Article

Field Network—A New Method to Detect Directional Object

Jin Liu and Yongjian Gao *

State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing,
Wuhan University, Wuhan 430079, China; jliu@sgg.whu.edu.cn

* Correspondence: gaoyongjian@whu.edu.cn

Received: 12 July 2020; Accepted: 27 July 2020; Published: 30 July 2020

Abstract: As the development of object detection technology in computer vision, identifying objects is always an active yet challenging task, and even more efficient and accurate requirements are being imposed on state-of-the-art algorithms. However, many algorithms perform object box regression based on RPN(Region Proposal Network) and anchors, which cannot accurately describe the shape information of the object. In this paper, we propose a new object detection method called Field Network (FN) and Region Fitting Algorithm (RFA). It can solve these problems by Center Field. Center field reflects the probability of the pixel approaching the object center. Different from the previous methods, we abandoned anchors and ROI technologies, and propose the concept of Field. Field is the intensity of the object area, reflecting the probability of the object in the area. Based on the distribution of the probability density of the object center in the visual field perception area, we add the Object Field in the output part. And we abstract it into an Elliptic Field with normal distribution and use RFA to fit objects. Additionally, we add two fields to predict the x,y components of the object direction which contain the neural units in the field array. We extract the objects through these Fields. Moreover, our model is relatively simple and have smaller size, which is only 73 M. Our method improves performance considerably over baseline systems on DOTA, MS COCO and PASCAL VOC datasets, with overall performance competitive with recent state-of-the-art systems.

Keywords: field network; object detection; direction prediction

1. Introduction

Owing to the continual development of computer vision technology in recent years, object detection has entered a new era [1–3]. However, we also have to face the complexity and cost of the resources [2]. These problems have been around for a long time and attracted much attention in the past decade [4–7].

Traditional two-stage algorithms mainly train two parts. The first step is to train the RPN(Region Proposal Network) network, and the second step is to train the network of object area detection [2,4,5,8]. Compared with one-stage algorithms, their network has high accuracy but relatively slow speed. On the other hand, one-stage algorithms are often fast but not accurate enough [1,3,9–13]. Although there are some algorithms that take both speed and accuracy into account, they are not satisfactory because they lack sufficient depth of semantic information [10,14–19].

In the experiment, we discover that when the grid density is large, the convolution network's ability to express the intensity of the object area will be improved correspondingly, but the ability to express the spatial information of the object will be reduced. The dense output means that the same length information representing the object needs to span more neurons. Since the single-layer convolution operation spans a limited number of neurons, this requires a deeper convolutional layer network to support, while deeper networks require more feature maps. Therefore, when the output

density increases, a large model is needed to support the coordinate regression of high precision object position.

Traditional algorithms do not have enough ability to directly describe the coordinate position of the object. Moreover, these algorithms use techniques such as anchors, NMS and ROI (Non-Maximum Suppression and Region of Interest)pooling [2,8,20,21]. However, these techniques are based on the horizontal recommendation box of RPN, and the object shape and direction are various, and contains many invalid areas. Meanwhile, semantic segmentation has strong learning ability for pixel-by-pixel classification and does not require very large models to support coordinate regression of high-precision object positions [5,6,22–24]. However, the classification of each pixel of semantic segmentation is isolated, and the same type of object will be connected [7].

To solve the aforementioned problems, in this paper we propose a new object detection model called Field Network (FN). Field is the intensity of the object area, reflecting the probability of the object in the area. The field is shown in Figure 1. We combine the advantages of object detection and semantics segmentation, effectively avoid their respective shortcomings, so that the detection speed and accuracy are greatly improved. Moreover, when we add a direction field to the object field, we can also get the direction. We choose to regress the direction vector instead of the direction angle to obtain the object direction. This is because the regression direction angle will have the angle circulation problem, for example, there is a considerable error between θ and $\theta + 2\pi$. And we extensively test and evaluate the FN algorithm on three public datasets for object detection in References [25–27], and compare it with state-of-the-art methods.

Figure 1. The presentation of object field. (a) is original image, (b) is Center Field and (c) is Direction Field. We can fit the object by (b). Also, we can get the direction through (c).

The main contributions of this paper are as follows.

- We propose the concept of Field. Based on the Field, our framework can distinguish the overlapping regions of the same object on the basis of Center Field. From this we can get the center coordinates, the range of the area, and the total number of objects for each one.
- We design a Field-based object Region Fitting Algorithm (RFA), which abandons some traditional techniques and makes the algorithm efficient and accurate for object detection.
- We can also get the direction of the object through the Direction Field by regressing the direction vector.

2. Related Work

Recent years have witnessed a vast amount of work on the computer vision. Among them, the fastest growing tasks can be divided into two classical categories—object detection and object segmentation.

The first category of popular object detection algorithms can be divided into two categories, two-stage and one-stage [1–5,7–10,14,28]. RCNN(Regions with CNN features) [4] is the pioneering of the two-stage algorithm. It used a convolutional neural network (CNN) for the first time in

the field of object detection, which greatly improved the effect of target detection. After several years of development, CNNs showed its strong vitality. The most representative of these is Faster-RCNN [2]. It generates region proposals from the RPN network and then classifies the regions proposals. It greatly improves the accuracy of object detection, but at the same time its speed is relatively unsatisfactory. After obtaining the region proposals, the calculation amount for each proposal classification is still relatively large. This affects its computational efficiency to some extent. The one-stage algorithm [1,3,9,10,14] is region-free, which converts the problem of object detection into a regression problem, but the speed is improved and the accuracy is not enough. Our method also discards the region proposal, and instead proposed the concept of Object Field, which can balance accuracy and efficiency.

Another type of algorithm is called object segmentation and the pioneering is FCN(Fully Convolutional Networks). What FCN [6] pursues is that the input is a picture, and the output is also a picture. It proposes a full convolutional neural network and learns pixel to pixel mapping and end-to-end mapping. The full convolutional network mainly uses three techniques, convolutional, upsample and skip layer. But there are still many problems that cannot be avoided, such as accuracy problems, insensitivity to details and ignoring spatial consistency, and so forth. U-Net [22] is used to solve simple problem segmentation of small samples. It is improved on the basis of FCN. U-Net uses excessive data augmentation by applying elastic deformations to the available training images, to some extent solves the problem of too few samples in some scenarios. Our algorithm uses it as a backbone, adds Object Field to the output, and then uses a fitting algorithm to detect the object.

3. Algorithm

3.1. Object Field

The convolutional neural network can be abstracted into a mathematical model $Y = F(W, X)$, where X is the input, Y is the output, and W is the convolution kernel parameter. CNN can be seen as a directed acyclic graph from X to Y . Its basic architecture consists of input layer, convolutional layer, pooling layer, upsample layer and output layer. Therefore, when designing the network structure F , it should be able to express Y more quickly and accurately. However, in the CNNs, the pooling layer extracts the intensity information of the object, and the spatial information, such as the maximum response neuron offset coordinate and the object width and height, cannot be transmitted by the pooling layer. Therefore, this convolutional network has a weak ability to express spatial information. So in order for the convolutional network to better express spatial information, we add an object output field to regress the probability of the objects Y appearing in the image. Because the values of the central field data is in the range of $[0, 1]$, we transform the output layer of the neural network into the final field output value through the logistic activation function. Through the object output field, we can further obtain the location information.

The object output field is the probability distribution map of the object appearing on the image. The probability of the object center is the largest, and the closer to the edge, the lower the probability. This field can be expressed by the two-dimensional normal distribution formula:

$$\begin{cases} f(x, y) = \frac{1}{2\pi\sigma_1\sigma_2\sqrt{1-\rho^2}} e^{-k} \\ k = \frac{1}{2(1-\rho^2)} \left[\frac{(x-\mu_1)^2}{\sigma_1^2} - 2\rho \frac{x-\mu_1}{\sigma_1} \cdot \frac{y-\mu_2}{\sigma_2} + \frac{(y-\mu_2)^2}{\sigma_2^2} \right] \end{cases} \quad (1)$$

We can get a maximum probability when the field coordinates are at the center of the ellipse. According to this definition, we use neural networks to regress this field probability information in an elliptical distribution.

On the basis of backbone, we added the object field to the output section. We abstract the object field into a normally distributed elliptical field containing two components, the Center Field and the Edge Field. The architecture is shown in Figure 2. We give the loss function as

$$loss = \lambda_s loss_{softmax} + \lambda_c loss_{center} + \lambda_d loss_{dir}, \quad (2)$$

where the $loss_{center}$, $loss_{direction}$ are defined in Equations (5) and (6) respectively.

Figure 2. The architecture of Field Network. We add an elliptical field to the output, which contains the Center Field and the Direction Field. The two fields respectively output C feature maps corresponding to the regional distribution field of the C-type object.

Center Field. The intensity of the normal distribution is related to the elliptic equation, so we use the elliptic equation to describe the distribution of an object on a two-dimensional image. The output value of the center field indicates the probability that the pixel is close to the target center, so we define the range of values for each output element to be $[0, 1]$. The output intensity of each pixel is calculated by

$$G_{ccp} = \begin{cases} e^{-ad_{cpi}^2} & d_{cpi}^2 \leq 1 \\ 0 & \text{other} \end{cases}, \quad (3)$$

where 'ccp' is an abbreviation of center class pixel. And G_{ccp} is the ground truth of the pixel P of feature map of class C in the center field. d_{cpi} is the distance from the pixel P in the class C feature map to the i-th object. Figure 3 shows the distribution of object intensity in image space. To build this mathematical model, we describe how close the pixel is to the center of the object by

$$\begin{cases} d_i^2 = \frac{(\cos \theta \cdot dx_i + \sin \theta \cdot dy_i)^2}{a^2} + \frac{(-\sin \theta \cdot dx_i + \cos \theta \cdot dy_i)^2}{b^2} \leq 1 \\ d_{x_i} = x_i - x_0 \\ d_{y_i} = y_i - y_0 \end{cases}, \quad (4)$$

and we give the loss function of Center Field as

$$loss_{center} = \sum_{c=1}^C \sum_{p \in field} (v_{ccp} - G_{ccp})^2. \quad (5)$$

Direction Field. The Direction Field is used to describe the direction information of the object and requires the training dataset to have direction information. We also add $2 \times C$ channels to output the x, y direction field of the C class object, then the direction vector of a certain neuron in the field is $\{Q_0(x, y), Q_1(x, y)\}$. The loss function of Direction Field is

$$loss_{dir} = \sum_c w_c \sum_{x,y} \delta(c) E_{xy}, \quad (6)$$

where $w_c = 1$ if an object has direction and $w_c = 0$ if not. Meanwhile, if x, y belongs to at least one object in the field, then $\delta(c) = 1$, otherwise $\delta(c) = 0$. We give E_{xy} as

$$E_{xy} = (v_{dxcp} - G_{dxcp})^2 + (v_{dycp} - G_{dycp})^2, \quad (7)$$

where $G_{d_{xcp}}$ and $G_{d_{ycp}}$ are ground truth of the x and y component of the Direction Field at the object point p of class c respectively.

Figure 3. Intensity distribution of object in the field. (a) is the direction diagram, (b) is the Direction Field. (c) is the two-dimensional graphs of the Center Field. (d) is the three-dimensional graph of the Center Field.

We define the default value of the back propagation weight of the Direction Field $\lambda_d = 2$ in Equation (2). According to the theory of constrained neural networks, we unitize Q_0 and Q_1 to obtain the direction vector q_0 and q_1 at x, y by

$$\begin{cases} q_0(x, y) = \frac{Q_0(x, y)}{L} \\ q_1(x, y) = \frac{Q_1(x, y)}{L} \\ L = \sqrt{Q_0^2(x, y) + Q_1^2(x, y)} \end{cases} \quad (8)$$

The object direction represented by rotation angle in regression will lead to the ambiguity of direction. To solve this problem, we use the unitization constraint algorithm to obtain the object unitization direction vector $\{q_0, q_1\}$, which is used to regress to the Ground Truth direction of the object. As shown in Figure 4, the output values of the two channels of the direction output layer of a neuron in the object area $\{Q_0, Q_1\}$ are converted to $\{q_0, q_1\}$ by unitization.

The direction of each iron atom in the magnet determines the direction of the magnetic field. For the same principle, we find the direction of each point in the Direction Field in the RFA to get the direction of the object. Then we can calculate the direction of the object by

$$d_{object} = \frac{\frac{1}{n} \sum_{j=1}^n \{q_{j0}, q_{j1}\}}{\left| \frac{1}{n} \sum_{j=1}^n \{q_{j0}, q_{j1}\} \right|}, \quad (9)$$

which is used to calculate the average direction of n points in the object area. The detailed description of the object points searching can be found in Section 3.2.

eight neighborhoods of the pixel. Then move to the position of this maximum value and repeat the search step until there is no greater value around it, then note the coordinates of the point (x_c, y_c) .

Getting the object edge point sets by searching the Center Field from the center point (x_c, y_c) . We use the center point as the starting point to get the point set of the edge area of each object through breadth-first search. As shown in Figure 3b, we spiral down from top to bottom to search the entire Center Field. The whole search process is as follows:

Step 1: Initializing a queue Q and put the starting point P_0 into Q.

Step 2: The head element P_i in Q is taken out, and then the 8 pixel neighborhood points $P_k \{k = 1, 2, 3, \dots, 8\}$ of P_i are pushed into Q by value V_k in Center Field in descending order. P_k must be a point that has not been searched.

Step 3: Repeat **Step 2** until Q is empty. In addition, if the average intensity of all points in Q is less than 0.5, the loop is exited. Finally, we can get all the point sets $\{x_j, y_j, v_j\}$ in Q corresponding to the starting point P_0 .

Step 4: We sample the point set in the Direction Field to get Q_0, Q_1 in the object, then get the unitized vector q_0, q_1 of each point by Equation (8), and then get the whole direction vector according to Equation (9).

Figure 6 is a diagram illustrating the above algorithm. It can be seen from the Figure 6b that as the iteration progresses, the search range gradually expands, and the center field strength of each object gradually decreases during the regional growth iteration process. When the average value is around 0.25, the center field intensity tends to be flat and there is a sufficient amount of sampled data. At this time, ellipse fitting can be performed, and the region growth process of a single object ends. This algorithm can converge quickly, and can collect a sufficient number of points that can regress to the object ellipse parameters.

Figure 6. Visual demonstration of Region Fitting Algorithm (RFA) algorithm. (a) is a graph of candidate edge points superimposed with multiple objects. (b) is the average intensity of the points in the queue during regional growth, and each of the different colored curves corresponds to an object in (a).

Calculating the elliptic equation of the object. An ellipse can effectively describe the regional distribution of an object of arbitrary aspect ratio in the image space. We substitute the edge points into Equation (4), and use the LM(Levenberg-Marquard) algorithm to solve the equations. In addition, we add a central restraint condition as

$$\lambda (x_0^2 + y_0^2) = 0. \quad (12)$$

Since the value interval of the center point (x_0, y_0) is $[0, 1]$, we define the default value $\alpha = 2000$ to have a better effect. According to Equations (3) and (4), we can get $d_i^2 = -\frac{\log(Y_{cpi})}{\alpha}$, where Y_{cpi} is the output of the neural network at the pixel p_i of the Center Field. Then we give the Jacobian matrix equations as shown in Equation (13). Where a and b are the major axis and minor axis of the ellipse respectively, and θ is the inclination angle of the ellipse. Because the ellipse is symmetric, the exact

direction of the object needs to be further determined by the direction field. x_0 and y_0 are the offset of the ellipse from the search center. And F_i is the value of the ellipse field at pixel p_i .

$$\begin{bmatrix} \frac{\partial F_1}{\partial a} & \frac{\partial F_1}{\partial b} & \frac{\partial F_1}{\partial x_0} & \frac{\partial F_1}{\partial y_0} & \frac{\partial F_1}{\partial \theta} \\ \dots & \dots & \dots & \dots & \dots \\ \frac{\partial F_n}{\partial a} & \frac{\partial F_n}{\partial b} & \frac{\partial F_n}{\partial x_0} & \frac{\partial F_n}{\partial y_0} & \frac{\partial F_n}{\partial \theta} \\ 0 & 0 & 2\lambda x_0 & 2\lambda y_0 & 0 \end{bmatrix} \begin{bmatrix} \Delta a \\ \Delta b \\ \Delta x_0 \\ \Delta y_0 \\ \Delta \theta \end{bmatrix} = \begin{bmatrix} d_1^2 - F_1 \\ \dots \\ d_n^2 - F_n \\ \lambda [0 - (x_0^2 + y_0^2)] \end{bmatrix}. \quad (13)$$

Then we define F_i as

$$F_i = \frac{(\cos \theta \cdot dx_i + \sin \theta \cdot dy_i)^2}{a^2} + \frac{(-\sin \theta \cdot dx_i + \cos \theta \cdot dy_i)^2}{b^2}, \quad (14)$$

and $e_i \{i = 1, 2, \dots, n\}$ is the intensity. We compute the params $\{a, b, x_0, y_0, \theta\}$ by minimizing the Mahalanobis distance:

$$\min_{a, b, x_0, y_0, \theta} \left[\sum_{i=1}^n (d_i - F_i)^T (d_i - F_i) + e^{-2\lambda^2} (x_0^2 + y_0^2)^2 \right]. \quad (15)$$

In addition, if an object has no direction, the ellipse fitting equation is defined as

$$\frac{(x_i - x_0)^2}{a^2} + \frac{(y_i - y_0)^2}{b^2} = d_i^2. \quad (16)$$

4. Experiments

4.1. Datasets

For experiments, we choose three datasets, known as DOTA, MS COCO, and PASCAL VOC for object detection.

DOTA [25,29]. It is the largest dataset for object detection in aerial images with oriented bounding box annotations. It contains 2806 large size images. There 15 categories, including Baseball diamond (BD), Ground track field (GTF), Small vehicle (SV), Large vehicle (LV), Tennis court (TC), Basketball court (BC), Storage tank (ST), Soccer-ball field (SBF), Roundabout(RA), Swimming pool (SP), and Helicopter (HC) [25,29]. The fully annotated DOTA images contain 188, 282 instances. We cut these images into subgraphs of size 416×416 and use these subgraphs as a collection of training samples.

MS COCO [27]. MS COCO is a large-scale object detection, segmentation, and captioning dataset. We used MS COCO 2014 dataset in our experiment. It contains 80 k training images, 40 k validation images and 40 k testing images.

PASCAL VOC [26]. The PASCAL Visual Object Classes is a world-class computer vision challenge that has emerged with many classic object detection and segmentation models. The most widely used datasets are VOC 2007 and VOC 2012. The VOC 2007 dataset consists of about 5k trainval images and 5 k test images over 20 object categories [2]. And the VOC 2012 has 11 k trainval images. In order to increase the amount of data, we combine these two datasets and then experiment based on this.

4.2. Implementation Details

We use the Darknet [1] framework for all training and inference. Darknet is an open source neural network framework written in C and CUDA. It is fast, easy to install, and supports CPU and GPU computation [30]. The classic object detection algorithm Yolo [1,9,10] is based on Darknet.

In the experiments, we trained two basic field models, U-Field-Net and FCN-Field, using U-Net and FCN as backbone respectively. For training, we firstly set up encode-decode structure network

to construct FCN-Field. Then we use the route layer to concat the output of upsample layer in the network decode part and the same size layer before maxpooling in the encode layer. Then the structure is constructed. And the batch size of FN is set to 8, the learning rate is set to 0.00001 for the first. Then it will be dropped by 10% at 100 and 50,000 batches respectively. The input image is resized to 224×224 .

4.3. Ablation Studies

We conduct a serial of ablation experiments on DOTA to find the appropriate settings of our proposed FN. And we use the U-Net and FCN as our baseline respectively. Then gradually change the settings. Table 1 summarizes the results of ablation studies at the training. It can be seen from the table that the U-Field-Net mAP is significantly higher than the Field-FCN. This is because the cross-connect between the encode and decode layers improves the ability to express network features. In addition, the mAP model with the batch normalize layer is higher overall. As described in Reference [9], batch normalize is a good way to prevent overfitting. We found that the larger the batch each subdivision, the better the model obtained, but at the cost of more memory resources consumed. The batch size is set to 64 which can get better performance under the same resolution, which is consistent with the configuration in Reference [9].

The output layer of U-Field-Net contains $2 \times C$ feature maps. If there are more categories, the model will be larger, so we designed a simplified model, as shown in the last row of Table 1. From this group of experiments, we can see that the highest accuracy can be achieved when batch normalize, batchsize = 64 and subdivision = 8 are enabled. We combine $2 \times C$ output fields into 2, and add a group of softmax layers composed of $C + 1$ output feature maps, so that the feature maps of the output layer are reduced from $2 \times C$ to $C + 3$.

Table 1. Results of ablation studies on the DOTA dataset at the training. We built 8 models by adding batch normalize to the convolutional layers and using different amounts of batch size in the Darknet.

Method	Batch Normalize	Batch Size	Subdivisions	Batch Each Subdivision	mAP
FCN-Field	✓	64	8	8	74.74
U-Field-Net	✓	64	8	8	75.18
	✓	16	4	4	75.05
	✓	8	1	8	75.12
	✓	4	1	1	74.97
		64	8	8	74.78
		16	4	4	74.56
		4	1	4	74.52
		8	1	8	74.76
U-Field-Net-1C	✓	64	8	8	63.72

At the inference phase, we also did ablation studies. Table 2 shows the results of the experiments according to Equation (13). We use hit precision (HP) to describe the accuracy of object detection at the inference. HP is defined as follows:

$$HP = \frac{TP}{TP + FP + TN}. \quad (17)$$

It can be seen from Table 2 that at inference, using Ransac, scaling the local graph and using the central constraint to solve the object elliptic equation can achieve the highest accuracy, which is significantly better than other methods not fully adopted.

Table 2. Results of ablation Algorithm (RFA) on the DOTA dataset at the training. Ransac means whether to use the ransac method when calculating ellipse parameters. Resize indicates whether the output field is enlarged by 2 times. Central restraint condition means whether to consider the central constraint by Equation (12).

Method	Ransac	Resize	Central Restraint Condition	HP
U-Field-Net	✓	✓	✓	78.43
		✓	✓	73.56
	✓		✓	73.14
	✓	✓		73.43
	Replaced by min AreaBox	✓	✓	73.15

In order to study the influence of tensor obj transform on the model, we compared several typical backbones in anchors, points and FN. The representatives of anchors are Yolo [10] and RoI Transform [29]; the representative algorithm of points is CenterNet [31]. Tables 3 and 4 show the mAP comparison between our FN method and other methods under the same backbone on VOC and DOTA. It can be seen from the table that the accuracy of FN method is significantly higher than that of other methods.

Table 3. Comparisons with different backbones on VOC dataset .

Backbone	anchors	points	FN
VGG16 [32]	77.65	73.37	79.92
FPN [15]	79.82	76.82	81.36
Hourglass [21]	79.64	76.05	80.25
DLA [33]	78.27	75.98	79.68

Table 4. Comparisons with different backbones on DOTA dataset.

Backbone	anchors	points	FN
VGG16 [32]	67.74	61.37	73.52
FPN [15]	69.56	63.16	75.18
Hourglass [21]	68.96	63.04	75.17
DLA [33]	68.37	62.98	75.03

4.4. Comparison with the State-of-the-Art Methods

We compared the performance of our proposed FN with the state-of-the-art algorithms on three datasets DOTA [25], MS COCO [27] and PASCAL VOC [26]. Yolo, SSD and Retinanet are all one stage algorithms, and anchors are used for regression. Faster-Rcnn is a two stage algorithm, which adopts anchors for RPN regression. Cornernet is the anchor free method, which uses the corner points in the upper left corner and the lower right corner to predict bbox. Different from the above methods, our method has the function of object detection and direction judgment through the regression of object field.

Performance on the DOTA dataset. In Table 5, we compare our method with state-of-the-art detectors on the DOTA dataset. As can be seen from this table, FN based on Field-FCN achieved the mAP of 74.74 for DOTA, it outperforms the previous RoI Trans(69.56) by 5.18 points. Furthermore, FN based on U-Field-Net also achieved the mAP of 75.18, which has improvement by 0.71 points. We give some qualitative results of FN on DOTA in Figures 7 and 8. The direction error is shown in Table 6. The previous methods can only find the quadrilateral of the object, but not the direction. Therefore, we give the accuracy index of the direction vector of the remote sensing objects.

Table 5. Comparisons with state-of-the-art detectors on DOTA [25]. The short names for each category can be found in Reference [29]. The object class marked with * has a directional attribute, that is $w_c = 1$ in the Equation (6). Selective regression of the directional fields of these categories can significantly improve accuracy.

method	Plane *	BD	Bridge	GTF	SV *	LV *	Ship *	TC	BC	ST	SBF	RA	Harbor *	SP	HC *	mAP
FR-O [25,29]	79.42	77.13	17.7	64.05	35.3	38.02	37.16	89.41	69.64	59.28	50.3	52.91	47.89	47.4	46.3	54.13
RRPN [34]	80.94	65.75	35.34	67.44	59.92	50.91	55.81	90.67	66.92	72.39	55.06	52.23	55.14	53.35	48.22	60.67
R2CNN [29,35]	88.52	71.2	31.66	59.3	51.85	56.19	57.25	90.81	72.84	67.38	56.69	52.84	53.08	51.94	53.58	61.01
R-DFPN [36]	80.92	65.82	33.77	58.94	55.77	50.94	54.78	90.33	66.34	68.66	48.73	51.76	55.1	51.32	35.88	57.94
Yang et al. [29,37]	81.25	71.41	36.53	67.44	61.66	50.91	56.6	90.67	68.09	72.39	55.06	55.6	62.44	53.35	51.47	62.29
ICN [38]	81.36	74.3	47.7	70.32	64.89	67.82	69.98	90.76	79.06	78.2	53.64	62.9	67.02	64.17	50.23	68.16
ROI Trans. [29]	88.64	78.52	43.44	75.92	68.81	73.68	83.59	90.74	77.27	81.46	58.39	53.54	62.83	58.93	47.67	69.56
FCN-Field	93.06	76.21	36.93	78.24	88.25	89.02	90.18	90.75	95.37	77.82	62.47	56.98	40.02	94.37	51.41	74.74
U-Field-Net	93.45	76.41	37.04	78.54	88.71	89.94	90.60	90.90	95.24	78.16	62.81	57.43	40.69	95.40	52.39	75.18

Figure 7. Visualizations of the Field's results on the DOTA dataset. The first column is the original image overlaying with the object ellipse and its bounding rectangle, the second column is the Center Field, the third column is the direction diagram and the last column is the Direction Field.

Table 6. Direction error on DOTA. λ_d is defined in Equation (2). Error with * indicates that we regress the Direction Field of the objects with * in the Table 5. Error with all means that we regress the Direction Field of all objects. Experiments show that for some objects that have no direction, or symmetric objects, such as storage tanks, direction regression will reduce the accuracy of overall direction prediction.

parameter	$\lambda_d = 2$	$\lambda_d = 10$	$\lambda_d = 20$
Error with * (degree)	8.5	4.1	2.3
Error with all (degree)	10.6	6.2	4.0
Percentage within 10 degrees	48.5	92.7	96.1

Performance on the MS COCO dataset. In Table 7, we compare our method with that of References [10,39,40] on the MS COCO dataset. Our method achieves the state-of-the-art performance on mAP. Specifically, based on our proposed method, the mAP can reach 61.2, which is the best performance among these methods.

Performance on the PASCAL VOC dataset. In Table 8, we also compare our method with the state-of-the-art methods on PASCAL VOC dataset. From the table we can see that our method also achieves the best performance, which is 82.0.

Figure 8. Visualizations of results on the DOTA dataset. The line segment starting from the object center point identifies the direction of the object. The category object marked with a main direction line in the area of the object oblique quadrilateral is the object type marked with * in Table 5.

Table 7. Comparisons with the state-of-the-art methods on MS COCO. The YO-v3 indicates the YOLOv3 method. The Re-Net indicates the RetinaNet method and the Cor means the CornerNet method.

method	YO-v3[10]	Re-Net[39]	Cor[40]	OURS
mAP	57.9	61.1	56.5	61.2

Table 8. Comparisons with the state-of-the-art methods on PASCAL VOC. The YO-v3 indicates the YOLOv3 method. The F-RCN means the Faster RCNN method.

method	YO-v3 [10]	F-RCN[2]	SSD [3]	OURS
mAP	63.4	70.0	76.8	79.1

4.5. Running Time

Given a 224×224 image, our method runs at 25 fps on a desktop with an Intel E5 3.5GHz CPU and a RTX 2080Ti GPU, which is efficient for real-time object detection. Table 9 shows the performance of FN. As we can see from the table, our model is only 73M. At the same time, when the input image size is 576×576 , our model detection time only needs 0.1s to achieve the mAP of 75.35.

Table 9. Performance testing of our U-Field-Net (including post process time) on DOTA. All the speed are tested on a single RTX 2080Ti.

Input Size	mAP	Test Speed	param	BFLOPs
224×224	75.18	0.025s	73M	24.31Bn
448×448	75.32	0.051s	73M	97.23Bn
576×576	75.35	0.106s	73M	198.56Bn

5. Conclusions

In this paper, we proposed an algorithm based on a field—called FN—for object detection, which can effectively balance speed and accuracy. The field can reflect the intensity of the object area. Our algorithm can not only detect the objects, but also determine the direction. Moreover, even if it is a big image, we can detect it by spray painting without cutting. Compared with the traditional ROI method, our method can describe the geometric distribution of the object space more accurately. At the same time, the directional field regression method proposed in this paper can be used to study the output direction field of directional object categories (such as aircraft, ship, car). In the future, we will consider using this method to achieve a probabilistic and directional semantic segmentation, and increase the probability and direction information based on the segmentation algorithm to improve the ability to understand the scene semantics. This method can be applied to many computer vision applications. Furthermore, we reported the state-of-the-art performances on three widely-used datasets and demonstrated the rationality of the proposed approach. The method proposed in this paper is not limited to air image for any scene image. And the ground objects captured by satellite images have significant two-dimensional directivity, which is convenient for us to carry out experimental tests. Also, our method has some limitations. When the input image resolution is large, the learning speed is slow, and a larger backbone is needed.

Author Contributions: Conceptualization, J.L.; methodology, J.L.; software, J.L.; validation, J.L. and Y.G.; formal analysis, J.L. and Y.G.; writing—original draft preparation, J.L.; writing—review and editing, J.L. and Y.G.; visualization, J.L.; supervision, J.L.; project administration, J.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by National Natural Science Foundation of China under Grant No. 41771457.

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

Abbreviations

The following abbreviations are used in this manuscript:

MDPI	Multidisciplinary Digital Publishing Institute
DOAJ	Directory of open access journals
TLA	Three letter acronym
LD	linear dichroism

References

1. 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, 26 June–1 July 2016.
2. 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 28*; Cortes, C., Lawrence, N.D., Lee, D.D., Sugiyama, M., Garnett, R., Eds.; Curran Associates, Inc.: New York, NY, USA, 2015; pp. 91–99.
3. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. SSD: Single Shot MultiBox Detector. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 11–14 October 2016.
4. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Columbus, OH, USA, 23–28 June 2014.
5. Girshick, R. Fast R-CNN. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 7–13 December 2015.
6. Long, J.; Shelhamer, E.; Darrell, T. Fully Convolutional Networks for Semantic Segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015.
7. He, K.; Gkioxari, G.; Dollar, P.; Girshick, R. Mask R-CNN. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017.
8. Cai, Z.; Vasconcelos, N. Cascade R-CNN: Delving Into High Quality Object Detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–22 June 2018.
9. Redmon, J.; Farhadi, A. YOLO9000: Better, Faster, Stronger. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017.
10. Redmon, J.; Farhadi, A. YOLOv3: An Incremental Improvement. *arXiv* **2018**, arXiv:1804.02767
11. Woniak, M.; Poap, D. Soft trees with neural components as image-processing technique for archeological excavations. *Pers. Ubiquitous Comput.* **2020**. [[CrossRef](#)]
12. Polap, D.; Wozniak, M. Bacteria shape classification by the use of region covariance and Convolutional Neural Network. In Proceedings of the 2019 International Joint Conference on Neural Networks (IJCNN), Budapest, Hungary, 14–19 July 2019.
13. Wozniak, M.; Polap, D. Object detection and recognition via clustered features. *Neurocomputing* **2018**, *320*, 76–84. [[CrossRef](#)]
14. Fu, C.Y.; Liu, W.; Ranga, A.; Tyagi, A.; Berg, A.C. DSSD: Deconvolutional Single Shot Detector. *arXiv* **2017**, arXiv:1701.06659.
15. Lin, T.Y.; Dollar, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature Pyramid Networks for Object Detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017.
16. Tang, Y.; Chen, M.; Wang, C.; Luo, L.; Zou, X. Recognition and Localization Methods for Vision-Based Fruit Picking Robots: A Review. *Front. Plant Sci.* **2020**, *11*, 510. [[CrossRef](#)] [[PubMed](#)]
17. Tang, Y.; Li, L.; Wang, C.; Chen, M.; Feng, W.; Zou, X.; Huang, K. Real-time detection of surface deformation and strain in recycled aggregate concrete-filled steel tubular columns via four-ocular vision. *Robot. Comput. Integr. Manuf.* **2019**, *59*, 36–46. [[CrossRef](#)]

18. Chen, M.; Tang, Y.; Zou, X.; Huang, K.; Lian, G. Three-dimensional perception of orchard banana central stock enhanced by adaptive multi-vision technology. *Comput. Electron. Agric.* **2020**, *174*, 105508. [\[CrossRef\]](#)
19. Chen, M.; Tang, Y.C.; Zou, X.; Huang, K.; He, Y. High-accuracy multi-camera reconstruction enhanced by adaptive point cloud correction algorithm. *Opt. Lasers Eng.* **2019**, *122*, 170–183. [\[CrossRef\]](#)
20. Cao, Z.; Simon, T.; Wei, S.E.; Sheikh, Y. Realtime Multi-Person 2D Pose Estimation Using Part Affinity Fields. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017.
21. Newell, A.; Yang, K.; Deng, J. Stacked Hourglass Networks for Human Pose Estimation. In *Computer Vision—ECCV 2016*; Leibe, B., Matas, J., Sebe, N., Welling, M., Eds.; Springer International Publishing: Cham, Switzerland, 2016; pp. 483–499.
22. Ronneberger, O.; Fischer, P.; Brox, T. U-Net: Convolutional Networks for Biomedical Image Segmentation. *Medical Image Computing and Computer-Assisted Intervention—MICCAI 2015*; Navab, N., Hornegger, J., Wells, W.M., Frangi, A.F., Eds.; Springer International Publishing: Cham, Switzerland, 2015; pp. 234–241.
23. Badrinarayanan, V.; Kendall, A.; Cipolla, R. SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 2481–2495. [\[CrossRef\]](#)
24. Zhao, H.; Shi, J.; Qi, X.; Wang, X.; Jia, J. Pyramid Scene Parsing Network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017.
25. Xia, G.S.; Bai, X.; Ding, J.; Zhu, Z.; Belongie, S.; Luo, J.; Datcu, M.; Pelillo, M.; Zhang, L. DOTA: A Large-Scale Dataset for Object Detection in Aerial Images. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–22 June 2018.
26. Vicente, S.; Carreira, J.; Agapito, L.; Batista, J. Reconstructing PASCAL VOC. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Columbus, OH, USA, 23–28 June 2014.
27. Lin, T.Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C.L. Microsoft COCO: Common Objects in Context. In *Computer Vision—ECCV 2014*; Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T., Eds.; Springer International Publishing: Cham, Switzerland, 2014; pp. 740–755.
28. Dai, J.; Li, Y.; He, K.; Sun, J. R-FCN: Object Detection via Region-based Fully Convolutional Networks. In *Advances in Neural Information Processing Systems 29*; Lee, D.D., Sugiyama, M., Luxburg, U.V., Guyon, I., Garnett, R., Eds.; Curran Associates, Inc.: New York, NY, USA, 2016; pp. 379–387.
29. Ding, J.; Xue, N.; Long, Y.; Xia, G.S.; Lu, Q. Learning RoI Transformer for Oriented Object Detection in Aerial Images. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 16–20 June 2019.
30. Redmon, J. Darknet: Open Source Neural Networks in C. 2013–2016. Available online: <http://pjreddie.com/darknet/> (accessed on 29 May 2020).
31. Duan, K.; Bai, S.; Xie, L.; Qi, H.; Huang, Q.; Tian, Q. CenterNet: Keypoint Triplets for Object Detection. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Korea, 27–28 October 2019.
32. Simonyan, K.; Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv* **2014**, arXiv:1409.1556.
33. Yu, F.; Wang, D.; Shelhamer, E.; Darrell, T. Deep Layer Aggregation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–22 June 2018.
34. Ma, J.; Shao, W.; Ye, H.; Wang, L.; Wang, H.; Zheng, Y.; Xue, X. Arbitrary-Oriented Scene Text Detection via Rotation Proposals. *IEEE Trans. Multimed.* **2018**, *20*, 3111–3122. [\[CrossRef\]](#)
35. Jiang, Y.; Zhu, X.; Wang, X.; Yang, S.; Li, W.; Wang, H.; Fu, P.; Luo, Z. R2CNN: Rotational Region CNN for Orientation Robust Scene Text Detection. *arXiv* **2017**, arXiv:1706.09579.
36. Yang, X.; Sun, H.; Fu, K.; Yang, J.; Sun, X.; Yan, M.; Guo, Z. Automatic Ship Detection in Remote Sensing Images from Google Earth of Complex Scenes Based on Multiscale Rotation Dense Feature Pyramid Networks. *Remote Sens.* **2018**, *10*, 132. [\[CrossRef\]](#)
37. Hsieh, M.R.; Lin, Y.L.; Hsu, W.H. Drone-Based Object Counting by Spatially Regularized Regional Proposal Network. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017.
38. Azimi, S.M.; Vig, E.; Bahmanyar, R.; Körner, M.; Reinartz, P. Towards Multi-class Object Detection in Unconstrained Remote Sensing Imagery. In *Computer Vision—ACCV 2018*; Jawahar, C.V., Li, H., Mori, G., Schindler, K., Eds.; Springer International Publishing: Cham, Switzerland, 2019; pp. 150–165.

39. Lin, T.Y.; Goyal, P.; Girshick, R.; He, K.; Dollar, P. Focal Loss for Dense Object Detection. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017.
40. Law, H.; Deng, J. CornerNet: Detecting Objects as Paired Keypoints. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018.

© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).