CONFIDENTIAL

BorderDet: Border Feature for Dense Object Detection

Han Qiu, Yuchen Ma , Zeming Li, Songtao Liu , and Jian Sun

yy/mm/dd

調和 太郎

北大発
認定ベンチャー企業
Hokkaido University Venture

株式会社
調和技研

Myself

- Name : Hossain Md Saddam
- Profile
  - July 2019, Joined Chowagiken Corporation as a ML Engineer.
  - I worked on both CV and NLP team.
- Today I am going to present a state of the art detection approach called borderDet.

Brief summary of the BorderDet

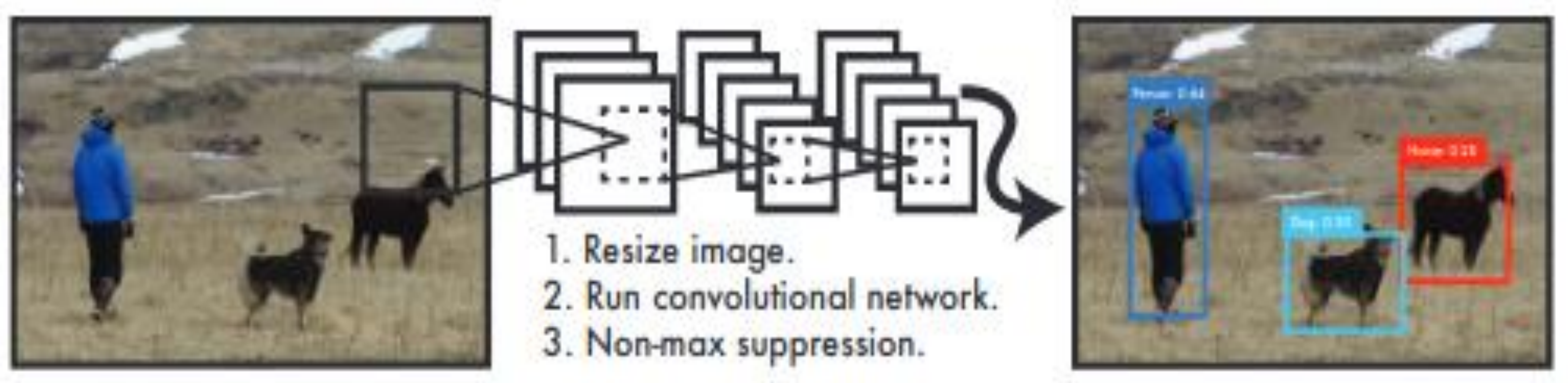
- There are two types of conventional approach for the detection task which are single-stage and two-stage detection.
- BorderDet added a module call BAM on both single-stage and two stage. Its achieved state of the art result.
- In this paper BorderDet proprosed a method which is called Border-Align to extract the border features which are extreme points of the border.
- With ResNet-50 backbone, this method improves single-stage detector FCOS by 2.8 AP gains (38.6 to 41.4). With the ResNeXt-101-DCN backbone, BorderDet obtains 50.3 AP, outperforming the existing state-of-the-art approaches

Established detection methods

Conventional Approach

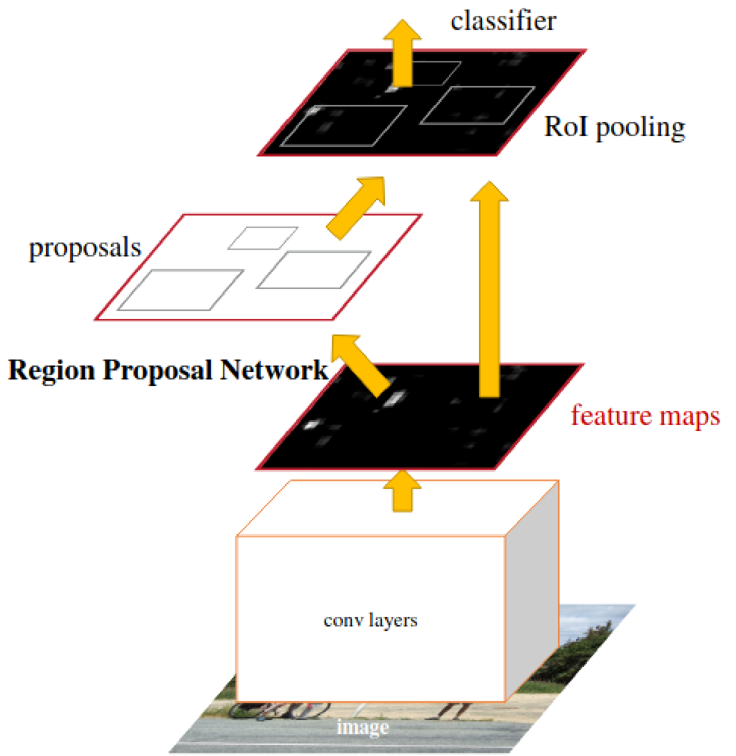
Densebox, YOLO, SSD, RetinaNet, and FCOS are the single-stage object detectors, have demonstrated the effectiveness to densely predict the classification and localization scores.

- YOLO object detection architecture



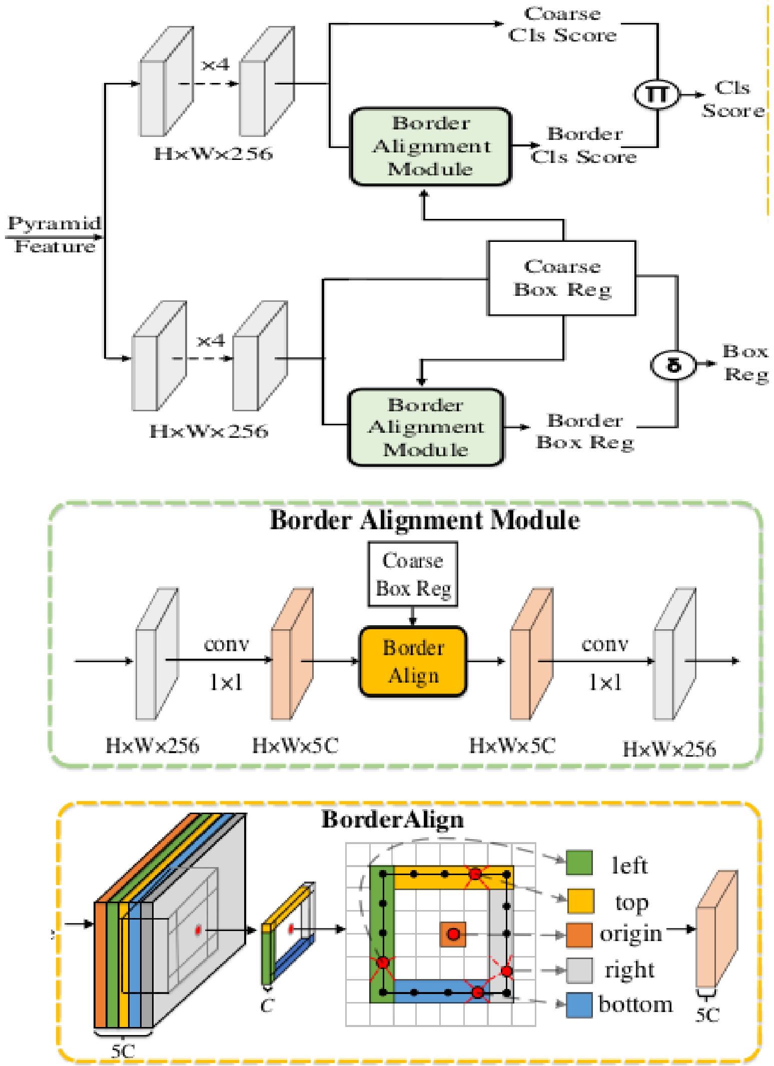
Conventional Approach

- For the two-stage, object Detectors, R-CNN series (R-CNN, Fast R-CNN, Faster R-CNN) adopt the region proposal network (RPN) that based on the sliding-window mechanism.
- To generate the initial proposals, and then a refinement stage that consists of a RoIAlign.
- R-CNN is performed to warp the feature maps of the region-of-interests (RoI) and generate the accurate predictions.



BorderDet(Single-Stage Architecture)

- Adopted a simple anchor-free object detector FCOS as baseline.
- Taking the pyramid feature maps as input. The BorderDet first predicts the coarse classification scores and coarse bounding box locations.
- The coarse bounding box locations and the feature maps are fed into the BAM to generate the feature maps which contain explicit border information.
- Finally, we apply a 1 × 1 convolutional layers to predict the border classification score and border locations.



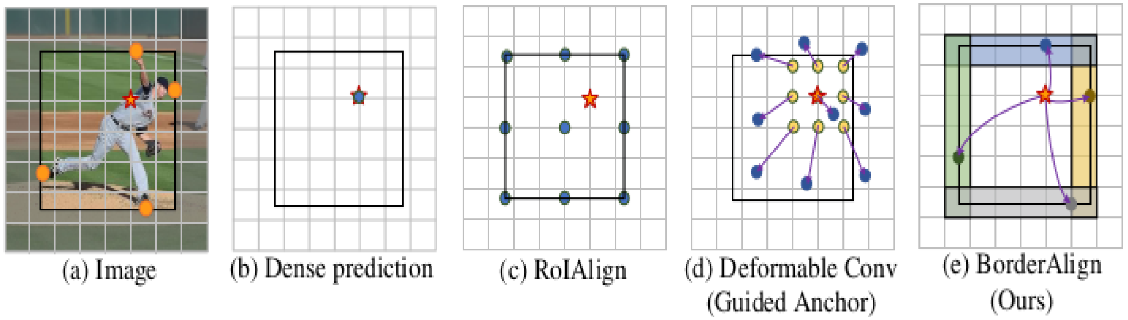
BorderDet Approach(Motivation)

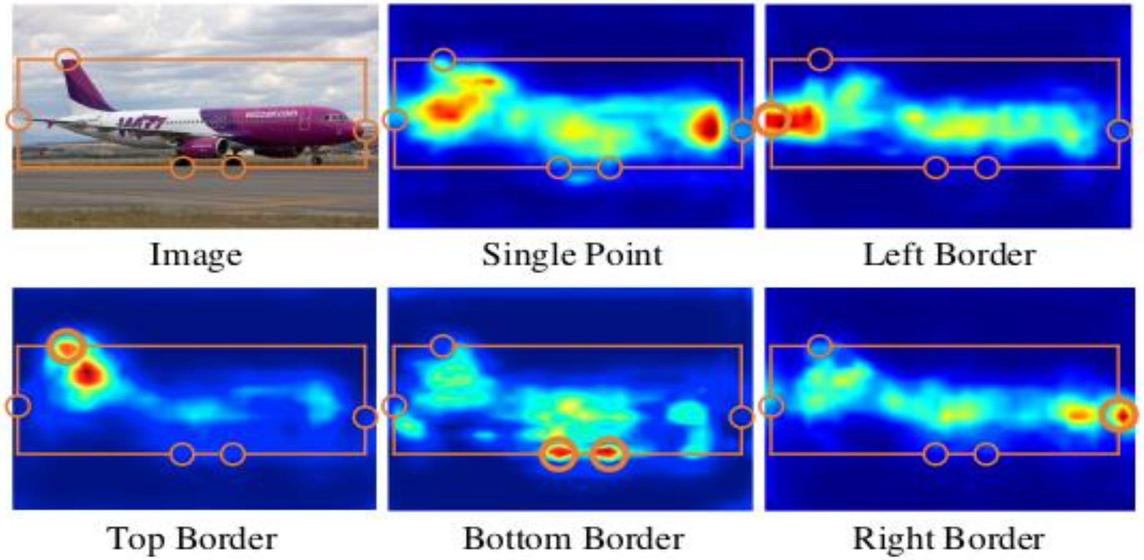
- As we can see in the table, only single point feature is not perform well compared to cosidering other features.
- It performs better than by 1.3 AP when consider region, border, middle border features.
- BorderDet comes into action behide this concepts.

F_{point}	F'_{point}	F_{region}	F_{border}	F_{middle}	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L	N
✓					38.6	57.2	41.7	23.5	42.8	48.9	0
✓	✓				38.9	57.7	42.1	23.7	43.1	49.3	1
✓	✓	✓			**39.9**	58.9	43.4	24.6	44.1	50.8	n^2+1
✓	✓		✓		39.6	58.5	43.2	24.2	43.8	50.4	$4n+1$
✓	✓			✓	**39.9**	58.7	43.4	24.8	44.0	50.4	$4+1$

Feature Extraction

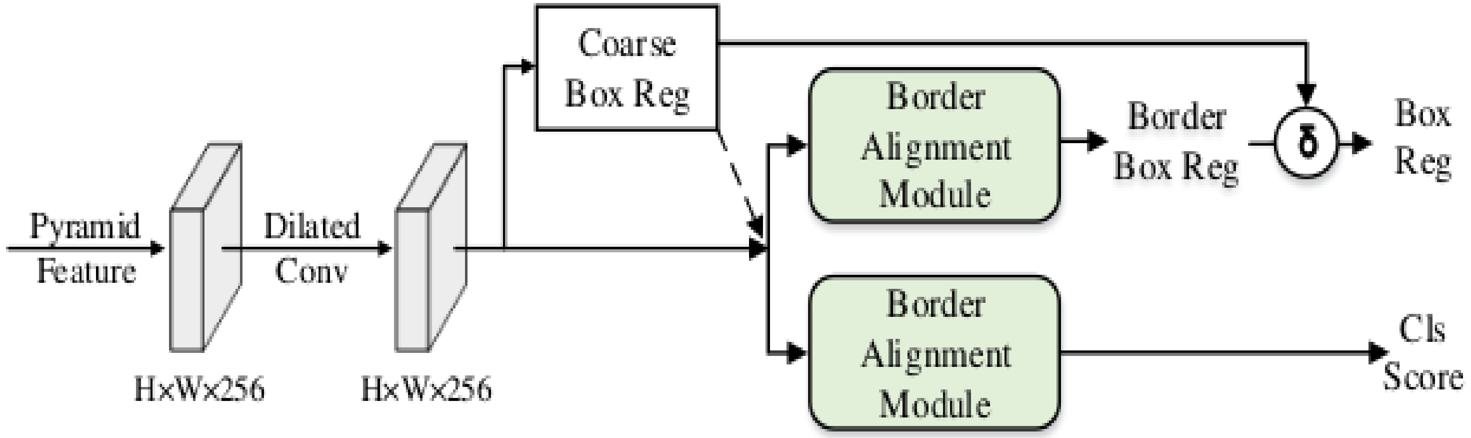
- First figure demonstrated different feature extraction strategy.
- Visualization of the border-sensitive feature maps. The orange circle on the border indicate the extreme points.





BorderRPN(two-stage)

- BorderDet method can also be served as a better proposal generator for the typical two-stage detectors.
- Added the border alignment module to RPN and denote the new structure as BorderRPN.



Loss function

- BorderDet defined training loss as follows

$$\mathcal{L} = \mathcal{L}_{cls}^{C} + \mathcal{L}_{reg}^{C} + \frac{1}{\mathcal{N}_{pos}} \sum_{x,y} \mathcal{L}_{cls}^{B}(\mathcal{P}^{B}, \mathcal{C}^{*}) + \mathcal{L}_{reg_{\{\mathcal{C}^{*}>0\}}}^{B}(\Delta, \Delta^{*}),$$

- In the implementation, focal loss and IoU loss are used as the classification loss and regression loss respectively, which are the same as FCOS.

Experiments

- BorderDet experiments are trained on COCO trainval35k set and evaluated on COCO minival set (5K images).
- As a standard settings, It used ResNet-50 with FPN as backbone network for all the experiments.
- With an initial learning rate of 0.01, BorderDet decrease it by a factor of 10 after 60k iterations and 80k iterations respectively.
- Weight decay of 0.0001 and momentum of 0.9 are used.
- Unless specified, the input images are resized to ensure their shorter edge being 800 and the longer edge less than 1333.

Experiments

- Below table demonstrated the difference between applying class-BAM and reg-BAM.
- With Border Alignment Module, the AP values are outperforming than without BAM.

Cls-BAM	Reg-BAM	AP	AP_{50}	AP_{60}	AP_{70}	AP_{80}	AP_{90}
		38.6	57.2	53.3	46.7	35.3	16.0
✓		39.7	58.4	54.8	48.5	36.2	15.9
	✓	39.7	57.3	53.3	47.3	36.9	18.6
✓	✓	**41.4**	**59.4**	**55.4**	**49.4**	**38.6**	**19.5**

Comparison with Other Feature Extraction Operators

- BorderAlign outperforms than other feature extraction operators by 1.0 AP at least.
- Below are the comparison of different feature extraction strategies.

Method	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L	fps
FCOS [21]	38.6	57.2	41.7	23.5	42.8	48.9	18.4
w/Iter-Box [5]	39.0	58.0	42.0	21.8	42.9	50.7	18.3
w/Adaptive Conv [23]	39.6	58.5	42.8	22.0	43.5	51.3	16.8
w/Deformable Conv [2]	39.5	58.5	42.9	22.0	43.5	52.0	16.8
w/RoIAlign [9]	40.4	58.6	43.6	22.6	44.1	53.1	12.6
w/**BorderAlign**	**41.4**	**59.4**	**44.5**	**23.6**	**45.1**	**54.6**	16.7

Generalization of BorderDet

- BorderDet can be easily integrated with the many popular object detectors, e.g. RetinaNet and FPN.
- To prove the generalization of the BorderDet, we first add the proposed border alignment module to the RetinaNet.
- BorderDet can consistently improve the RetinaNet by 2.3 AP.
- For the two-stage method FPN, our experiments show that the proposed BorderRPN gains 3.6 AP improvement.

Method	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
Retinanet [16]	36.1	55.0	38.4	19.1	39.6	48.2
BD-Retinanet	**38.4**	56.5	55.5	22.4	41.6	51.0
FPN [20]	37.1	58.7	40.3	21.1	40.3	48.6
BD-FPN	**40.7**	57.8	44.3	21.9	43.7	54.8

Comparisons with State-of-the-art Detectors

- The BorderDet, based on FCOS, ResNet-101 backbone, is the state-of-the-art methods in below table under standard setting and advanced setting.
- With the standard setting, the proposed BorderDet achieves an AP of 43.2.
- By adopting advanced settings, BorderDet reaches 50.3 AP, the state of the art among existing one-stage methods and two-stage methods.

Method	Backbone	Iter.	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
FPN [15]	ResNet-101-FPN	180k	36.2	59.1	39.0	18.2	39.0	48.2
Mask R-CNN [9]	ResNet-101-FPN	180k	38.2	60.3	41.7	20.1	41.1	50.2
Cascade R-CNN [1]	ResNet-101	280k	42.8	62.1	46.3	23.7	45.5	55.2
RefineDet512 [27]	Resnet-101	280k	41.8	62.9	45.7	25.6	45.1	54.1
RetinaNet [16]	ResNet-101-FPN	135k	39.1	59.1	42.3	21.8	42.7	50.2
FSAF [30]	ResNet-101-FPN	135k	40.9	61.5	44.0	24.0	44.2	51.3
FCOS [21]	ResNet-101-FPN	180k	41.5	60.7	45.0	24.4	44.8	51.6
FCOS-imprv [21]	ResNet-101-FPN	180k	43.0	61.7	46.3	26.0	46.8	55.0
CenterNet [3]	Hourglass-104	500k	44.9	62.4	48.1	25.6	47.4	57.4
BorderDet	ResNet-101-FPN	90k	43.2	62.1	46.7	24.4	46.3	54.9
BorderDet†	ResNet-101-FPN	180k	45.4	64.1	48.8	26.7	48.3	56.5
BorderDet†	ResNeXt-64x4d-101	180k	46.5	65.7	50.5	29.1	49.4	57.5
BorderDet†	ResNet-101-DCN	180k	47.2	66.1	51.0	28.1	50.2	59.9
BorderDet†	ResNeXt-64x4d-101-DCN	180k	48.0	67.1	52.1	29.4	50.7	60.5
BorderDet‡	ResNeXt-64x4d-101-DCN	180k	50.3	68.9	55.2	32.8	52.8	62.3

Details and Implementation

- Paper : https://arxiv.org/abs/2007.11056
- Implementation:https://github.com/Megvii-BaseDetection/BorderDet

Thank you!