Article

Augmenting Crop Detection for Precision Agriculture with Deep Visual Transfer Learning—A Case Study of Bale Detection

Wei Zhao [1], William Yamada [1], Tianxin Li [2], Matthew Digman [1] and Troy Runge [1,*]

1   Department of Biological System Engineering, University of Wisconsin-Madison, Madison, WI 53706, USA; wzhao97@wisc.edu (W.Z.); wyamada@wisc.edu (W.Y.); digman@wisc.edu (M.D.)
2   Cockrell School of Engineering, University of Texas at Austin, Austin, TX 78712, USA; tianxinli@utexas.edu
*   Correspondence: trunge@wisc.edu; Tel.: +1-608-890-3143

Abstract: In recent years, precision agriculture has been researched to increase crop production with less inputs, as a promising means to meet the growing demand of agriculture products. Computer vision-based crop detection with unmanned aerial vehicle (UAV)-acquired images is a critical tool for precision agriculture. However, object detection using deep learning algorithms rely on a significant amount of manually prelabeled training datasets as ground truths. Field object detection, such as bales, is especially difficult because of (1) long-period image acquisitions under different illumination conditions and seasons; (2) limited existing prelabeled data; and (3) few pretrained models and research as references. This work increases the bale detection accuracy based on limited data collection and labeling, by building an innovative algorithms pipeline. First, an object detection model is trained using 243 images captured with good illimitation conditions in fall from the crop lands. In addition, domain adaptation (DA), a kind of transfer learning, is applied for synthesizing the training data under diverse environmental conditions with automatic labels. Finally, the object detection model is optimized with the synthesized datasets. The case study shows the proposed method improves the bale detecting performance, including the recall, mean average precision (mAP), and F measure (F1 score), from averages of 0.59, 0.7, and 0.7 (the object detection) to averages of 0.93, 0.94, and 0.89 (the object detection + DA), respectively. This approach could be easily scaled to many other crop field objects and will significantly contribute to precision agriculture.

Keywords: domain adaptation; computer vision; illimitation change; adverse weather conditions; generative adversarial network (GAN)

1. Introduction

According to the United Nations population estimates and projections, the growing world population will be nearly 10 billion in 2050 [1]. By 2050, there will be an increase in food demand of 59–98% [2]. To increase crop production while minimizing inputs, the adoption of advanced computing technologies, including computer vision, machine learning, and big data analytics, have recently gained interests among researchers in the fields of agriculture. Precision agriculture takes advantage of advanced computing technologies to minimize the inputs required, to improve the crop quality, and to increase yields.

With the reduction of equipment costs, increase in computing power, and availability of non-destructive food assessment methods, the efforts of many researchers and practitioners to improve the crop quality and yields have focused on computer vision and machining learning [3]. Computer vision helps with object detection and machine learning allows for useful information to be extracted from the collected data, showing tremendous advantages over the traditional methods applied in agriculture [4].

A number of research efforts has shown that the combination of computer vision and machine learning techniques regarding the multiple periods of crop production and harvesting are promising [5]. Computer vision in agriculture can be applied easily to

analyze digital images collected from the field and to provide high-level understandable information to the users [6]. For example, computer vision not only detects the weeds fast and effortlessly, but also accurately applies treatment with the help of a ground robot [7]. In addition, computer vision can detect the diseases on crops and inform users for them to take action [8,9].

Image Acquisitions: To collect the images as inputs for the computer vision, using an unmanned aerial vehicle (UAV) is an efficient approach, which has been widely used in precision agriculture as well as many other fields, such as path planning, design, and wildlife rescuing [10,11]. A UAV combined with computer vision can also contribute to remote sensing to help inform farmers about the geo-specific crop yield and identify crop diseases [12,13]. Sometimes, decisions are required to be made off-board once the data have been collected and processed by the UAV, based on the information provided by the images processed from the computer vision technique [14,15]. For example, UAVs can be used to detect a potential issue, and then obtain high-resolution images or inspect and apply treatments correspondingly.

Bale detection challenges: When it comes to object detection, associated methods are commonly sensitive to the illumination and object and background domain change. A non-robust model can easily fail if it was not taking into account the variation in light conditions [16,17]. Because of the diversity of illumination situations, seasons, and weather conditions, object detection in the outdoor environment is more complicated than in the indoor environment, since humans can manipulate a consistent environment, as is shown in Table 1.
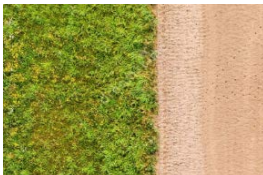
To emphasize, the illumination and hue change are the most significant factors impacting the bale detection model performance. Illumination variation, including the change in light conditions, with/without shadow covering, plays a significant role in object detection in the context of outdoor practices. Patrício and Rieder [18] suggested that consistent light conditions between the source domain and target domain will decrease the difficulties of shaping accurate classification models built on the deep learning architecture. A similar conclusion has been drawn by Hornberg [19], in that adequate lighting in the environment can increase the reliability of the performance of the models based on the collected images.

Hue change due to the season transition and variation in light conditions is another key factor to be considered in precision agriculture. During the growing season of vineyards, when the light is not strong enough, Baweja et al. [20] added extra light when collecting images by using a strobe lighting mounted on a ground robot image-capturing machine to compensate for the hue variation, to build a reliable deep learning model.

Since the deep learning-based object detection model always needs a large number of images labeled as the ground truths before training a supervised object detection model, the accuracy and thus detecting performance is impacted by the quality of the labeled data. One approach to improve the quality of the labeled data is to include balanced data, by including various images from the target domains, listed in Table 1. However, if we want to guarantee the quality, i.e., for each condition, the total number of images required to be manually labeled could be large and take significant resources to complete.

To reduce the task of labeling the objects manually, style transferring methods have been developed. To minimize the discrepancy between the source domain and target domain regarding the domain distribution, we propose a model by combining the convolutional neural network (CNN)-based YOLOv3 model and domain adaption (DA), a representative method in transfer learning. Domain adaption works very well where the tasks are similar, except for the domain distribution between the source domain and target domain [21]. In methodology section, we illustrate the proposed biomass detection model on the basis of CNN and DA. Since it has the strengths of accuracy and speed for object detection, YOLOv3 was selected to build the CNN model [22]. To realize the DA, an unpaired translation method, cycle generative adversarial networks (CycleGAN), was used to tackle the image difference due to the illumination, hue, and clarity discrepancy.

Table 1. Object detection challenges in complex unconstrained outdoor environments.

Environment Diversity	Conditions	Example	Agriculture Information	Technical Challenges for Bale Detection
Illumination Diversity (Target Domain 1)	Lighting condition		To gain the efficiency of agriculture, different process routines are conducted to crops in the morning, afternoon, and night.	Decreasing the difficulties of shaping an accurate classification models built on the deep learning architecture.
	Shadow		Shadow is commonly seen during daytime. This always happens in the rainy season. The images taken by UAV includes shadows in certain months.	Shadows crossing the objects decrease the accuracy of the classification on these kinds of objects. The scale of the background and bale size also makes it worse.
Seasonal Change (Target Domain 2)	Hue change		Farms with different plants have various harvest seasons. As a result, the bales and backgrounds vary in different season.	The inconsistent changes in background and bale with season trigger the decrease of bale detection performance.
Adverse Weather Conditions (Target Domain 3)	Haze		Haze weather sometimes happen with a temperature drop or precipitation change. This may cause a grain lifecycle adjustment, which needs to be monitored.	The high performance of supervised learning and semi-supervised learning (object detection) in haze weather is always a challenge.
	Snow covered		Tracking bales in winter and in a snow environment is also important for continuously feeding livestock.	Restoration-based algorithms may mislead or overfit the object compared to the original one. The snow weather reduces the features of the objects in the images.

The present study sought to test the proposed method by collecting data from the field by a UAV equipped with RGB cameras, including 243 images captured with good illumination conditions in the fall and 150 images in other conditions, with the baled biomass also collected. Manually labeling each baled biomass from these conditional images is essential to train the YOLOv3 model. In addition, we also needed to manually label the images collected from other conditions to test the accuracy of the prediction given by the model and provide validation of the method.

In addition to use our proposed model, we also apply the traditional background subtraction algorithm developed by Li et al. [23], using the same data. The results show that our method gained the best F scores, indicating that it performs well when dealing with the discrepancy of domain distribution due to the different outdoor environments. Part of the images was manually labeled, while the rest of the images with different illumination contexts share the same labels by implementing CycleGAN for domain transferring. The processed images were used as inputs for the proposed YOLOv3 model to perform bale

detection. The goal was to show that our proposed model, a combination of computer vision and domain adaption, could improve the accuracy and efficiency of bale detection.

The key contributions of this work are listed in the following three points:

i.   For bale detection under illumination conditions, a YOLOv3 model was built. The associate training dataset will be released under conditions with the current work to fill the voids in the bale training dataset, with labels as the ground truths.

ii.   We constructed an innovative object detection approach (algorithms pipeline), including YOLOv3 and domain adaptation (DA). Additionally, this approach improves the capability of bale detection.

iii.   We augmented the labeled training data with more scenarios using domain adaptation. Combined with our manually labeled data, we are able to provide a valuable training dataset of over 1000 bale images, which is publicly available after this publication.

2. Related Work

2.1. Computer Vision in Precision Agriculture

A number of research studies have investigated the application of computer vision in different key steps in agriculture, including observing crop growing, detecting diseases, and facilitating crop harvest [24].

Crop Growth Monitoring: Computer vision techniques have been used to collect the nutritional status of plants. Romualdo et al. [25] conducted research on maize plants to realize the diagnosis of the nitrogen nutritional status by implementing the computer vision technique at different development stages. Compared to the traditional method that relies on human observations, the computer vision technique improves the detection efficiency and accuracy. Pérez-Zavala et al. [26] proposed a computer vision approach to detect the grape bunches in vineyard scenes relying on the shape, texture descriptors, and bunch separation strategy to realize automatic monitoring of grapevine growth. Chandel et al. [27] applied deep learning models to monitor the water condition of crops and identified the water stress with over 90% accuracy. Parra et al. [28] compared various edge detection filters for weed recognition in lawns and identified that the sharping filters provided the best results with low computing requirements.

Disease Detection: Computer vision techniques also help with disease detection in agriculture. Oberti et al. [29] implemented computer vision to detect powdery mildew on grapevine leaves and the accuracy has been improved significantly by adjusting the view angles from 40 to 60 degrees, hence improving the overall quality of the plants. Pourreza et al. [30] explored the application of a computer vision technique to detect Huanglongbing disease on trees infected by a citrus psyllid. To analyze the performance of our model, laboratory and field experiments were taken and the results showed that the new method improve the target disease detection accuracy from 95.5% to 98.5%. Instead of identifying a single disease, the computer vision technique also contributes to the classification of multiple diseases of crops. Maharlooei et al. [31] applied image processing technology on detecting and counting soybean aphids to achieve the identification and enumeration of mites with lower costs and a high accuracy in strong light conditions. Toseef and Khan [32] used a fuzzy inference system to generate an intelligent mobile application to help rural farmers diagnose diseases that commonly occur on wheat and cotton crops with a 99% accuracy, reducing the loss of farmers due to crop diseases and dramatically improving the crop yields. Rustia et al. [33] applied an image and environmental sensor network to automatically detect greenhouse insect pests and achieved a 93% average temporal accuracy in terms of counting insect pests.

Crop harvest: Crop harvest is another aspect that benefits from the computer vision techniques. Barnea et al. [34] developed crop harvesting robots by using a color-agnostic shape-based 3D fruit detection technique on a registered image and depth to address the localization issue in precision agriculture, due to shape variations and occlusions. Lehnert et al. [35] designed an approach based on effective vision algorithms for harvesting

sweet pepper and protecting the cropping system, demonstrated to be successful by the experiments of harvesting sweet peppers from modified and unmodified crops.

Dealing with the biomass after crop harvesting is essential. Biomass collection can provide economic benefits and, in certain cases, may also benefit future crops [36]. Biomass from crop fields are usually baled to a compact form before collection and transportation. In addition, stacking the bales to utilize the efficient bale-hauling equipment is desired. Other benefits of putting bales into stacks include efficiently clearing the crop field for the next grow cycle; avoiding bales, as they can be a hindrance that adversely affect the mechanical crop management operations; and shortening the time costs between harvest and planting.

2.2. Transfer Learning and Domain Adaptation

Transfer learning is a popular machine learning technique that aims to help with repetitive tasks by using the existing developed model. When it comes to situations where labeled data are only available in a source domain, domain adaption (DA), a common technique in transfer learning, as shown in Figure 1, can be applied. A little distribution change or domain shift, due to illumination, pose, and image quality, between the source and target domains can lead to a degraded performance of the machine learning models. Domain adaption (DA) provides an opportunity to mimic the human vision system that allows to perform new tasks in a target domain by using the labeled data from more relevant source domains. A number of research studies have recently addressed the issue of domain shift.
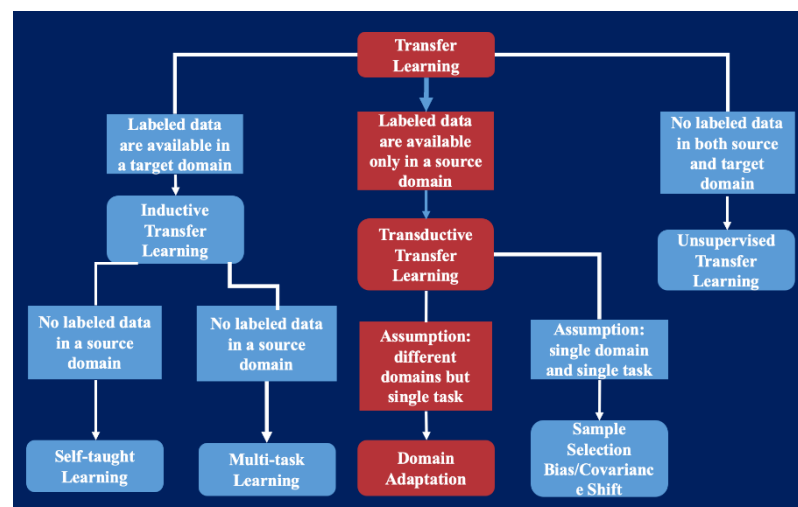

Figure 1. Relationship between transfer learning and domain adaptation.

To implement CNN techniques, a large images dataset with manually labeled targets is required, which is expensive and challenging [37]. By synthesizing images through use of the DA techniques, one can reduce the images needed to be collected from the field and solve the problem when the labeled data cannot be acquired from the target domain [38]. Various research has been conducted and has achieved promising results. Ganin et al. [39] used unlabeled images from the target domains based on labeled images from the source domains for a deep learning architecture, based on a few standard layers and an additional gradient reversal layer. Othman et al. [40] designed a domain adaption network to overcome the issues of a domain shift in classification scenarios where the labeled images from the source domain and unlabeled ones from the target have completely different geographical features. Overall, when it comes to the problems of domain shift between the source and target domains, the DA technique can not only reduce the costs of data preparation, but also improve image recognition [41,42].

3. Methodology

Bale detection method pipeline summary: Figure 2 shows the completed structure of the bale detection method, from image acquisition to creating the model, and then to augment the model proposed in this work. We divided this pipeline into three steps, as follows: Step 1 trains a primary object detection model with YOLOv3, only based on the manually labeled initial condition images. Step 2 demonstrates the method how we use the manually labeled ground truth images to generate more ground truth images with automatic labels. Then, in Step 3, we augment the object detection model in Step 1 with the mixed labeled ground truth images as the training data.
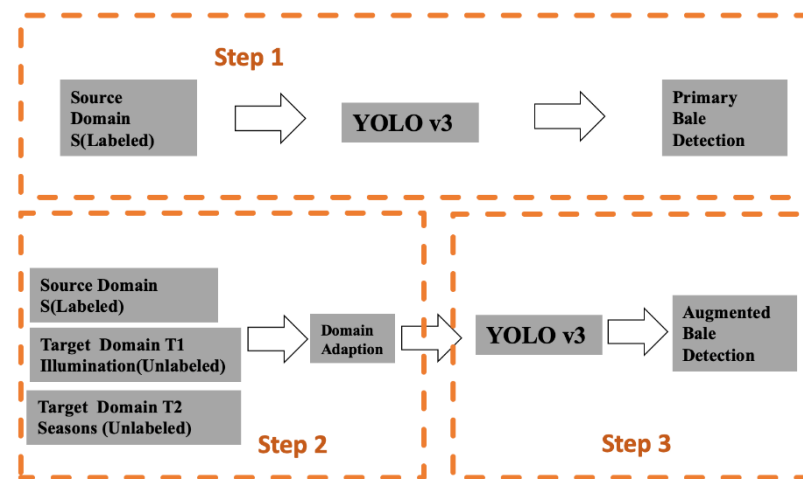

Figure 2. Framework summary of the proposed bale detection method pipeline.

Step 1: Primary object detection

A YOLOv3 model was trained for primary bale detection using 243 images captured with good illumination conditions in the fall. We define these labeled images as the source domain. CNN-based object detection methods, such as Faster R-CNN, YOLO, and Mask R-CNN, gained popularity among researchers and have been proved to be efficient [43–45]. YOLOv3 was released by Redmon and Farhadi in 2018, extended from the previous YOLO versions [46]. In this paper, YOLOv3 is implemented in the baled detection process, taking advantages of its accuracy and fast speed on object detection. Instead of using multiple networks for analysis, YOLOv3, indicated by its name You Only Look Once, passes the input image once to a convolutional neural network, lowering the costs and improving the performance significantly. In addition, the network splits the input into multiple regions and works on each one with the bounding boxes and their classification probabilities. By focusing on the global context of the image, YOLOv3 decreases the possibility of making a location classification error.

To implement YOLOv3, we used PyTorch to train the model and to make inferences, based on Darknet-53 (an architecture that consists of 53 convolutional neural networks). The initial weights between the layers were provided by the Darknet-53 backbone [46]. YOLOv3 relies on a deeper architecture to extract features; the backbone here is "Darknet-53" with 53 convolutional layers. Leaky ReLU activation as well as normalization were added to every layer. Instead of using any form of pooling, often contributing to a loss of low-level features, we applied a stride of 2 in convolutional layers to reduce the size of the samples of the feature maps. Stride refers to the factor between the applications of the filter to the input image. An image of size 416×416, for instance, can be down-sampled to 13×13 by a stride of 32. The shape of the input images is (m, 416, 416, 3). The output consists of bounding boxes, representing the recognized classes. Each bounding box is defined by 6 numbers $(p_c, b_x, b_y, b_h, b_w, c)$. With augmenting cc (class) to an 80 dimensions vector, 85 numbers are used to describe every single bounding box, as shown in Figure 3.
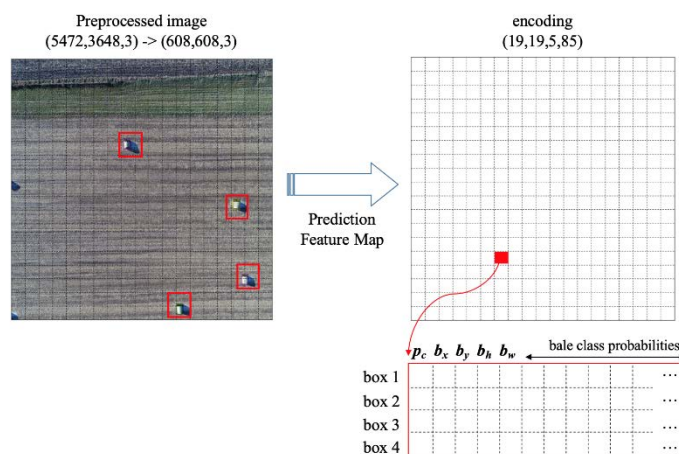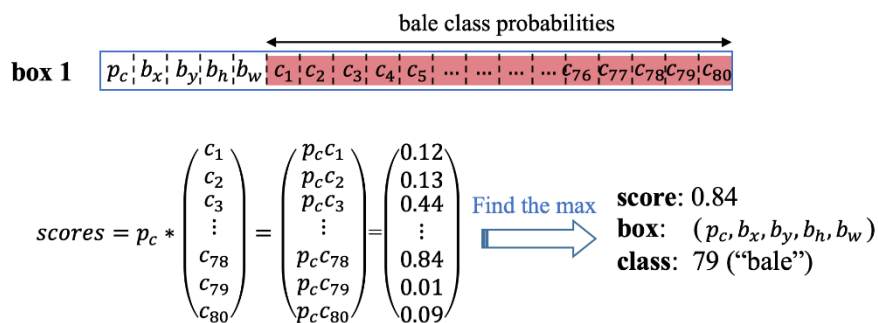
Figure 3. Explanation of encoding in the YOLOv3 architecture.

Similar to the object detectors, features learned by the convolutional layers are filtered to predict the detection, such as the coordinates of the bounding boxes and the class label. YOLO v3 is based on a 1-to-1 convolution to predict, so the prediction map has the same size as the input. Each cell in the prediction map represents a fixed number of bounding boxes, as shown in Figure 4.

$$scores = p_c * \begin{pmatrix} c_1 \\ c_2 \\ c_3 \\ \vdots \\ c_{78} \\ c_{79} \\ c_{80} \end{pmatrix} = \begin{pmatrix} p_c c_1 \\ p_c c_2 \\ p_c c_3 \\ \vdots \\ p_c c_{78} \\ p_c c_{79} \\ p_c c_{80} \end{pmatrix} = \begin{pmatrix} 0.12 \\ 0.13 \\ 0.44 \\ \vdots \\ 0.84 \\ 0.01 \\ 0.09 \end{pmatrix}$$

Find the max

score: 0.84
box: $(p_c, b_x, b_y, b_h, b_w)$
class: 79 ("bale")

the box (b_x, b_y, b_h, b_w) has detected c = 79("bale") with probabilities score:0.84

Figure 4. Example of generating the probability of a certain class in each bounding box.

Step 2: Augmenting the training data with domain adaptation

Domain adaptation, as a kind of transfer learning, is designed for augmenting the training data scenarios with automatic labels. As shown in the lower left in Figure 2, more than two conditional images are listed as Target Domain 1, 2, etc. Traditionally, all the targeting objects in the images need to be manually labeled. However, our proposed method, combining YOLOv3 with DA, decreases the laborious manual identification work but also ensure the performance of the model by applying style transferring. This method is created by referring to other state-of-the-art research that uses a similar structured approach; e.g., Song et al. [47] proposed an advanced subspace alignment algorithm combining convolutional neural networks, in order to classify remote sensing images, with domain adaptation on a theoretical level. Another fundamental work [48] proposes an approach with a pipeline of algorithms, including FasterR-CNN, DA, and H-divergence theory. While this research validates their application in cityscapes and other public datasets, it is without a practical-use scenario. Khodabandeh et al. [49] inserts noise during pre-processing of the training datasets and DA, which makes the object detection model be resilient to random noise. However, since the idea of object detection with DA is still under development, most related researches are based on a few public datasets with limited practical implementation scenarios, especially in the agriculture domain. There is no similar approach or concept related to bale detection. The following approach breaks the ice for augmenting the bale detection ability, taking advantage of state-of-the-art algorithms.

We only labeled the inputs from the images with one condition and then we collected more images with diverse illuminations, hues, and styles under different environments. Then we built a domain transferring model to convert the images of the initial condition to new images of the other conditions. Instead of manually labeling all the inputs required by the model, only part of the images was manually processed and the rest of the inputs shared the same label automatically because of the style transfer. In this way, a more robust YOLOv3 model that performs accurately on augmented styles of images could be achieved.

The DA technique is applied to shape the translation mapping from the source domain (*S*) in the initial environment to the target domain *T* in the other environments, and vice versa, as shown in Figure 5. The images from two different domains were not related in any way. CycleGAN [50] was implemented to transfer the styles between the two domains to synthesize the target domain images from the source domain (*S*).
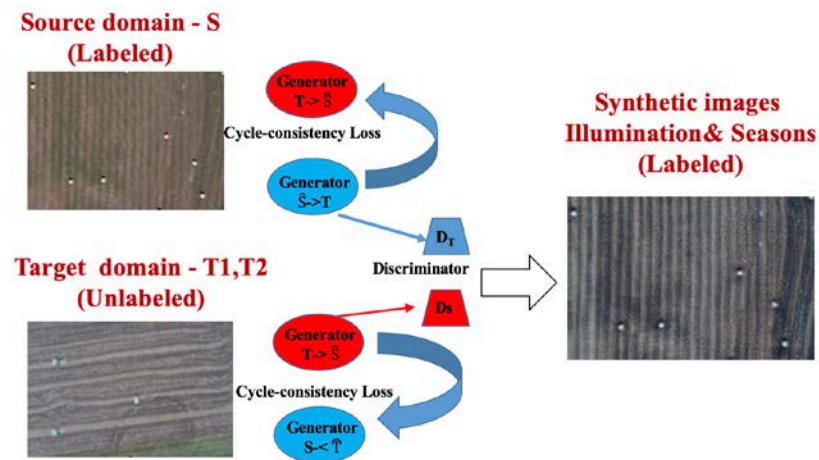

Figure 5. Framework of the domain adaption algorithm in Step 2.

Two GANs were used for applying the CycleGAN in the style transfer. Each one includes one generator and one adversarial discriminator. The generator, $Gen_{(S,T)}$, in the first GAN translates images from the source domain (*S*) to the target domain (*T*), while the adversarial discriminator D_T outputs the likelihood that the images taken from the target domain (*T*) are real images. Similarly, the generator $Gen_{(S,T)}$, in the other GAN, translates images from the target domain (*T*) to the source domain (*S*), and its adversarial discriminator D_S outputs the likelihood that the images taken from the source-domain (*S*) is real images from the source domain (*S*). I_S and I_t represents images from domain (*S*) and (*T*), respectively. Given $i_S \in I_S$ and $i_t \in I_t$, it represents images in domain (*S*) and (*T*), respectively.

$\hat{T}$ represents the domains of the images synthesized in Figure 2. It represents the domain of the diverse seasons and illuminations of the synthetic images generated from the real initial environment images, while $\hat{S}$ denotes the initial synthetic images generated from the real other environment images. By applying $Gen_{(S,T)}$, images $i_S \in I_S$ is transferred to synthetic images in $\hat{T}$, while the corresponding adversarial discriminator improves the model by encouraging the translated image hardly distinguishable from the domain (*T*). Ideally, when the translated image from the source domain (*S*) to the target domain (*T*) is translated back from the target domain (*T*) to the source domain (*S*), we should get identical images. However, learning models are not perfect, and two different images will be obtained. The difference between the two images is measured by the cycle consistency loss, as defined below:

$$
\begin{aligned}
Ls_{Cycle}\Big(Gen_{(S,T)}, Gen_{(T,S)}, D_S, D_T, S, T\Big) &= \lambda Ls_{Cycle}\Big(Gen_{(S,T)}, Gen_{(T,S)}, S, T\Big) + \\
&\quad Ls_{GAN}\Big(Gen_{(T,S)}, D_S, T, S\Big) + \\
&\quad Ls_{GAN}\Big(Gen_{(S,T)}, D_T, S, T\Big)
\end{aligned}
\tag{1}
$$

In Equation (1), λ is the balance weight, Ls_{Cycle} measures the cycle consistency loss, and Ls_{GAN} represents the loss function of the adversarial training. The cycle consistency loss used in the GAN training penalize Ls_1 in the cycle architecture, defined as

$$Ls_{Cycle}\left(Gen_{(S,T)}, Gen_{(T,S)}, S, T\right) \quad = E_{i_T \sim I_T}\left[\left\|Gen_{(T,S)}\left(Gen_{(S,T)}(i_S)\right) - i_S\right\|_1\right] + \\ E_{i_T \sim I_T}\left[\left\|Gen_{(S,T)}\left(Gen_{(T,S)}(i_T)\right) - i_T\right\|_1\right] \tag{2}$$

Equation (3) defines the loss in adversarial training:

$$Ls_{Cycle}\left(Gen_{(S,T)}, Gen_{(T,S)}, S, T\right) = E_{i_T \sim I_T}[\log(D_T(i_T))] + E_{i_T \sim I_T}\left[\log\left(1 - Gen_{(S,T)}(i_S)\right)\right] \tag{3}$$

To train these generators and discriminators, we need to solve

$$\begin{matrix} Gen_{(S,T)}^* \\ Gen_{(T,S)}^* \end{matrix} = \mathrm{argmin}_{Gen_{(S,T)}, Gen_{(T,S)}} \max_{D_S, D_T} Ls_{CycleGAN}\left(Gen_{(S,T)}, Gen_{(T,S)}, D_S, D_T, S, T\right) \tag{4}$$

Gradient descent is first applied to Equation (4), followed by backpropagation to allow the generator $Gen_{(S,T)}$ to complete the style transfer between the real initial-style images and synthetic other-style images, without changing the spatial relationship between the biomass in the images.

Step 3: Optimize the YOLOv3 model with the extended datasets from Step 2.

There are two optional methods we can apply to optimize the performance of the model. One is retraining the model, and the other one is fine-tuning. Retraining a model using extended data with a proper preprocessing is a straightforward and robust way, however it takes longer than fine-tuning.

A commonly used manner to transfer the trained model to the new dataset is fine-tuning, which is more efficient when the size of the new dataset is small. Fine-tuning trained models can not only reduce the probability of overfitting, but also provides better generalization if the original dataset and new dataset share similar domains. In this research, we applied both methods, keeping the better results of the two.

4. Experiment Design and Data Association

4.1. Experiment Equipment

The input data, the baled housing biomass, were collected from the fields by a drone from the Arlington Research Station (Arlington, WI, USA). The drone, equipped with a 1-inch Exmor R CMOS sensor and a gimbal stabilizer that handles the lateral and vertical vibration, allowed us to collect images from different heights, as shown in Figure 5. Through each campaign, the locations of the baled biomass were identified by a Global Navigation Satellite System (GNSS) and their corresponding centers were surveyed by a Carlson Surveyor 2. These two additional systems are for validation of the location accuracy and as a contribution to the public database for future research.

4.2. Bales Data Collection and Description

All the images collected of the bale biomass in the fields were taken with one drone model. Images from two different heights, 200 ft and 400 ft, were captured through seven campaigns to provide different resolutions to test our model performance. The size of the collected images was 5472×3648 pixels, corresponding to a 20-megapixel resolution, as shown in Figure 6. In addition, we created a second dataset by rescaling the collected images to 1080×720 with a 3:2 ratio, simulating a camera with a less than 1-megapixel resolution. The image numbers specifications used in the experiments are shown in Table 2, as "Initial condition". There were a total of 300 images used for the training, validation, and testing. All these images were collected in the fall under good illumination conditions, without shadows. We also collected 128 real images under the other conditions as ground truths for both training the CycleGAN model and testing the performance. We used independent training, validation, and testing datasets when we conducted this research.

We tested the various cases without using their original images anywhere in the training or validation. More images under the other conditions were generated by the CycleGAN model.

Figure 6. Zenmuse-X4S camera-equipped DJI-Inspire-2 unmanned aerial vehicle (UAV).

Table 2. Experimental data collection distributed under different environmental conditions (note: there are images cross counted in different conditions).

Environment	Condition	Training	Validation	Testing
Initial condition	Good illumination, fall, w/o shadow	243	27	30
Diverse illumination (Target Domain 1)	w/Lighting condition change	160	20	20
	w/Shadow	158	20	20
Seasonal change (Target Domain 2)	Hue change (summer)	185	19	19
	Hue change (early winter)	187	12	12
Adverse weather conditions (Target Domain 3)	w/Haze	159	20	20
	w/Snow covered	150	19	19

Figure 7 provides an example of the images used in the model. The baled biomass and streets in all the collected images were annotated in both the MS COCO and YOLO data formats by using the Computer Vision Annotation Tools (CVATS) LabelImg and LabelMe [51,52].
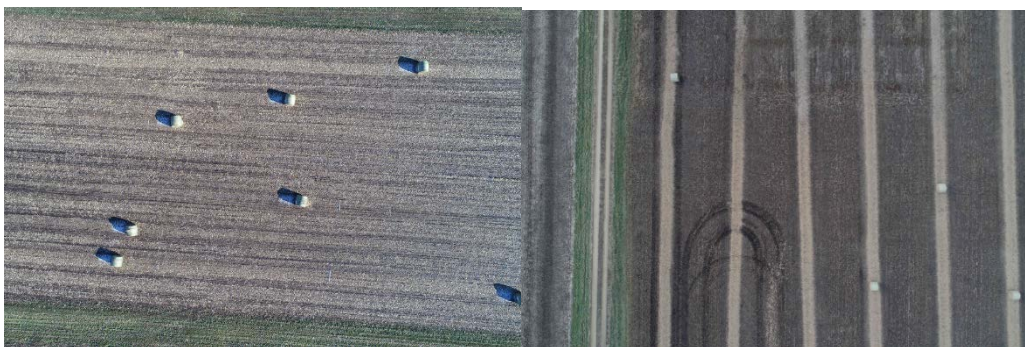

Figure 7. Example of the drone-collected images under good lighting conditions in the fall (the initial condition).

5. Result and Discussion

5.1. Primary Bale Detection with YOLOv3 Corresponding to Step 1

The YOLOv3 detector trained with only initial condition images in Step 1 was applied to detect bales in the real images. Although the training process does not include images under extended conditions, we still include these images in the testing results for comparison with the optimized detection model. The testing results, in terms of precision, recall, mAP, and F1 score for each scenario, are presented in Table 3. The high value of precision indicates a low incidence of false positives, meaning that the algorithm did not detect a

bale where there was not any. On the other hand, the low recall means the algorithm fails to see some of the bales inside the image.

The prediction performance on the initial condition images achieved high values (0.92) of precision, recall, mAP, and F1 score. However, these four indices vary in the negative way for the extended conditions. The precision values for all conditions, except shadow, are over 0.85. As is shown in Figure 8b, the bales usually failed to be detected inside or partially covered by the shadow. The other three indices (recall, mAP, and F1 score) are all lower than expected for the extended conditions (average values are less than 0.59, 0.7, and 0.7 respectively). The F1 score is the harmonic average between precision and recall. Since the last one was low, the F1 score also got low. The mean average precision (mAP) was low and it varies through the different simulated scenarios. However, mAP was high in the haze condition since all the images with haze condition are collected with minor haze or fog, which may cause a significant blur on the background instead of bales. The general results are as expected, since there are few image samples in the extended conditions in the training datasets.

Table 3. YOLOv3 model (trained in Step 1) performance for detecting bales in different conditions without being trained with synthetic images.

Data	Images [1]	Precision	Recall	mAP@0.5	F1
All conditions	148	0.859	0.599	0.780	0.746
Initial condition	30	0.929	0.993	0.987	0.960
Illumination	20	0.881	0.587	0.848	0.735
Shadow	20	0.675	0.456	0.622	0.621
Hue change (summer)	19	0. 917	0.644	0.853	0.783
Hue change (early winter)	19	0.929	0.605	0.751	0.852
Haze	20	0.910	0.871	0.975	0.931
Snow	19	0.874	0.456	0.584	0.621

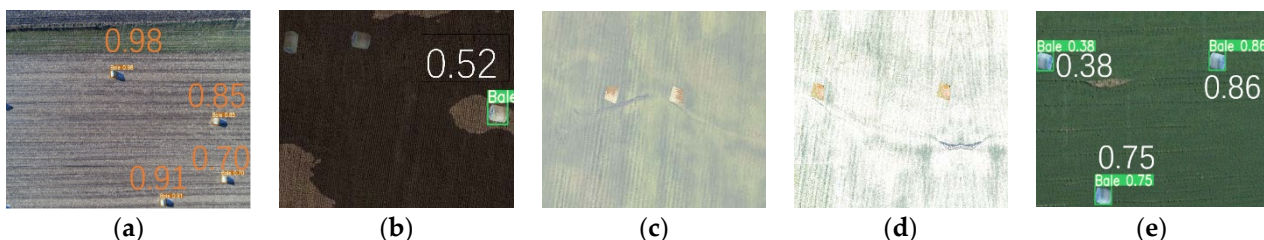[1] We used real images for testing.

| (a) | (b) | (c) | (d) | (e) |

Figure 8. Example of tested bale images under multiple environmental conditions using the primary bale detection model (trained in Step 1): (**a**) initial condition with good illumination and no shadow in fall; (**b**) extended condition—early winter with shadow; (**c**) summer with haze; (**d**) winter with snow cover; (**e**) summer with good illumination conditions.

Some examples of the tested bale images under multiple environmental conditions using the model trained in Step 1 are shown in Figure 8. These typical results under different conditions display the same trend with Table 3. There are some undetected bales and low confidence scores listed in Figure 8.

5.2. Augmenting the Training Data with CycleGAN Corresponding to Step 2

During Step 2, we built a CycleGAN model to convert the real images to synthetic/fake images, as shown in Figure 9a. In the figure, real_A and real_B are real images, fake_B is the synthetic/fake image from real_A, and rec_A stands for the reconstructed image A based on fake_B. The second row has the same idea as the first one. With this CycleGAN model, 1200 synthetic images were generated and will be used as the extended training dataset in Step 3.
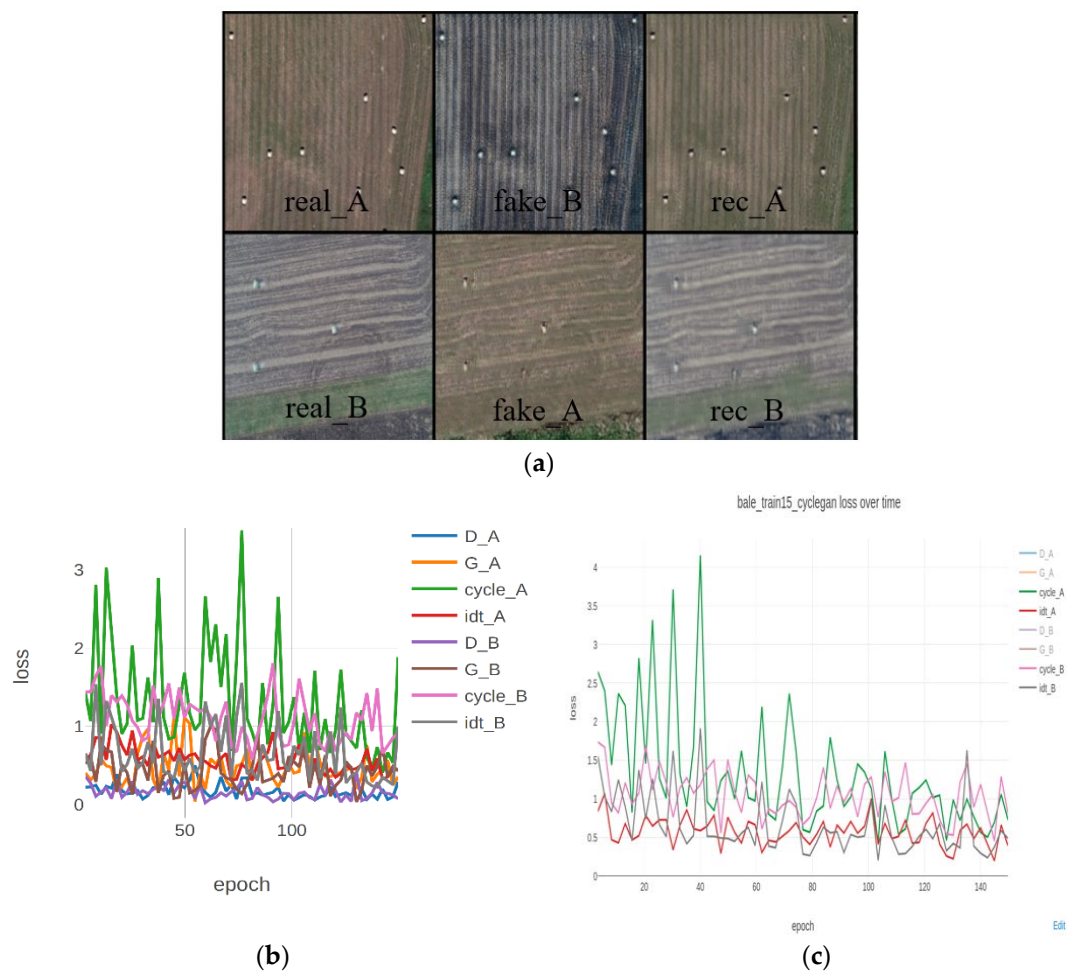
Figure 9. Description of the CycleGAN model: (**a**) examples of the real images, fake images, and reconstructed images; (**b**,**c**) loss tracking during training of the CycleGAN model, during repeated training.

Identity loss is the index when measuring the discrepancy due to translating one style of image to another style image, regulating the generator to generate images with high fidelity translated from the real samples in the target domain. No extra change is needed for the images that are almost distinguishable from the target domain. Generally, a greater identity loss value will be applied for unknown content. Figure 9b,c are the loss values during different and repeated training processes. These two tables show a slight reduction in some losses, especially cycle_A in the green color, as expected. So, the six loss values and four loss values are all partial references for the training status, but not the decision of a successful training process. Based on the purpose of these loss values, we do not expect a growing loss. A stable trend of loss with a flat plot or decreasing plot is expected. More information about the model parameters and logic can be found in Zhu et al. [45]. Figure 10 shows some examples of the augmented bale images with multiple environmental conditions.
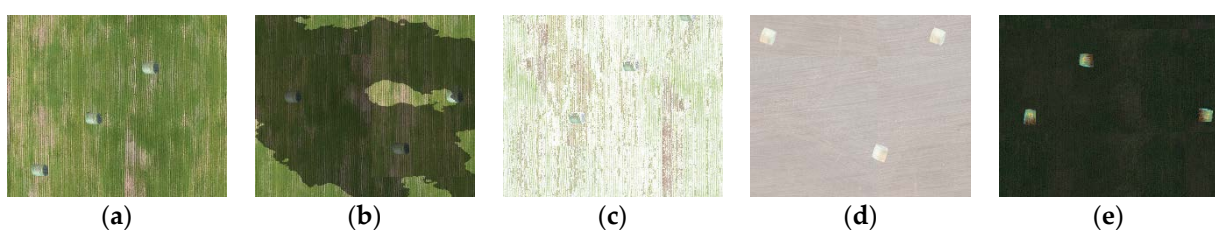

Figure 10. Examples of augmented bale images with multiple environmental conditions: (**a**) summer w/good illumination; (**b**) summer w/shadow; (**c**) winter w/snow; (**d**) early winter w/haze; (**e**) summer w/dark illumination.

5.3. Optimized YOLOv3 Model with Extended Datasets Corresponding to Step 3

The optimized YOLOv3 detector, trained with both the real images and synthetic images in Step 3, was applied to the same testing datasets. Table 4 shows the testing results, which will be compared with the performance of the primary YOLOv3 model in Step 1. YOLOv3 in Steps 1 and 3 have a similar performance for bale image detection under the initial condition, as shown in the line "Initial condition" in Tables 3 and 4. The generic testing results, in terms of precision, recall, mAP@0.5, and F1 score, for each scenario, are presented in Table 4. In most cases, the recall, mAP, and F1 score are obviously improved from average (0.59, 0.7, and 0.7) to average (0.93, 0.94, and 0.89), respectively. All the significantly increased values are marked in green. The increment in the recall indicates that most of the bales that cannot be detected in Step 1 are detected in Step 3. Meanwhile, the precision value keeps a similar level with occasional reduction because of the occasional increased false positives and true positives. This result is strong evidence that using synthetic images from transfer learning is a reasonable approach to enhance the detection capability with images under new conditions.

Table 4. YOLOv3 model (trained in Step 3) performance for detecting bales in different conditions after being trained with the synthetic images from Step 2.

Data [1]	Images [2]	Precision	Recall	mAP@0.5	F1
All conditions	148	0.869	0.927	0.941	0.892
Initial condition	30	0.913	0.980	0.990	0.945
Illumination	20	0.847	0.926	0.959	0.885
Shadow	20	0.847	0.933	0.854	0.888
Hue change (summer)	19	0.836	0.933	0.954	0.882
Hue change (early winter)	19	0.905	0.893	0.969	0.831
Haze	20	0.831	0.867	0.895	0.848
Snow	19	0.926	0.878	0.941	0.901

[1] All the significantly increased values are marked in green; [2] We used real images for testing.

The same examples of the tested bale images under multiple environmental conditions using the model trained in Step 3 are shown in Figure 11. These typical results under different conditions show the improvement compared to Figure 8.

Figure 11. Examples of the tested bale images under multiple environmental conditions using our framework (YOLOv3 + DA): (**a**) initial condition with good illumination and no shadow in fall; (**b**) extended condition—early winter with shadow; (**c**) summer with haze; (**d**) winter with snow cover; (**e**) summer with good illumination conditions.

5.4. Comparison and Advantages

To better understand the detection improvement on images under different environmental conditions, we plotted the F1 value between Step 1 and Step 3 under each condition separately, as shown in Figure 12. It was discovered, clearly, that under most conditions, the performance increases a lot, except for the initial condition, hue change (early winter), and haze; this is because the optimization curve generally slows down after the accuracy is over 80% when improving the object detection performance. What we aim for is improving the detection accuracy of the conditions with a lower accuracy (less than 80%). So, in our case, we expect to see a big jump for conditions such as illumination, shadow, hue

change (summer), and snow, which all have a less than 75% accuracy. The following results analysis shows that we not only kept the original high performance but also increased the performance of some conditions that originally had a low accuracy.
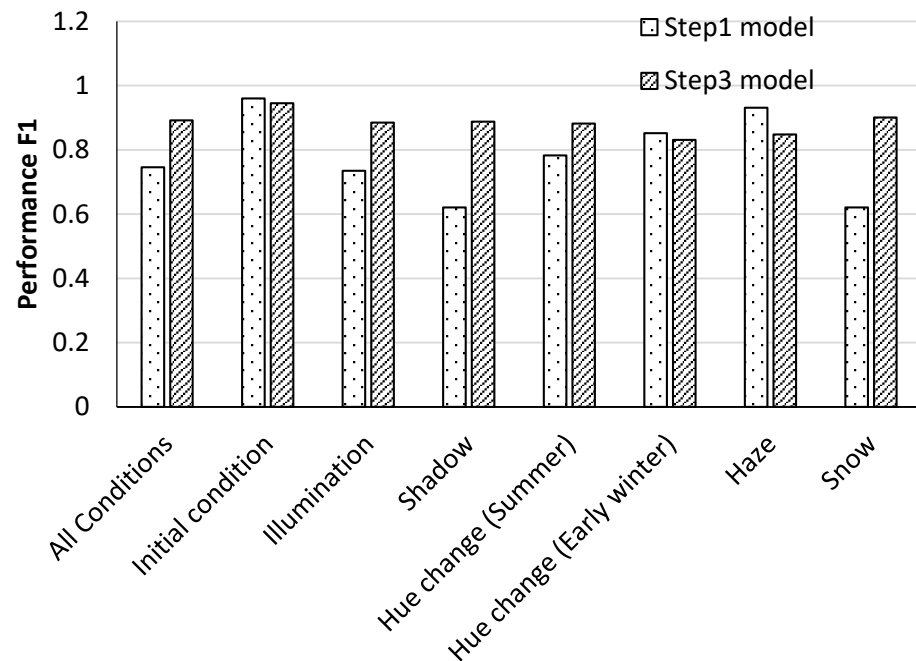

Figure 12. F1 score comparison between Step 1 (primary YOLOv3 model) and Step 3 (optimized YOLOv3 + DA model) under each condition and mixed conditions.

Firstly, the initial condition already has a high accuracy of over 93% with either model. Similarly, the hue change (early winter) condition also keeps a relevantly high performance around 90% before and after our approach. This method maintains the high accuracy score with a slight change during the enhancement of the training dataset volume and the false-negative samples. Meanwhile, the haze images enlarge the base number when calculating the F1 score. Although the haze condition accuracy is high, we can still make improvements by collecting more better-quality images season by season. However, this would be a long period and would entail continuous collecting work for our lab, which is not the core contribution of this algorithm research. Secondly, for conditions like, illumination, shadow, hue change (summer), and snow, our method significantly ameliorates the detection accuracy by around 15%, 26%, 10%, and 28%, respectively. Since it includes more images to train the model, the performance of the detection accuracy in the initial condition is slightly compromised, while its performance for other environmental conditions were improved significantly, in that the minimal F1 measure is at least above 80%. Adding more images from various conditions means more diverse bale types are considered. This may increase the false positives and false negatives, resulting in the slightly degraded measure of precision, recall, and F1. This phenomenon is common in the object-detection, deep-learning practice. Generally, this YOLOv3 + DA model proves its advantages in augmenting the detection ability with at least 80% accuracy for all conditions.

Moreover, we estimated the time cost of manually labeling bales in all images, as shown in Table 5. Step 1 only needs image labeling under the initial condition, for around 90 h. After that, we have two options to augment the bale detection model. One is to label every new image under all extended conditions with 260 extra hours of work, the other one is to train a CycleGAN model without extra labeling other than the first 90 h. Since the general F1 score, precision, recall, and mAP from the proposed approach are all over 0.9, this is sufficient for this specific task. Thus, the proposed method provides additional advantages of time and labor saving.

Table 5. Manual labeling cost for bale detection with different approaches.

Train Approach	Time Cost (Hours)
w/Initial condition images	90
w/Domain adaption images	90
w/Labeled all conditions images [1]	350

[1] "Labeled all conditions images" means manually labeling real images under all conditions and then training a model with these labeled data.

6. Conclusions

A YOLOv3 bale detection model combined with the domain adaptation approach is proposed in this paper, augmenting the ability for crop/bale detection in three seasons, different illumination conditions, and diverse weather conditions. This method is advantageous as it needs limited manual-labeling tasks. In this work, only the images captured under the initial condition needed to be manually labeled as the source-domain data. Then the domain adaptation approach, CycleGAN models, were trained to transfer the source-domain images to the target domain (images under other conditions) with the same labeled annotation file. We have effectively augmented the training datasets under extended conditions but without extra manual-labeling tasks. After these two steps, we trained the YOLOv3 model again with augmented training datasets. The optimized YOLOv3 model shows a significant improvement in general detecting performance. This approach decreases the labor and time cost by way of improving the crop quality and yields. It also shows strong scalability to many other crops and will significantly reduce the cost of precision agriculture. Future work should include collecting more real images under more specific conditions, generating more synthetic images associated with these conditions, and combining the activate learning method with the CycleGAN model, making the whole pipeline of the algorithm more robust and easier to use.

Author Contributions: Conceptualization, W.Z., T.R. and M.D.; methodology W.Z.; software, W.Y.; validation, T.L. and W.Y.; formal analysis, W.Z.; investigation, T.L.; resources, T.R.; data curation, M.D.; writing—original draft preparation, W.Z.; writing—review and editing, T.R.; visualization, W.Y.; supervision, T.R.; project administration, T.R.; funding acquisition, T.R. All authors have read and agreed to the published version of the manuscript.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available on request from the corresponding author. The data are not publicly available due to other ongoing researches.

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

References

1. Census Bureau. U.S. and World Population Clock. Available online: https://www.census.gov/data-tools/demo/idb/#/country?YR_ANIM=2050&COUNTRY_YEAR=2050 (accessed on 16 December 2020).
2. Valin, H.; Sands, R.D.; Van Der Mensbrugghe, D.; Nelson, G.C.; Ahammad, H.; Blanc, E.; Bodirsky, B.; Fujimori, S.; Hasegawa, T.; Havlík, P.; et al. The future of food demand: Understanding differences in global economic models. *Agric. Econ.* **2014**, *45*, 51–67. [CrossRef]
3. Mahajan, S.; Das, A.; Sardana, H.K. Image acquisition techniques for assessment of legume quality. *Trends Food Sci. Technol.* **2015**, *42*, 116–133. [CrossRef]
4. Barbedo, J. A review on the main challenges in automatic plant disease identification based on visible range images. *Biosyst. Eng.* **2016**, *144*, 52–60. [CrossRef]
5. Story, D.; Kacira, M. Design and implementation of a computer vision-guided greenhouse crop diagnostics system. *Mach. Vis. Appl.* **2015**, *26*, 495–506. [CrossRef]
6. Rieder, R.; Rieder, R. Computer vision and artificial intelligence in precision agriculture for grain crops: A systematic review. *Comput. Electron. Agric.* **2018**, *153*, 69–81. [CrossRef]

7.   Tillett, N.; Hague, T.; Grundy, A.; Dedousis, A. Mechanical within-row weed control for transplanted crops using computer vision. *Biosyst. Eng.* **2008**, *99*, 171–178. [CrossRef]
8.   Mohanty, S.P.; Hughes, D.P.; Salathé, M. Using Deep Learning for Image-Based Plant Disease Detection. *Front. Plant Sci.* **2016**, *7*, 1419. [CrossRef] [PubMed]
9.   Rastogi, A.; Arora, R.; Sharma, S. Leaf disease detection and grading using computer vision technology & fuzzy logic. In Proceedings of the 2015 2nd International Conference on Signal Processing and Integrated Networks (SPIN), Noida, India, 19–20 February 2015; Institute of Electrical and Electronics Engineers (IEEE): New York, NY, USA, July 2015; pp. 500–505.
10.   Choi, H.; Geeves, M.; Alsalam, B.; Gonzalez, F. Open source computer-vision based guidance system for UAVs on-board decision making. In Proceedings of the 2016 IEEE Aerospace Conference, Big Sky, MT, USA, 5–12 March 2016; pp. 1–5. [CrossRef]
11.   Ward, S.L.; Hensler, J.; Alsalam, B.H.Y.; Duncan, C.; Felipe, G. Autonomous UAVs Wildlife Monitoring and Tracking Using Thermal Imaging and Computer vision. In Proceedings of the IEEE Aerospace Conferece, Big Sky, MT, USA, 4–11 March 2016.
12.   Xiang, H.; Tian, L. Method for automatic georeferencing aerial remote sensing (RS) images from an unmanned aerial vehicle (UAV) platform. *Biosyst. Eng.* **2011**, *108*, 104–113. [CrossRef]
13.   Hunt, E.R., Jr.; Cavigelli, M.; Daughtry, C.S.T.; McMurtrey, J.E.; Walthall, C.L. Evaluation of Digital Photography from Model Aircraft for Remote Sensing of Crop Biomass and Nitrogen Status. *Precis. Agric.* **2005**, *6*, 359–378. [CrossRef]
14.   Rieke, M.; Foerster, T.; Geipel, J.; Prinz, T. High-Precision Positioning and Real-Time Data Processing of UAV-Systems. *ISPRS Int. Arch. Photogramm. Remote. Sens. Spat. Inf. Sci.* **2012**, *38*, 119–124. [CrossRef]
15.   Zhao, W.; Yin, J.; Wang, X.; Hu, J.; Qi, B.; Runge, T. Real-Time Vehicle Motion Detection and Motion Altering for Connected Vehicle: Algorithm Design and Practical Applications. *Sensors* **2019**, *19*, 4108. [CrossRef] [PubMed]
16.   Cheng, F.-C.; Huang, S.-C.; Ruan, S.-J. Illumination-Sensitive Background Modeling Approach for Accurate Moving Object Detection. *IEEE Trans. Broadcast.* **2011**, *57*, 794–801. [CrossRef]
17.   Zhao, W.; Xu, L.; Xi, S.; Wang, J.; Runge, T. A Sensor-Based Visual Effect Evaluation of Chevron Alignment Signs' Colors on Drivers through the Curves in Snow and Ice Environment. *J. Sens.* **2017**, *2017*, 1–10. [CrossRef]
18.   Zhao, W.; Wang, X.; Qi, B.; Runge, T. Ground-level Mapping and Navigating for Agriculture based on IoT and Computer Vision. *IEEE Access* **2020**, *8*, 221975–221985. [CrossRef]
19.   Hornberg, A. (Ed.) *Handbook of Machine and Computer Vision: The Guide for Developers and Users*; Wiley-VCH: Weinheim, Germany, 2017.
20.   Baweja, H.S.; Parhar, T.; Nuske, S. Early-season Vineyard Shoot and Leaf Estimation Using Computer Vision Techniques. *2017 Spokane Wash.* **2017**. [CrossRef]
21.   Lin, Y.; Chen, J.; Cao, Y.; Zhou, Y.; Zhang, L.; Tang, Y.Y.; Wang, S. Cross-Domain Recognition by Identifying Joint Subspaces of Source Domain and Target Domain. *IEEE Trans. Cybern.* **2017**, *47*, 1090–1101. [CrossRef]
22.   Ren, S.; He, K.; Girshick, R.; Jian, S. R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2016**, *39*, 91–99. [CrossRef]
23.   Li, S.; Yu, H.; Yang, K.; Zhang, J.; Bin, R. Video-based traffic data collection system for multiple vehicle types. *IET Intell. Transp. Syst.* **2014**, *8*, 164–174. [CrossRef]
24.   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]
25.   Romualdo, L.M.; Luz, P.H.D.C.; Devechio, F.F.S.; Marin, M.A.; Zúñiga, A.M.G.; Bruno, O.M.; Herling, V.R. Use of artificial vision techniques for diagnostic of nitrogen nutritional status in maize plants. *Comput. Electron. Agric.* **2014**, *104*, 63–70. [CrossRef]
26.   Pérez-Zavala, R.; Torres-Torriti, M.; Cheein, F.A.; Troni, G. A pattern recognition strategy for visual grape bunch detection in vineyards. *Comput. Electron. Agric.* **2018**, *151*, 136–149. [CrossRef]
27.   Chandel, N.S.; Chakraborty, S.K.; Rajwade, Y.A.; Dubey, K.; Tiwari, M.K.; Jat, D. Identifying crop water stress using deep learning models. *Neural Comput. Appl.* **2020**, *1–15*, 1–15. [CrossRef]
28.   Parra, L.; Marin, J.; Yousfi, S.; Rincón, G.; Mauri, P.V.; Lloret, J. Edge detection for weed recognition in lawns. *Comput. Electron. Agric.* **2020**, *176*, 105684. [CrossRef]
29.   Oberti, R.; Marchi, M.; Tirelli, P.; Calcante, A.; Iriti, M.; Borghese, A.N.; Oberti, R.; Marchi, M.; Tirelli, P.; Calcante, A.; et al. Automatic detection of powdery mildew on grapevine leaves by image analysis: Optimal view-angle range to increase the sensitivity. *Comput. Electron. Agric.* **2014**, *104*, 1–8. [CrossRef]
30.   Pourreza, A.; Lee, W.S.; Ehsani, R.; Schueller, J.K.; Raveh, E. An optimum method for real-time in-field detection of Huanglongbing disease using a vision sensor. *Comput. Electron. Agric.* **2015**, *110*, 221–232. [CrossRef]
31.   Maharlooei, M.; Sivarajan, S.; Bajwa, S.G.; Harmon, J.P.; Nowatzki, J. Detection of soybean aphids in a greenhouse using an image processing technique. *Comput. Electron. Agric.* **2017**, *132*, 63–70. [CrossRef]
32.   Toseef, M.; Khan, M.J. An intelligent mobile application for diagnosis of crop diseases in Pakistan using fuzzy inference system. *Comput. Electron. Agric.* **2018**, *153*, 1–11. [CrossRef]
33.   Rustia, D.J.A.; Lin, C.E.; Chung, J.-Y.; Zhuang, Y.-J.; Hsu, J.-C.; Lin, T.-T. Application of an image and environmental sensor network for automated greenhouse insect pest monitoring. *J. Asia-Pacific Èntomol.* **2020**, *23*, 17–28. [CrossRef]
34.   Barnea, E.; Mairon, R.; Ben-Shahar, O. Colour-agnostic shape-based 3D fruit detection for crop harvesting robots. *Biosyst. Eng.* **2016**, *146*, 57–70. [CrossRef]

35.   Lehnert, C.F.; English, A.; McCool, C.; Tow, A.W.; Perez, T. Autonomous Sweet Pepper Harvesting for Protected Cropping Systems. *IEEE Robot. Autom. Lett.* **2017**, *2*, 872–879. [CrossRef]
36.   Brechbill, S.C.; Tyner, W.E.; Ileleji, K.E. The Economics of Biomass Collection and Transportation and Its Supply to Indiana Cellulosic and Electric Utility Facilities. *BioEnergy Res.* **2011**, *4*, 141–152. [CrossRef]
37.   Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. In Proceedings of the 25th International Conference on Neural Information Processing Systems, Lake Tahoe, NV, USA, 1 December 2012; Volume 1, pp. 1097–1105.
38.   Richter, S.R.; Vineet, V.; Roth, S.; Koltun, V. *Playing for Data: Ground Truth from Computer Games*; Springer Science and Business Media LLC: Heidelberg, Germany, 2016; pp. 102–118.
39.   Ganin, Y.; Lempitsky, V. Unsupervised domain adaptation by backpropagation. In *International Conference on Machine Learning*; PMLR: New York, NY, USA, 27 June 2015; pp. 1180–1189.
40.   Othman, E.; Bazi, Y.; Melgani, F.; Alhichri, H.; Alajlan, N.; Zuair, M. Domain Adaptation Network for Cross-Scene Classification. *IEEE Trans. Geosci. Remote. Sens.* **2017**, *55*, 4441–4456. [CrossRef]
41.   Li, X.; Ye, M.; Fu, M.; Xu, P.; Li, T. Domain adaption of vehicle detector based on convolutional neural networks. *Int. J. Control. Autom. Syst.* **2015**, *13*, 1020–1031. [CrossRef]
42.   Qi, B.Z.; Liu, P.; Ji, T.; Wei, Z.; Suman, B. *Augmenting Driving Analytics with Multi-Modal Information*; IEEE Vehicular Networking Conference (VNC): Taipei, Taiwan, 2018.
43.   Ren, S.; He, K.; Girshick, R.; Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems*; Neural Information Processing Systems Foundation Inc.: San Diego, CA, USA, 2015; pp. 91–99.
44.   Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.
45.   He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 16 February 2017; pp. 2961–2969.
46.   Redmon, J.; Farhadi, A. Yolov3: An incremental improvement. *arXiv* **2018**, arXiv:1804.02767.
47.   Song, S.; Yu, H.; Miao, Z.; Zhang, Q.; Lin, Y.; Wang, S. Domain Adaptation for Convolutional Neural Networks-Based Remote Sensing Scene Classification. *IEEE Geosci. Remote. Sens. Lett.* **2019**, *16*, 1324–1328. [CrossRef]
48.   Ren, S.; He, K.; Girshick, R.; Jian, S. Faster R-CNN: Towards real-time object detection with region proposal networks. In Proceedings of the 29th Annual Conference on Neural Information Processing Systems, Montreal, QC, Canada, 11–12 December 2015. [CrossRef]
49.   Khodabandeh, M.; Vahdat, A.; Ranjbar, M.; Macready, W. A Robust Learning Approach to Domain Adaptive Object Detection. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Korea, 27 October 2019; Institute of Electrical and Electronics Engineers (IEEE), 2019; pp. 480–490.
50.   Zhu, J.-Y.; Park, T.; Isola, P.; Efros, A.A. Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks. In Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; Institute of Electrical and Electronics Engineers: New York, NY, USA, 2017; pp. 2242–2251.
51.   Darrenl, T. Labelimg. Available online: https://github.com/tzutalin/labelImg (accessed on 15 November 2015).
52.   Kentaro Wada. labelme: Image Polygonal Annotation with Python. Available online: https://github.com/wkentaro/labelme (accessed on 30 September 2016).