Increasing Shape Bias to Improve the Precision of Center Pivot Irrigation System Detection

Jiwen Tang 1,2, Zheng Zhang 1, Lijun Zhao 1 and Ping Tang 1,*

Abstract: Irrigation is indispensable in agriculture. Center pivot irrigation systems are popular means of irrigation since they are water-efficient and labor-saving. Monitoring center pivot irrigation systems provides important information for the understanding of agricultural production, water resources consumption and environmental change. Deep learning has become an effective approach for object detection and semantic segmentation. Recent studies have shown that convolutional neural networks (CNNs) are prone to be texture-biased rather than shape-biased, and increasing shape bias can improve the robustness and performance of CNNs. In this study, a simple yet effective method was proposed to increase shape bias in object detection networks to improve the precision of center pivot irrigation system detection. We extracted edge images of training samples and integrated them into the training data to increase shape bias in the networks. With the proposed shape increasing training scheme, we evaluated and compared PVANET and YOLOv4. Experiments with the images in Mato Grosso have shown that both PVANET and YOLOv4 achieved improved performance, which demonstrated the validity of the proposed method.

Keywords: center pivot irrigation systems; object detection; deep learning; convolutional neural networks; shape bias

1. Introduction

Irrigation is an important part of agriculture. Irrigation systems are icons of the modernization of agricultural production and intensification management [1]. On the other hand, irrigation also consumes a lot of water resources and has an impact on the environment. Mapping irrigation systems has important implications for understanding agricultural production, water resources consumption and environmental impact. Center pivot irrigation systems have relatively high efficiency and are able to irrigate in a wide range. They are popular in countries like the USA, Brazil and Israel. Yet, center pivot irrigation systems are big consumers of water and energy, 496 L of diesel oil and 3,806,511 L of water are needed per hectare per year by this method [2]. It is easy to recognize center pivot irrigation systems in satellite images by visual interpretation because the crop fields approximate to circular shape. Center pivot irrigation systems can be mapped with satellite images to get their location and distribution, which is important information about ongoing trends in agriculture and environmental change.

Mapping center pivot irrigation systems with satellite images is low-cost. With a circular shape, center pivot irrigation systems can be mapped by detecting circles. Hough transform [3] is a basic method to detect circles automatically. Hough transform converts every edge point from image space to parameter space and then detect circles by voting. Though effective for directly detecting circular objects in satellite images, Hough transform has low precision, long computational time and complex parameter setting.
In recent years, deep learning has emerged as an effective method for image recognition tasks like image classification [4,5], object detection [6,7] and image segmentation [8,9]. Especially, convolutional neural network (CNN) has become a powerful model for these tasks. Several deep-learning-based approaches have been proposed to map center pivot irrigation systems. Zhang et al. [10] presented a CNN approach to identify center pivot irrigation systems in North Colorado (USA) using Landsat images, achieving a precision of 95.85% and a recall of 93.33%. However, the proposed method relied on the Crop Data Layer (CDL) to filter non-cropland areas in the preprocessing. In addition, the processing was slow since it used a sliding window to the test images. Saraiva et al. [11] implemented a modified U-Net [12] to detect center pivot irrigation systems in a study area in Brazil. They achieved a precision of 99% and a relatively low recall of 88%. Likewise, Albuquerque et al. [13] used U-Net to map center pivot irrigation systems in three study sites of central Brazil with a precision of 98.26% and a higher recall of 94.57%. Nevertheless, the training samples of center pivot irrigation systems needed accurate delineation, which was work-intensive and time-consuming. In addition, the method had a slow processing speed, especially when the overlapping windows were large, which was essential for high performance. Tang et al. [14] proposed a joint method of PVANET-Hough to detect and delineate center pivot irrigation systems. In the proposed method, a lightweight real-time object detection network PVANET was used to detect center pivot irrigation systems, and then Hough transform was applied to delineate their shape and eliminate the false detections in the candidates of PVANET. The proposed method achieved a precision of 88.1% and a recall of 91.0%. The processing was fast and the training samples were easier to build, which only needed bounding box annotation of center pivot irrigation systems. However, Hough transform still had a high false detection rate and the parameter setting was complex.

Recent studies have shown that CNNs are prone to rely excessively on object textures rather than object shapes to make predictions [15,16], which is in sharp contrast to human vision. Texture-biased CNNs have poorer generalization and robustness when faced with images that have a different distribution to training images, because object textures are easily distorted and less distinguishable. This is one of the reasons why the object detection network PVANET has a high false detection rate when detecting center pivot irrigation systems [14]. In contrast, shape representations are more robust and distinguishable since object shapes are more stable, which also explains the robustness of human vision against distortion and why humans can recognize line drawing as it relies more on shape. When a dataset can be solved with high accuracy using only texture features, CNNs trained on the dataset tend to be texture-biased. CNNs that learn a texture-biased representation are able to learn a shape-biased representation when trained on a suitable dataset, such as Stylized-ImageNet which replaces the object-related local texture with the uninformative style of randomly selected artistic paintings. Incorporating stylized images in the training data improves the robustness and accuracy of CNNs.

Inspired by these findings, a simple yet effective method was proposed in this study to improve the performance of the detection of center pivot irrigation systems. The proposed method can be essentially regarded as a method of data augmentation. In this method, edge images of training images were incorporated as samples to training data. Edge images only had shape information and no texture information. With edge images mixed as samples in training data, object detection networks were forced to learn to detect objects based on object shapes, thus relieving the texture bias in the network and increasing shape bias to improve performance. PVANET [17] and YOLOv4 [18] were evaluated and compared in this study. Benefiting from the increase of shape bias learned from the augmented training data, the proposed method has a lower false detection rate. Moreover, in the proposed training scheme, edge images were extracted directly from training images, thus no extra data collection and annotation were required. With the data augmentation of edge images, center pivot irrigation systems detection performance of object detection models can be improved without any architectural change, preprocessing and post-processing.
2. Materials and Methods

2.1. Study Area

The study area is located in Mato Grosso, a Brazilian state in the center-west region and the south of the Amazon basin (Figure 1). Mato Grosso is the third largest state in Brazil, which spans 903,357 square kilometers. With agriculture as its main economic activity, Mato Grosso is one of the largest producers of soybean and corn in Brazil [19]. The number of center pivot irrigation systems increased significantly during 2010–2017 in Mato Grosso [20], the monitoring of which has important implications for analyzing the agricultural intensification and expansion in the region.

![Figure 1. The location of Mato Grosso in Brazil.](image)

2.2. Data

We used True Color Image (TCI) images of Sentinel-2 to detect the center pivot irrigation systems. The spatial resolution of TCI images is 10 m. The images used in the study cover three major Amazon watersheds in Mato Grosso: Juruena, Teles Pires and Xingu river. The coverage area is 750,000 square kilometers and 2/3 of Mato Grosso. There were 77 image tiles in total, acquired between June and August 2017, the images with low cloud cover were selected. The image size is $10,980 \times 10,980$ pixels.

2.3. Training Data

Images ($500 \times 500$ pixels) with center pivot irrigation systems were annotated and then used as samples to train the models. To obtain these samples, images with the size of $500 \times 500$ were randomly cropped from one Sentinel-2 image tile of the 77 image tiles in Mato Grosso acquired in July 2017, whose acquisition time was different from the test image tiles. A total of 613 images with center pivot irrigation systems were selected, and then annotated to be the samples. Examples of the samples are shown in Figure 2a,b.
Figure 2. Examples of the samples. (a,b) Ordinary samples. (c,d) Edge image samples extracted using Canny edge detector. (e,f) Edge image samples extracted using Sobel edge detector. (g,h) Edge image samples extracted using Laplacian edge detector. (i,j) Edge image samples extracted using HED. (k,l) Edge image samples extracted using DexiNed.
2.4. Methods

In this study, edge images were extracted from the training images and incorporated as samples to the training data. PVANET and YOLOv4 were used and compared as the object detection network. With the augmented shape-bias-increasing training data, PVANET and YOLOv4 were trained and then evaluated in the test image tiles. We trained and compared the models with and without edge image samples in the training data. Then we increased the number of edge image samples in the training data to see if it leads to better accuracy. We also trained the models separately with edge image samples extracted using different edge extraction methods in the training data to see if different edge image samples lead to different accuracy. Finally, we experimented to train the models with only edge image samples in the training data.

2.4.1. Edge Image Sample Data Augmentation to Increase Shape Bias

In order to increase shape bias in the network to achieve better performance, edge images of the sample images were integrated into the training data as samples. For every sample image, edge images were extracted from it and then used as samples with the corresponding annotation migrated. Examples of edge image samples are shown in Figure 2c–l. We used 5 different methods to extract edge images: Canny edge detector, Sobel edge detector, Laplacian edge detector, HED (Holistically-Nested Edge Detection) [21] and DexiNed (Dense Extreme Inception Network for Edge Detection) [22]. HED and DexiNed are two edge extraction methods based on deep learning. HED is an end-to-end edge detection method that leverages multi-scale and multi-level feature learning. DexiNed is a deep-learning-based edge detector that generates high-quality thin edge maps. In edge images, textures are removed and only shapes remain. With edge image samples in the training data, we expect that the networks will be forced to learn to detect objects based on their shapes, so that the shape bias in the networks will increase.

2.4.2. PVANET

PVANET is a lightweight real-time object detection network based on Faster R-CNN [6] with modification on the feature extraction part. The pipeline of PVANET follows Faster R-CNN, which is first extracting features with CNN, then getting region proposals with region proposal network (RPN) and finally classifying and regressing the coordinates of region proposals based on their RoI features. PVANET adopted several building blocks including concatenated rectified linear unit (C.ReLU) [23], Inception [24] and HyperNet [25] to the feature extraction network, making it deep and thin. C.ReLU is motivated from an observation of CNNs that in the early stage, filters tend to be paired such that for every filter there is another filter that is almost the opposite phase. Inspired by this observation, C.ReLU reduces the number of convolution channels by half, then simply multiplies the outputs by $-1$ and concatenates it to the outputs, thus leading to 2 times speed-up in the early stage without losing accuracy. Inception is an effective module that clusters different sizes of kernels to the convolution layers, so that the receptive fields of the features in CNNs can capture both the small and large objects in an input image. The combination of shadow fine-grained details and deep highly-abstracted information provides abundant features, which have been proven to be beneficial for many deep learning tasks [8,25,26]. PVANET integrates the last layer and two intermediate layers whose scales are 2x and 4x of the last layer, respectively. The middle-size layer was chosen to be the reference layer. The 4x-scaled layer was down-scaled by pooling and the last layer was up-scaled by linear interpolation, then concatenated to the middle-size layer. The combination of the last layer and two intermediate layers benefits the following RPN and classification network.

2.4.3. YOLOv4

YOLOv4 is a state-of-art object detection network with real-time speed and high accuracy. Based on YOLOv3 [7], YOLOv4 integrates some state-of-art methods to improve both the speed and accuracy. As a one-stage method, YOLOv4 first extracts features using CNN
and then predicts bounding boxes using anchor boxes. YOLOv4 uses CSPDarknet53 [27] as the backbone network, which additionally adds SPP block [28] to increase the receptive field and adopts PANet as the parameter aggregation method. Furthermore, cross-stage partial connections (CSP) [27] and multi-input weighted residual connections (MiWRC) [29] are employed to integrate different feature pyramid. Spatial Attention Module (SAM) [30] is used to improve the power of the backbone network. Mish activation is adopted as the activation function. As for the loss function, CIoU-loss (Complete Intersection over Union loss) [31] is used to achieve better convergence speed and accuracy on the BBox regression problem. DIoU (Distance Intersection over Union) [31] is employed as the NMS method, which considers the distance between central points of two bounding boxes when suppressing redundant boxes, making it more robust for the cases with occlusions. For the data augmentation methods, CutMix [32] and Mosaic data augmentation, Self-Adversarial Training [33] and Class label smoothing [34] are adopted to improve the robustness and generalization of the model. DropBlock [35] is used as the regularization method. For the training, multiple anchors are used for a single ground truth. Moreover, Cross mini-Batch Normalization (CmBN), Cosine annealing scheduler [36] and random training shapes are adopted to improve the training. With all these improvements, YOLOv4 is faster and more accurate.

2.4.4. Training of PVANET and YOLOv4

With the integrated training data, PVANET and YOLOv4 were both fine-tuned from the corresponding pre-trained model from ImageNet [37]. Both the learning rates were set to be 0.001. PVANET was implemented using Caffe [38] and YOLOv4 was implemented using Darknet [39]. Both the training and evaluation were done in a machine with a 32 cores Intel Core Xeon E5-2620 CPU, 126 GB RAM and 4 NVIDIA TITAN Xp graphics cards. All the settings were kept the same for all the experiments except training data. The training of PVANET took 8 h, while the training of YOLOv4 took 12 h.

2.4.5. Evaluation

We used the 77 Sentinel-2 image tiles (10,980 × 10,980 pixels each) in Mato Grosso to evaluate PVANET and YOLOv4 which were trained using the shape-bias-increasing scheme. Every image tile was cropped into 500 × 500 grids with an overlap of 200 pixels between the neighboring grids, all the grids were fed into the network to detect center pivot irrigation systems. After all the grids were detected, the duplicate detections in the overlap were removed to get the final result of the whole image tile.

We adopted two quantitative indexes to evaluate the performance: precision and recall, or false detection rate and missed detection rate, as false detection rate = 1 − precision and missed detection rate = 1 − recall. Precision is defined as the number of correct detections over the number of correct detections plus the number of false detections (Equation (1)), which tells us how many of the detections are correct. Recall is defined as the number of correct detections over the number of ground truth (Equation (2)), which tells us how many of the ground truth are detected.

\[
\text{precision} = \frac{TP}{TP + FP}, \quad (1)
\]

\[
\text{recall} = \frac{TP}{TP + FN} \quad (2)
\]

where \(TP\) is true positive, \(FP\) is false positive and \(FN\) is false negative. All the center pivot irrigation systems were manually mapped in the test image tiles as ground truth. In total, there were 641 center pivot irrigation systems in the 77 image tiles of Mato Grosso.
3. Results

3.1. Edge Image Samples Extracted Using Canny Edge Detector in the Training Data

For simplicity, we trained PVANET and YOLOv4 using the ordinary samples and the edge image samples extracted using canny edge detector, which were referred to as Shape-Biased-PVANET and Shape-Biased-YOLOv4, respectively. For comparison, we trained PVANET and YOLOv4 with only the ordinary samples in the training data, referred to as PVANET and YOLOv4, respectively. We also compared the combination method of object detection model and Hough transform [14], referred to as PVANET-Hough and YOLOv4-Hough, respectively. The combination method first used a detection network to detect center pivot irrigation system candidates and then applied Hough transform to reduce the false detection. We trained PVANET and YOLOv4 using only the ordinary samples in PVANET-Hough and YOLOv4-Hough. The results are shown in Table 1.

Table 1. The results of PVANET and YOLOv4 with edge image samples extracted using Canny edge detector in the training data.

|                     | Detected Candidate Pivots | Correctly Detected Pivots | Precision | Recall |
|---------------------|---------------------------|---------------------------|-----------|--------|
| PVANET              | 846                       | 619                       | 73.2%     | 96.6%  |
| PVANET-Hough        | 737                       | 614                       | 83.3%     | 95.8%  |
| Shape-Biased-PVANET | 706                       | 627                       | 88.8%     | 97.8%  |
| YOLOv4              | 707                       | 623                       | 88.1%     | 97.2%  |
| YOLOv4-Hough        | 682                       | 614                       | 90.0%     | 95.8%  |
| Shape-Biased-YOLOv4 | 680                       | 624                       | 91.8%     | 97.3%  |

As can be seen from the results, in the result of Shape-Biased-PVANET, there were 706 detected candidates of center pivot irrigation systems, 627 of the detected candidates were correct, 79 of the detected candidates were false and 14 center pivot irrigation were missed. The precision was 88.8%, the recall was 97.8%, the false detection rate was 11.2% and the missed detection rate was 2.2%.

In the result of PVANET, there were 846 detected candidates of center pivot irrigation systems, 619 of the detected candidates were correct, 227 of the detected candidates were false and 22 center pivot irrigation were missed. The precision was 73.2%, the recall was 96.6%, the false detection rate was 26.8% and the missed detection rate was 3.4%. PVANET had a high false detection rate, many forest patches and river banks that were similar to center pivot irrigation systems were falsely detected as them by PVANET. Examples of the false detections are shown in Figure 3.

One of the reasons why PVANET had a high false detection rate was that the fine-tuned model from the pre-trained model of ImageNet was texture-biased [16]. It relied more on texture features to make predictions, which was not able to distinguish objects like forest patches and river banks from center pivot irrigation systems as they had similar textures. With the proposed shape-bias-increasing training scheme, the edge image samples in the training data forced the network to learn to make predictions based on object shapes. Therefore, the false detection rate was decreased from 26.8% to 11.2%. The recall also had a small increase.

Hough transform in PVANET-Hough decreased the false detection rate of PVANET from 26.8% to 16.7%. However, the recall had a small decrease since 5 center pivot irrigation systems were missed by Hough transform. In addition, Hough transform added an extra post-processing step to the method. Moreover, parameters needed to be tuned for Hough transform to have optimal performance, it was less adaptive to different images. Shape-Biased-PVANET achieved better precision and recall than PVANET-Hough without adding any preprocessing or post-processing step to the model. There was no parameter tuning needed. The edge images were extracted from the sample images, and the annotations were also migrated. Therefore, no extra data collecting and annotation were needed.
In the result of Shape-Biased-YOLOv4 (Table 1), there were 680 detected candidates of center pivot irrigation systems, 624 of the detected candidates were correct, 56 of the detected candidates were false and 17 center pivot irrigation were missed. The precision was 91.8%, the recall was 97.3%, the false detection rate was 8.2% and the missed detection rate was 2.7%.

In the result of YOLOv4 (Table 1), there were 707 detected candidates of center pivot irrigation systems, 623 of the detected candidates were correct, 84 of the detected candidates were false and 18 center pivot irrigations were missed. The precision was 88.1%, the recall was 97.2%, the false detection rate was 11.9% and the missed detection rate was 2.8%.

With a more powerful backbone network and state-of-art data augmentation methods, as well as other state-of-art methods, YOLOv4 had better precision and recall than PVANET. Using the proposed shape-bias-increasing training scheme, Shape-Biased-YOLOv4 decreased the false detection rate from 11.9% of YOLOv4 to 8.2%. The recall was had a minor increase. Shape-Biased-YOLOv4 achieved better precision and recall than YOLOv4-Hough without any extra step or architectural change.

In conclusion, Shape-Biased-PVANET and Shape-Biased-YOLOv4 both achieved improved precision and recall using the simple yet effective shape-bias-increasing training scheme, without adding any extra step or changing the architect. Shape representations helped the models to have better robustness and lower false detect rates. The processing was also fast. For the detection of an image tile with the size of $10,980 \times 10,980$ pixels, Shape-Biased-PVANET took 83 s, while Shape-Biased-YOLOv4 took 40 s.

3.2. More Edge Image Samples in the Training Data

In the previous experiments, the number of edge image samples and the number of ordinary samples were the same. We added more edge image samples to the training data to see if it leads to better accuracy. Apart from the edge images extracted using canny edge detector, 4 more edge images were extracted from each ordinary sample using Sobel edge detector, Laplacian edge detector, HED and DexiNed. We successively added the 4 sets of edge image samples to the training data to make the ratio between edge image samples
and ordinary samples to be 2:1, 3:1, 4:1 and 5:1, then trained PVANET and YOLOv4 using the obtained training data, referred to as Shape-Biased-PVANET-2 to Biased-PVANET-5 and Shape-Biased-YOLOv4-2 to Shape-Biased-YOLOv4-5, respectively. All the settings were kept the same with the previous experiments except training data. The results are shown in Table 2.

Table 2. The results of PVANET and YOLOv4 with more edge image samples in the training data.

|                         | Detected Candidate Pivots | Correctly Detected Pivots | Precision | Recall  |
|-------------------------|---------------------------|---------------------------|-----------|---------|
| PVANET                  | 846                       | 619                       | 73.2%     | 96.6%   |
| Shape-Biased-PVANET     | 706                       | 627                       | 88.8%     | 97.8%   |
| Shape-Biased-PVANET-2   | 709                       | 629                       | 88.7%     | 98.1%   |
| Shape-Biased-PVANET-3   | 660                       | 632                       | 95.8%     | 98.6%   |
| Shape-Biased-PVANET-4   | 699                       | 632                       | 90.4%     | 98.6%   |
| Shape-Biased-PVANET-5   | 712                       | 626                       | 87.9%     | 97.7%   |
| YOLOv4                  | 707                       | 623                       | 88.1%     | 97.2%   |
| Shape-Biased-YOLOv4     | 680                       | 624                       | 91.8%     | 97.3%   |
| Shape-Biased-YOLOv4-2   | 677                       | 629                       | 92.9%     | 98.1%   |
| Shape-Biased-YOLOv4-3   | 675                       | 630                       | 93.3%     | 98.3%   |
| Shape-Biased-YOLOv4-4   | 673                       | 632                       | 93.9%     | 98.6%   |
| Shape-Biased-YOLOv4-5   | 678                       | 625                       | 92.2%     | 97.5%   |

As shown in Table 2, for PVANET, when the number of edge image samples was three times the number of ordinary samples, Shape-Biased-PVANET-3 achieved the best precision of 95.8% and the best recall of 98.6%. Further increasing the number of edge image samples in the training data did not lead to better accuracy. For YOLOv4, when the number of edge image samples was four times the number of ordinary samples, Shape-Biased-YOLOv4-4 achieved the best precision of 93.9% and the best recall of 98.6%. Further increasing the number of edge image samples in the training data did not lead to better accuracy either.

3.3. Edge Image Samples Extracted Using Different Methods in the Training Data

Different edge detection methods generate different edge images (Figure 2). Traditional edge detectors Canny, Sobel and Laplacian are simple and generate high-quality edge images. Edge images extracted using Sobel edge detector and Laplacian edge detector have more noises. Deep-learning-based methods HED and DexiNed generate higher-quality edge images, but they are more complex and need more computation resources. We trained the models with different edge image samples extracted using different methods in the training data to see if they lead to different performance. We separately trained PVANET and YOLOv4 with the edge image samples extracted using Canny edge detector, Sobel edge detector, Laplacian edge detector, HED and DexiNed, referred to as Shape-Biased-PVANET-Canny, Shape-Biased-PVANET-Sobel, Shape-Biased-PVANET-Laplacian, Shape-Biased-PVANET-HED and Shape-Biased-PVANET-DexiNed, respectively. YOLOv4 models were referred to like PVANET models. The results are shown in Table 3.

For PVANET, when trained with the edge image sample extracted using DexiNed in the training data, Shape-Biased-PVANET-DexiNed achieved the comparatively best accuracy, with a precision of 92.4% and a recall of 98.0%. For YOLOv4, when trained with the edge image sample extracted using HED in the training data, Shape-Biased-YOLOv4-HED achieved the comparatively best accuracy, with a precision of 94.4% and a recall of 97.8%. When trained with the edge image sample extracted using Sobel edge detector in the training data, Shape-Biased-YOLOv4-Sobel achieved a high precision of 93.5%, but the recall was comparatively lower, which was 97.0%.
Table 3. The results of PVANET and YOLOv4 with edge image samples extracted using different methods in the training data.

| Model                           | Detected Candidate Pivots | Correctly Detected Pivots | Precision | Recall  |
|---------------------------------|---------------------------|---------------------------|-----------|---------|
| PVANET                          | 846                       | 619                       | 73.2%     | 96.6%   |
| Shape-Biased-PVANET-Canny       | 706                       | 627                       | 88.8%     | 97.8%   |
| Shape-Biased-PVANET-Sobel       | 690                       | 630                       | 91.3%     | 98.3%   |
| Shape-Biased-PVANET-Laplacian   | 692                       | 627                       | 90.6%     | 97.8%   |
| Shape-Biased-PVANET-HED         | 692                       | 629                       | 90.9%     | 98.1%   |
| Shape-Biased-PVANET-DexiNed     | 680                       | 628                       | 92.4%     | 98.0%   |
| YOLOv4                          | 707                       | 623                       | 88.1%     | 97.2%   |
| Shape-Biased-YOLOv4-Canny       | 680                       | 624                       | 91.8%     | 97.3%   |
| Shape-Biased-YOLOv4-Sobel       | 665                       | 622                       | 93.5%     | 97.0%   |
| Shape-Biased-YOLOv4-Laplacian   | 683                       | 629                       | 92.1%     | 98.1%   |
| Shape-Biased-YOLOv4-HED         | 664                       | 627                       | 94.4%     | 97.8%   |
| Shape-Biased-YOLOv4-DexiNed     | 679                       | 626                       | 92.2%     | 97.7%   |

3.4. Only Edge Image Samples in the Training Data

We conducted extra experiments to further verify the generalizability of shape representations in the networks. We trained PVANET and YOLOv4 using only the edge image samples extracted using Canny edge detector, referred to as PVANET-edge and YOLOv4-edge, respectively. Then we successively added the edge image samples extracted using Sobel edge detector and Laplacian edge detector to the training data, referred to as PVANET-edge-2 and PVANET-edge-3, respectively. YOLOv4 models were referred to like PVANET models. All the settings were kept the same with the previous experiments except training data. The results are shown in Table 4.

Table 4. The results of PVANET and YOLOv4 with only edge image samples in the training data.

| Model                           | Detected Candidate Pivots | Correctly Detected Pivots | Precision | Recall  |
|---------------------------------|---------------------------|---------------------------|-----------|---------|
| PVANET                          | 846                       | 619                       | 73.2%     | 96.6%   |
| Shape-Biased-PVANET             | 706                       | 627                       | 88.8%     | 97.8%   |
| PVANET-edge                     | 661                       | 568                       | 85.9%     | 88.6%   |
| PVANET-edge-2                   | 630                       | 581                       | 92.2%     | 90.6%   |
| PVANET-edge-3                   | 658                       | 580                       | 88.1%     | 90.5%   |
| YOLOv4                          | 707                       | 623                       | 88.1%     | 97.2%   |
| Shape-Biased-YOLOv4             | 680                       | 624                       | 91.8%     | 97.3%   |
| YOLOv4-edge                     | 728                       | 572                       | 78.6%     | 89.2%   |
| YOLOv4-edge-2                   | 635                       | 585                       | 89.3%     | 91.3%   |
| YOLOv4-edge-3                   | 664                       | 582                       | 87.7%     | 90.8%   |

As we can see from the results (Table 4), even trained with only the edge image samples extracted using Canny edge detector, PVANET-edge achieved a precision of 85.9% and a recall of 88.6%, the precision of which was higher than PVANET which was trained using the ordinary samples. With the edge image samples extracted using Sobel edge detector added to the training data, the precision increased to 92.2% and the recall increased to 90.6%. Further increasing the number of edge image samples did not bring accuracy improvement. When trained with only the edge image samples extracted using Canny edge detector, YOLOv4-edge achieved a precision of 78.6% and a recall of 89.2%. With the edge image samples extracted using Sobel edge detector added to the training data, the precision increased to 89.3% and the recall increased to 91.3%. Further increasing the number of edge image samples did not bring accuracy improvement either. These results further demonstrated the robustness and generalizability of shape representations and shape-biased networks.
4. Conclusions

In this study, we proposed a simple yet effective method to accurately detect center pivot irrigation systems. Edge images of the image samples in the training data were extracted, and then integrated into the training data to increase shape bias in the object detection networks to improve robustness and accuracy. PVANET and YOLOv4 were trained and evaluated in the study. With the proposed shape-bias-increasing training scheme, both PVANET and YOLOv4 achieved better performance, especially, lower false detection rate. Compared to PVANET trained using only the ordinary samples, Shape-Biased-PVANET which was trained with the edge image samples extracted using Canny edge detector in the training data, increased the precision from 73.2% to 88.8% and increased the recall from 96.6% to 97.8%. Compared to YOLOv4 trained using only the ordinary samples, Shape-Biased-YOLOv4 which was trained with the edge image samples extracted using Canny edge detector in the training data, increased the precision from 88.1% to 91.8%. With more edge image samples added to the training data, PVANET and YOLOv4 gained more performance improvements. For PVANET, when the number of edge image samples was three times the number of ordinary samples, Shape-Biased-PVANET-3 achieved the best precision of 95.8% and the best recall of 98.6%. For YOLOv4, when the number of edge image samples was four times the number of ordinary samples, Shape-Biased-YOLOv4-4 achieved the best precision of 93.9% and the best recall of 98.6%. However, further increasing the number of edge image samples in the training data did not bring further improvements. Different edge image samples extracted using different edge detection methods in the training data led to a difference in accuracy of the models. When trained with the edge image samples extracted using DexiNed in the training data, Shape-Biased-PVANET-DexiNed achieved the comparatively best accuracy, with a precision of 92.4% and a recall of 98.0%. For YOLOv4, when trained with the edge image samples extracted using HED in the training data, Shape-Biased-YOLOv4-HED achieved the comparatively best accuracy, with a precision of 94.4% and a recall of 97.8%. Even trained using only the edge image samples extracted using Canny edge detector and Sobel edge detector in the training data, PVANET-edge-2 achieved a precision of 92.2% and a recall of 90.6%, the precision of which was higher than PVANET which was trained using the ordinary samples. When trained using only the edge image samples extracted using Canny edge detector and Sobel edge detector in the training data, YOLOv4-edge-2 achieved a precision of 89.3% and a recall of 91.3%. The performance improvements demonstrated the robustness of shape representations. The proposed shape-bias-increasing training scheme improved the performance of the object detection model without introducing any preprocessing or post-processing step. In future work, we will research introducing shape prior to object detection networks.

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

Funding: This research was funded and supported by the Strategic Priority Research Program of the Chinese Academy of Sciences, Grant No. XDA19080301 and National Natural Science Foundation of China, Grant No. 41701399.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data available in a publicly accessible repository.
Acknowledgments: We would like to thank Damien Arvor and Thomas Corpetti for their help in the data acquisition and selection, as well as the discussion.

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

References
1. Arvor, D.; Meirelles, M.; Dubreuil, V.; Bégué, A.; Shimabukuro, Y.E. Analyzing the agricultural transition in Mato Grosso, Brazil, using satellite-derived indices. Appl. Geogr. 2012, 32, 702–713. [CrossRef]
2. Rundquist, D.C.; Hoffman, R.O.; Carlson, M.P.; Cook, A.E. The Nebraska Center-Pivot Inventory: An Example of Operational Satellite Remote Sensing on a Long-Term Basis. Photogramm. Eng. Remote Sens. 1989, 55, 587–590.
3. Duda, R.O.; Hart, P.E. Use of the Hough transformation to detect lines and curves in pictures. Commun. ACM 1972, 15, 11–15. [CrossRef]
4. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778. [CrossRef]
5. Tan, M.; Le, Q.V. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. arXiv 2020, arXiv:1905.11946.
6. 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]
7. Redmon, J.; Farhadi, A. YOLOv3: An Incremental Improvement. arXiv 2018, arXiv:1804.02767.
8. Shelhamer, E.; Long, J.; Darrell, T. Fully Convolutional Networks for Semantic Segmentation. arXiv 2016, arXiv:1605.06211.
9. Tang, J.W.; Arvor, D.; Corpetti, T.; Tang, P. PVANET-Hough: Detection and location of center pivot irrigation systems from Remotely Sensed Data. Remote Sens. 2020, 12, 2139. [CrossRef]
10. Zhang, C.; Yue, P.; Di, L.; Wu, Z. Automatic Identification of Center Pivot Irrigation Systems from Landsat Images Using Convolutional Neural Networks. Agriculture 2018, 8, 147. [CrossRef]
11. Saraiva, M.; Protas, E.; Salgado, M.; Souza, C. Automatic Mapping of Center Pivot Irrigation Systems from Satellite Images Using Deep Learning. Remote Sens. 2020, 12, 558. [CrossRef]
12. Ronneberger, O.; Fischer, P.; Brox, T. U-Net: Convolutional Networks for Biomedical Image Segmentation. arXiv 2015, arXiv:1505.04597.
13. De Albuquerque, A.O.; de Carvalho Júnior, O.A.; Carvalho, O.L.F.D.; de Bem, P.P.; Ferreira, P.H.G.; de Moura, R.D.S.; Silva, C.R.; Trancoso Gomes, R.A.; Fontes Guimarães, R. Deep Semantic Segmentation of Center Pivot Irrigation Systems from Remotely Sensed Data. Remote Sens. 2020, 12, 2139. [CrossRef]
14. Geirhos, R.; Rubisch, P.; Michaelis, C.; Bethge, M.; Wichmann, F.A.; Brendel, W. ImageNet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness. arXiv 2019, arXiv:1811.12231.
15. Kim, K.H.; Hong, S.; Roh, B.; Cheon, Y.; Park, M. PVANET: Deep but Lightweight Neural Networks for Real-time Object Detection. arXiv 2016, arXiv:1608.08021.
16. Bochkovskiy, A.; Wang, C.Y.; Liao, H.Y.M. YOLOv4: Optimal Speed and Accuracy of Object Detection. arXiv 2020, arXiv:2004.10934.
27. Wang, C.Y.; Liao, H.Y.M.; Yeh, I.H.; Wu, Y.H.; Chen, P.Y.; Hsieh, J.W. CSPNet: A New Backbone that can Enhance Learning Capability of CNN. *arXiv* 2019, arXiv:1911.11929.

28. He, K.; Zhang, X.; Ren, S.; Sun, J. Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* 2015, 37, 1904–1916. [CrossRef]

29. Abdi, M.; Nahavandi, S. Multi-Residual Networks: Improving the Speed and Accuracy of Residual Networks. *arXiv* 2017, arXiv:1609.05672.

30. Woo, S.; Park, J.; Lee, J.Y.; Kweon, I.S. CBAM: Convolutional Block Attention Module. *arXiv* 2018, arXiv:1807.06521.

31. Zheng, Z.; Wang, P.; Liu, W.; Li, J.; Ye, R.; Ren, D. Distance-IoU Loss: Faster and Better Learning for Bounding Box Regression. *arXiv* 2019, arXiv:1911.08287.

32. Yun, S.; Han, D.; Chun, S.; Oh, S.J.; Yoo, Y.; Choe, J. CutMix: Regularization Strategy to Train Strong Classifiers With Localizable Features. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Korea, 27 October–2 November 2019; pp. 6022–6031. [CrossRef]

33. Chou, C.; Chien, J.; Chen, H. Self Adversarial Training for Human Pose Estimation. In Proceedings of the 2018 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC), Honolulu, HI, USA, 12–15 November 2018; pp. 17–30. [CrossRef]

34. Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; Wojna, Z. Rethinking the Inception Architecture for Computer Vision. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 2818–2826. [CrossRef]

35. Ghiasi, G.; Lin, T.Y.; Le, Q.V. DropBlock: A regularization method for convolutional networks. *arXiv* 2018, arXiv:1810.12890.

36. Loshchilov, I.; Hutter, F. SGDR: Stochastic Gradient Descent with Warm Restarts. *arXiv* 2017, arXiv:1608.03983.

37. Deng, J.; Dong, W.; Socher, R.; Li, L.J.; Li, K.; Li, F.-F. ImageNet: A large-scale hierarchical image database. In Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 20–25 June 2009; pp. 248–255. [CrossRef]

38. Jia, Y.; Shelhamer, E.; Donahue, J.; Karayev, S.; Long, J.; Girshick, R.; Guadarrama, S.; Darrell, T. Caffe: Convolutional Architecture for Fast Feature Embedding. In Proceedings of the ACM International Conference on Multimedia (MM ’14), Orlando, FL, USA, 3–7 November 2014; ACM Press: Orlando, FL, USA, 2014; pp. 675–678. [CrossRef]

39. Darknet: Open Source Neural Networks in C. Available online: https://pjreddie.com/darknet/ (accessed on 2 January 2021).