MDPI

Article

EFN: Field-based Object Detection for Aerial Images

Jin Liu [1], Haokun Zheng [2]*

[1] LIESMARS, Wuhan University; jliu@sgg.whu.edu.cn
[2] LIESMARS, Wuhan University; 2019286190101@whu.edu.cn
* Correspondence: zackrt@foxmail.com

Version September 11, 2020 submitted to Journal Not Specified

Abstract: In this paper, We propose a field-based network for object detection: Ellipse Field Network(EFN). It is a elegant way to detect the objects that is cluttered and rotated. EFN works with the probability fields which can preserves the information of object distribution in image space during forward propagation. It is for object detection in aerial images, and also work well in natural images detection. The extensive experiments have validated that EFN can work with a light weight model and doesn't sacrificing performance. We achieve state-of-the-art results in aerial images test, and a good score in natural images.

Keywords: Aerial Image; Object detection; Semantic segmentation; Probability field

1. Introduction

Remote sensing imaging technique has opened a door for people to better understand the earth. In recent years, as the resolution of remote sensing images has increased, remote sensing target detection (e.g., the detection of airplane, ship, oil-pot, etc), has become a research hot-spot[1–5]. Remote sensing target detection has broad applications, such as military investigation, disaster rescue, and urban traffic management. Different from natural images taken from low-altitude perspectives, aerial images are taken with bird views, which implies that objects in aerial images are arbitrary oriented. Moreover, in many circumstances, the background is complex, the targets are densely arranged and vary in shape and orientation. These problems make the target detection of aerial image very challenging. Traditional methods[2,4,6–8] based on the horizontal region proposals always get an unsatisfactory result because of the mismatches between the Region of Interests (RoIs) and objects. This leads to the common misalignment between the final object classification confidence and localization accuracy. Primitively, rotated anchors have been used to tackle this problem[9,10], but it comes with several times of computational complexity. Subsequently, two-stage approach frameworks[1,11–13] are proposed, it still regresses the horizontal bounding box at first, and then trains a light weight transformer to transform the Horizontal Region of Interest (HRoI) into a Rotated Region of Interest (RRoI).

We start with the structure of the network. The features in deeper layers of a CNN are beneficial for category recognition, but it is not conducive to localizing objects. The pooling layers in CNN preserve the strength information and broaden the receptive field meanwhile loses the location information to a great extent. The relative works[14–17] show that the descriptors have very weak ability to locate the objects relative to the center of the filter. On the other hand, many related works[18–20] shows that learning with segmentation has done a great job. Edges and boundaries are the basic elements that constitute human visual cognition[21,22]. As the feature of semantic segmentation tasks well captures the boundary of an object, segmentation may be helpful for category recognition. Secondly, the ground-truth bounding box of an object is determined by its well-defined boundary. For some objects with a non-rectangle shape (e.g., a slender ship), it will be difficult to predict high IoU locations. As object boundaries can be well encoded in semantic segmentation features, learning with segmentation would be helpful for accurate object localization.

To this end, We propose EFN, a detector with more efficient training method that takes the advantage of preserving location. EFN can be regarded as the supplement and upgrade of traditional semantic segmentation, which has a strong ability of pixel by pixel classification that can get the fine-grained region division. But the problem is that the pixels of same kind are connected, and the classification of each pixel is isolated, so we can not get the overall information of a object, nor the number of object, that is difficult to carry out effective semantic understanding. EFN proposed the concept of "Object Field" which is defined as a probability density function describing the distribution of objects in image space, it is composed of "Center Field" and "Edge Field". Luo et al.[23] shows that the effective receptive field follows the Gaussian distribution. In combination with practical results, We consider that 2D-Gaussian distribution is an appropriate choice for object field[24]. The intensity distribution of the Gaussian distribution is related to the elliptic equation(as shown in Fig.1(b)), so we call this method Ellipse Field Network(EFN). In addition, we designed a special post-processing for Object Field: Ellipse Region Fitting(ERF), it combine the center field and the edge field, and finally get the ellipse region set of objects. To sum up, our work has several advantages:

- Field-based, preserves the location information.
- Anchor-free, reduces computation cost.
- A robust post-processing makes result more reliable.
- The framework is one-stage, get rotated boxes directly.

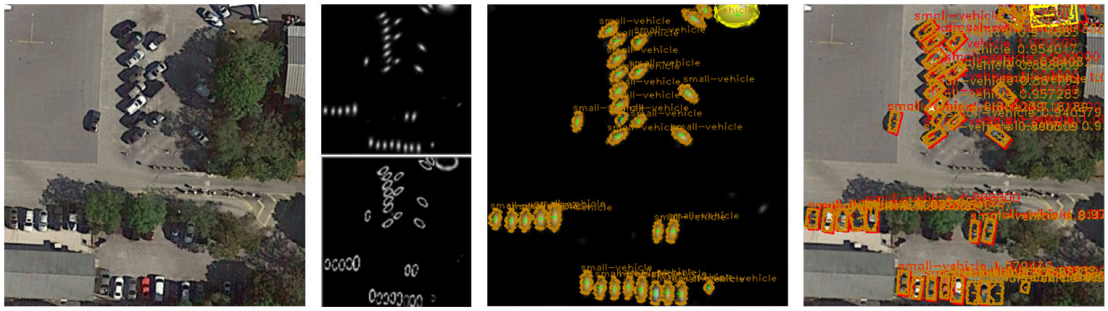


Figure 1. Overall pipeline. (a) is the input. (b) is the Center Field (top) and Edge Field (bottom). (c) is the post-process by ERF. (d) is the detection result.

2. Ellipse Field Network Methodology

Fig.2 illustrates the EFN architecture. The EFN usually takes 418*418 images as input, so it need to be cropped when the input is high-resolution. The network first process the input image with several convolutional, max pool and concatenate layer. Then, branch into two sibling output layer: one is the Object Field and another is the Edge Field. Each output layer has several channels corresponding to the number of categories. After that, the ERF algorithm will process the output to get the center points and edge points of each object and finally figures out the ellipse. As in (1), we use 5 parameters: x_0, y_0, a, b and θ to define an ellipse, where the x_0, y_0 are the coordinates of the center point of the ellipse, a, b are the semi-major axis and the semi-minor axis, θ is the angles of rotation. The function $F(x, y;\ x_0, y_0, a, b, \theta)$ describes the relationship between a point and an ellipse.

$$F(x,y;\ x_0,y_0,a,b,\theta) = \frac{[\cos\theta\cdot(x-x_0)+\sin\theta\cdot(y-y_0)]^2}{a^2} + \frac{[-\sin\theta\cdot(x-x_0)+\cos\theta\cdot(y-y_0)]^2}{b^2} - 1 \tag{1}$$

2.1. Center Filed

The Center Field represents the distribution of intensity which describe the distance between the pixels and center point of objects. According to (1), If a pixel is inside of an object or on the edge, $F < 0$.

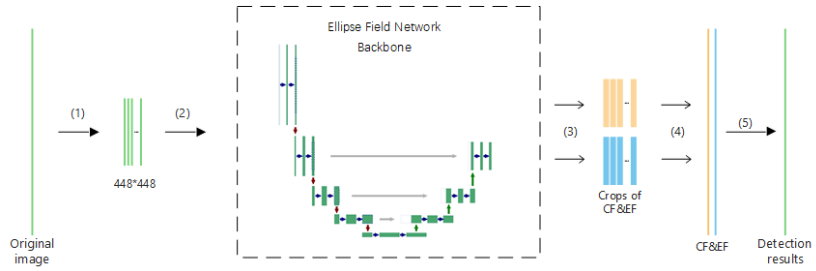


Figure 2. The Model. Our system (1) crops the images into pieces, (2) feed it into the network, (3) output with two-branch: Center Field and Edge Field. (4) Reconstruct the crops using Spray Painting and upsample them using bilinear interpolation. Finally, (5) Ellipse Region Fitting algorithm generate accurate detection results.

Based on that, We define the Center Field intensity $G_{c,p}$ of pixel (x, y) is calculated as in (2), where α is a coefficient we call center field decay index and set the default value 2.5, it determines the decay rate. When objects are densely packed, some points may be in more than one ellipse, we choose the ellipse which has minimum distance with the point. Intensity decays from 1 at the center of an object to $e^{-\alpha}$ at the edge with a certain rate. In areas containing no objects, the intensity is 0. Fig.3(a) shows the intensity distribution.

$$G_{c,p} = \begin{cases} e^{-\alpha F}, & \text{if } F \leq 0 \\ 0, & \text{else} \end{cases} \tag{2}$$

2.2. Edge Field

Similarly, the Edge Field represents the distribution of edge intensity which describe the distance between the pixels and edge of objects. According to (1), the sufficient and necessary conditions of the pixels on an edge is $F = 0$. Based on that, We define the Center Field intensity $G_{e,p}$ of pixel (x, y) is calculated as in (3). Theoretically, the edge of an object is an elliptic boundary formed by a sequence of connected pixels. In other words, the edge is pretty slim, which make it difficult for detection. To reduce the impact of that, we define a parameter ω called edge width to adjust the width of the edges. We set the default value 0.1. The visualization of the edge field is shown in Fig.3(b)

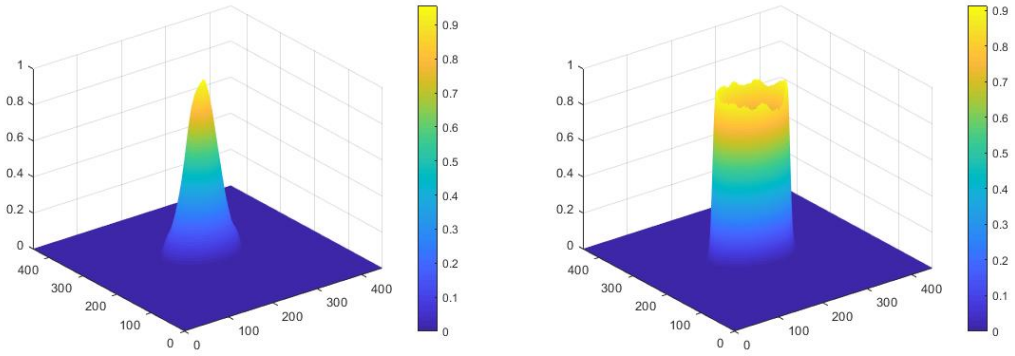
$$G_{e,p} = \begin{cases} 1, & \text{if } |F| < \omega \\ 0, & \text{else} \end{cases} \tag{3}$$

2.3. The process of Training

Because the small object is easy to be submerged, we give more weight to small objects. When network is training, We first get the area $A_{obj(p)}$ of the rectangle to which the pixel belongs, and then set the weight according to the reciprocal of the area size, as in (4), where ϵ is a bias, we set it to 0.1 by default.

$$\lambda_{c,p} = \lambda_{e,p} = \frac{1}{\epsilon + A_{obj(p)}} \tag{4}$$

Finally, the output layer scans the whole output array to determine the distance between each pixel and each ellipse, and then get the loss as in (5). Where p represents the pixel of the images, $v_{c,p}$ and $v_{e,p}$ are the center intensity and edge intensity predicted by the network, $G_{c,p}$ and $G_{e,p}$ are the ground truth. In this way, the impact of object size can be reduced to some extent.



(a) (b)

Figure 3. (a) is the visualization of a center intensity in image space. (b) is the visualization of an edge intensity in image space.

$$\begin{aligned} Loss_{CF} &= \sum_{p} \lambda_{c,p}(v_{c,p} - G_{c,p})^2 \\ Loss_{EF} &= \sum_{p} \lambda_{e,p}(v_{e,p} - G_{e,p})^2 \\ Loss &= Loss_{softmax} + Loss_{CF} + Loss_{EF} \end{aligned} \tag{5}$$

3. Post-process

3.1. Spray ing

Limited by the storage capacity of Memory, the input images cannot be too large. For the high-resolution aerial images, them must be cropped into pieces. There will be a lot of object split in junction when crops. To this end, we use a new method, it takes the advantage of the characteristic of EFN whose ouputs are fields. The process of traditional methods can be described as $NMS[\cup_i(Trans(nms(FPN(crop_i))))]$, it has high computational complexity and in the margin of patches, incomplete objects maybe missing or repeat. To improve this defect, we propose a unique method for the object field, we call it Spray Painting, the process of out method can be described as $ERF[mosaic_i(EFN(crop_i))]$ Fig.4 demonstrate the process. Concretely, we first generate the blocks with overlapped edge which set to 0.2 times the widths of the large image by default. Each junction may be composed by 2-4 cropped images, we use linear weighted algorithm, as in (6), to fuse them. It can get near seamless large image.

$$f(x,y) = \frac{\sum_i f_i(x,y)e_i(x,y)}{\sum_i e_i(x,y)} \tag{6}$$

where $f_i(x,y)$ is the value of field intensity, $e_i(x,y)$ is the shortest distance from the edge of the point mapped to the i-th cropped image at (x,y) of the large image.

3.2. Ellipse Region Fitting

ERF algorithm is to figure out the parameters of all ellipses according to the Center Field and Edge Field, and describes the output information of network with mathematical language, obtain oriented bounding boxes (see Fig.1(d), it is minimum enclosing rectangle of the ellipse). There are three main steps in this algorithm: acquiring initial center points, acquiring points on the edges, and figuring out the reliable parameters.

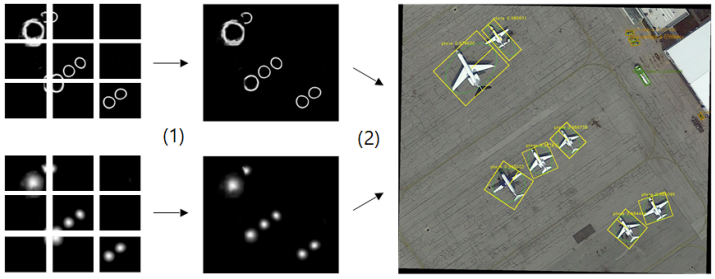
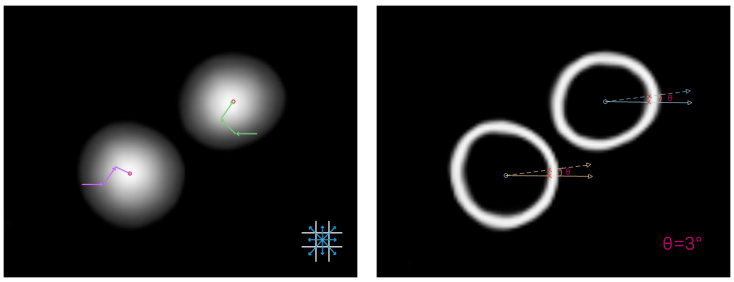


Figure 4. (1) is the Spray Painting and (2) is the Ellipse Region Fitting



(a) (b)

Figure 5. Solution procedure. (a) approximately demonstrate the way to find the center point of fields. (b) shows two points found by our method, can deduce the rest from this.

The first step is to acquire the initial coordinates of the center points according to the Center Field which represents the object intensity distribution. For each channel, we scan the elements sequentially. As shown in Fig.5(a), if the intensity $v_{c,p} \leq e^{-\alpha}$, we search for the maximum intensity in the eight pixel around the current one. If the maximum intensity is greater than $v_{c,p}$, we keep searching from the new pixel until there is no pixel greater and record its coordinates (x_c, y_c). After the processing of a channel, we get a group of local maximum coordinates of the pixels, which are the initial coordinates of the center points of a category object.

The second step is to acquire the points on the edges. Edge Field represents the edge intensity distribution. We start from the initial center points acquired from first step, producing a ray every three degrees from 0° to 360°. As shown in Fig.5(b), along with the rays, the $v_{e,p}$ jumps somewhere and the $v_{c,p}$ decays from the center point. We start from the center points and scan pixel along with the rays, if the $v_{c,p}$ of a pixel is less than $e^{-\alpha}$, or the $v_{e,p}$ is more than 0.4 greater than that of the prior pixel, we think it is one of the points on the edge. After the processing of each ray, we get 120 points on an edge, and use parametric equations to record these points, as in (7).

$$\begin{aligned} x_k &= x_c + t_k \cos \beta_k \\ y_k &= y_c + t_k \cos \beta_k \end{aligned} \tag{7}$$

where t_k is the length from the center point to the edge point, $\beta_k = k(\pi/60)$, $k = 0, 1, 2, \ldots, 119$.

The final step is to figure out the parameters. The elliptic equation is nonlinear that contains five parameters, as in (1), requires at least five points to solve. In the former steps, we get 120 edge points for each object. Accordingly, we pick five points and employ the Levenberg-Marquardt (LM) method[25] to figure out the parameters. Since the initial value of the central point is selected by the local maximum of the central field, the deviation will not be large, we set a center constrained condition: $\lambda(x_0^2 + y_0^2) = 0$ (λ is a coefficient, we ues λ=2000). The constrained condition and the elliptic function, for each variable to find partial derivative, then we get Jacobian matrix. Generally, the equation established by the edge point with higher strength is more reliable and should be given greater weight, so we use $v_{e,p}$ (the value of edge field) of each edge point to compose a weight diagonal matrix, as in

(8). Based on the above, we get the formula, as in (9). This formula can calculate the correction of five parameters. We can get reliable results by iterative correction within a certain threshold.

$$\Lambda = \begin{bmatrix} v_{e,p_1} & & & \\ & \ddots & & \\ & & v_{e,p_n} & \\ & & & \frac{1}{n}\sum_{i=1}^{n} v_{e,p_i} \end{bmatrix} \tag{8}$$

$$\Lambda \cdot \begin{bmatrix} \frac{\partial F_1}{\partial a} & \frac{\partial F_1}{\partial b} & \frac{\partial F_1}{\partial x_0} & \frac{\partial F_1}{\partial y_0} & \frac{\partial F_1}{\partial \theta} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \frac{\partial F_n}{\partial a} & \frac{\partial F_n}{\partial b} & \frac{\partial F_n}{\partial x_0} & \frac{\partial F_n}{\partial y_0} & \frac{\partial F_n}{\partial \theta} \\ 0 & 0 & 2\lambda x_0 & 2\lambda y_0 & 0 \end{bmatrix} \cdot \begin{bmatrix} \Delta a \\ \Delta b \\ \Delta x_0 \\ \Delta y_0 \\ \Delta \theta \end{bmatrix} = \Lambda \cdot \begin{bmatrix} -F_1 \\ \vdots \\ -F_n \\ -\lambda(x_0^2 + y_0^2) \end{bmatrix} \tag{9}$$

To improve the fault tolerance, we use the Algorithm 1 to optimize. a and b are within a certain range in a specific dataset. Semi-axis, such as the DOTA is $(0.001, 0.7)$, beyond the range is false positive and should be eliminated.

Algorithm 1: The process of optimization

Input: N Points
Output: (a, b, x_0, y_0, θ)

set the parameters: $count = 0$, $max\_inlier = 0$, $\xi = 0.05$;
repeat
  Randomly choose five from N points;
  solve (a, b, x_0, y_0, θ) with the five points;
  figure out the errors $E_i = \|F_i - 1\|^2$ for N points with Equation;
  n = The number of errors E_i that is less than ξ;
  if $max\_inlier < n$ **then** $max\_inlier = n$;
  $p = [1 - (max\_inlier/N)^5]^t$;
  $count = count + 1$;
until $p < 0.0001$;
for all inliers, solve (a, b, x_0, y_0, θ);

4. Experiments

Tests are implemented by Darknet[26] on a PC with Nvidia GeForce RTX 2080Ti GPU and 16G memory. We mainly test our method with aerial images in DOTA[5] and HRSC2016[27] as a supplement. To verify the generality of EFN we also test on the natural images in VOC2012[28]. Note the experiments on the VOC2012 are regress with horizontal box and no image crops due to the small size and common perspective. The EFN is orthogonal to specific network backbone, we use U-Net[29] and FCN[30] as backbone for the experiments section.

4.1. Datasets

DOTA[5] is the largest dataset for object detection in aerial images with oriented bounding box annotations. It contains 2806 images with different sizes. There are objects of 15 categories, including Plane, Bridge, Ship, Harbor, Baseball diamond, Ground track field, Small vehicle, Large vehicle, Tennis court, Basketball court, Storage tank, Soccer-ball field, Roundabout, Swimming pool, and Helicopter, containing 188282 annotated instances. The dataset provides the evaluation server. DOTA provides annotation labels in the manner of pixel coordinates of oriented bounding boxes. To make it fit with our method, we convert the original labels to elliptic equation parameters. DOTA has divided all images into a training set (1411 images), validation set (485 images) and testing set (937 images). We crop a

series of 448*448 patches from the training set and do limited data augmentation for categories with a small number of samples. Specifically, we do stochastic translation and rotation. Besides, we resize objects according to their sizes, large objects are randomly scaled-down, small objects are randomly scaled up. With all these processes, we obtain 81917 patches for training, which are much less than that in the official baseline implements (150,342 patches). For validation, we also crop the original images into 448*448 patches.

High resolution ship collections 2016 (HRSC2016)[27] is a data set used for scientific research, all of the images in HRSC2016 were collected from Google Earth. The ships were annotated with rotated bounding box on three levels including object class, class category and class type. It contains 1061 images and more than 20 categories of ships in various appearances. The image size ranges from 300*300 to 1500*900. The training, validation and test set include 436 images, 181 images and 444 images, respectively.

4.2. Implementation details

The visualization of detection results of DOTA and HRSC2016 is shown in Fig. 6 and Fig. 8. Apart from detecting patches, our detector can also obtain accurate detection results in whole images using the spary printing, as is shown in Fig. 7. It indicates that our detector can precisely locate and identify the instances in scenes with ellipses and oriented bounding boxes.

4.3. Comparison with the state-of-the-art

Table. 1 shows the comparison of accuracy in DOTA. EFN outperforms the official baseline FR-O[5] by 21.14 points and surpasses all the state-of-the-art methods, which proves that our method is more suitable for oriented object detection in aerial images. We argue that there are two reasons: 1) traditional frameworks first generate proposal boxes, then analyze boxes one by one to discriminate whether a box is correct. It is easy to make wrong discrimination. EFN predicts fields, the intensity of the object region is high while that of the no-object region is low, which is more similar to how the human visual system works. 2) traditional frameworks regress bounding boxes. However, usually objects only occupy small parts of images, which makes these models more nonuniform. Compared to them, EFN can better model the distribution of the objects in aerial images by regressing fields. Although our method was originally designed for aerial image object detection, experiments show that it also works well in conventional images, and can be used in a variety of scenes with great potential. We also make a comparison on memory, as shown in Table. 3. traditional frameworks use deep backbones like ResNet[31] to extract features and rely on pre-trained models. As a consequence, such models are memory-consuming. EFN uses U-Net as backbone, which does not rely on pre-trained model, and have a relatively shallow backbone. So the model much more memory-saving.

Table. 2 shows the comparison of accuracy in HRSC2016. The HRSC2016 contains a lot of thin and long ship instances with arbitrary orientation. Based on our proposed method, the mAP can reach 86.6, which is in line with the current state-of-the-art methods. As the visualization of detection result shows, there are many many ships arrange closely, they are long and narrow, the horizontal rectangles are hard to distinguish. While our EFN can effectively deal with this situation, which proves the strong stability of our method.

4.4. Ablation study

In our research, we find that many factors have an impact on the performance of EFN. Here we have ablation studies of various aspects on DOTA. Specifically, we examine the cases of (1) training the models with different input sizes, (2) using different backbones to construct EFN, whether to add batch normalization (BN)[39] after convolutional layers and setting different batch sizes, (3) training the models with different center field decay index α and edge width ω.

Table. 4 shows the accuracy and average testing speed comparison between different input sizes of EFN. Compared to EFN-112 and EFN-224, EFN-448 outperforms them on accuracy to a great extent.

Table 1. Comparison with the state-of-the-art method on DOTA1.0. The short names for categories are defined as: PL-Plane, BD-Baseball diamond, BR-Bridge, GTF-Ground field track, SV-Small vehicle, LV-Large vehicle, SH-Ship, TC-Tennis court, BC-Basketball court, ST-Storage tank, SBF-Soccer-ball field, RA-Roundabout, HB-Harbor, SP-Swimming pool, and HC-Helicopter.

method	mAp	PL	BD	BR	GTF	SV	LV	SH	TC	BC	ST	SBF	RA	HB	SP	HC
FR-O [5]	54.13	79.42	77.13	17.70	64.05	35.30	38.02	37.16	89.41	69.64	59.28	50.30	52.91	47.89	47.40	46.30
RRPN [32]	61.01	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
R2CNN [33]	60.67	88.52	71.20	31.66	59.30	51.85	56.19	57.25	90.81	72.84	67.38	56.69	52.84	53.08	51.94	53.58
R-DFPN [34]	57.94	80.92	65.82	33.77	58.94	55.77	50.94	54.78	90.33	66.34	68.66	48.73	51.76	55.10	51.32	35.88
DFPN [35]	62.29	81.25	71.41	36.53	67.44	61.16	50.91	56.60	90.67	68.09	72.39	55.06	55.60	62.44	53.35	51.47
LR-O [12]	58.31	81.06	76.81	27.22	69.75	38.99	39.07	38.30	89.97	75.53	65.74	63.48	59.37	48.11	56.86	44.46
LROFPN [12]	66.90	88.02	76.99	36.70	72.54	70.15	61.79	75.77	90.14	73.81	85.04	56.57	62.63	53.30	59.54	41.91
DPSRP [12]	63.89	81.18	77.42	35.48	70.41	56.74	50.42	53.56	89.97	79.68	76.48	61.99	59.94	53.34	64.04	47.76
RT [12]	67.74	88.53	77.91	37.63	74.08	66.53	62.97	66.57	90.5	79.46	76.75	59.04	56.73	62.54	61.29	55.56
RT-FPN [12]	69.56	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
SCRDet[13]	72.61	89.98	80.65	52.09	68.36	68.36	60.32	72.41	90.85	87.94	86.86	65.02	66.68	66.25	68.24	65.21
EFN	**75.27**	93.44	76.38	37.05	78.47	88.75	89.96	90.58	90.91	94.89	78.02	63.87	57.41	40.73	95.49	53.13

Table 2. Comparisons with the state-of-the-art methods on HRSC2016

method	CP[36]	BL2[36]	RC1[36]	RC2[36]	R^2PN[37]	RRD[38]	LRT[12]	**EFN**
mAP	55.7	69.6	75.7	75.7	79.6	84.3	86.2	**86.6**

EFN-576 achieves higher mAP, but the promotion is modest and the testing speed is much lower. Therefore, we think that setting the input size to 448*448 is most appropriate.

We choose two backbones, the FCN[30] and the U-Net. Based on the U-Net, we train eight models using different batch sizes, four with BN and four without BN. The comparison is displayed in Table 5. From the table, we can see that the U-Net achieves higher mAP. The FCN is a sequential architecture while there is a connection between encoder and decoder in the U-Net, such connection ensures that the gradient information can be passed directly to the upper layers, which helps the gradients propagate and improves the network's performance. Though the performance of the FCN backbone is inferior to the U-Net, it outperforms many prior works. This indicates that the EFN is compatible with different backbones and improve detection performance.

The U-Net with BN performs better than without that, for BN can reduce overfitting, avoid gradient vanishing and accelerate training. To a certain extent, larger batch size results in better performance but consumes more memory. Further experiments find that training with small batch size in the preliminary phase then with larger batch size later is an effective strategy. The small batch size makes for faster convergence while the larger batch size makes for fine optimization.

There are two important parameters in the training phase, center field decay index α and edge width ω. The two parameters have significant impacts on the performance. To find out the best values, we train models with a set of values. Table 6 shows the comparison of models trained with different α and ω. Both of them should be set to appropriate values. If the value of α is too low, the CF decays suddenly from center to edge, which may cause the wrong object center found in the ERF. The too-high value of α makes the CF decays rapidly, leading to difficulty for detecting object center. A low value of ω will make the EF inconspicuous. In this case, the detection of points on edge will be inaccuracy and some points may be left out. A high value of ω may cause edge overlap of adjacent objects, which will

Table 3. Comparison on memory.

Method	**Backbone**	**mAP**	**Param**
LR-O[12]	ResNet101	58.3	273MB
DPSRP[12]	ResNet101	63.89	273.2MB
RT[12]	ResNet101	67.74	273MB
EFN	U-Net	75.27	73MB

Table 4. Accuracy and average testing speed comparison between different input sizes of EFN.

Input size	mAP	EFN	ERF	Testing speed
EFN-112	71.63	2ms	32ms	34ms
EFN-224	75.18	9ms	83ms	92ms
EFN-448	75.27	39ms	204ms	243ms
EFN-576	75.30	68ms	374ms	442ms

Table 5. Comparison of models trained with different configures. All the models are trained with input size 224*224. "√" means adding batch normalization after convolutional layers.

Backbone	W/BN	Batch size	mAP
FCN	√	64	74.74
U-Net	√	64	**75.18**
	√	16	75.05
	√	8	75.12
	√	4	74.97
		64	74.48
		16	74.56
		8	74.76
		4	74.52

bring about disturbance to the edge distinction of different objects. In our experiments, we find that setting α to 2.5 and ω to 0.1 achieves high performance, which may serve as the default values.

Table 6. Comparison of models trained with different α and ω. All the models are trained with input size of 224*224. The batch size is set to 64.

Backbone	α	ω	mAP
FCN	2.5	0.1	74.74
U-Net	**2.5**	**0.1**	**75.18**
	1	0.1	72.02
	5	0.1	72.23
	2.5	0.3	75.16
	2.5	0.05	74.85

4.5. Experiments on Natural Images

To verify the universality of our model, we further test the proposed techniques on generic datasets: PASCAL VOC 2012 challenge[28]. We use VOC2007 train+val+test sets (9963 images) and VOC2012 train+val sets (11540 images) train the network and then use the VOC2012 test sets to test. Unlike the aerial images, the images of VOC are in the low altitude perspective and low resolution, we no longer crop the images and predict the angle of the object. In other words, we only predict four parameters, predefine $\theta = 0$, as shown in (10). Table. 7 shows the results. We achieve 84.7% mAP. The visualization of detection results is shown in Fig. 9. Compare to the state-of-the-art methods, EFN which specialize in different application scenarios still gets a not bad scores.

$$F(x, y; x_0, y_0, a, b) = \frac{(x - x_0)^2}{a^2} + \frac{(y - y_0)^2}{b^2} - 1 \tag{10}$$

5. Conclusion

In this paper, we proposed a novel method for detecting objects in aerial images. Unlike typical region proposal method, We introduced the concept of field into networks. We remold a common network to calculate Center Field and Edge Field, and use the robust Ellipse Region Fitting algorithm to identify object precisely. Only the first step is related to the learning data, the second step can be applied to any target, which greatly reduces the difficulty of network training. Experiments on the

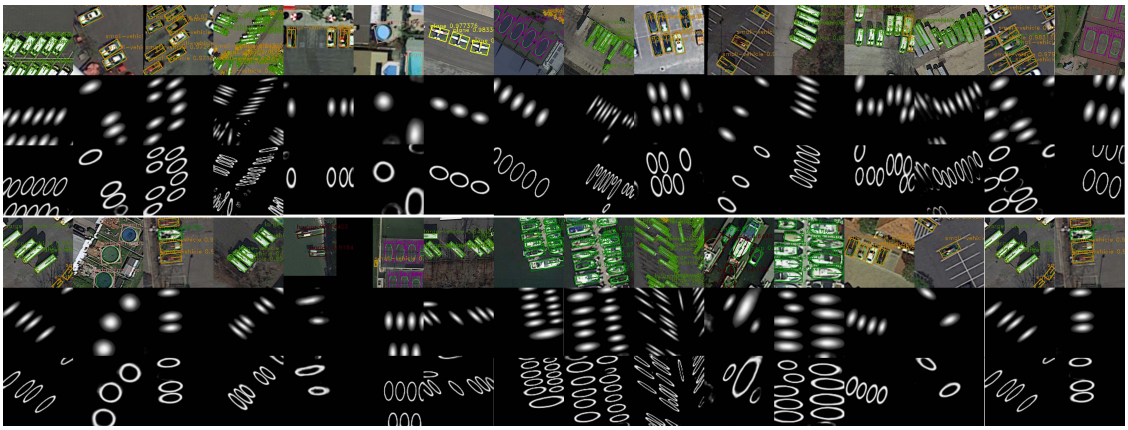


Figure 6. Visualization of detection results in patches of the DOTA testing set. From top to bottom are two groups of detection results, center fields and edge fields. For the images with multiple categories which correspond to several object fields, we only demonstrate one of them.

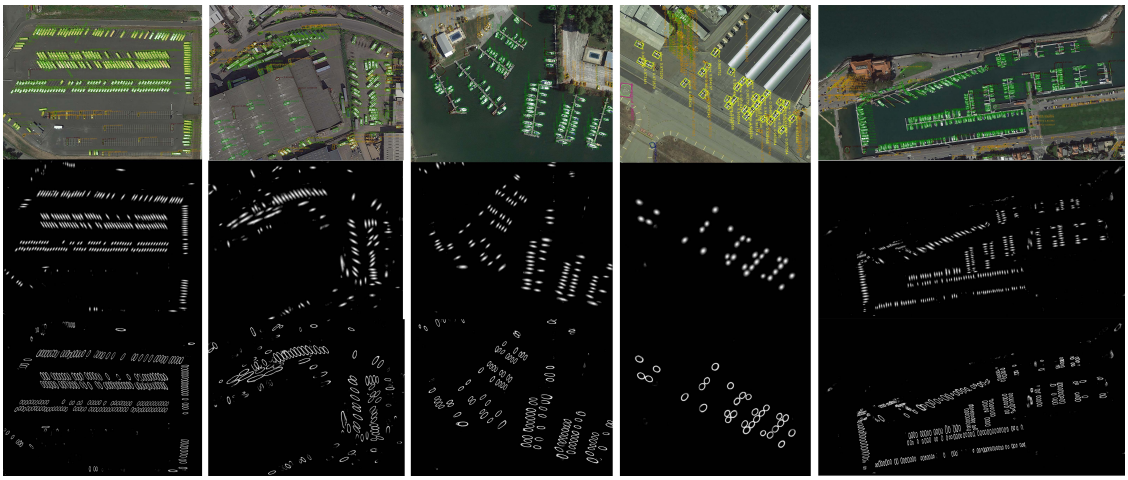


Figure 7. Detection results in whole images of DOTA testing set. All of them have high resolution and cover a large area.

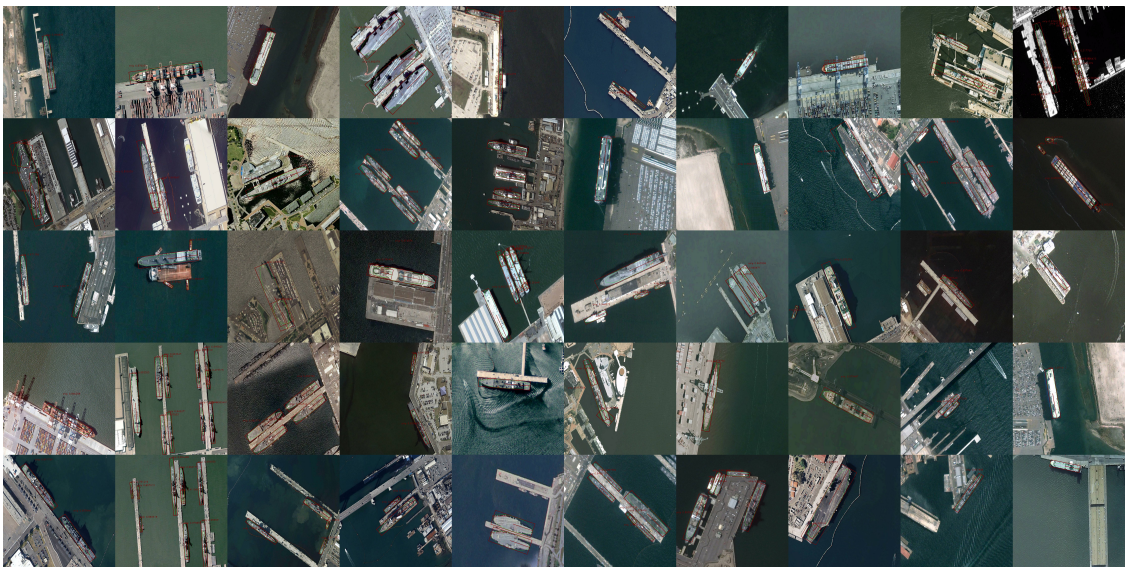


Figure 8. Visualization of detection results from EFN in HRSC2016.

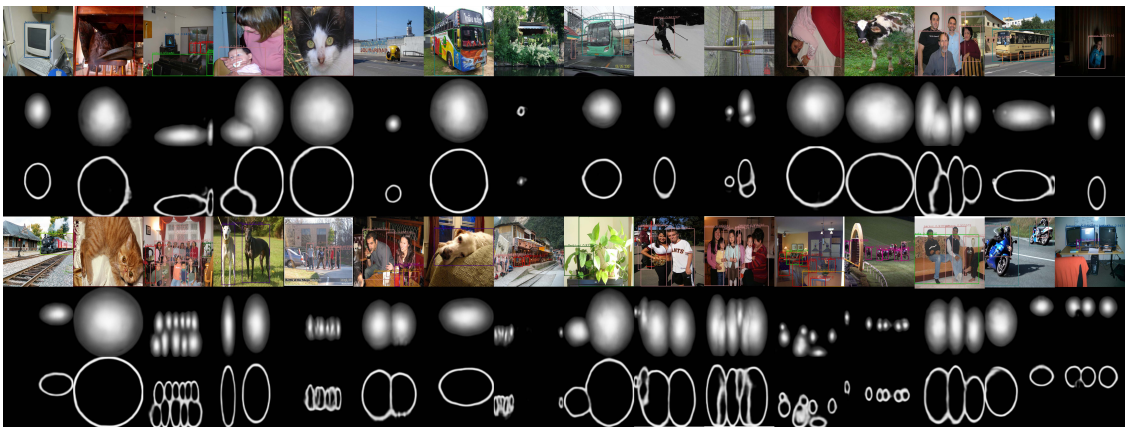


Figure 9. Visualization of detection results in patches of the PASCAL VOC testing set. From top to bottom are two groups of detection results, center fields and edge fields. For the images with multiple categories which correspond to several object fields, we only demonstrate one of them.

Method	Backbone	VOC 2012 test mAP(%)
Faster R-CNN[14]	VGG-16	75.9
OHEM++[40]	VGG-16	80.1
R-FCN[41]	ResNet-101	82.0
SSD300[8]	VGG-16	79.3
SSD512[8]	VGG-16	82.2
RefineDet320[42]	VGG-16	82.7
RefineDet512[42]	VGG-16	85.0
PSPNet[43]	ResNet50	85.4
EFN	FCN	**83.4**
EFN	U-Net	**84.7**

Table 7. Comparison with prior works on VOC2012, all methods are trained on VOC 2007 and VOC 2012 trainval sets plus VOC 2007 test set, and tested on VOC 2012 test set.

DOTA and HRSC2016 dataset shows that EFN outperforms many prior methods both on speed and accuracy. And our test on VOC2012 verify the generality of our techniques. Besides, we do ablation studies, discuss how different factors influence the performance of EFN and find out an appropriate way to train it. In the future, we will do more research on EFN and continuously improve our method.

Author Contributions: conceptualization, J. L.; methodology, J. L.; software, J. L.; validation, J. L.; formal analysis, J. L. and H. Z.; writing–original draft preparation, J. L.; writing–review and editing, J. L.; visualization, J. L.; supervision, J. L.; project administration, J. L.

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. Cheng, G.; Zhou, P.; Han, J. Learning Rotation-Invariant Convolutional Neural Networks for Object Detection in VHR Optical Remote Sensing Images. *IEEE Transactions on Geoscience and Remote Sensing* **2016**, *54*, 7405–7415. doi:10.1109/TGRS.2016.2601622.
2. Long, Y.; Gong, Y.; Xiao, Z.; Liu, Q. Accurate object localization in remote sensing images based on convolutional neural networks. *IEEE Transactions on Geoscience and Remote Sensing* **2017**, *55*, 2486–2498.
3. Wang, G.; Wang, X.; Fan, B.; Pan, C. Feature extraction by rotation-invariant matrix representation for object detection in aerial image. *IEEE Geoscience and Remote Sensing Letters* **2017**, *14*, 851–855.
4. Deng, Z.; Sun, H.; Zhou, S.; Zhao, J.; Zou, H. Toward fast and accurate vehicle detection in aerial images using coupled region-based convolutional neural networks. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* **2017**, *10*, 3652–3664.
5. 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. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 3974–3983.
6. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. Proceedings of the IEEE conference on computer vision and pattern recognition, 2014, pp. 580–587.
7. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 779–788.
8. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. Ssd: Single shot multibox detector. European conference on computer vision. Springer, 2016, pp. 21–37.
9. Liu, Z.; Hu, J.; Weng, L.; Yang, Y. Rotated region based CNN for ship detection. 2017 IEEE International Conference on Image Processing (ICIP). IEEE, 2017, pp. 900–904.
10. Liu, K.; Mattyus, G. Fast multiclass vehicle detection on aerial images. *IEEE Geoscience and Remote Sensing Letters* **2015**, *12*, 1938–1942.
11. Cheng, G.; Zhou, P.; Han, J. Rifd-cnn: Rotation-invariant and fisher discriminative convolutional neural networks for object detection. Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 2884–2893.
12. Ding, J.; Xue, N.; Long, Y.; Xia, G.S.; Lu, Q. Learning roi transformer for oriented object detection in aerial images. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 2849–2858.

13. Yang, X.; Yang, J.; Yan, J.; Zhang, Y.; Zhang, T.; Guo, Z.; Sun, X.; Fu, K. Scrdet: Towards more robust detection for small, cluttered and rotated objects. Proceedings of the IEEE International Conference on Computer Vision, 2019, pp. 8232–8241.
14. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. Advances in neural information processing systems, 2015, pp. 91–99.
15. Cai, Z.; Vasconcelos, N. Cascade r-cnn: Delving into high quality object detection. Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 6154–6162.
16. Cao, Z.; Hidalgo, G.; Simon, T.; Wei, S.E.; Sheikh, Y. OpenPose: realtime multi-person 2D pose estimation using Part Affinity Fields. *arXiv preprint arXiv:1812.08008* **2018**.
17. Newell, A.; Yang, K.; Deng, J. Stacked hourglass networks for human pose estimation. European conference on computer vision. Springer, 2016, pp. 483–499.
18. Brahmbhatt, S.; Christensen, H.I.; Hays, J. StuffNet: Using 'Stuff'to improve object detection. 2017 IEEE Winter Conference on Applications of Computer Vision (WACV). IEEE, 2017, pp. 934–943.
19. Shrivastava, A.; Gupta, A. Contextual priming and feedback for faster r-cnn. European conference on computer vision. Springer, 2016, pp. 330–348.
20. Gidaris, S.; Komodakis, N. Object detection via a multi-region and semantic segmentation-aware cnn model. Proceedings of the IEEE international conference on computer vision, 2015, pp. 1134–1142.
21. Bell, A.J.; Sejnowski, T.J. The "independent components" of natural scenes are edge filters. *Vision research* **1997**, *37*, 3327–3338.
22. Olshausen, B.A.; Field, D.J. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature* **1996**, *381*, 607–609.
23. Luo, W.; Li, Y.; Urtasun, R.; Zemel, R. Understanding the effective receptive field in deep convolutional neural networks. Advances in neural information processing systems, 2016, pp. 4898–4906.
24. Soong, T.T. *Fundamentals of probability and statistics for engineers*; John Wiley & Sons, 2004.
25. Ma, C.; Jiang, L. Some research on Levenberg–Marquardt method for the nonlinear equations. *Applied mathematics and Computation* **2007**, *184*, 1032–1040.
26. Redmon, J. Darknet: Open source neural networks in c **2013**.
27. Liu, Z.; Wang, H.; Weng, L.; Yang, Y. Ship rotated bounding box space for ship extraction from high-resolution optical satellite images with complex backgrounds. *IEEE Geoscience and Remote Sensing Letters* **2016**, *13*, 1074–1078.
28. Everingham, M.; Van Gool, L.; Williams, C.K.I.; Winn, J.; Zisserman, A. The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results. http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html.
29. Ronneberger, O.; Fischer, P.; Brox, T. U-net: Convolutional networks for biomedical image segmentation. International Conference on Medical image computing and computer-assisted intervention. Springer, 2015, pp. 234–241.
30. Long, J.; Shelhamer, E.; Darrell, T. Fully convolutional networks for semantic segmentation. Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 3431–3440.
31. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770–778.
32. Ma, J.; Shao, W.; Ye, H.; Wang, L.; Wang, H.; Zheng, Y.; Xue, X. Arbitrary-oriented scene text detection via rotation proposals. *IEEE Transactions on Multimedia* **2018**, *20*, 3111–3122.
33. 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 preprint arXiv:1706.09579* **2017**.
34. 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 Sensing* **2018**, *10*, 132.
35. Yang, X.; Sun, H.; Sun, X.; Yan, M.; Guo, Z.; Fu, K. Position detection and direction prediction for arbitrary-oriented ships via multitask rotation region convolutional neural network. *IEEE Access* **2018**, *6*, 50839–50849.
36. Liu, Z.; Wang, H.; Weng, L.; Yang, Y. Ship rotated bounding box space for ship extraction from high-resolution optical satellite images with complex backgrounds. *IEEE Geoscience and Remote Sensing Letters* **2016**, *13*, 1074–1078.

37. Zhang, Z.; Guo, W.; Zhu, S.; Yu, W. Toward arbitrary-oriented ship detection with rotated region proposal and discrimination networks. *IEEE Geoscience and Remote Sensing Letters* **2018**, *15*, 1745–1749.
38. Liao, M.; Zhu, Z.; Shi, B.; Xia, G.s.; Bai, X. Rotation-sensitive regression for oriented scene text detection. Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 5909–5918.
39. Normalization, B. Accelerating deep network training by reducing internal covariate shift. *CoRR.–2015.–Vol. abs/1502.03167.–URL: http://arxiv. org/abs/1502.03167* **2015**.
40. Shrivastava, A.; Gupta, A.; Girshick, R. Training region-based object detectors with online hard example mining. Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 761–769.
41. Dai, J.; Li, Y.; He, K.; Sun, J. R-fcn: Object detection via region-based fully convolutional networks. Advances in neural information processing systems, 2016, pp. 379–387.
42. Zhang, S.; Wen, L.; Bian, X.; Lei, Z.; Li, S.Z. Single-shot refinement neural network for object detection, 2018.
43. Zhao, H.; Shi, J.; Qi, X.; Wang, X.; Jia, J. Pyramid scene parsing network. Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 2881–2890.

Sample Availability: Samples of the compounds ..... are available from the authors.