Article

Aerial Imagery Paddy Seedlings Inspection Using Deep Learning

Mohamed Marzhar Anuar ¹, Alfian Abdul Halin ², Thinagaran Perumal ² and Bahareh Kalantar ^{3,*}

¹ Agriculture Engineering Division, Department of Agriculture Malaysia, Universiti Putra Malaysia, Serdang 43400, Malaysia; marzhar80@gmail.com

² Department of Multimedia, Faculty of Computer Science & Information Technology, Universiti Putra Malaysia, Serdang 43400, Malaysia; alfian@upm.edu.my (A.A.H.); thinagaran@upm.edu.my (T.P.)

³ RIKEN Center for Advanced Intelligence Project, Goal-Oriented Technology Research Group, Disaster Resilience Science Team, Tokyo 103-0027, Japan

* Correspondence: bahareh.kalantar@riken.jp; Tel.: +81-362252482

Abstract: In recent years complex food security issues caused by climatic changes, limitations in human labour, and increasing production costs require a strategic approach in addressing problems. The emergence of artificial intelligence due to the capability of recent advances in computing architectures could become a new alternative to existing solutions. Deep learning algorithms in computer vision for image classification and object detection can facilitate the agriculture industry, especially in paddy cultivation, to alleviate human efforts in laborious, burdensome, and repetitive tasks. Optimal planting density is a crucial factor for paddy cultivation as it will influence the quality and quantity of production. There have been several studies involving planting density using computer vision and remote sensing approaches. While most of the studies have shown promising results, they have disadvantages and show room for improvement. One of the disadvantages is that the studies aim to detect and count all the paddy seedlings to determine planting density. The defective paddy seedlings' locations are not pointed out to help farmers during the sowing process. In this work we aimed to explore several deep convolutional neural networks (DCNN) models to determine which one performs the best for defective paddy seedling detection using aerial imagery. Thus, we evaluated the accuracy, robustness, and inference latency of one- and two-stage pretrained object detectors combined with state-of-the-art feature extractors such as EfficientNet, ResNet50, and MobilenetV2 as a backbone. We also investigated the effect of transfer learning with fine-tuning on the performance of the aforementioned pretrained models. Experimental results showed that our proposed methods were capable of detecting the defective paddy rice seedlings with the highest precision and an F1-Score of 0.83 and 0.77, respectively, using a one-stage pretrained object detector called EfficientDet-D1 EfficientNet.

Keywords: paddy seedlings; computer vision; object detection; deep learning; convolutional neural networks

Citation: Anuar, M.M.; Halin, A.A.; Perumal, T.; Kalantar, B. Aerial Imagery Paddy Seedlings Inspection Using Deep Learning. *Remote Sens.* **2022**, *14*, 274. <https://doi.org/10.3390/rs14020274>

Academic Editors: Vasit Sagan, Sidike Paheding, Maitiniyazi Maimaitijiang and Matthew Maimaitiyiming

Received: 30 November 2021

Accepted: 3 January 2022

Published: 7 January 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

The biggest challenge in today's agriculture industry is to ensure that the growing population is sufficiently supplied by current levels of food production [1,2]. Rice is the primary, and the world's second largest, staple food in Asian countries and contributes 90% of the total food production [1,3]. In recent years complex food security issues caused by climate changes, the limitations of human labour, and increasing production costs require a strategic approach in addressing the problem [1,4]. The emergence of artificial intelligence due to the capability of recent advances in computing architecture could become a new alternative to existing solutions. Deep learning algorithms in computer vision for image classification and object detection can facilitate the agriculture industry, especially in paddy cultivation, to alleviate human efforts in laborious, burdensome, and repetitive tasks [5].

We have seen numerous successes in applied deep convolutional neural network (DCNN) algorithms for agricultural research in areas dealing with issues related to pest and disease control, weed management, crop recognition, plant health, and planting density [5,6]. DCNN has also contributed to the emergence of high spatial remote sensing and temporal remote sensing applications using satellite and unmanned aerial vehicle (UAV) imagery for a wide range of agriculture applications as popular and cost-effective solutions. Remote sensing images in RGB (red, green, blue), multispectral, and hyperspectral applications have been widely used for land classification tasks and precision agriculture [7]. Multispectral technologies such as normalized difference vegetation index (NDVI), enhanced vegetation index (EVI) and thermal inspection have shown enormous success in measuring vegetation, land use classification, and marine monitoring [8]. However, due to the characteristics of remote sensing data, there are some limitations and practical challenges. Recently, new DCNN approaches that combine remote sensing applications have achieved significant breakthroughs, offering novel opportunities for research and development in remote sensing images on automated image classification and object detection [9]. This new technique can be applied to agricultural industries for solving various issues, especially in precision agriculture.

Optimal planting density is a crucial factor in paddy cultivation as it will influence the quality and quantity of production [10]. Thus, it is necessary for the farmers to inspect the paddy fields to search for defective paddy seedlings at approximately 14 days after planting. Then the defective paddy seedlings are replaced or replanted manually. This process is called sowing and is known to be labour-intensive as well as being prone to disease outbreaks [11]. There have been several studies involving planting density using computer vision and remote sensing approaches. While most of the studies have shown promising results, they have disadvantages and leave room for improvement. One of the disadvantages is that the studies aim to detect and count all the paddy seedlings to determine planting density. The defective paddy seedling locations are not pointed out to help farmers during the sowing process. As a solution, semi-automated or fully automated methods for detecting defective paddy seedlings need to be developed to enhance the current method. All the previous studies used two-stage method object detectors like Faster RCNN and Mask RCNN combined with different types of feature extractor architectures, such as VGG16, ResNet, Inception, and MobileNet. Even though this type of object detector showed high accuracy, it has a high inference time in real-time detection. The fact that near or real-time defective crop detection may lead to the development of an autonomous mobile transplanter is another reason why we are looking into this work. There have been several studies that have used multispectral technologies like NDVI and EVI associated with high-resolution remote sensing imagery to determine plant health. However, for this work we decided to use RGB images to accommodate farmers' accessibility constraints in technologies and expensive equipment. To the best of our knowledge with regards to paddy seedling detection, only one study has used a one-stage method object detector. In this work, we aimed to explore several DCNN models to determine which one performs the best for defective paddy seedling detection using aerial imagery. Thus, we evaluated the accuracy, robustness, and inference latency of one- and two-stage pretrained object detectors combined with state-of-the-art feature extractors such as EfficientNet, ResNet50 and MobilenetV2 as a backbone. We also investigated the effect of transfer learning with fine-tuning on the performance of the aforementioned pretrained model.

2. Related Work

Studies have been conducted by researchers around the globe for the past few years that have utilized DCNN in object detection to recognize various types of crops [12]. The main idea behind these studies is to determine the capability of pretrained models to detect new objects via transfer learning with a fine-tuning method. Most of the studies have shown promising results. Pearlstein et al. [13] applied convolutional neural networks (CNN) with

the adoption of AlexNet to detect weeds in lawn grass based on synthetic imagery used as the training image and achieved a test accuracy of 100%. Li et al. [14] proposed a deep learning-based framework to detect and count oil palm trees using high-resolution remote sensing images and achieved 98% test accuracy. Baweja et al. [15] developed a novel pipeline that accurately detected object regions using dense semantic segmentation for extracting stalk counts and stalk width. The stalk count was 30 times faster and the stalk width measurement was 270 times faster compared to previous work. Consequently, Liu et al. [16] proposed a gamma count model to determine wheat crop planting density and quantify the plant spacing heterogeneity. Wu et al. [17] proposed an efficient method that used computer vision to count rice seedlings in digital images. It adopted a regression network to estimate the rice seedlings in an image captured using UAV and achieved a test accuracy higher than 93%. Ma et al. [18] achieved a test accuracy rate of 92.7% by proposing SegNet, that used a semantic segmentation algorithm to locate rice seedlings and weeds. Then Neupane et al. [19] used high-resolution RGB images captured from UAV to train a deep learning model with three image processing techniques for the detection and counting of banana plants. They achieved accuracy rates of 96.4%, 85.1% and 75.8%, respectively. Kitano et al. [20] developed a technique to count corn plants based on UAV images with a network that used a blob detector algorithm called U-Net. Desai et al. [21] proposed a simple pipeline to detect regions that contain flowering panicles from ground level RGB images of paddy rice using DCNN. Madec et al. [22] explored the potential use of a double stage method object detector called Faster R-CNN combined with Tessel-Net as a feature extractor to detect ear density using nadir high spatial resolution RGB images and discovered that Faster R-CNN was more robust when applied to a dataset at a later stage with ears and background showing a different aspect due to higher plant maturity. Jiang et al. [23] built a deep learning-based approach to count plant seedlings in fields. The models achieved an F1 score of 0.727 (at IoUall) and 0.969 (at IoU0.5). Valente et al. [24] proposed crop plant counting using very high-resolution images based on excess green index and Otsu's method together with transfer learning to identify and count plants. Cao et al. [25] then proposed EfficientDet network to detect the wheat ear image. The objective was to frame out the wheat ear image and then count the detected target number for automatic counting of wheat ears. They gained an average accuracy of 95.8%. Liu et al. [26] used a deep learning-based approach called the scale-fusion counting classification network (SFC2 Net) that integrates several types of network. Particularly, SFC2 Net addressed appearance and illumination changes by applying a multicolumn pretrained network with multilayer feature fusion to enhance feature extractions. The work in paper [27] proposed the recognition method of no-seedlings grids of trays based on a pretrained model named Inception-V3 and gained average test accuracy of 81.5%. Recently, Zhang et al. [28] proposed an intersection over union (IoU) method to calculate image segmentation thresholds for detecting defective cabbage seedlings. They achieved precision, recall, and accuracy of 97.6%, 97.4%, and 99.8%. Li et al. [29] proposed an automatic approach for detecting seedlings per hill. They used an algorithm that detected the endpoints of skeletonized seedling hills to represent the leaf tips of seedlings as endpoints. The overall detection accuracy was up to 93.5%. Then Lin et al. [30] developed a plant row detection algorithm that served as a navigation system of a rice transplanter based on CNN to locate rice seedlings from field images. Bah et al. [31] obtained a crop row detection of 93.58% by introducing a method called CRowNet based on CNN. In paper [32] they developed a system that detected the entire crop row instead of individual crops using geometric descriptor information with deep neural networks. Yu et al. [33] proposed crop row segmentation and detection algorithms for complex paddy fields. Lately, explorations on the potential of pretrained object detectors via transfer learning have become more favorable. Deng et al. [34] used this method with a target detection framework to detect weeds at the seedling stage in paddy fields. This was followed by Yarak et al. [35], who used Faster RCNN for the automatic detection and health classification of palm oil trees. Xu et al. [36] developed a weed density detection based on an absolute feature corner

point (ACFP) algorithm. The test accuracy reached 93%. Anami et al. [37] designed a DCNN framework for the automatic recognition and classification of various biotic and abiotic paddy crop stresses using field images by adopting a pretrained VGG-16. The model achieves an accuracy of 92.89%. In recent years, DCNN integrated with remote sensing potential has been explored to solve various agriculture issues. Ramanath et al. [38] compared hand-crafted features based on NDVI and feature learning from CNN for land cover classification of aerial images, which resulted in CNN with batch normalization to achieve the best performance. An et al. [39] combined hyperspectral remote sensing data with four advance machine learning techniques to estimate the chlorophyll content of rice. They concluded that random forest regression is the best machine learning technique for the task. Ammar et al. [40] proposed an original deep learning framework for the automated counting and geolocation of palm trees from aerial images using CNN. The result showed that YOLOv4 and EfficientDet-D5 yielded the best trade-off between accuracy and speed, with up to 99% mean average precision and 7.4 FPS. Although all the works mentioned above achieved good accuracy by using CNN to either classify an image or detect a target object there is no work that focuses on detecting defective paddy seedlings to determine planting density. Therefore, we proposed a method that not only detects the defective paddy seedlings but also points out their locations.

3. Study Area, Materials, and Methods

3.1. Study Area and Image Acquisition

Several paddy fields (as denoted by the red square in Figure 1) located in the sub-district of Tanjung Karang within the Kuala Selangor District (3.3491° N Latitude North, 101.2467° E Longitude East) in the state of Selangor, Malaysia were chosen for data (image) collection. The location is shown in Figure 1. The planting date for all fields was the 4th of January 2021. In all, 197 images of paddy seedlings were collected on the 10th to 14th days after planting. All the images were collected using the multirotor DJI Phantom 4 drone using an on-board RGB camera with the specification details as follows:

<i>Sensors</i>	1/2.3 CMOS Effective pixels: 12.4 M
<i>Lens</i>	FOV 94° 20 mm (35 mm format equivalent) f/2.8 focus at ∞
<i>ISO Range</i>	100–320 (Video) 100–1600 (Photo)
<i>Electronic Shutter Speed</i>	8–1/8000 s
<i>Image Size</i>	4000 × 3000

The flight altitude was five meters whereas flight speed was five meters per second. The camera angle was set to face downwards.

Figure 1. The location of the study area: (a) map of Malaysia; (b) aerial images of the study area. The red square shows the location of the several paddy fields chosen.

3.2. Data Annotations

The targeted object in each training image was annotated within its respective bounding box. Basically, the area of interest in every image containing the unhealthy paddy seedlings are ‘marked’ to be within specific rectangular coordinates, forming the bounding boxes (Figure 2). Each bounding box is hence the labelled targeted class, which acts as a training input for our deep learning networks. Using the labelImg software (<https://github.com/tzutalin/labelImg> (accessed on 4 June 2021), we annotated a total of 3980 bounding boxes labelled as ‘unhealthy’ from 197 images. The images were then split into training and testing sets with a ratio of 80:20, respectively. Table 1 shows the number of images and bounding boxes in the training and testing datasets.

Figure 2. The process of data annotations using labelImg Software.

Table 1. Number of images and bounding boxes in the training and testing datasets.

	Training Dataset	Testing Dataset
Number of Images	157	40
Percentage	80%	20%
Bounding Boxes Labelled “Unhealthy”	3140	840

3.3. Methodology Applied

3.3.1. Image Pre-Processing

Image pre-processing is a common step to prepare the image dataset for input into CNN, which includes noise removal, intensity manipulation, and resizing. In this study, the image pre-processing steps we employed were:

brightness adjustment and contrast enhancement: to reduce the factor influences of the background;

- (i) resizing all images to 640×640 px from the original size of 4000×3000 px to match the default input image size of all the pretrained models selected.
- (ii) An example of the image before and after pre-processing is presented in Figure 3.

Figure 3. Example of: (a) original image; and (b) after pre-processing.

3.3.2. Proposed Method

In this work, we used one- and two-stage method pretrained object detectors from the TensorFlow2 object detection zoo. For the one-stage method we chose EfficientDet-D1 and Single Shot Detector (SSD). For the two-stage method, Faster R-CNN was selected. These object detectors were combined with different state-of-the-art feature extractors such as EfficientNet, ResNet50, and MobileNetV2 as a backbone. In this work, we experimented with the following combinations of object detector + feature extractor:

- (i) EfficientDet-D1 + EfficientNet;
- (ii) SSD + MobileNetV2;
- (iii) SSD + ResNet50;
- (iv) Faster R-CNN + ResNet50.

Pre-Trained Object Detector

EfficientDet-D1

EfficientDet-D1 belongs to the family of EfficientDet and is a one-stage method network. It proposes a weighted bi-directional feature pyramid network (BiFPN) which is an enhancement from conventional feature pyramid networks (FPN). The main difference is that it allows for fast multi-scale feature fusion and uses a compound method to uniformly scale the resolution, depth, and width for all backbone, feature network, and box/class prediction networks at the same time [41]. It also can efficiently scale concerning hardware resources constraints.

Single Shot Detector (SSD)

SSD was oriented for real-time object detection and is also a one-stage network. It accelerates the detection process by removing the dependence on a region proposal network and proposes some improvements with multi-scale features and default boxes to maintain accuracy. These improvements allowed it to match the Faster R-CNN's accuracy using lower resolution images, and accelerated the speed of the whole process [42].

Faster R-CNN

In contrast with the one-stage, Faster R-CNN is a two-stage method. It contains two networks: a region proposal network (RPN) for generating region proposals and a feature extractor network that uses these proposals to classify an object inside an image [43]. A selective search to generate region proposals is what differentiates Faster R-CNN from Fast R-CNN. The RPN time cost of generating region proposal is much smaller than selective search, when most computation is shared with the object detection network. Anchors or the RPN ranks region boxes will propose the one that is most likely to contain an object.

Feature Extractor

The feature extractor is DCNN, and consists of an input layer, hidden layer, and output layer. Hidden layers receive information from the previous layer and send information to the next layer. Thus, a hidden layer must have an activation function. All our selected feature extractors used rectified linear unit (*ReLU*) as an activation function and can be described with Equation (1):

$$ReLU = \max(0, x) \quad (1)$$

Filters consist of some numbers called weights with random initialization that connect one layer to another layer in the DCNN. These weights are optimized through forward and backward propagations. Filters are used to extract feature maps from our two-dimensional paddy seedlings input image using a convolutional operation, as denoted using Equation (2), where I is an input image and filters as K . It was then followed by a pooling process to reduce variance and dimension from the convolutional layer to the output neurons because it only selects the maximum value in each feature map. The pooling process is also known to prevent overfitting.

$$S(i, j) = conv(I * K)(i, j) = \sum_m \sum_n I(m, n) K(i - m, j - n) \quad (2)$$

The processes of the convolutional operation and pooling were repeated several times, depending on the complexity and depth of the network, before feeding it to the fully connected layer or output layer for classification tasks. We presented the overview of feature extractor structure in Figure 4.

We have selected three DCNN architectures as feature extractors. These are ResNet50, MobileNetV2, and EfficientNet. Resnet50 consist of 50 layers and identity blocks in five stages. Each identity block has three convolutional layers, and it has more than 23 million trainable parameters. MobileNetV2 uses a method called depth-wise separable convolutions to replace the standard convolutions found in earlier model architectures to make it lighter. Two new global hyperparameters were introduced, a width multiplier and a resolution multiplier which allows a trade-off between accuracy for speed and low size [44]. It was based on an inverted residual structure where the residual connections were between the bottleneck layers. The intermediate expansion layer used lightweight depth-wise convolutions to filter features as a source of non-linearity. The architecture of

MobileNetV2 contains the initial full convolution layer with 32 filters, followed by 19 residual bottleneck layers. On the other hand, EfficientNet-D1 uses a compound coefficient to uniformly scale all dimensions of depth/width/resolution. The so-called compound scaling method is justified by the intuition that the bigger the input image, the more layers are needed for the networks to increase receptive fields, and more channels are needed to capture more fine-grained patterns. EfficientNet network is based on the inverted bottleneck residual blocks of MobileNetV2 [39]. The properties of all the feature extractors selected are presented in Table 2.

Figure 4. The overview of feature extractor structure.

Table 2. Properties of the selected feature extractor.

Feature Extractor	Parameters (Million)	Number of Layers
Resnet 50	25	50
MobileNet V2	3.4	53
EfficientNet	6.5	237

We proposed a method to detect defective paddy seedlings using a pretrained model with image pre-processing and parameter tuning. The earlier mentioned combinations were chosen because of their comparability in terms of mean average precision (mAP) and speed-based on a Microsoft common object in context (MS COCO) dataset. All the models have the same input size of 640×640 px and were trained via transfer learning with a fine-tuning approach. The overall framework is shown in Figure 5.

Figure 5. The Overall framework.

Transfer Learning

A list of object detection models in Tensorflow2 object detection zoo was selected which in our perception was the best for our task, with a good understanding and knowledge for each model. These pretrained models were then used via transfer learning for feature extraction and new detection with some parameter tuning involved. Generally, transfer

learning means reusing a pretrained model which has been trained previously to detect a new object. In common practice, weights in the early and middle layers of the network are frozen, and only the last layer is retrained. Freezing a layer means the layer does not need any adjustment made on the weights and does not receive any updates during the training process. The networks will leverage the weights from the large amount of labelled data in which it was initially trained, and it will learn the pattern that separates previous objects from the new objects. By training our model via transfer learning, the weights of the pretrained model, which has already learned to extract vital features or information such as color, shape and texture from previous training tasks, is then applied to our new detection task. The model will basically be exploiting what has previously been learned in a previous detection task to improve the generalization in our new task. Reasonably, the pretrained model will transfer as much information gained from its previous training process to the new detection task. The number of layers that need to be retrained depends on the complexity of the new task. In this work, we examined the complexity of our new detection task and decided to only re-train the last layer. We used the transfer learning approach because of its ability to re-train the networks to achieve a reasonably high detection accuracy with a small amount of data. One more benefit by leveraging this method is that our learning process during training does not start from scratch but instead uses previous learning patterns to reduce computational time.

This approach is the most widely used nowadays in the fields of deep learning for computer vision. With this approach the network itself extracts the features and is capable of learning which features are important, and which are not, from the defective paddy seedlings input images. It is also capable of discovering a good combination of features within a few iterations or epochs, thereby eliminating the need for expert hand-crafted feature engineering which is important in a machine learning approach to improve overall performance.

Training the Model with Fine Tuning

Training the pretrained model for our new detection task was quite resource-intensive because the algorithms were required to analyze the data (images) and learn their patterns. For this study, the pretrained model saved weights which were trained with MS COCO dataset and used as an initialization in the feature extractor network to make the training process time and resources efficient for our new detection task. In our training process, we used a Dell Alienware Laptop with Intel®Core™ i7-8750H CPU @ 2.20 GHz 2.21 GHz, 16 GB RAM and NVIDIA GeForce GTX 1070 GPU. We made a minor adjustment to the pretrained model network to achieve the level of desired performance. The fine-tuning parameter involved in this study changed some parts in the configuration pipeline, such as using different training batch sizes and learning rates for each pretrained model. The hyperparameter and details of the pretrained object detector selected are in Table 3. All the pretrained model code repositories are available at https://github.com/tensorflow/models/blob/master/research/object_detection (accessed on 20 July 2021).

Table 3. Hyperparameter details of each pretrained model.

Object Detector	Feature Extractor	Input Size	Learning Rate	Batch Size	Number of Steps
EfficientDet-D1	EfficientNet	640×640	1×10^{-3}	3	6400
SSD	MobileNetV2	640×640	2.6×10^{-2}	6	4800
SSD	ResNet50	640×640	1.3×10^{-2}	4	5600
Faster R-CNN	ResNet50	640×640	1.3×10^{-2}	1	3200

Evaluating the Performance Metrics

For the model performance, we looked at the intersection over union (IoU) that was introduced in the pattern analysis, statistical modelling, and computational learning visualize object classes (Pascal VOC) challenge. By calculating the IoU, we were able to derive

the more descriptive metrics of precision, recall and F1 Score. All these three, furthermore, are described in Equations (3)–(5). IoU is the percentage of overlap between the predicted box with the ground truth box (Figure 6). It can be quantified using Equation (6). Due to the complexity of the detection problems, we chose (IoU0.50) as our minimum threshold.

Figure 6. The intersection over union (IoU) method of overlapping percentage.

Within the context of this work, the following applies:

- (i) True Positive (TP)—the number of correctly detected unhealthy paddy seedlings;
- (ii) False Positive (FP)—the number of healthy paddy seedlings detected as unhealthy;
- (iii) False Negative (FN)—the number of unhealthy paddy seedlings that are not detected;
- (iv) F1 score—a measure of the model’s accuracy on a dataset and can be defined as the harmonic mean of precision and recall.

$$Precision = \frac{TP}{TP + FP} \quad (3)$$

$$Recall = \frac{TP}{TP + FN} \quad (4)$$

$$F1 \text{ Score} = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (5)$$

$$IoU = \frac{area(\beta p \cap \beta gt)}{area(\beta p \cup \beta gt)} \quad (6)$$

4. Results and Discussion

4.1. Overall Performance

In this study, we aimed to detect and count defective paddy seedlings from aerial imagery using four different combinations of a pretrained CNN object detector model based on DCNN with its respective feature extractor. We trained four models with different combinations of object detectors and feature extractors via transfer learning with fine-tuning. Training a model means optimizing the values for all the weights and biases from labelled examples. Loss indicates how well the model is fitting the data. Thus, it is important to monitor the loss during training. To avoid overfitting, we stopped the training when the loss reached between 0.15–0.20 for each model. The total loss of the models is shown in Figure 7.

Figure 7. The total loss of each model during the training process.

The total loss decrease rate depends on the complexity of the model architecture. This is clearly visualized in Figure 7, which shows that the SSD Resnet 50 and SSD MobileNetV2 from the one-stage method object detector took fewer steps to reach a total loss of 0.15–0.20 compared to the EfficientDet-D1 EfficientNet which was built using the same one-stage method but with a more complex architecture. Faster-RCNN from the two-stage method meant that it was the fastest model to reach the optimum loss value because the training batch size was limited to one and the region proposal networks split the training images into smaller pieces before sending it to the classifier. This also proved that transfer learning methods with fine-tuning significantly reduce training times. The average training time of all the pretrained models used ranged between 2–5 h. Then we evaluated and compared the performance of the model as presented in Table 4.

Table 4. The overall performance of the models.

No.	Model		Performance Metrics		
	Object Detector	Feature Extractor	Average Precision	Average Recall	F1-Score
1	EfficientDet-D1	EfficientNet	0.83	0.71	0.77
2	SSD	MobileNetV2	0.77	0.57	0.66
3	SSD	ResNet50	0.86	0.49	0.62
4	Faster R-CNN	ResNet50	0.82	0.64	0.72

EfficientDet-D1 + EfficientNet- showed the highest overall performance with average precision and an F1-Score of 0.83 and 0.77. Faster R-CNN + Resnet50, which is a two-stage method, showed a comparable achievement with EfficientDet + EfficientNet-D1, a one-stage method object detector with average precision and an F1-Score of 0.82 and 0.72, respectively. Both SSD object detectors with different feature extractors showed quite a lower performance value compared to EfficientDet and Faster-RCNN. However, SSD ResNet50 showed the highest average precision of 0.86 among all four pretrained models but with the lowest recall and F1-Score. Figure 8 compares the overall performance of all the tested pretrained models.

The two-stage method object detector Faster R-CNN ResNet50 used in this study was expected to give a better result when compared to the one-stage method. However, EfficientDet-D1 EfficientNet, which is a one-stage method, achieved the highest F1 score of 0.77 compared to Faster R-CNN ResNet50 and all other networks that were mostly used in previous related studies. This showed that the bi-directional feature pyramid network (BiFPN) can be enhanced with fast normalization and feature fusion which also allows multi-scale feature fusion and is capable of achieving high precision with a low inference time. The model also showed an outstanding capability on counting performance

and works well for real-time detection. Figure 9 shows the detection of defective paddy seedlings using the proposed pretrained object detector.

Figure 8. Showing comparison of the performance metrics on all tested pretrained model.

Detecting defective paddy seedlings can be categorized as small object detection. It is important to ensure that all the characteristics of defective paddy seedlings are captured, and that the noise caused from the illumination background is minimized for the purpose of training of the DCNN object detector. Thus, the flight altitude was set to low at five meters above the crop during image acquisition. The Resnet50 feature extractor seems to perform well on this task with a good precision on both one-stage and two-stage object detectors as shown in Table 4 and Figure 8. The confidence rate showed how likely the model predicted that the boxes containing the desired object, in this case a defective paddy seedling. To score how well the prediction performed depended on both classification and bounding box regression based on IoU. Unlike detecting common objects like persons or vehicles, defective paddy seedlings come in various shapes, sizes, and colors with few similarities. Thus, the confidence threshold was set at minimum of 0.30 on the test images of all four of the pretrained models tested so that it detected as many defective paddy seedlings as possible. There were a few possible contributing factors in the small confidence score and FN or misdetection. First, annotation of bounding boxes was not properly centered, the boxes overlapped and were too close to each other, leading to a poor training example. Secondly, there were insufficient training images to generalize the defective paddy seedlings. Third, even though pre-processing minimizes factors caused by variance and inconsistency, perhaps more augmentation techniques, such as flipping, rotation, shearing, cropping and zooming should be considered to tackle large variation issues. Increasing the input image resolution will also help to increase the model's performance, especially in small object detection. False negatives may also have resulted from a smaller IoU than the minimum threshold, and different background textures or colors. Figure 10 shows an example of TP, FP, and FN or misdetection.

Despite achieving considerably good results, constructing or re-engineering an object detection model or feature extractor specifically for the purpose of detecting defective paddy seedlings to increase performance is an open debate for future works and beyond the scope of this study. Parameter tuning enhanced the overall performance of the model and helped with overfitting problems. It also reduced the computational time and cost without hurting the performance of the model. Both one-stage and two-stage pretrained object detectors were showing a good capability in detecting defective paddy seedlings with relatively small inferencing times by using transfer learning with fine-tuning. The overall performance of the pretrained object detector and the state-of-the-art feature extractor

showed promising results in detecting defective paddy seedlings using aerial imagery for large fields.

Figure 9. Detection result of defective paddy seedlings using one-stage and two-stage method pretrained object detectors: (a) EfficentDet-D1+EfficientNet; (b) Faster R-CNN+ResNet50; (c) SSD+MobileNetV2; and (d) SSD+ResNet50.

Figure 10. Showing an example of true positive, false positive and false negative. Red bounding boxes represent the annotation of labelled bounding boxes while the green boxes are the predicted boxes of the model. Boxes with color overlapping in (b) are the correctly detected defective paddy seedlings or true positive, the non-overlapping red boxes is the false negative or misdetection and the non-overlapping green boxes are false positive, or healthy paddy seedlings detected as unhealthy or defective. (a) Input Image, (b) Ground Truth, (c) Output Images.

4.2. Comparison with Other Approaches

Performance comparison from the experimental results of the selected pretrained model demonstrated the capability of each model in detecting defective paddy seedlings using aerial images in RGB form at a low altitude. Hence, the results were encouraging for exploration with deep learning-based object detector for detecting new objects via transfer learning with fine-tuning. Discussing several other potential methods could expand the study of detecting defective paddy seedlings using aerial imagery. First, using multispectral sensors technologies such as NDVI where a dimensional index is used to describe the difference between visible and near infra-red reflectance could determine the defective paddy seedlings by comparing the average NDVI index value between healthy and defective seedlings without requiring a lot of labelled training images. Although this method is easier to perform, the price of a multispectral camera can be considered as expensive to most farmers. Thus, such technologies may not be transferable to farmers. Defective paddy seedlings are detected based on shape, color, and size, which cannot be determined with only a reflectance index value. Different NDVI index values with the healthy paddy seedlings may not only lead to defective paddy seedlings but also to nutrient deficiency and pest- or disease-related issues which do not require replanting processes.

Second, high-resolution spatial imagery is also another interesting subject in detecting defective paddy seedlings, especially for large areas. This high-resolution spatial image can be integrated with a deep learning model using tools in software such as ArcGIS pro and train to detect new desired objects. Furthermore, it can also automatically determine the geolocation (latitude and longitude) of the detected object. Although it seems like the most promising state-of-the-art approach for accessing and working with high-resolution images, images of below 50 cm may face an availability problem, making it less suitable for capturing the characteristics of small object.

Although training a deep learning model for object detection using tools in GIS software is faster and less of a hassle, it has limitations due to the unavailability of some deep learning object detectors and feature extractor models. On the other hand, using Tensorflow object detection API, which was built as an open-source framework, was easier to deploy, transferable to a single board computing device, and with edge computing all the data were connected for further use.

5. Conclusions

Fine-tuning a pretrained model improved the accuracy of transfer learning and proved to be efficient in training for the detection of new objects. It simplified the process and reduced the training time. It also provided a useful feature for the new model, making it much more reliable and robust. Fine-tuning pushed the evolution of deep learning by making the new algorithm developments more efficient in many aspects. In this study we evaluated the accuracy, robustness, and inference latency of one-stage and two-stage object detectors combined with a different feature extractor. Four combinations were used, with the EfficientDet-D1 EfficientNet outperforming all other models. The aim of the study was to find the most suitable model for detecting defective paddy seedlings. Results and

comparisons of all the pretrained models selected in this experimental study demonstrated how well the deep learning-based object detector detected defective paddy seedlings via transfer learning with fine-tuning. We hope that our studies will make a significant contribution to the field of computer vision and open a new page of endless possibilities for determining planting density. However, for generalization and future work, more training images are required due to the high variation in the shape, size, and color of the defective paddy seedlings, and different techniques for image pre-processing should be applied for larger test images. Constructing a new or re-engineered present object detector model specifically for detecting defective paddy seedlings may also increase the performance of the model.

Author Contributions: Conceptualization, M.M.A. and A.A.H.; methodology, M.M.A. and A.A.H.; software, M.M.A. and A.A.H.; validation, A.A.H. and T.P.; formal analysis, A.A.H. and T.P.; investigation, A.A.H., T.P. and B.K.; resources M.M.A. and A.A.H.; data curation, M.M.A., A.A.H. and T.P.; writing—original draft preparation, M.M.A.; writing—review and editing, A.A.H., T.P. and B.K.; visualization, B.K.; supervision, A.A.H. and T.P., B.K.; project administration, A.A.H. and T.P.; funding acquisition, A.A.H. and T.P. All authors have read and agreed to the published version of the manuscript.

Funding: The APC is supported by the RIKEN Centre for Advanced Intelligence Project (AIP), Tokyo, Japan.

Institutional Review Board Statement: Not applicable.

Data Availability Statement: The data used to support the findings of this study are available from the corresponding author upon request.

Acknowledgments: The authors would like to thank the Universiti Putra Malaysia, Malaysia and RIKEN Centre for AIP, Japan, for providing all facilities during the research.

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

References

1. Firdaus, R.B.R.; Tan, M.L.; Rahmat, S.R.; Gunaratne, M.S. Paddy, rice and food security in Malaysia: A review of climate change impacts. *Cogent Soc. Sci.* **2020**, *6*, 1818373. [\[CrossRef\]](#)
2. Patel, P.P.; Vaghela, D.B. Crop Diseases and Pests Detection Using Convolutional Neural Network. In Proceedings of the 2019 3rd IEEE International Conference on Electrical, Computer and Communication Technologies (ICECCT), Coimbatore, India, 20–22 February 2019; pp. 2019–2022. [\[CrossRef\]](#)
3. Maclean, B.H.; Dawe, J.L.; Hardy, D.C. *Rice Almanac*; IRRI: Los Banos, Philippines, 2013.
4. Food and Agriculture Organization of the United Nations. *The Future of Food and Agriculture: Trends and Challenges*; Food and Agriculture Organization of the United Nations: Rome, Italy, 2017; Volume 4, ISBN 1815-6797.
5. Kamilaris, A.; Prenafeta-Boldú, F.X. Deep learning in agriculture: A survey. *Comput. Electron. Agric.* **2018**, *147*, 70–90. [\[CrossRef\]](#)
6. Silva, M.F.; Lima, J.L.; Reis, L.P.; Sanfeliu, A.; Tardioli, D. Correction to Robot 2019: Fourth Iberian Robotics Conference. *Adv. Intell. Syst. Comput.* **2020**, *1092*, C1. [\[CrossRef\]](#)
7. Tsouros, D.C.; Bibi, S.; Sarigiannidis, P.G. A review on UAV-based applications for precision agriculture. *Information* **2019**, *10*, 349. [\[CrossRef\]](#)
8. Liu, J.; Xiang, J.; Jin, Y.; Liu, R.; Yan, J.; Wang, L. Boost Precision Agriculture with Unmanned Aerial Vehicle Remote Sensing and Edge Intelligence: A Survey. *Remote Sens.* **2021**, *13*, 4387. [\[CrossRef\]](#)
9. Li, Y.; Zhang, H.; Xue, X.; Jiang, Y.; Shen, Q. Deep learning for remote sensing image classification: A survey. *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.* **2018**, *8*, e1264. [\[CrossRef\]](#)
10. Calero Hurtado, A.; Pérez Díaz, Y.; Quintero Rodríguez, E.; González-Pardo Hurtado, Y. Densidades de plantas adecuadas para incrementar el rendimiento agrícola del arroz. *Cent. Agrícola* **2021**, *48*, 28–36.
11. Kumar, M.; Dogra, R.; Narang, M.; Singh, M.; Mehan, S. Development and Evaluation of Direct Paddy Seeder in Puddled Field. *Sustainability* **2021**, *13*, 2745. [\[CrossRef\]](#)
12. Yang, X.; Sun, M. A survey on deep learning in crop planting. In *IOP Conference Series: Materials Science and Engineering*; IOP Publishing: Bristol, UK, 2019; Volume 490, p. 062053. [\[CrossRef\]](#)
13. Pearlstein, L.; Kim, M.; Seto, W. Convolutional neural network application to plant detection, based on synthetic imagery. *Proc.-Appl. Imag. Pattern Recognit. Work* **2017**, 1–4. [\[CrossRef\]](#)
14. Li, W.; Fu, H.; Yu, L.; Cracknell, A. Deep Learning Based Oil Palm Tree Detection and Counting for High-Resolution Remote Sensing Images. *Remote Sens.* **2017**, *9*, 22. [\[CrossRef\]](#)

15. Baweja, H.S.; Parhar, T.; Mirbod, O.; Nuske, S. StalkNet: A Deep Learning Pipeline for High-Throughput Measurement of Plant Stalk Count and Stalk Width. In *Field and Service Robotics*; Springer: Cham, Switzerland, 2018; pp. 271–284. [\[CrossRef\]](#)
16. Liu, S.; Baret, F.; Allard, D.; Jin, X.; Andrieu, B.; Burger, P.; Hemmerlé, M.; Comar, A. A method to estimate plant density and plant spacing heterogeneity: Application to wheat crops. *Plant Methods* **2017**, *13*, 38. [\[CrossRef\]](#)
17. 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. [\[CrossRef\]](#)
18. Ma, X.; Deng, X.; Qi, L.; Jiang, Y.; Li, H.; Wang, Y.; Xing, X. Fully convolutional network for rice seedling and weed image segmentation at the seedling stage in paddy fields. *PLoS ONE* **2019**, *14*, e0215676. [\[CrossRef\]](#)
19. Neupane, B.; Horanont, T.; Hung, N.D. Deep learning based banana plant detection and counting using high-resolution red-green-blue (RGB) images collected from unmanned aerial vehicle (UAV). *PLoS ONE* **2019**, *14*, e0223906. [\[CrossRef\]](#)
20. Kitano, B.T.; Mendes, C.C.T.; Geus, A.R.; Oliveira, H.C.; Souza, J.R. Corn Plant Counting Using Deep Learning and UAV Images. *IEEE Geosci. Remote Sens. Lett. (Early Access)* **2019**, 1–5. [\[CrossRef\]](#)
21. Desai, S.V.; Balasubramanian, V.N.; Fukatsu, T.; Ninomiya, S.; Guo, W. Automatic estimation of heading date of paddy rice using deep learning. *Plant Methods* **2019**, *15*, 76. [\[CrossRef\]](#)
22. Madec, S.; Jin, X.; Lu, H.; De Solan, B.; Liu, S.; Duyme, F.; Heritier, E.; Baret, F. Ear density estimation from high resolution RGB imagery using deep learning technique. *Agric. For. Meteorol.* **2018**, *264*, 225–234. [\[CrossRef\]](#)
23. Jiang, Y.; Li, C.; Paterson, A.H.; Robertson, J.S. DeepSeedling: Deep convolutional network and Kalman filter for plant seedling detection and counting in the field. *Plant Methods* **2019**, *15*, 141. [\[CrossRef\]](#) [\[PubMed\]](#)
24. Valente, J.; Sari, B.; Kooistra, L.; Kramer, H.; Mucher, S. Automated crop plant counting from very high-resolution aerial imagery. *Precis. Agric.* **2020**, *21*, 1366–1384. [\[CrossRef\]](#)
25. Cao, L.; Zhang, X.; Pu, J.; Xu, S.; Cai, X.; Li, Z. The Field Wheat Count Based on the Efficientdet Algorithm. In Proceedings of the 2020 IEEE 3rd International Conference on Information Systems and Computer Aided Education (ICISCAE), Dalian, China, 27–29 September 2020; pp. 557–561. [\[CrossRef\]](#)
26. Liu, L.; Lu, H.; Li, Y.; Cao, Z. High-Throughput Rice Density Estimation from Transplantation to Tillering Stages Using Deep Networks. *Plant Phenomics* **2020**, *2020*, 1375957. [\[CrossRef\]](#)
27. Xiao, Z.; Liu, X.; Tan, Y.; Tian, F.; Yang, S.; Li, B. Recognition Method of No-seedling Grids of Trays based on Deep Convolutional Neural Network. In Proceedings of the 2019 Chinese Control Conference (CCC), Guangzhou, China, 27–30 July 2019; pp. 8695–8700. [\[CrossRef\]](#)
28. Zhang, G.; Wen, Y.; Tan, Y.; Yuan, T.; Zhang, J.; Chen, Y.; Zhu, S.; Duan, D.; Tian, J.; Zhang, Y.; et al. Identification of Cabbage Seedling Defects in a Fast Automatic Transplanter Based on the maxIOU Algorithm. *Agronomy* **2020**, *10*, 65. [\[CrossRef\]](#)
29. Li, H.; Li, Z.; Dong, W.; Cao, X.; Wen, Z.; Xiao, R.; Wei, Y.; Zeng, H.; Ma, X. An automatic approach for detecting seedlings per hill of machine-transplanted hybrid rice utilizing machine vision. *Comput. Electron. Agric.* **2021**, *185*, 106178. [\[CrossRef\]](#)
30. Lin, S.; Jiang, Y.; Chen, X.; Biswas, A.; Li, S.; Yuan, Z.; Wang, H.; Qi, L. Automatic Detection of Plant Rows for a Transplanter in Paddy Field Using Faster R-CNN. *IEEE Access* **2020**, *8*, 147231–147240. [\[CrossRef\]](#)
31. Bah, M.D.; Hafiane, A.; Canals, R. CRowNet: Deep Network for Crop Row Detection in UAV Images. *IEEE Access* **2020**, *8*, 5189–5200. [\[CrossRef\]](#)
32. Pang, Y.; Shi, Y.; Gao, S.; Jiang, F.; Veeranampalayam-Sivakumar, A.-N.; Thompson, L.; Luck, J.; Liu, C. Improved crop row detection with deep neural network for early-season maize stand count in UAV imagery. *Comput. Electron. Agric.* **2020**, *178*, 105766. [\[CrossRef\]](#)
33. Yu, Y.; Bao, Y.; Wang, J.; Chu, H.; Zhao, N.; He, Y.; Liu, Y. Crop Row Segmentation and Detection in Paddy Fields Based on Treble-Classification Otsu and Double-Dimensional Clustering Method. *Remote Sens.* **2021**, *13*, 901. [\[CrossRef\]](#)
34. Deng, X.; Liang, S.; Xu, Y.; Gong, K.; Zhong, Z.; Chen, X.; Chen, Y. Object Detection of Alternanthera Philoxeroides at Seedling Stage in Paddy Field Based on Faster R-CNN. *IEEE Adv. Inf. Technol. Electron. Autom. Control Conf.* **2021**, *5*, 1125–1129. [\[CrossRef\]](#)
35. Yarak, K.; Witayangkurn, A.; Kritiyutanont, K.; Arunplod, C.; Shibasaki, R. Oil Palm Tree Detection and Health Classification on High-Resolution Imagery Using Deep Learning. *Agriculture* **2021**, *11*, 183. [\[CrossRef\]](#)
36. Xu, Y.; He, R.; Gao, Z.; Li, C.; Zhai, Y.; Jiao, Y. Weed Density Detection Method Based on Absolute Feature Corner Points in Field. *Agronomy* **2020**, *10*, 113. [\[CrossRef\]](#)
37. Anami, B.S.; Malvade, N.N.; Palaiah, S. Deep learning approach for recognition and classification of yield affecting paddy crop stresses using field images. *Artif. Intell. Agric.* **2020**, *4*, 12–20. [\[CrossRef\]](#)
38. Ramanath, A.; Muthusrinivasan, S.; Xie, Y.; Shekhar, S.; Ramachandra, B. NDVI Versus CNN Features in Deep Learning for Land Cover Classification of Aerial Images. In Proceedings of the IGARSS 2019—2019 IEEE International Geoscience and Remote Sensing Symposium, Yokohama, Japan, 28 July–2 August 2019; pp. 6483–6486. [\[CrossRef\]](#)
39. An, G.; Xing, M.; He, B.; Liao, C.; Huang, X.; Shang, J.; Kang, H. Using Machine Learning for Estimating Rice Chlorophyll Content from In Situ Hyperspectral Data. *Remote Sens.* **2020**, *12*, 3104. [\[CrossRef\]](#)
40. Ammar, A.; Koubaa, A.; Benjdira, B. Deep-Learning-Based Automated Palm Tree Counting and Geolocation in Large Farms from Aerial Geotagged Images. *Agronomy* **2021**, *11*, 1458. [\[CrossRef\]](#)
41. 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.

42. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.-Y.Y.; Berg, A.C. SSD: Single Shot MultiBox Detector. In *Computer Vision—ECCV 2016, Proceedings of the 14th European Conference, Amsterdam, The Netherlands, 11–14 October 2016*; Leibe, B., Matas, J., Sebe, N., Welling, M., Eds.; Springer International Publishing: Cham, Switzerland, 2016; Volume 9905, pp. 21–37. [[CrossRef](#)]
43. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 1137–1149. [[CrossRef](#)] [[PubMed](#)]
44. Howard, A.G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; Adam, H. MobileNets: Efficient convolutional neural networks for mobile vision applications. *arXiv* **2017**, arXiv:1704.04861.