0.95 values, respectively were 1, 1, 0.9952, 0.6358, with a total processing time of 0.858 hours. At the testing stage, Bounding boxes were detected at confidence thresholds of 0.1 and 0.2 and at confidence 0.3, 0.4, 0.5 were able to detect a bounding box of 80%. In this study, the UAV camera angle was still pointing downwards (90 degrees). The use of camera tilt variations during data acquisition can be continued from this study.

Acknowledgments. This research was financially supported by Skema Penelitian Dasar DIPA Universitas Pendidikan Ganesha 2021.

References

- [1] I. M. G. Sunarya, M. R. Al Affan, A. Kurniawan, and E. M. Yuniarno, "Digital Map Based on Unmanned Aerial Vehicle," *CENIM 2020 - Proceeding Int. Conf. Comput. Eng. Network, Intell. Multimed. 2020*, no. Cenim, pp. 211–216, 2020, doi: 10.1109/CENIM51130.2020.9297883.
- [2] N. Ikhwana and D. R. Hapsari, "Aplikasi Drone Wawasan Tani untuk Pertanian di Simpang Lima, Sungai Besar, Selangor," *J. Pus. Inov. Masy.*, vol. 1, no. 1, pp. 99–104, 2019.
- [3] H. Khoirunisa and F. Kurniawati, "Penggunaan Drone dalam Mengaplikasikan Pestisida di Daerah Sungai Besar, Malaysia," *J. Pus. Inf. Masy.*, vol. 1, no. 1, pp. 87–91, 2019.
- [4] M. Wardani and A. Yudhana, "Rancang Bangun Penyemprot Pestisida untuk Pertanian Padi Berbasis Quadcopter," *J. Ilm. Tek. Elektro Komput. dan Inform.*, vol. 3, no. 2, p. 132, 2018, doi: 10.26555/jiteki.v3i2.7479.
- [5] Fajri, "Pengembangan sistem penyemprotan pada," *Pengemb. Sist. PENYEMPROTAN PADA Platf. PESAWAT TANPA Awak Berbas. QUADROPTER UNTUK MEMBANTU PETANI MENGURANGI BIAYA Pertan. DALAM MENDORONG KONSEP Pertan. Pint. (SMART FARMING) Djarot*, vol. 9, no. 2, pp. 49–56, 2017.
- [6] I. M. G. Sunarya *et al.*, "Deteksi Arteri Karotis pada Citra Ultrasound B-Mode Berbasis Convolution Neural Network Single Shot Multibox Detector," *J. Teknol. dan Sist. Komput.*, vol. 7, no. 2, pp. 56–63, 2019, doi: 10.14710/jtsiskom.7.2.2019.56-63.
- [7] R. A. Pangestu, B. Rahmat, and F. T. Anggraeny, "Implementasi Algoritma CNN untuk Klasifikasi Citra Lahan dan Perhitungan Luas," *Inform. dan Sist. Inf.*, vol. 1, no. 1, pp. 166–174, 2020.
- [8] Q. Wu and Y. Zhou, "Real-time object detection based on unmanned aerial vehicle," *Proc. 2019 IEEE 8th Data-Driven Control Learn. Syst. Conf. DDCLS 2019*, pp. 574–579, 2019, DOI: 10.1109/DDCLS.2019.8908984.
- [9] M. Liu, X. Wang, A. Zhou, X. Fu, Y. Ma, and C. Piao, "Uav-yolo: Small object detection on unmanned aerial vehicle perspective," *Sensors (Switzerland)*, vol. 20, no. 8, pp. 1–12, 2020, DOI: 10.3390/s20082238.
- [10] K. Boudjit and N. Ramzan, "Human detection based on deep learning YOLO-v2 for real-time

- UAV applications,” *J. Exp. Theor. Artif. Intell.*, vol. 00, no. 00, pp. 1–18, 2021, DOI: 10.1080/0952813X.2021.1907793.
- [11] J. Lu *et al.*, “A Vehicle Detection Method for Aerial Image Based on YOLO,” *J. Comput. Commun.*, vol. 06, no. 11, pp. 98–107, 2018, DOI: 10.4236/jcc.2018.611009.
- [12] B. Y. Suprpto, A. Wahyudin, H. Hikmarika, and S. Dwijayanti, “The Detection System of Helipad for Unmanned Aerial Vehicle Landing Using YOLO Algorithm,” *J. Ilm. Tek. Elektro Komput. dan Inform.*, vol. 7, no. 2, pp. 193–206, 2021, doi: 10.26555/jiteki.v7i2.20684.
- [13] Glenn-jocher, “Overview of model structure about YOLOv5,” 2021. <https://github.com/ultralytics/yolov5/issues/4518> (accessed Aug. 30, 2021).
- [14] C. Goutte and E. Gaussier, “A Probabilistic Interpretation of Precision, Recall and F-Score, with Implication for Evaluation,” *Lect. Notes Comput. Sci.*, vol. 3408, no. June 2014, pp. 345–359, 2005, DOI: 10.1007/978-3-540-31865-1_25.