



Article

Rice Seedling Detection in UAV Images Using Transfer Learning and Machine Learning

Hsin-Hung Tseng ^{1,2}, Ming-Der Yang ^{1,2,3,*}, R. Saminathan ³, Yu-Chun Hsu ^{1,2}, Chin-Ying Yang ^{2,4} and Dong-Hong Wu ⁵

- ¹ Department of Civil Engineering, National Chung Hsing University, Taichung 402, Taiwan; d108062001@mail.nchu.edu.tw (H.-H.T.); d107062002@mail.nchu.edu.tw (Y.-C.H.)
- ² Smart Multidisciplinary Agriculture Research and Technology Center, National Chung Hsing University, Taichung 402, Taiwan; emiyang@nchu.edu.tw
- ³ Innovation and Development Center of Sustainable Agriculture, National Chung Hsing University, Taichung 402, Taiwan; sami2k50@yahoo.co.in
- ⁴ Department of Agronomy, National Chung Hsing University, Taichung 402, Taiwan
- ⁵ Crop Science Division, Taiwan Agricultural Research Institute, Taichung 413, Taiwan; dhwu@tari.gov.tw
- * Correspondence: mdyang@nchu.edu.tw; Tel.: +886-4-22840440 (ext. 214)

Abstract: To meet demand for agriculture products, researchers have recently focused on precision agriculture to increase crop production with less input. Crop detection based on computer vision with unmanned aerial vehicle (UAV)-acquired images plays a vital role in precision agriculture. In recent years, machine learning has been successfully applied in image processing for classification, detection and segmentation. Accordingly, the aim of this study is to detect rice seedlings in paddy fields using transfer learning from two machine learning models, EfficientDet-D0 and Faster R-CNN, and to compare the results to the legacy approach—histograms of oriented gradients (HOG)-based support vector machine (SVM) classification. This study relies on a significant UAV image dataset to build a model to detect tiny rice seedlings. The HOG-SVM classifier was trained and achieved an F1-score of 99% in both training and testing. The performance of HOG-SVM, EfficientDet and Faster R-CNN models, respectively, were measured in mean average precision (mAP), with 70.0%, 95.5% and almost 100% in training and 70.2%, 83.2% and 88.8% in testing, and mean Intersection-over-Union (mIoU), with 46.5%, 67.6% and 99.6% in training and 46.6%, 57.5% and 63.7% in testing. The three models were also measured with three additional datasets acquired on different dates to evaluate model applicability with various imaging conditions. The results demonstrate that both CNN-based models outperform HOG-SVM, with a 10% higher mAP and mIoU. Further, computation speed is at least 1000 times faster than that of HOG-SVM with sliding window. Overall, the adoption of transfer learning allows for rapid establishment of object detection applications with promising performance.

Keywords: UAV; machine learning; deep learning; object detection; EfficientDet; Faster R-CNN; transfer learning



Citation: Tseng, H.-H.; Yang, M.-D.; Saminathan, R.; Hsu, Y.-C.; Yang, C.-Y.; Wu, D.-H. Rice Seedling Detection in UAV Images Using Transfer Learning and Machine Learning. *Remote Sens.* **2022**, *14*, 2837. <https://doi.org/10.3390/rs14122837>

Academic Editor: Saeid Homayouni

Received: 6 May 2022

Accepted: 8 June 2022

Published: 13 June 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Demand for agricultural products urges agriculture sectors to accommodate technology to overcome production challenges [1,2]. Prominently, population growth creates constant pressure on the agricultural system to supply more food to fulfil global demand, which drives farmers to adopt modern technologies (such as precision agriculture) in food-crop production [3–6]. Globally, precision agriculture plays an important role in increasing the quality of crop production, sustaining crop production and making decisions based on analyzing large amounts of data and information about crop status obtained from farms. Moreover, it is used for effective fertilizer management and irrigation as well as for labor reduction [7–9]. In practice, remote sensing has been widely used to support precision agriculture in recent years. Sarvia et al., 2021, analyzed the inconsistency between airborne and

satellite sensors by using K-means clustering on NDVI map-derived multispectral remote sensing data and found a high correlation for visible bands. Airborne images provide centimeter geometric resolution and 3D measurement potential for precision agriculture [10]. With the advantages of technology in capturing high-resolution images, particularly by using unnamed aerial vehicle (UAV), large amount of remote sensing data can easily be obtained for analyzing crop yield in precision agriculture.

Through the development of Internet of Things (IoT) and computer vision, sensors and cameras along with machine learning, deep learning and image processing techniques have been getting increasing attention for capturing information and further processing for smart farming to help maintain the sustainability of agricultural production [11]. Smart farming plays a vital role in the agricultural process based on adjusting various agricultural management measures. It provides suggestions and insights for more efficient and effective agriculture production and to solve the challenges in agriculture systems [12]. Several studies have discussed smart farming techniques that were practically implemented to reduce fertilizers, pesticides and herbicides [13] and to estimate optimum crop planting dates [14]. Moreover, computer vision plays a key role to extract useful information from the collected image dataset for management of smart farming tasks [11,15]. In recent agricultural operations, machine learning in computer vision has been applied for various object detection and classification tasks through extracting information from images to significantly promote intelligent agriculture [16–20].

As mentioned above, the developments in IoT provide a good platform to collect a large amount of image data with many objects to make meaningful image analysis [21]. To collect image data in agriculture sectors, UAVs or drones are widely used in precision agriculture and many other fields, such as path planning and design, wildlife rescue, weed classification, harvesting, livestock counting and crop and aquatic products damage assessment [22–27]. UAVs can be used to detect potential issues and then obtain high-resolution images to inspect and apply treatments correspondingly. The combination of UAVs and computer vision helps farmers make correct decision by obtaining information from the images [15]. This study focusses on monitoring sowing area via UAVs for identifying rice seedlings and counting them for decision-making regarding the progress of rice seedlings in paddy fields.

Deep learning, which is one branch of machine learning, in object detection can deal with high-density scenes with complex and small objects in images [28,29]. Object detection in computer vision is widely used for various applications. By training with large amounts of image data, object detection can accurately identify the targeted objects and their spatial locations in the images, classify objects from the specified varieties, such as human, animals, crops, plants, and vehicles, and mask the objects within bounding boxes by well-developed algorithms [30–34]. All existing object detection models can generally perform better to detect either big objects or small objects in the large part of an image. Moreover, the challenging task in computer vision is to detect the small objects in an image that lack appearance information to distinguish them from background and similar categories. The precision requirement is higher for accurately locating small objects. A recent review reported detailed information about the use of convolutional neural networks (CNNs) for small-object detection. Their results, based on popularly existing datasets, showed better performance for detecting small objects in terms of multi-scale feature learning, data augmentation, training strategy, context-based detection and GAN-based detection methods [35]. Based on this evidence, this study aims to employ an object-detection model to monitor a single small object in paddy fields using UAVs. In particular, this study focusses on the rice seedlings in paddy fields as very tiny objects that can hardly be observed by the human eye to find displacements or missing rice seedlings and count and locate rice seedlings.

Object detection based on machine learning in computer vision has improved enormously in accuracy and speed compared to traditional detection algorithms with feature extraction [36]. It is used for classifying and locating objects in automatic image detection

processes based on statistical and geometric features. Traditionally, object detection consists of two stages: feature extraction and object detection [33,37]. Feature extraction models, such as Haar-like feature [38], scale-invariant feature transform (SIFT) [39], histogram oriented gradient feature (HOG) [40], principal components analysis (PCA) [41], Viola–Jones feature [31] and local binary patterns, are used to generate regional proposals. Generally, static objects are detected in images using background subtraction algorithms, and dynamic objects are detected by subtracting two adjacent frames with frame difference algorithms. Then, the extracted features are inserted into support vector machine (SVM), logistic regression (LR) and/or random forest (RF) to classify the objects. Several researchers have proposed various object detection models, for example conjugate building feature detection using SIFT [42], human face detection and moving vehicle detection using Haar–Feature combined with AdaBoost [38,43,44], dangerous animal detection using local binary pattern adopting AdaBoost [45], HOG with SVM for human detection with high detection accuracy and long detection time [46–48] and a deformable parts model based on reducing the dimensions of HOG features with PCA [49]. This study adopts HOG features with SVM to build a two-stage model to detect rice seedlings.

The advantage of directly using images in object detection applications is that it allows CNNs to avoid manual feature extraction. In other words, CNNs are one of the most effective algorithms for object detection due to directly extracting features and detecting objects from images. In recent years, impressive improvements have been achieved in CNNs to address the problem of object detection by proposing many algorithms in which the network models are trained by combining local regional perception and feature extraction with a classification process. Tong et al. [35] and Zhang et al. [29] provided detailed reviews about the recent progress of CNN algorithms for objection detection. The development stages of CNN-based object detection models are shown in Figure 1 [31]. These object detection algorithms based on deep learning are divided into one-stage and two-stage detection algorithms. In one-stage algorithms, features for bounding box regression and class classification are directly extracted by the convolution operations on the output features of backbone networks. Object detection algorithms based on feature map convolution include YOLO [50], SSD [51], SqueezeDet [52], RetinaNet [53], CornerNet [54] and EfficientDet [55]. In two-stage algorithms, regional proposal modules are used to propose targeted object boundary boxes, and features are subsequently extracted from them to predict categories and masking objects. Object detection algorithms based on regional proposal, such as R-CNN [56], Fast R-CNN [57], Faster R-CNN [58], Mask R-CNN [59] and FPN [60], perform better and achieved high mean average precision (mAP). Among them, Mask-RCNN could be used to predict an exact mask within the bounding box of objects to detect single objects in images. This study adopts two state-of-the-art models, EfficientDet as a one-stage algorithm and Faster R-CNN as a two-stage algorithm, to detect rice seedlings in paddy fields due to the advantages of high efficiency, high localization and high precision for object detection.

This study overcomes a data scarcity problem with lightweight architecture and transfer learning in deep learning for precision agriculture. Besides, this study chooses EfficientDet and Faster R-CNN due to their architectures being capable of handling huge variations of feature scales for small-object detection. Overall, this study adopts one-stage and two-stage object detection architectures to develop tiny-object detection in UAV images to identify rice seedlings for precision agriculture, which has never been done for traditional rice cultivation, and the legacy HOG-SVM approach was applied for comparison. This study aims to achieve the following purposes:

- (1) adopting legacy, one-stage and two-stage machine learning algorithms to precisely detect small objects in UAV images,
- (2) rapidly establishing object detection models from prior knowledge as a transfer learning approach to overcome a data scarcity problem with lightweight architecture and verifying the applicability on the unseen data, and

- (3) evaluating performance and computation cost of rice seedling detection by three machine learning models to observe rice seedling growth.

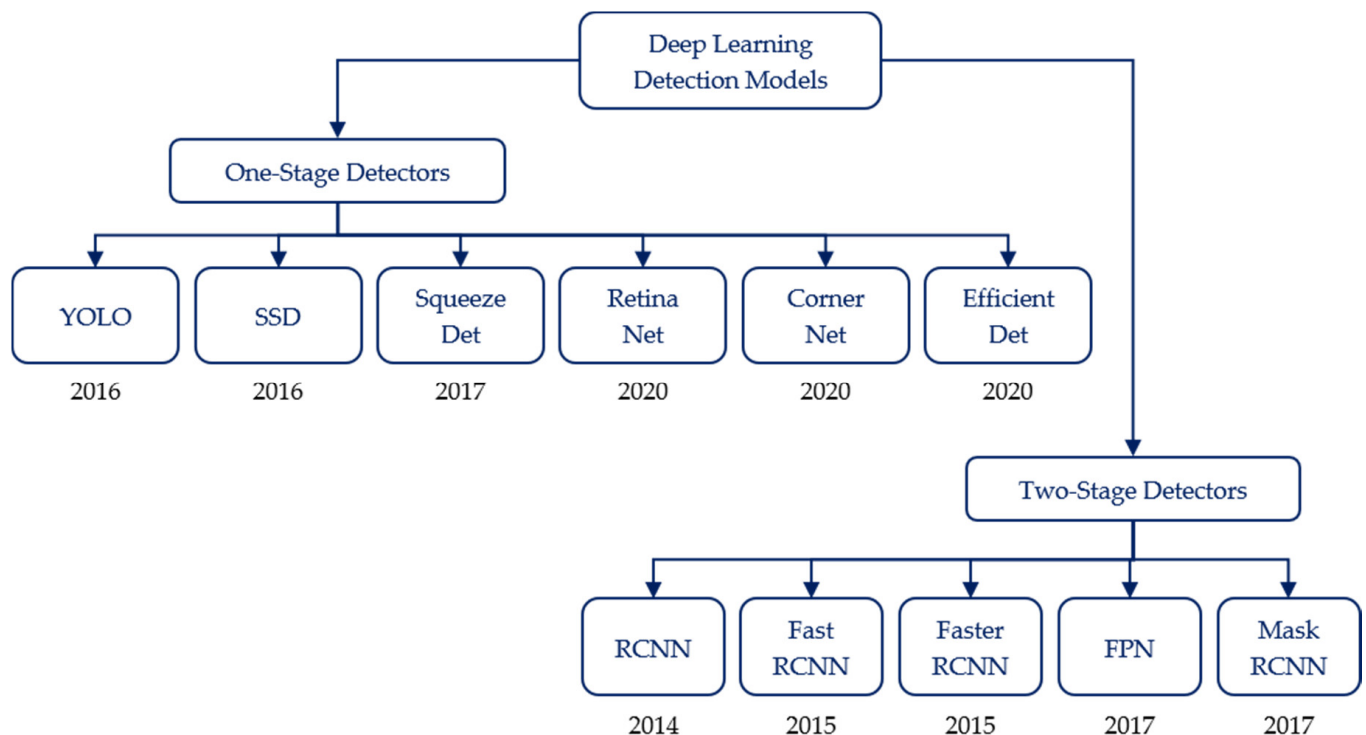


Figure 1. Development stages of deep learning object detection [31].

2. Materials and Methods

2.1. Data Introduction

UAVs can be used to help farmers broadly monitor rice growth in the early stage. The field images of rice seedlings were collected by UAVs equipped with cameras and downloaded from an open dataset [61]. The detailed information of camera, UAV and calibration settings to take images is given in Tables 1 and 2. The study area is located at the Taiwan Agriculture Research Institute, Wu-Feng District, Taichung City, Taiwan, where a long-term field investigation and observation, including UAV imaging and field survey, has been conducted for rice cultivation management experiments. Counting rice seedlings in paddies is one of the keys to calculate density and estimate grain yield. The study field is shown in Figure 2, in which the cyan bounding area represents the area for the deep learning training–testing dataset, the green and cyan bounding area represent the area for the HOG-SVM training–testing dataset, and the red bounding area represents the area for additional grass sub-images. To generate orthorectified mosaic images such as Figure 2, a commercial image-based-modeling software, Agisoft Metashape, is applied. The sample images of paddy fields are shown in Figure 3. Each of these images contains numerous tiny rice seedlings at low resolutions.

Table 1. UAV imaging sensor details [61].

Sensor Description	DJI Phantom 4 Pro
Resolution (H pixel \times V pixel)	5472 \times 3648
FOV (H° \times V°)	73.7° \times 53.1°
Focal Length (mm)	8.8
Sensor Size (H mm \times V mm)	13.2 \times 8.8
Pixel Size (μ m)	2.41 \times 2.41
Image Format	JPG
Dynamic Range	8 bit

Table 2. Details of flight mission [61].

Parameter	Setting			
Sensor	DJI Phantom 4 Pro			
Acquisition Date	7 August 2018	14 August 2018	12 August 2019	20 August 2019
Time	07:19–07:32	07:03–07:13	14:23–14:44	08:16–08:36
Weather	Mostly clear	Mostly cloudy	Mostly cloudy with occasional rain	Partly cloudy
Avg. Temperature (°C)	28.9	26.8	26.6	27.5
Avg. Press (hPa)	997.7	992.0	994.1	996.4
Flight Height (m)	21.4	20.8	18.6	19.1
Spatial Resolution (mm/pixel)	5.24	5.09	4.62	4.78
Forward Overlap (%)	80	80	85	85
Side Overlap (%)	75	75	80	80
Collected Images	349	299	615	596

Overview of study site – Reference System – TWD97 TM2 ZONE 121 (EPSG 3826)

Easting (Meter)

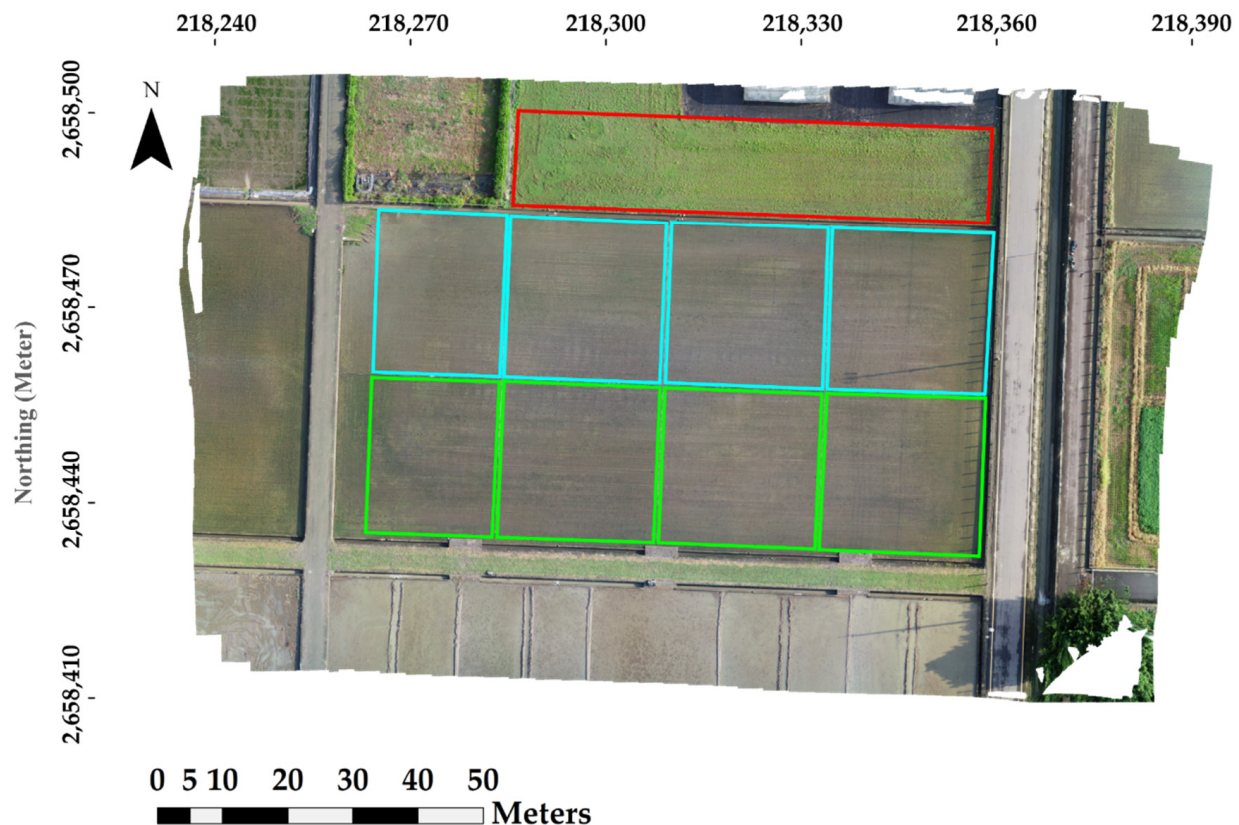


Figure 2. An overview of UAV image acquired on 7 August 2018. Reference system is TWD97 TM2 zone 121, EPSG: 3826.

The framework for processing the RGB images of rice fields using HOG-SVM and two deep learning models to detect and count rice seedlings is shown in Figure 4. The framework includes four phases: image pre-processing, sub-image generation, object detection with three approaches, and detection result and evaluation. The first phase is orthorectifying and mosaicking images captured by UAVs. The second phase is generating sub-images from orthorectified mosaic images due to the GPU's memory limitation. The third phase shows three object detection approaches: the legacy approach—HOG-SVM, two-stage object detection architecture—Faster-RCNN, and one-stage architecture—EfficientDet. All

three approaches generate detection results with classification and localization predictions that are evaluated with ground truth in the fourth phase. Each rice seedling in sub-images is manually annotated by agricultural experts. A training dataset is used to obtain the best model weights for rice seedling detection and counting.

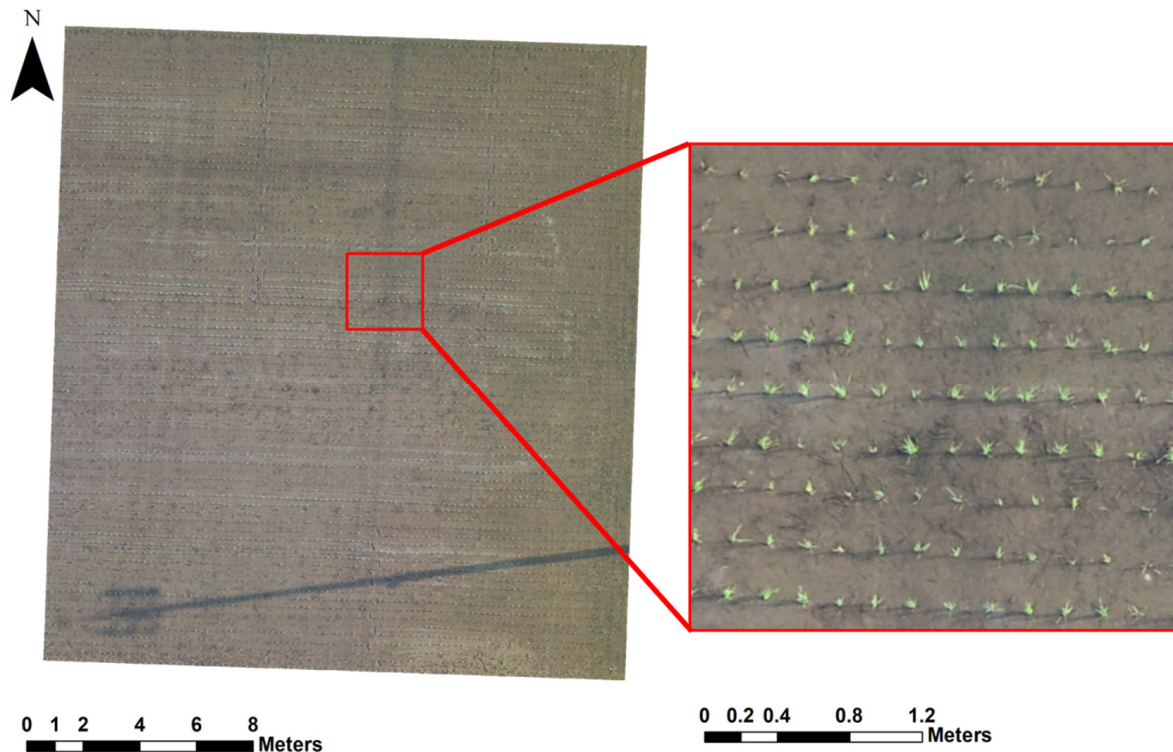


Figure 3. Sample images of rice seedlings in a paddy field image dataset [61].

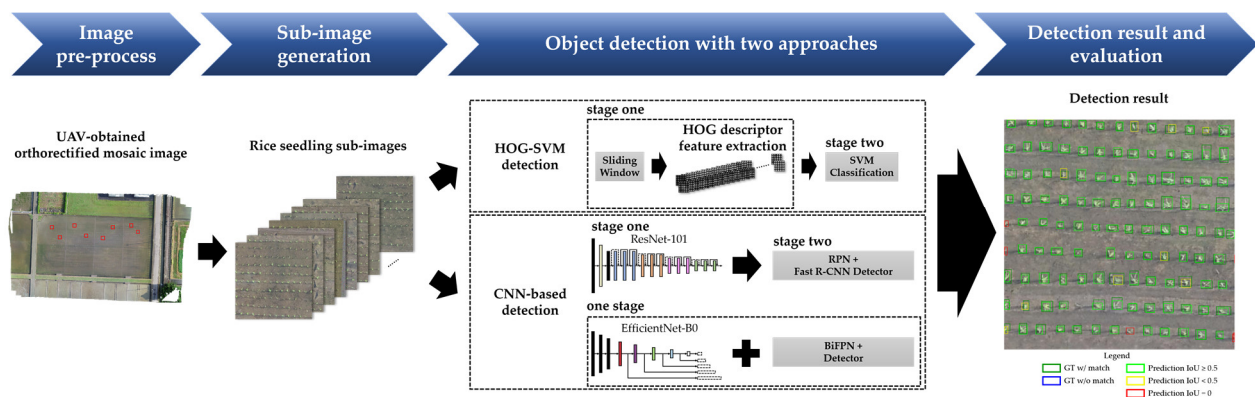


Figure 4. The framework of rice seedling detection.

2.2. Training and Testing Datasets

2.2.1. HOG-SVM Model

This study adopted a labeled dataset (RiceSeedlingClassification.tgz, accessed on 1 July 2021) from the open dataset [61] and cropped one of 8 full, large images into 54,628 sub-images with size of $48 \times 48 \times 3$ pixels and includes 26,581 bare land images and 28,047 rice seedling images, and externally added 18,757 grass sub-images from outside the paddy as shown in the red bounding region in Figure 2. From the 73,385 images, 80% were used to build the image classification method using feature extraction and classification methods, whereas the remaining 20% were used for testing. The example image of the three classes is shown in Figure 5.

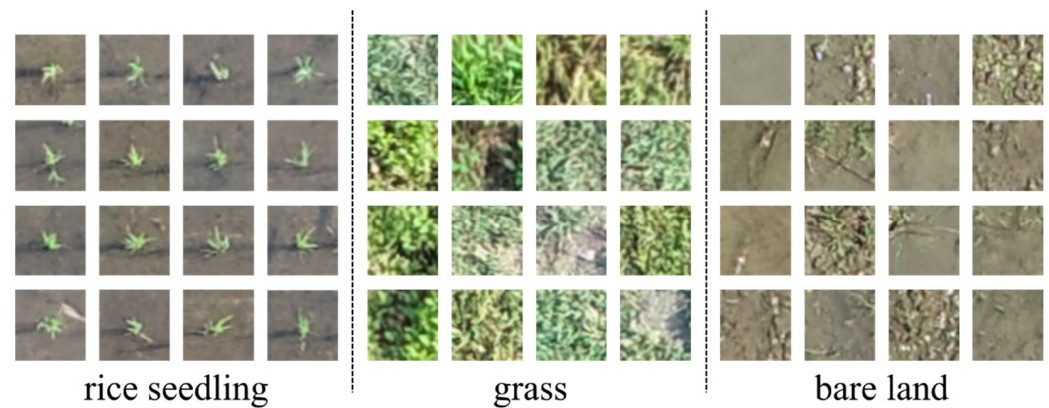


Figure 5. Examples of annotated images for model training.

2.2.2. CNN Models

Dataset collection is the essential part in object detection. This study used a total of four UAV images of rice paddy fields to train the model for CNN-based rice seedling detection. Each rice seedling in every image was manually labeled using *labellmg*, an open-source graphical image annotation tool in a pixel-basis with a single separate category (i.e., rice seedlings) from background. In order to have a sufficient dataset with a number of rice seedlings in each image, each image was split into several sub-images with each side 512 pixels. A total of 297 sub-images were generated from the four field images, and a training–test split ratio of 80–20 was applied to generate 273 sub-images for training and 60 sub-images for testing. In addition, three separate test datasets acquired on 14 August 2018, 12 August 2019 and 20 August 2019 with 72, 100 and 100 images, respectively, were also included to evaluate the model’s applicability to various imaging conditions. Annotating rice seedlings in every sub-image is time-consuming (needing a huge number of person-hours), thus a semi-auto preprocess was adopted for rapid annotation. Datasets can be used to determine the accuracy of rice seedlings detection and counting. The expected study results are looking for raw counts and also illustrating the spatial distribution of rice seedlings. Figure 6 shows annotated images of rice seedlings.

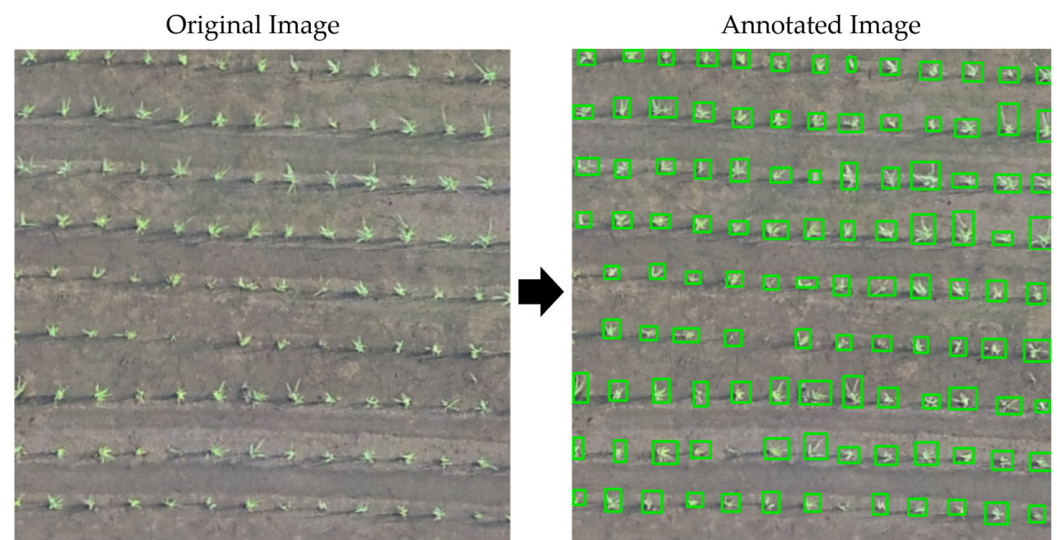


Figure 6. Rice seedlings annotated with bounding boxes in green on sub-images.

2.3. EfficientDet Model Training

EfficientDet is a highly scalable one-stage object detection architecture. The network inherits the prior-developed model EfficientNet [62] as the backbone for feature extraction. To comprehend the contextual features, a bi-directional feature pyramid network (BiFPN) is introduced as a feature network to aggregate multi-scale features. Compared to the prior proposed architectures, such as FPN, PANet and NAS-FPN, BiFPN performs with both better efficiency and higher accuracy. Moreover, EfficientDet also inherits the compound scaling method from EfficientNet, which allows deeper and wider network scaling without significantly changing the network architecture. In this study, the compound coefficient $\phi = 0$ (EfficientDet-D0) is chosen as the detection model, and the visualized architecture is shown in Figure 7.

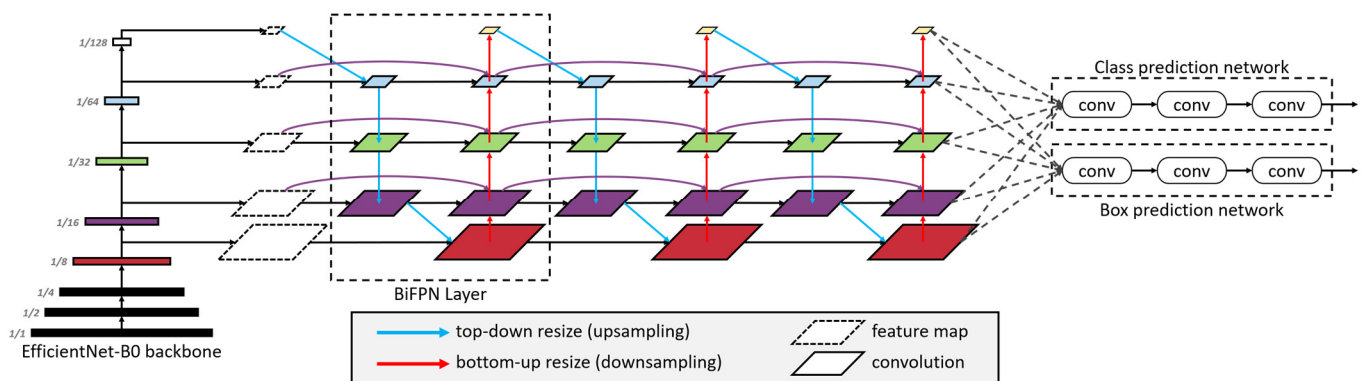


Figure 7. Architecture of EfficientDet-D0.

In order to rapidly establish a specific application, transfer learning is applied by adopting a pretrained weight based on prior knowledge for faster convergence. Thus, COCO 17 pretrained weight for EfficientDet-D0 is imported in the model training stage. Model training used the officially released scripts by TensorFlow on GitHub. The changes of parameters in the configuration are the number of classes, training step, batch size and the max number of detection boxes as 1, 30,000, 16 and 200, respectively.

2.4. Faster R-CNN Model Training

Faster R-CNN is a two-stage object detection network that inherits the robustness of the R-CNN family with a precise detecting capability. To speed up detection, a regional proposal network (RPN) is proposed to replace the selective search algorithm, which has poor GPU utilization. RPN is a fully convolutional network that generates both box-regression and box-classification features with a set of predefined anchors as the regression target. After the RPN generates proposals, both proposals from RPN and features from the backbone are passed into the Fast R-CNN detector. Because the design of the anchors already considers multi-scale and multi-ratio patterns, the architecture can feasibly be trained on single-scale images. The visualized architecture of Faster R-CNN ResNet-101 is shown in Figure 8.

A transfer learning strategy is also employed to rapidly establish the detection application. In this study, the ResNet-101 [63] backbone with input image size 640×640 pretrained weight is chosen for training the Faster R-CNN model. The changes of parameters in the configuration are the number of classes, training step, batch size and the max number of detection box, which are 1, 25,000, 8 and 200 respectively.

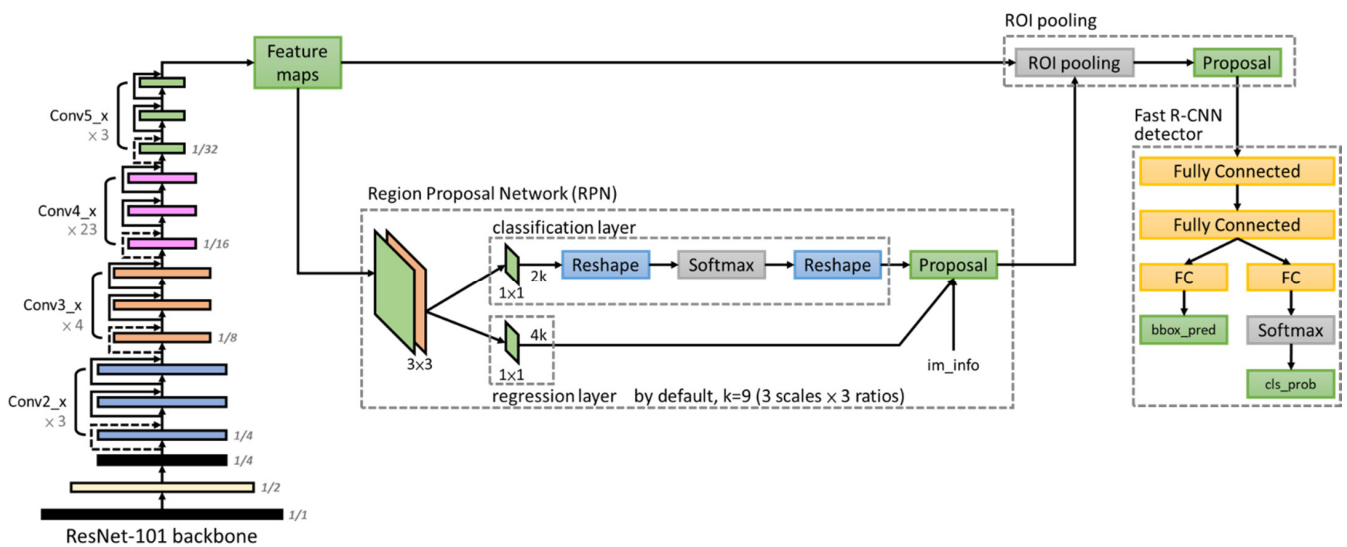


Figure 8. Architecture of Faster R-CNN ResNet-101.

2.5. HOG-SVM Model Training

SVMs are mainly used to separate data into two or more classes by hyperplane. The support vectors are located on the margin of the optimal hyperplane obtained with cost and kernel functions [64]. Instead of original image features, HOG features can be used in training models to achieve better performance with a large amount of UAV images. A HOG presents an object by estimating the magnitudes and gradient orientations from a specific set of pixel blocks. In this study, the HOG descriptor from OpenCV library is adopted, and the parameter settings are listed in Table 3. With an input image of 48×48 pixels, the HOG descriptor generates a vector comprised of 1296 elements. The SVM classifier is adopted from the NVIDIA GPU supported library ThunderSVM, which was claimed to be 10 to 100 times faster than the LibSVM library. To find an optimal hyperplane, a grid search with cross-validation approach is applied. The parameters of the grid search for SVM training are listed in Table 4. For SVM classification, the complexity can be expressed as Equation (1),

$$\text{Complexity} = n \cdot S \quad (1)$$

where n depicts the sample size and S depicts the number of support vectors. However, the computation cost of solving SVM problem has both a quadratic and cubic component; this leads the complexity to grow to at least n^2 to n^3 [65].

Table 3. Parameter settings of HOG descriptor.

Parameter	HOG Descriptor
orientation bins	9
cell size	8×8 pixel
block size	3×3 cell
block stride	8×8 pixel
window size	48×48 pixel

Table 4. Parameters of grid search for SVM training.

Parameter	SVM Classifier
kernel	linear
C	1, 2, 4, 8, 16, 32, 64, 128
decision_function_shape	ovr (One-vs-Rest)
max_iter	3000

Object features obtained from HOG are used in an SVM classifier to separate pixels into objects or background. Further, the trained SVM classifier can use sliding window to detect objects. The sliding window approach is an effective technique to localize objects with varying sizes in the image with time consumption. Figure 9 is a diagram of rice seedling detection based on HOG and SVM. SVM determines whether the detected objects belong to rice seedlings or the rest of the classes.

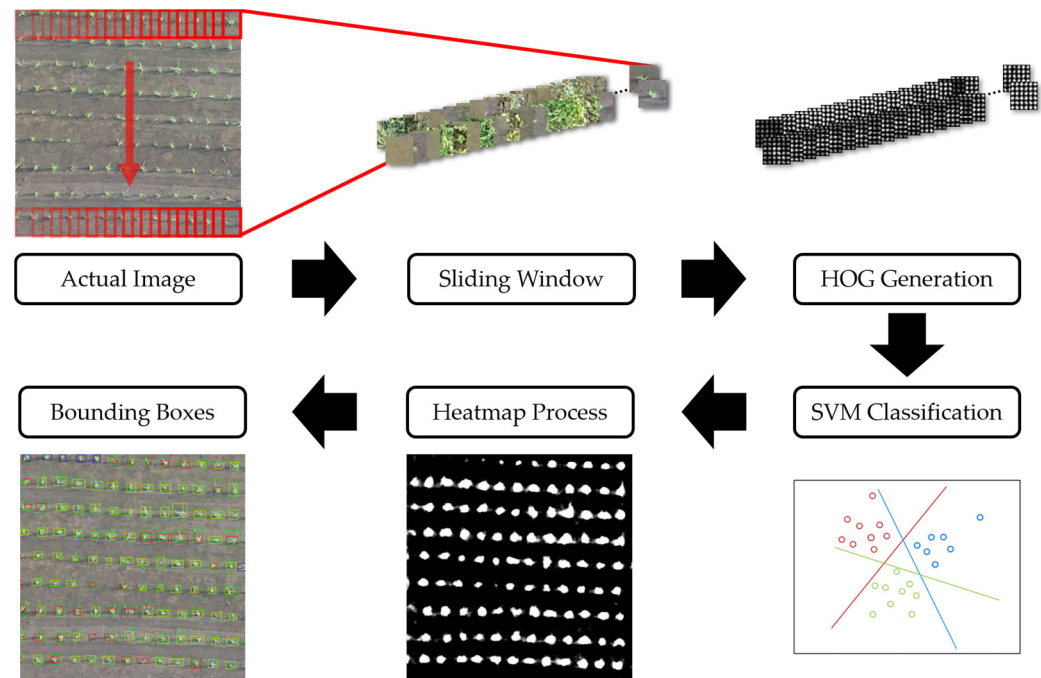


Figure 9. A diagram of rice seedling detection based on HOG and SVM.

2.6. Evaluation Metrics

To evaluate the localization of prediction boxes, intersection over union (IoU) is calculated. High IoU represents precisely predicted object location compared to the ground truth. In practice, average precision (AP) is commonly used and is defined as the average detection precision under different recalls in a category-specific manner. The mean average precision (mAP) is evaluated by the average AP score across classes and is commonly used to evaluate many object-detection datasets [35] and the performance of all models in this study. For object detection, AP is calculated equal or greater than a certain IoU threshold. From the definition of metrics evaluation of COCO datasets, three types of AP metrics related to IoU are proposed: AP, $AP^{IoU=0.50}$ and $AP^{IoU=0.75}$. The first AP is the mean of AP with IoU ranging from 0.5 to 0.95 in a 0.05 interval, which are 10 intervals total. $AP^{IoU=0.50}$ and $AP^{IoU=0.75}$ are the APs at an IoU threshold equal or greater than 0.5 and 0.75, respectively. The larger the IoU threshold, the lower the AP. $AP^{IoU=0.50}$ is calculated in this study.

Further, the performance of the HOG-SVM classification model is evaluated based on precision, recall, F1-score and overall accuracy (OA), representing the precision of the prediction, the accuracy of the prediction to the real data, the robustness of the prediction and the accuracy across all categories, respectively.

3. Results and Discussion

3.1. HOG-SVM

To rapidly establish a GPU-capable computing environment, Taiwan Computing Cloud (TWCC) provided by the National Center for High-Performance Computing (NCHC, Taiwan) is used. The service provides a variety of containerized computing environments to run experiments using powerful computing resources. In this study, the containerized

image tensorflow-19.08 is used to satisfy the requirement for running ThunderSVM in CUDA version 10.1. The container runs on the hardware specification of one NVIDIA Tesla V100 GPU, four cores of Intel Xeon Gold 6154 CPU, and 90 GB of host memory.

In the experiments, features are extracted by HOG descriptor and then are used in SVM for image classification. As mentioned in Section 2.5, the grid search approach is applied to find the optimal parameters of the SVM classifier. After a search sequence, the parameter with $C = 4$ is chosen for the best classification capability. The evaluation metrics are given in Table 5, and the confusion matrix is given in Figure 10. The SVM based on HOG features has an overall accuracy of 93.9% and 93.1% for training and test data, respectively. In the metrics of separate classes, all reach above 85%, especially the rice seedling class, reaching above 99%. This shows that the model is capable of distinguishing rice seedlings from images robustly.

Table 5. HOG-SVM training and test evaluation metrics.

	Training				Test			
	Precision	Recall	F1-Score	OA	Precision	Recall	F1-Score	OA
bare land	91.7	91.3	91.5		90.3	90.7	90.5	
grass	87.8	88.4	88.1	93.9	87.2	86.7	86.9	93.1
rice seedling	99.9	99.9	99.9		99.7	99.6	99.6	

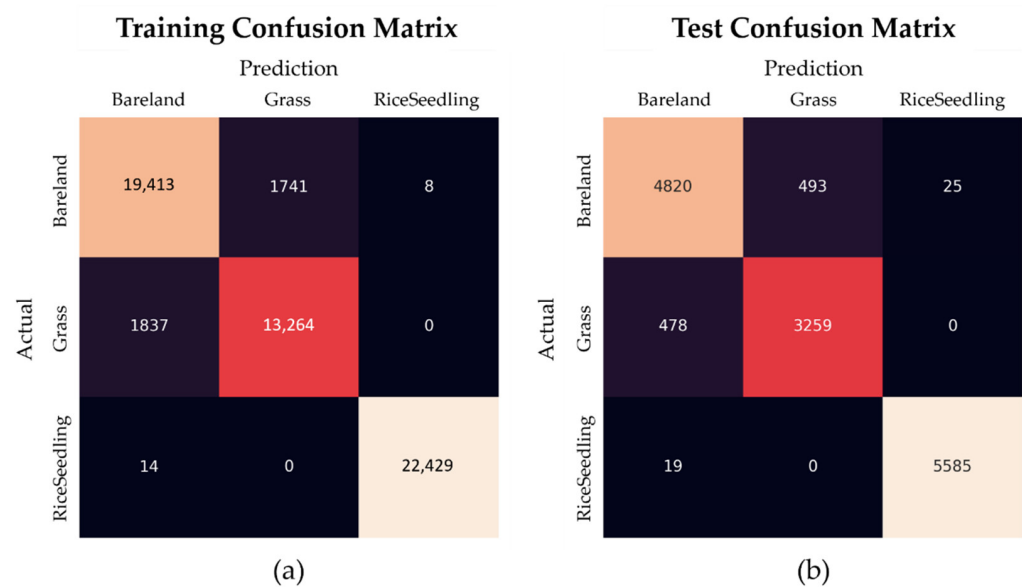


Figure 10. HOG-SVM confusion matrix of (a) training dataset and (b) test dataset.

To detect rice seedlings in the paddy field images, a sliding window approach is implemented, where the highly overlapping tiles are firstly computed as HOG vectors and classified as probabilities in three classes. Then, the output probabilities are gathered and ordered as the shape of the input image to form the probability map. The probability map is passed to a gaussian filter for smoothing, and then a threshold of confidence is applied to keep the positive pixels. Finally, the process of finding contours is applied to identify every closed object and to generate the bounding boxes.

To evaluate the performance of the sliding window approach of detection, two common metrics are calculated, AP and IoU. Prior to calculating AP, the IoU of every object is calculated to proceed the matching. If the predicting box overlaps with ground truth boxes and the IoU is above the threshold, the object is counted as a correct prediction. If the predicting box does not overlap with ground-truth boxes, or the overlapping IoU is below the threshold, the object is counted as misprediction. Then, AP is calculated in a descending power manner from the most confident box. An example of visualized detection results is shown in Figure 11. The dark green boxes present the ground truth boxes matching with

prediction boxes, and the blue boxes present the ground truth boxes matching with no prediction box. The light green boxes present the prediction boxes matching with ground truth boxes with the IoU equal or over 0.5, the yellow boxes present the prediction boxes matching with ground truth boxes with the IoU under 0.5, and the red boxes present the prediction boxes matching with no ground truth boxes. An example of a visualized recall–precision curve (AP curve) is shown in Figure 12. By evaluating all training and test data, HOG-SVM performs 0.700 mAP and 0.465 mIoU on training data and 0.702 mAP and 0.466 mIoU on test data.

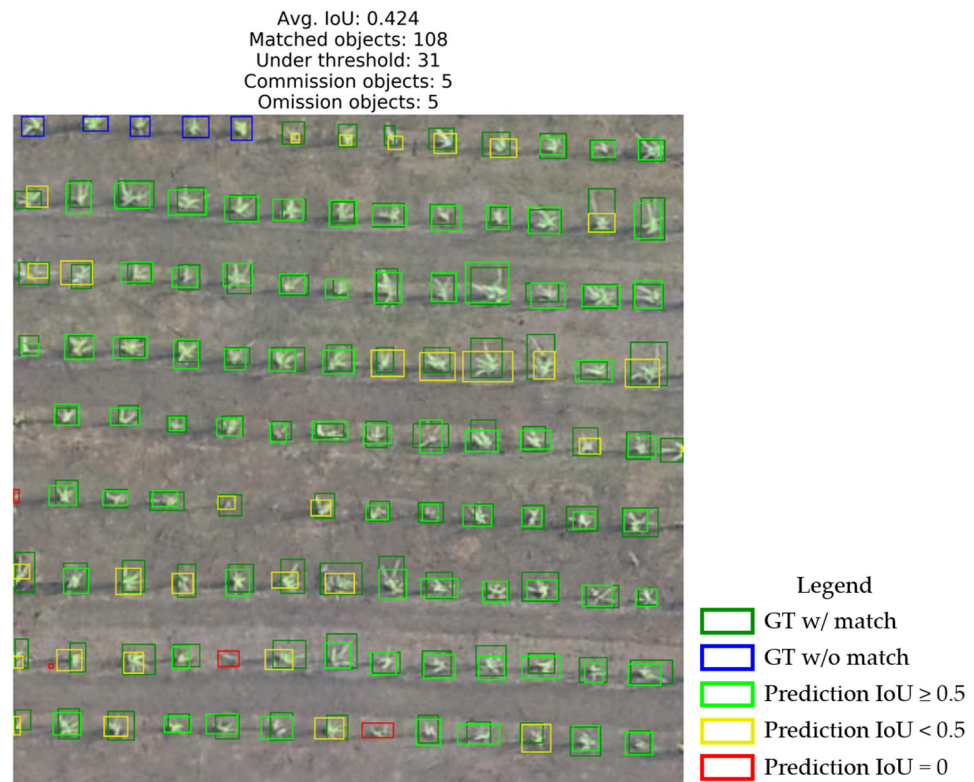


Figure 11. An example of HOG-SVM detection result on test data.

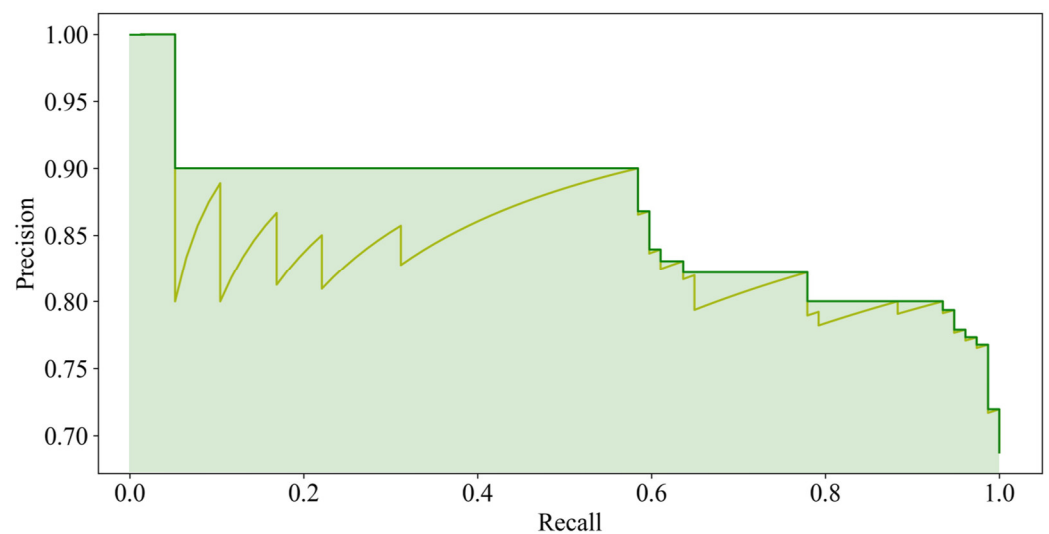


Figure 12. Visualization of recall–precision curve of the test image for AP calculation. The yellow curve depicts the precision metric in a descending power manner from the most-confident box. The area under the green curve represents AP (0.842).

3.2. CNN Models

The experiments of the two CNN-based detection models are also implemented on TWCC. The containerized image for the experiments is tensorflow-21.06-tf2 with TensorFlow version 2.5 to have the latest function support. The hardware specifications are the same as for the HOG-SVM experiment.

Starting from the model building, both experiments apply the same scripts from the officially released object detection example of TensorFlow. The only two changes are the configuration files and the pretrained weights. Detailed documentation of usage can be accessed from TensorFlow on GitHub. Both detection models are initialized with the COCO 2017 pretrained weights, which provides prior knowledge of object features to give a better and faster model convergence. Evaluation of the two detection models is the same as for HOG-SVM and is skipped herein. Figure 13 shows the visualized detection results, and Figure 14 shows the visualized AP curve. For all the training and test data, EfficientDet-D0 performs 0.955 mAP and 0.676 mIoU on the training data and 0.837 mAP and 0.575 mIoU on the test data. Faster R-CNN performs 1.000 mAP and 0.996 mIoU on the training data and 0.888 mAP and 0.637 mIoU on the test data. The metrics between training and testing show a big gap that could be caused by overfitting during training. The comparison of all the models will be mentioned in Section 3.3.

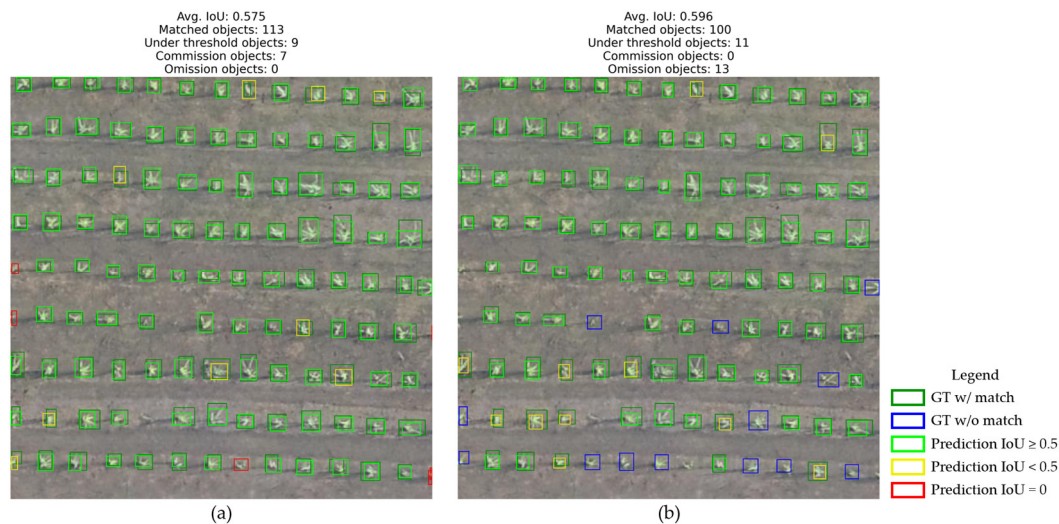


Figure 13. Example of visualized detection results of (a) EfficientDet-D0 and (b) Faster R-CNN.

3.3. Comparison of Model Performance

Three detection models are compared by evaluation metrics and computation costs (i.e., execution time) to illustrate the tradeoffs between the models from different viewpoints of applications. The workflows of detection between the HOG-SVM model and CNN-based models are different, so the comparison of computational costs is separated into three segments: preprocess, inference and visualization. The comparison is listed in Table 6.

Table 6. Comparison of model performance and computational cost (Best performance marked in bold).

Model	Training		Test		Computational Cost				
	mAP	mIoU	mAP	mIoU	Preprocess	Inference	Visualization	Total	Total
HOG-SVM	0.700	0.465	0.702	0.466	28.659 s	46.644 s	0.020 s	75.323 s	0.013 fps
EfficientDet	0.955	0.676	0.837	0.575	0.005 s	0.026 s	0.003 s	0.034 s	29.412 fps
Faster R-CNN	1.000	0.996	0.888	0.637	0.005 s	0.042 s	0.003 s	0.050 s	20.000 fps

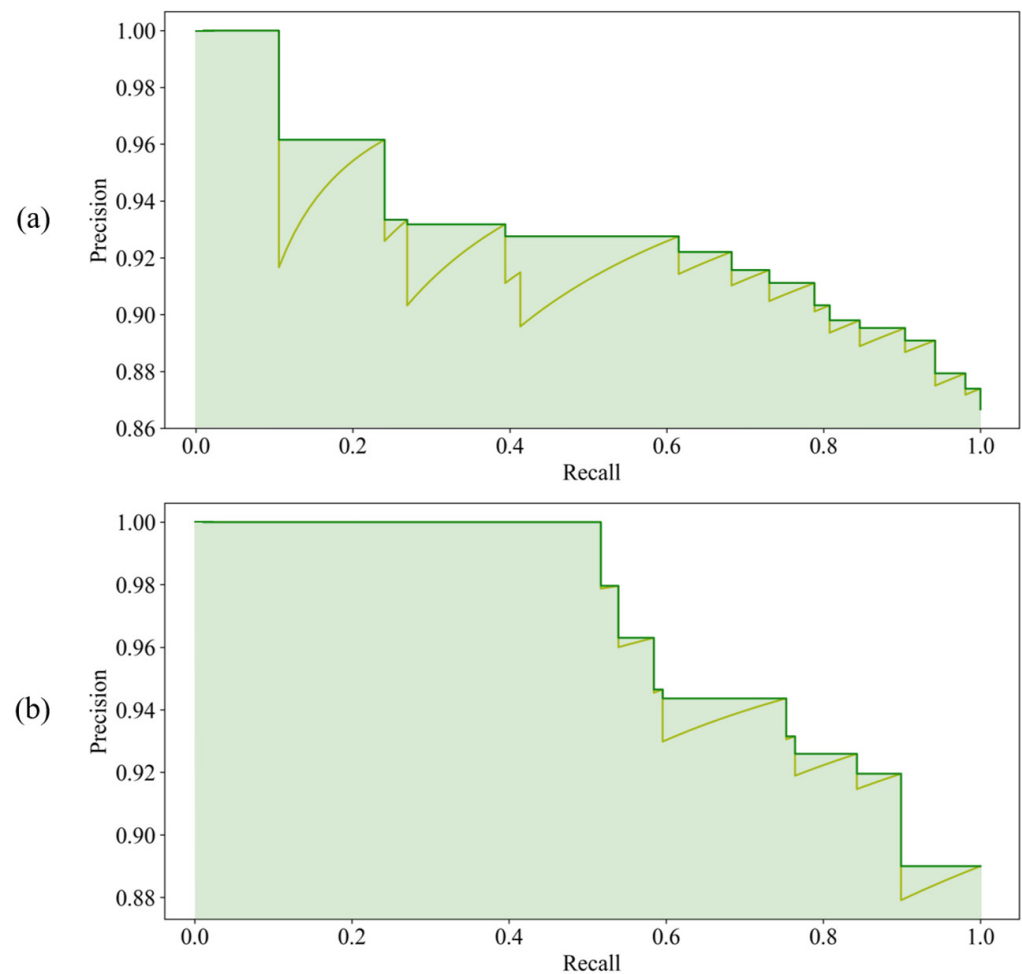


Figure 14. AP curves of the detection results of (a) EfficientDet-D0 with AP 0.924 and (b) Faster R-CNN with AP 0.964.

To simulate the inference for real-time scenarios, images are loaded one by one from the disk for all three models. In HOG-SVM, images are processed through the sliding window approach and HOG feature computation to generate the input vectors for SVM classification. This is a tedious process that does not take advantage of the GPU to perform parallel processing. Different to HOG-SVM, the CNN-based detection models highly utilize the GPU to extract features and detect objects parallelly. The comparison shows an obvious gap of computation time between HOG-SVM and CNN-based models, especially in the preprocessing and inference segments. The huge gap in computational time is due to the adoption of the sliding window approach that processes data in an exponential explosion time if parallel computing is not applied. Our previous study (Yang et al., 2021) [61] used a simple CNN classification model with a sliding window approach to detect rice seedlings on the open dataset. The results showed the classification model performed well, with an F1-score of 0.99, but showed less localization accuracy, similar to the results of the HOG-SVM approach in this study. In this study, deep learning methods with one stage and two stages were employed to enhance localization accuracy. Wu et al., 2019 [20], used fully convolutional architecture to count rice seedlings in 40 UAV images, resulting in high correlation to the ground truth count ($R^2 = 0.94$), but false positive counting was not considered and adjusted for. Moreover, the size of detected objects was not detected, so it was unable to estimate localization accuracy. In addition to rice seedling counting, the proposed methods in this study can detect the position and size of the rice seedlings. Etienne et al., 2021 [34], aimed to detect monocot and dicot weeds in corn and soybean fields using YOLOv3 with different sets of images captured in two spatial resolutions (1.5 cm

and 0.5 cm). The results showed that both the AP at IoU 0.5 threshold reached the highest number, 65.37% and 45.13%, for monocot and dicot weeds in the dataset with the finest spatial resolution (0.5 cm). In general, YOLOv3 is a popular one-stage detection architecture, whereas EfficientDet adopted in this study is a new one-stage detection architecture with less computation and better accuracy (Tan et al., 2020) [55]. Moreover, this study applied transfer learning to reduce the need for a great amount training data and training time (less than 1 h for 500 epochs). Overall, the EfficientDet and Faster R-CNN models show the capability of real-time inference, as they performed around 30 fps and 20 fps detection, respectively. However, the computational cost of the CNN-based model is overstated, because normally the images are read from the camera cache through the bus instead of the disk. Further, the CNN models will be optimized to a faster and lighter runtime package to satisfy various deployment environments.

3.4. Model Evaluation on Different Datasets

Three detection models were all evaluated with four test datasets, which were different in the planting year, growth day, location and environmental conditions. The AP and IoU metrics of three models are listed in Table 7 and visualized in Figure 15. To evaluate the robustness of the models, metrics of precision, recall and F1-score were also evaluated, the results are listed in Table 8 and visualized in Figure 16.

Table 7. Evaluation of AP and IoU on four datasets.

Model Date	EfficientDet		Faster R-CNN		HOG-SVM	
	AP	IoU	AP	IoU	AP	IoU
7 August 2018	0.837	0.575	0.888	0.637	0.702	0.466
14 August 2018	0.965	0.631	0.981	0.686	0.732	0.315
12 August 2019	0.903	0.537	0.986	0.871	0.476	0.156
20 August 2019	0.744	0.357	0.739	0.382	0.335	0.092

Comparisons of AP and IoU metrics of test results

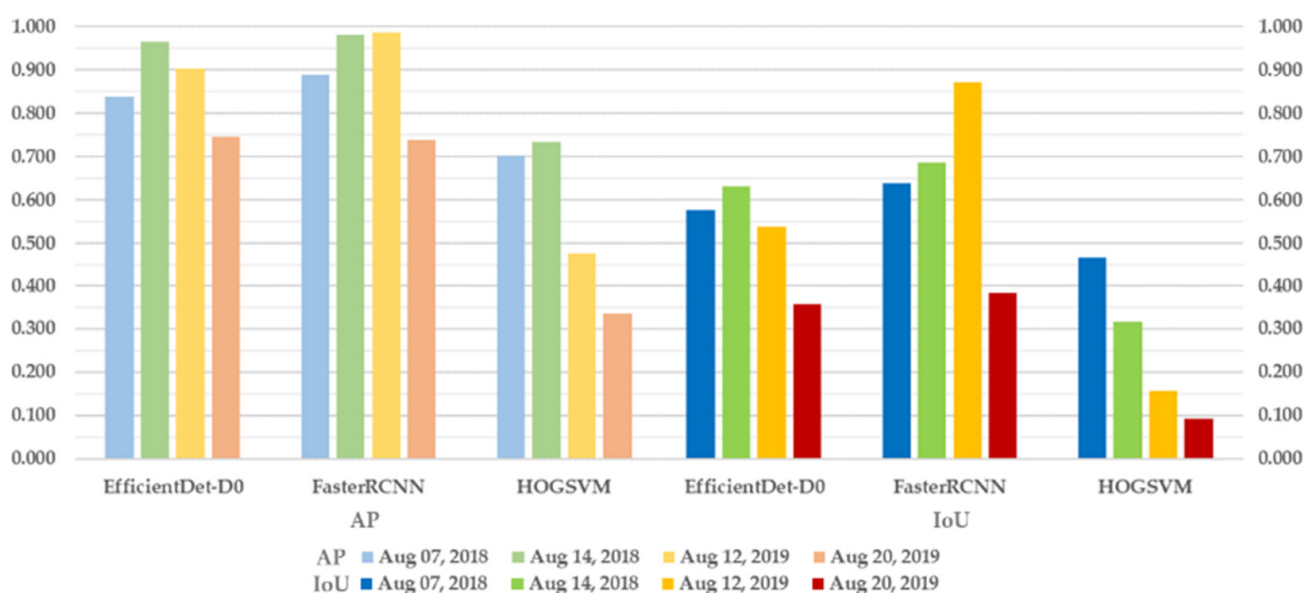
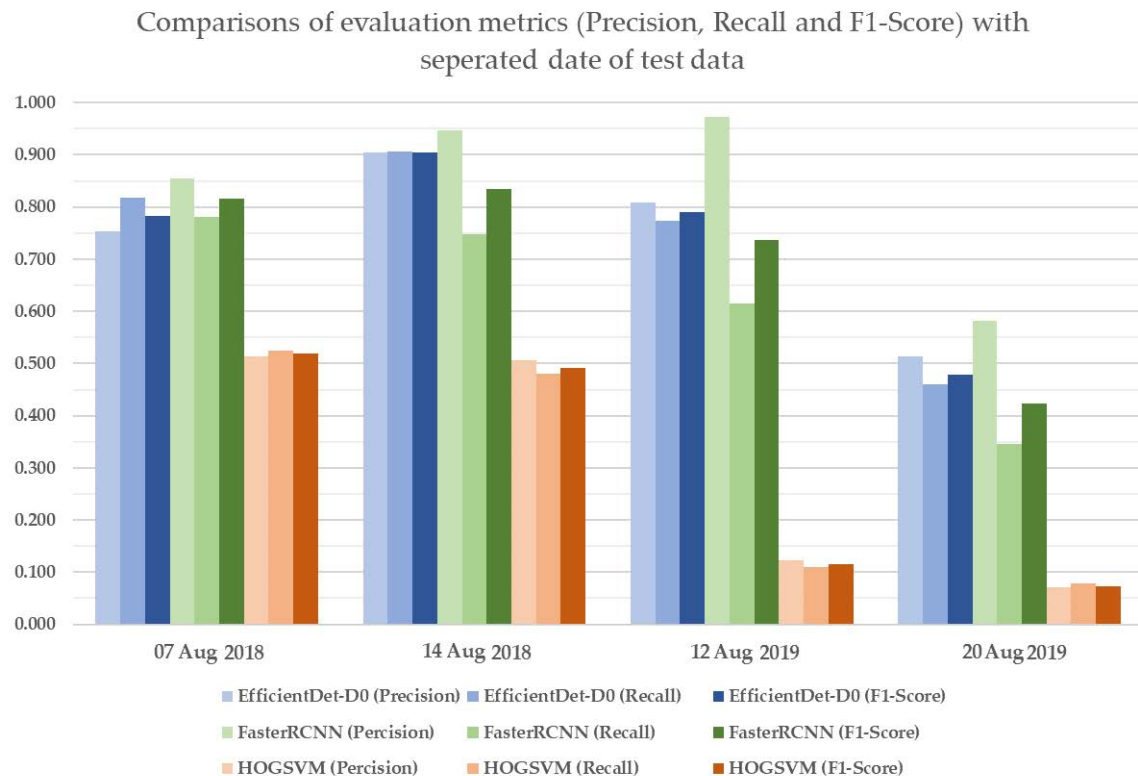


Figure 15. A vertical chart comparing evaluation results with AP and IoU metrics.

Table 8. Evaluation of model performance on four datasets.

Model Date	EfficientDet			Faster R-CNN			HOG-SVM		
	Precision	Recall	F1-Score	Precision	Recall	F1-Score	Precision	Recall	F1-Score
7 August 2018	0.753	0.817	0.783	0.855	0.780	0.815	0.514	0.524	0.518
14 August 2018	0.904	0.905	0.904	0.948	0.747	0.834	0.506	0.480	0.491
12 August 2019	0.809	0.774	0.790	0.972	0.615	0.737	0.123	0.110	0.115
20 August 2019	0.515	0.460	0.480	0.583	0.345	0.422	0.071	0.079	0.073

**Figure 16.** Comparison of evaluation metrics (precision, recall and F1-score) on four datasets.

According to the test results, the performance of the CNN-based detection models surpassed the HOSVM model with at least 10% higher performance in all metrics. The AP and IoU metrics of Faster RCNN were higher than EfficientDet-D0. Oppositely, the recall of Faster RCNN was lower than EfficientDet-D0 on all the test sets. To discuss this issue, an example of the test images on four different datasets was selected and visualized (Figure 17), and comparison of the test results is visualized as Figure 18. Figure 17 shows the variances of paddy environment, seedling size and illumination. The variance of seedling sizes is due to the different image acquisition date, which can be categorized into three sizes of side length 20, 25 and 30 pixels. The paddy environment can be categorized into four situations with combinations of ponding management and growth of algae. The illumination conditions can be categorized into object with shadow and object without shadow.

An example of the detection results with comparisons between the three models is shown in Figure 18. Precision was calculated by the number of predictions for which the IoU is equal or greater than 0.5 (light green boxes) divided by the total number of predictions. Recall was calculated by the number of light green boxes divided by the total number of ground truth boxes.

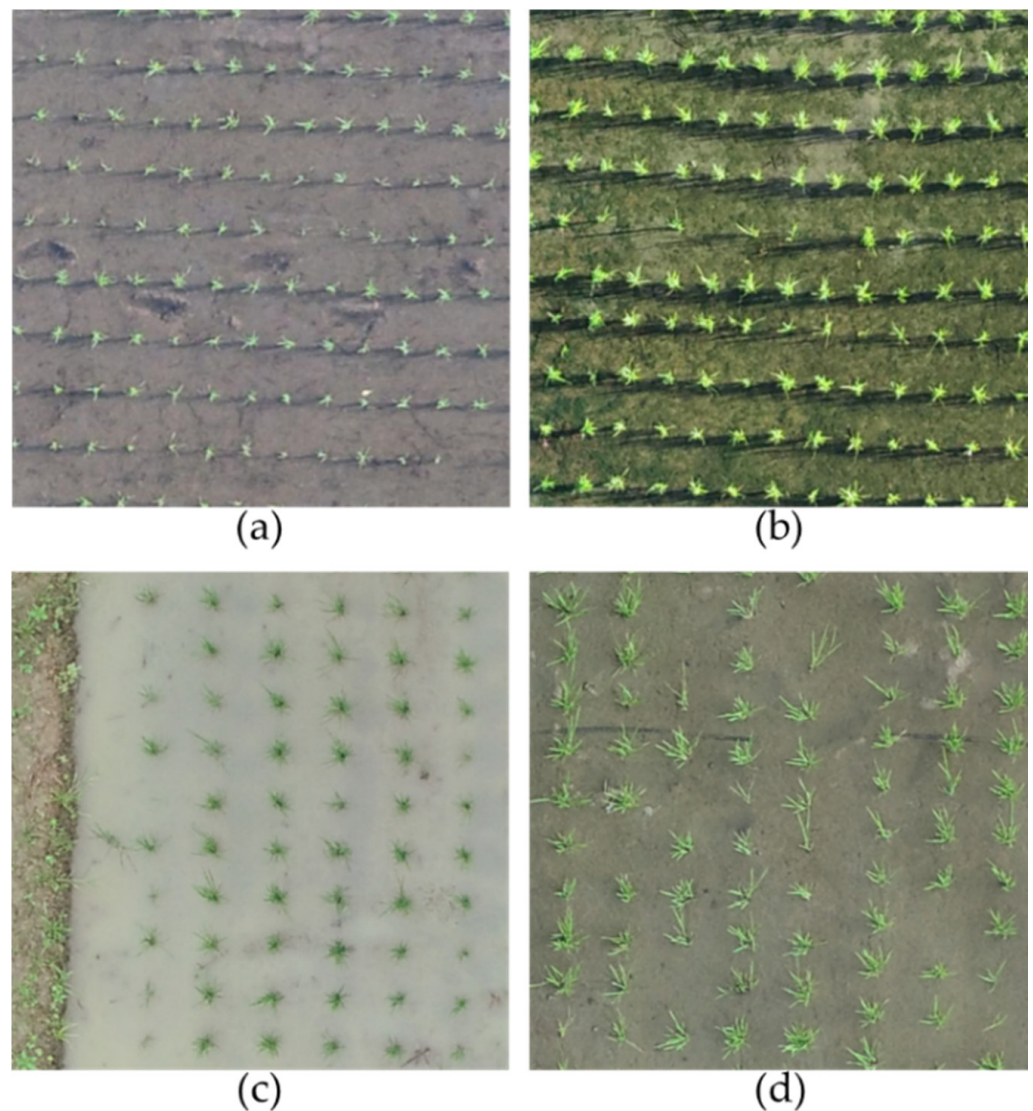


Figure 17. An example of the test images on four datasets on (a) 7 August 2018, (b) 14 August 2018, (c) 12 August 2019 and (d) 20 August 2019.

The detection results of Faster RCNN on the dataset acquired on 12 August 2019 (center column of Figure 18c) shows a large percentage of omission objects, which causes a lower recall rate. This issue could be relatively low light intensity on rice seedlings contrasting with the high reflectance of turbid water. All three models yield a lower precision and recall on the dataset acquired in 20 August 2019 (Figure 18d), which was due to the significant difference in seedling size between the training samples (Figure 6) and test samples. The data of 12 August and 20 August 2019 were acquired on 17 and 25 days after seedling transplantation, respectively. Further, the field was fertilized 19 days after seedling transplantation. According to the rice growth calendar [66], the rice seedlings on 20 August 2019 were in the middle of the active-tillering stage, during which the rice seedlings were growing rapidly with more canopy cover. Therefore, the size of rice seedlings from the nadir perspective is obviously different from the training data, reducing the precision, recall and f1 score.

According to the evaluation in Table 8 and Figure 16, EfficientDet-D0 shows similar prediction results on the first three datasets. Figure 18 also shows a stable prediction capability of EfficientDet, although differences in image tone and contrast, object size and surrounding reflectance appear between the test datasets and the training dataset. Thus, EfficientDet has the highest robustness to minimize the impact of image variance.

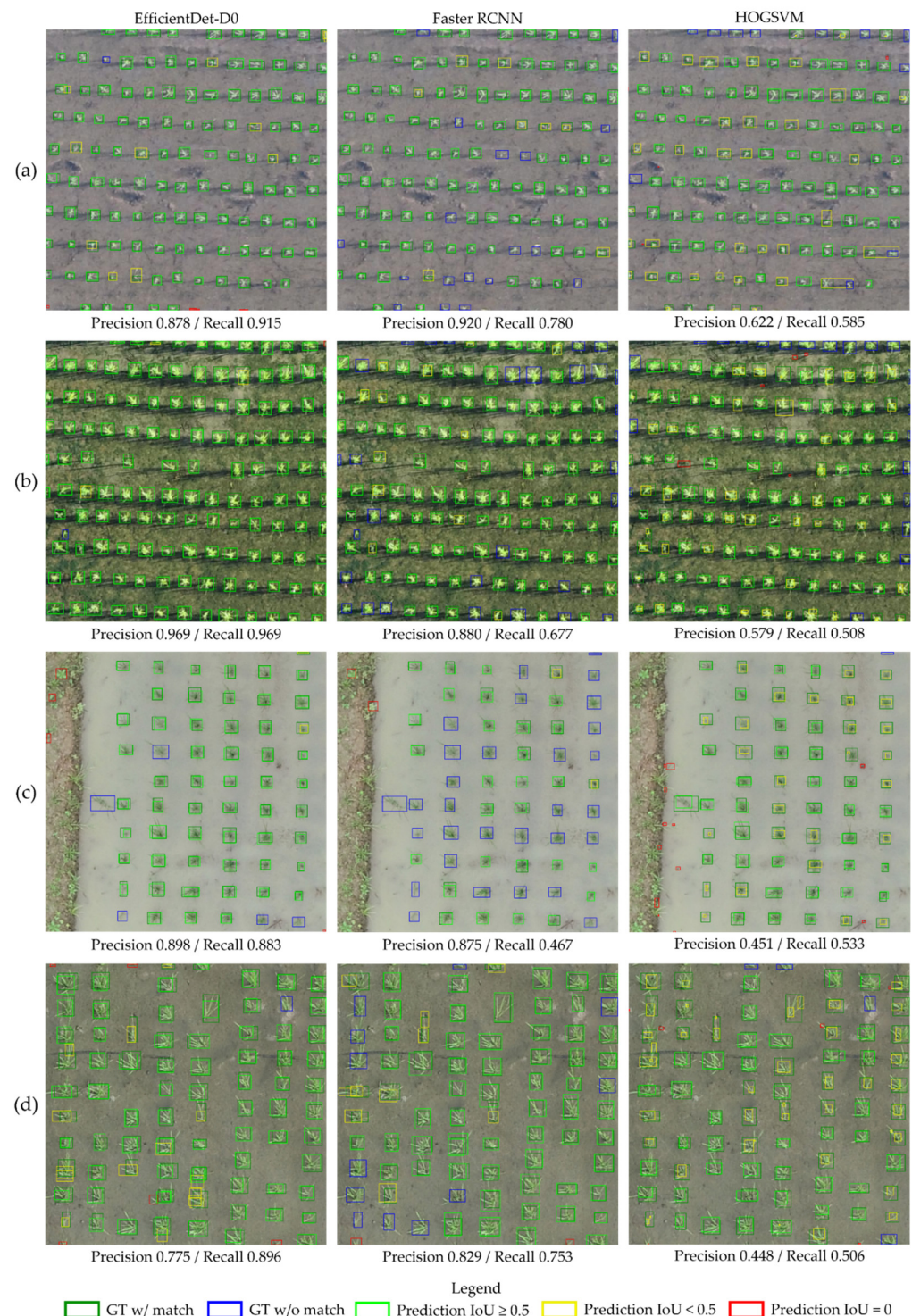


Figure 18. An example of detection results with precision and recall metrics on four datasets: (a) 7 August 2018, (b) 14 August 2018, (c) 12 August 2019 and (d) 20 August 2019.

4. Conclusions

Tiny-object detection in UAV images is a challenging task in practical applications. Long computation time and slow speed due to memory consumption are the first-priority causes. Complex background and scenes, high density areas and random textures of fields can also decrease the performance of small-object detection. In this study, small rice seedlings are presented in highly noisy environments that influence the detection using deep learning on UAV imagery.

This study presents three machine learning models, HOG-SVM, EfficientDet and Faster R-CNN on UAV images to detect tiny rice seedlings. The datasets are semi-annotated with preprocessing of image processing and manual verification to reduce the cost of labor. This approach is sure to generate usable datasets rapidly. The combination of HOG descriptor and SVM classifier gives a robust result for rice seedling classification, which achieves 99.9% F1-score in training and 99.6% F1-score in testing. The rest of the two classes also achieve above 85% F1-score in both training and testing. However, explosive data growth is one of problems that needs to be solved in practical applications. For SVM classification, computational complexity grows exponentially with sample size. This is a significant drawback for such an algorithm that cannot efficiently process such big data for practical applications. In this study, two CNN models were transferred from pretrained models to develop a well-generalized model with high detection accuracy and rapidity. The pretrained models were well-trained by splitting four paddy images into 297 sub-images (each image sized $512 \times 512 \times 3$) by annotating each rice seedling in every sub-image.

To verify model applicability with various imaging conditions, HOG-SVM, EfficientDet and Faster R-CNN were applied to the rest of the images and three additional datasets acquired on different dates for model testing. The test results of HOG-SVM, EfficientDet and Faster R-CNN to detect rice seedlings showed that Faster R-CNN has the best detection performance, with mAP of 0.888, 0.981 and 0.986 and mIoU of 0.637, 0.686 and 0.871 on the first three test datasets. Further, EfficientDet had promising results, with mAP of 0.837, 0.965 and 0.903 and mIoU of 0.575, 0.631 and 0.537 on the first three test datasets, and it had the fastest computation speed at nearly 30 fps. Moreover, the two CNN-based models had acceptable detection results, with 0.744 mAP and 0.357 mIoU (EfficientDet) and 0.739 mAP and 0.382 mIoU (Faster RCNN) on the fourth dataset, even though huge variances exist between test datasets and training datasets. EfficientDet especially showed the highest robustness to minimize the impact of image variance. Overall, rice seedlings can be well-detected using both CNN-based models with real-time computation performance. In contrast, HOG-SVM gave a merely adequate result with long computation time.

Further study will focus on detecting rice seedlings with more variety in imaging conditions, such as illumination, tone, color temperature, blur and noise. The models can be retrained using these additional images to adapt to more image changes. Further, optimizing model parameters to reduce computational time and increase prediction accuracy is needed to enable models to be deployed in environments with tight resources.

Author Contributions: Conceptualization, M.-D.Y., H.-H.T., Y.-C.H., C.-Y.Y. and D.-H.W.; methodology, M.-D.Y., H.-H.T., R.S., Y.-C.H., C.-Y.Y. and D.-H.W.; software, H.-H.T. and R.S.; formal analysis, M.-D.Y., R.S. and Y.-C.H.; writing—original draft preparation, H.-H.T. and R.S.; writing—review and editing, M.-D.Y.; visualization, H.-H.T.; supervision, M.-D.Y.; funding acquisition, M.-D.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This work was funded in part by the Ministry of Science and Technology, Taiwan, under grant numbers MOST 110-2634-F-005-005 and MOST 110-2634-F-005-006.

Data Availability Statement: In this study, the training data are openly available in “Rice Seedling Datasets” at doi:10.3390/rs13071358 [61].

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Lencucha, R.; Pal, N.E.; Appau, A.; Thow, A.-M.; Drope, J. Government policy and agricultural production: A scoping review to inform research and policy on healthy agricultural commodities. *Glob. Health* **2020**, *16*, 11. [[CrossRef](#)] [[PubMed](#)]
2. Bochtis, D.D.; Sørensen, C.G.C.; Busato, P. Advances in agricultural machinery management: A review. *Biosyst. Eng.* **2014**, *126*, 69–81. [[CrossRef](#)]
3. Josephson, A.L.; Ricker-Gilbert, J.; Florax, R.J.G.M. How does population density influence agricultural intensification and productivity? Evidence from Ethiopia. *Food Policy* **2014**, *48*, 142–152. [[CrossRef](#)]
4. Ricker-Gilbert, J.; Jumbe, C.; Chamberlin, J. How does population density influence agricultural intensification and productivity? Evidence from Malawi. *Food Policy* **2014**, *48*, 114–128. [[CrossRef](#)]

5. Fróna, D.; Szenderák, J.; Harangi-Rákos, M. The challenge of feeding the world. *Sustainability* **2019**, *11*, 5816. [\[CrossRef\]](#)
6. Le Mouél, C.; Lattre-Gasquet, D.; Mora, O. *Land Use and Food Security in 2050: A Narrow Road*; Éditions Quae: Versailles, France, 2018.
7. Zhang, Y. The Role of Precision Agriculture. *Resource* **2019**, *19*, 9.
8. Singh, P.; Pandey, P.C.; Petropoulos, G.P.; Pavlides, A.; Srivastava, P.K.; Koutsias, N.; Deng, K.A.K.; Bao, Y. 8-Hyperspectral remote sensing in precision agriculture: Present status, challenges, and future trends. In *Hyperspectral Remote Sensing*; Pandey, P.C., Srivastava, P.K., Balzter, H., Bhattacharya, B., Petropoulos, G.P., Eds.; Elsevier: Amsterdam, The Netherlands, 2020; pp. 121–146.
9. Yang, C.Y.; Yang, M.D.; Tseng, W.C.; Hsu, Y.C.; Li, G.S.; Lai, M.H.; Wu, D.H.; Lu, H.Y. Assessment of Rice Developmental Stage Using Time Series UAV Imagery for Variable Irrigation Management. *Sensors* **2020**, *20*, 5354. [\[CrossRef\]](#)
10. Sarvia, F.; DePetris, S.; Orusa, T.; Borgogno-Mondino, E. *MAIA S2 Versus Sentinel 2: Spectral Issues and Their Effects in the Precision Farming Context*; Springer International Publishing: Berlin/Heidelberg, Germany, 2021. [\[CrossRef\]](#)
11. Tian, H.; Wang, T.; Liu, Y.; Qiao, X.; Li, Y. Computer vision technology in agricultural automation—A review. *Inf. Process. Agric.* **2020**, *7*, 1–19. [\[CrossRef\]](#)
12. Saiz-Rubio, V.; Rovira-Más, F. From smart farming towards agriculture 5.0: A review on crop data management. *Agronomy* **2020**, *10*, 207. [\[CrossRef\]](#)
13. Carolan, M. Publicising Food: Big data, precision agriculture, and co-experimental techniques of addition. *Sociol. Rural.* **2017**, *57*, 135–154. [\[CrossRef\]](#)
14. López, I.D.; Corrales, J.C. A Smart Farming Approach in Automatic Detection of Favorable Conditions for Planting and Crop Production in the Upper Basin of Cauca River. In Proceedings of the Advances in Information and Communication Technologies for Adapting Agriculture to Climate Change, Popayán, Colombia, 22–24 November 2017; pp. 223–233.
15. Zhao, W.; Yamada, W.; Li, T.; Digman, M.; Runge, T. Augmenting crop detection for precision agriculture with deep visual transfer learning—A case study of bale detection. *Remote Sens.* **2021**, *13*, 23. [\[CrossRef\]](#)
16. Gomes, J.F.S.; Leta, F.R. Applications of computer vision techniques in the agriculture and food industry: A review. *Eur. Food Res. Technol.* **2012**, *235*, 989–1000. [\[CrossRef\]](#)
17. Rose, D.C.; Chilvers, J. Agriculture 4.0: Broadening responsible innovation in an era of smart farming. *Front. Sustain. Food Syst.* **2018**, *2*, 87. [\[CrossRef\]](#)
18. Deng, R.; Jiang, Y.; Tao, M.; Huang, X.; Bangura, K.; Liu, C.; Lin, J.; Qi, L. Deep learning-based automatic detection of productive tillers in rice. *Comput. Electron. Agric.* **2020**, *177*, 105703. [\[CrossRef\]](#)
19. Vasconez, J.P.; Delpiano, J.; Vougioukas, S.; Auat Cheein, F. Comparison of convolutional neural networks in fruit detection and counting: A comprehensive evaluation. *Comput. Electron. Agric.* **2020**, *173*, 105348. [\[CrossRef\]](#)
20. Wu, J.; Yang, G.; Yang, X.; Xu, B.; Han, L.; Zhu, Y. Automatic counting of in situ rice seedlings from UAV images based on a deep fully convolutional neural network. *Remote Sens.* **2019**, *11*, 691. [\[CrossRef\]](#)
21. Wolfert, S.; Ge, L.; Verdouw, C.; Bogaardt, M.-J. Big data in smart farming—A review. *Agric. Syst.* **2017**, *153*, 69–80. [\[CrossRef\]](#)
22. Yang, M.D.; Boubin, J.G.; Tsai, H.P.; Tseng, H.H.; Hsu, Y.C.; Stewart, C.C. Adaptive autonomous UAV scouting for rice lodging assessment using edge computing with deep learning EDANet. *Comput. Electron. Agric.* **2020**, *179*, 105817. [\[CrossRef\]](#)
23. Ward, S.; Hensler, J.; Alsalam, B.H.; Gonzalez, L. Autonomous UAVs wildlife detection using thermal imaging, predictive navigation and computer vision. In Proceedings of the IEEE Aerospace Conference, IEEE Aerospace Conference, Big Sky, MT, USA, 5–12 March 2016; pp. 1–8.
24. Yang, M.D.; Huang, K.S.; Kuo, Y.H.; Tsai, H.P.; Lin, L.M. Spatial and spectral hybrid image classification for rice lodging assessment through UAV imagery. *Remote Sens.* **2017**, *9*, 583. [\[CrossRef\]](#)
25. Driessen, C.; Heutinck, L. Cows desiring to be milked? Milking robots and the co-evolution of ethics and technology on Dutch dairy farms. *Agric. Hum. Values* **2014**, *32*, 3–20. [\[CrossRef\]](#)
26. Li, D.; Sun, X.; Elkhouchlaa, H.; Jia, Y.; Yao, Z.; Lin, P.; Li, J.; Lu, H. Fast detection and location of longan fruits using UAV images. *Comput. Electron. Agric.* **2021**, *190*, 106465. [\[CrossRef\]](#)
27. Soares, V.H.A.; Ponti, M.A.; Gonçalves, R.A.; Campello, R.J.G.B. Cattle counting in the wild with geolocated aerial images in large pasture areas. *Comput. Electron. Agric.* **2021**, *189*, 106354. [\[CrossRef\]](#)
28. Zhang, Y.; Chu, J.; Leng, L.; Miao, J. Mask-Refined R-CNN: A network for refining object details in instance segmentation. *Sensors* **2020**, *20*, 1010. [\[CrossRef\]](#) [\[PubMed\]](#)
29. Zhang, Q.; Liu, Y.; Gong, C.; Chen, Y.; Yu, H. Applications of deep learning for dense scenes analysis in agriculture: A review. *Sensors* **2020**, *20*, 1520. [\[CrossRef\]](#)
30. Kamilaris, A.; Prenafeta-Boldú, F.X. Deep learning in agriculture: A survey. *Comput. Electron. Agric.* **2018**, *147*, 70–90. [\[CrossRef\]](#)
31. Murthy, C.B.; Hashmi, M.F.; Bokde, N.D.; Geem, Z.W. Investigations of object detection in images/videos using various deep learning techniques and embedded platforms—A comprehensive review. *Appl. Sci.* **2020**, *10*, 3280. [\[CrossRef\]](#)
32. Zou, X. A review of object detection techniques. In Proceedings of the 2019 International Conference on Smart Grid and Electrical Automation (ICSGEA), Xiangtan, China, 10–11 August 2019; pp. 251–254.
33. Liu, L.; Ouyang, W.; Wang, X.; Fieguth, P.; Chen, J.; Liu, X.; Pietikäinen, M. Deep learning for generic object detection: A survey. *Int. J. Comput. Vis.* **2020**, *128*, 261–318. [\[CrossRef\]](#)
34. Etienne, A.; Ahmad, A.; Aggarwal, V.; Saraswat, D. Deep Learning-Based Object Detection System for Identifying Weeds Using UAS Imagery. *Remote Sens.* **2021**, *13*, 5182. [\[CrossRef\]](#)

35. Tong, K.; Wu, Y.; Zhou, F. Recent advances in small object detection based on deep learning: A review. *Image Vis. Comput.* **2020**, *97*, 103910. [\[CrossRef\]](#)
36. Yang, M.D.; Tseng, H.H.; Hsu, Y.C.; Tsai, H.P. Semantic segmentation using deep learning with vegetation indices for rice lodging identification in multi-date UAV visible images. *Remote Sens.* **2020**, *12*, 633. [\[CrossRef\]](#)
37. Youzi, X.; Tian, Z.; Yu, J.; Zhang, Y.; Liu, S.; Du, S.; Lan, X. A review of object detection based on deep learning. *Multimed. Tools. Appl.* **2020**, *79*, 23729–23791. [\[CrossRef\]](#)
38. Viola, P.; Jones, M. Rapid object detection using a boosted cascade of simple features. In Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Kauai, HI, USA, 8–14 December 2001; pp. 1–9.
39. Lowe, D.G. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vis.* **2004**, *60*, 91–110. [\[CrossRef\]](#)
40. Tsai, G. Histogram of oriented gradients. *Univ. Mich.* **2010**, *1*, 1–17.
41. Faruqe, O.; Hasan, M.A. Face recognition using PCA and SVM. In Proceedings of the International Conference on Anti-Counterfeiting, Security, and Identification in Communication, Hong Kong, China, 20–22 August 2009; pp. 97–101.
42. Yang, M.D.; Su, T.C.; Lin, H.Y. Fusion of infrared thermal image and visible image for 3D thermal model reconstruction using smartphone sensors. *Sensors* **2018**, *18*, 2003. [\[CrossRef\]](#)
43. Sharifara, A.; Rahim, M.; Anisi, Y. A general review of human face detection including a study of neural networks and Haar feature-based cascade classifier in face detection. In Proceedings of the 2014 International Symposium on Biometrics and Security Technologies (ISBAST), Kuala Lumpur, Malaysia, 26–27 August 2014. [\[CrossRef\]](#)
44. Moghimi, M.M.; Nayeri, M.; Pourahmadi, M.; Moghimi, M.K. Moving vehicle detection using AdaBoost and haar-like feature in surveillance videos. *arXiv* **2018**, arXiv:1801.01698.
45. Zhou, D. *Real-Time Animal Detection System for Intelligent Vehicles*; Université d'Ottawa/University of Ottawa: Ottawa, ON, Canada, 2014.
46. Pang, Y.; Yuan, Y.; Li, X.; Pan, J. Efficient HOG human detection. *Signal Process.* **2011**, *91*, 773–781. [\[CrossRef\]](#)
47. Tu, R.; Zhu, Z.; Bai, Y. Improved pedestrian detection algorithm based on HOG and SVM. *J. Comput.* **2020**, *31*, 211–221.
48. Dalal, N.; Triggs, B. Histograms of oriented gradients for human detection. In Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), San Diego, CA, USA, 20–26 June 2005; pp. 886–893.
49. Felzenszwalb, P.F.; Girshick, R.B.; McAllester, D. Cascade object detection with deformable part models. In Proceedings of the 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Francisco, CA, USA, 13–18 June 2010; pp. 2241–2248.
50. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788. [\[CrossRef\]](#)
51. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. SSD: Single shot multibox detector. In *European Conference on Computer Vision*; Springer: Berlin/Heidelberg, Germany, 2016; pp. 21–37. [\[CrossRef\]](#)
52. Wu, B.; Iandola, F.; Jin, P.H.; Keutzer, K. SqueezeDet: Unified, Small, Low Power Fully Convolutional Neural Networks for Real-Time Object Detection for Autonomous Driving. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Honolulu, HI, USA, 21–26 July 2017; pp. 446–454. [\[CrossRef\]](#)
53. Lin, T.Y.; Goyal, P.; Girshick, R.; He, K.; Dollár, P. Focal Loss for Dense Object Detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, *42*, 318–327. [\[CrossRef\]](#)
54. Law, H.; Deng, J. CornerNet: Detecting Objects as Paired Keypoints. *Int. J. Comput. Vis.* **2020**, *128*, 642–656. [\[CrossRef\]](#)
55. Tan, M.; Pang, R.; Le, Q.V. EfficientDet: Scalable and efficient object detection. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 10778–10787. [\[CrossRef\]](#)
56. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Columbus, OH, USA, 23–28 June 2014; pp. 580–587. [\[CrossRef\]](#)
57. Girshick, R. Fast R-CNN. In Proceedings of the IEEE International Conference on Computer Vision (ICCV'15), Washington, DC, USA, 7–13 December 2015; pp. 1440–1448. [\[CrossRef\]](#)
58. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards real-time object detection with region proposal networks. *arXiv* **2015**, arXiv:1506.01497. [\[CrossRef\]](#)
59. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask R-CNN. In Proceedings of the IEEE International Conference on Computer Vision (ICCV'17), Venice, Italy, 22–29 October 2017; pp. 2961–2969. [\[CrossRef\]](#)
60. Lin, T.-Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature pyramid networks for object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'17), Kalakaua Ave, HI, USA, 21–26 July 2017; pp. 2117–2125. [\[CrossRef\]](#)
61. Yang, M.D.; Tseng, H.H.; Hsu, Y.C.; Yang, C.Y.; Lai, M.H.; Wu, D.H. A UAV Open Dataset of Rice Paddies for Deep Learning Practice. *Remote Sens.* **2021**, *13*, 1358. [\[CrossRef\]](#)
62. Tan, M.; Le, Q.V. EfficientNet: Rethinking model scaling for convolutional neural networks. In Proceedings of the 36th International Conference on Machine Learning, PMLR, Long Beach, CA, USA, 9–15 June 2019; pp. 10691–10700.

-
63. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR'16), Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778. [[CrossRef](#)]
 64. Mountrakis, G.; Im, J.; Ogole, C. Support vector machines in remote sensing: A review. *ISPRS J. Photogramm. Remote Sens.* **2011**, *66*, 247–259. [[CrossRef](#)]
 65. Bottou, L.; Lin, C. Support Vector Machine Solvers. *Large Scale Kernel Mach.* **2007**, *3*, 301–320. [[CrossRef](#)]
 66. Yoshida, S. *Fundamentals of Rice Crop Science*; International Rice Research Institute: Malina, Philippines, 1981; ISBN 978-971-104-052-9.