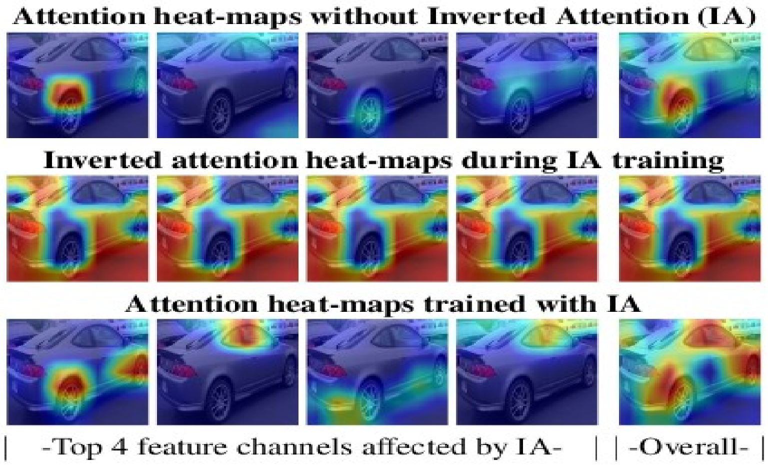
Improving Object Detection with Inverted Attention

INTRODUCTION

Improving object detectors against image defects such as occlusion, blur and noise is a critical step to deploy detectors in real applications. Object detector is improved using a highly efficient and fine-grain mechanism, Inverted Attention (IA). IA is implemented as a simple module added to the standard back-propagation operation. In every training iteration, IA computes the gradient of the feature maps produced at the feature extraction network using object classification scores, and iteratively invert the attention of the network. Different from the original detector network that only focuses on the small parts of objects, the neural network with IA puts more attention on complementary spatial parts of the original network, feature channels and even the context. The IA module only changes the network weights in training and does not change any computation in inference.


Attention heat-maps without Inverted Attention (IA)
Inverted attention heat-maps during IA training
Attention heat-maps trained with IA
| -Top 4 feature channels affected by IA- | | -Overall- |

INVERTED ATTENTION FOR OBJECT DETECTION

An Inverted Attention Generation Module is added to the R-CNN detection network, and operates on the ROI featrue maps. And it consists of only few simple operations such as pooling, threshold and element-wise product.

Inverted Attention Generation Module

The Inverted Attention Generation Module consists of two simple operations:
(1) Gradient-Guided Attention Generation: computing the the gradientsat feature maps, by back-warding only the classification score on the ground-truth category,
(2) Attention Inversion: reversing element values of the attention tensor to produce IA heat-maps.

INVERTED ATTENTION NETWORK (IAN)

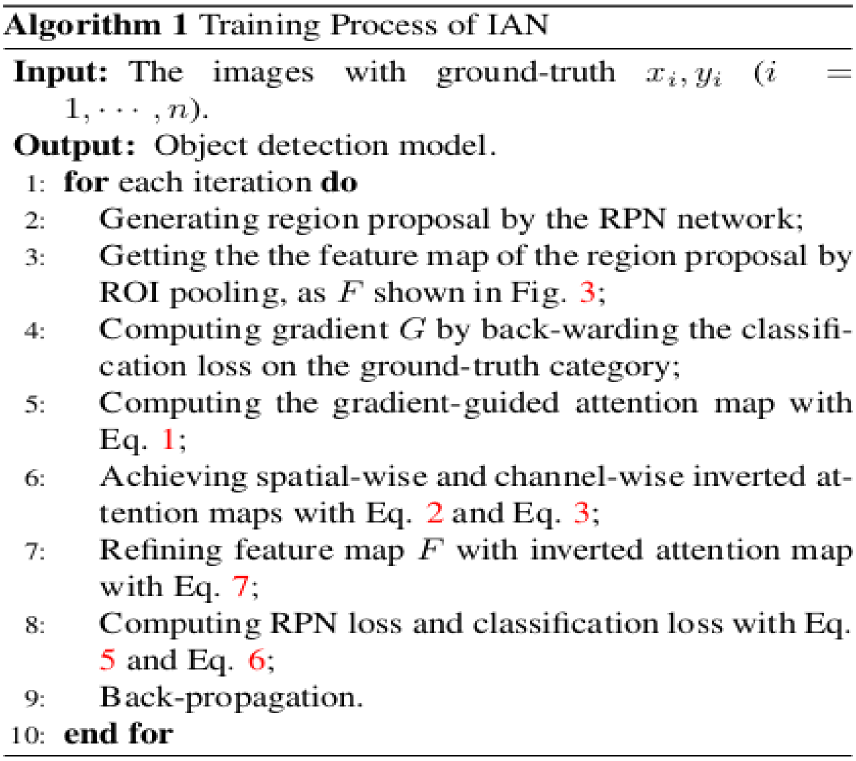
(IAN) is basically built by adding Inverted Attention Generation Module to the R-CNN based detection network, and operates on the ROI feature maps.
Given an input image, the backbone of the R-CNN framework, i.e., VGG or ResNet, takes the whole image as an input and produces feature maps. The region proposals are generated from these feature maps by region proposal network (RPN) or pre-computed region proposal candidates.
The R-CNN can be trained end-to-end by optimizing the following two loss functions:

$$L_{rpn} = L_{cross-entropy} + L_{rpn\_reg},$$

$$L_{rcnn} = L_{softmax} + L_{rcnn\_reg},$$

where L cross−entropy and L rpn reg are the cross-entropy loss and L1 loss for RPN network. L cross−entropy and L rpn reg are the softmax loss and L1 loss for RCNN network. L rpn + L rcnn are jointly optimized in the Faster-RCNN framework, and L rcnn is optimized in the Fast-RCNN framework.

ALGORITHM



Algorithm 1 Training Process of IAN

```
Input: The images with ground-truth $x_i, y_i$ $(i = 1, \cdots, n)$.
Output: Object detection model.
for each iteration do
    Generating region proposal by the RPN network;
    Getting the the feature map of the region proposal by ROI pooling, as $F$ shown in Fig. 3;
    Computing gradient $G$ by back-warding the classification loss on the ground-truth category;
    Computing the gradient-guided attention map with Eq. 1;
    Achieving spatial-wise and channel-wise inverted attention maps with Eq. 2 and Eq. 3;
    Refining feature map $F$ with inverted attention map with Eq. 7;
    Computing RPN loss and classification loss with Eq. 5 and Eq. 6;
    Back-propagation.
end for
```

In the backward stage, the gradient is computed by backpropagating the classification loss only on the ground-truth category, which is used for inverted attention generation module. With the generated Inverted Attention map, an element-wise product layer between feature maps and IA heat-maps is used for feature refinement, as

$$F_{new} = F. * A,$$

where . * indicates element-wise multiplication. The refinement is conducted at element-level, i.e., along both the spatial and channels dimensions of the feature maps.

ADVANTAGES

Highly Efficient and fine-grain mechanism : A highly efficient and fine-grain mechanism that significantly improves the object detection.

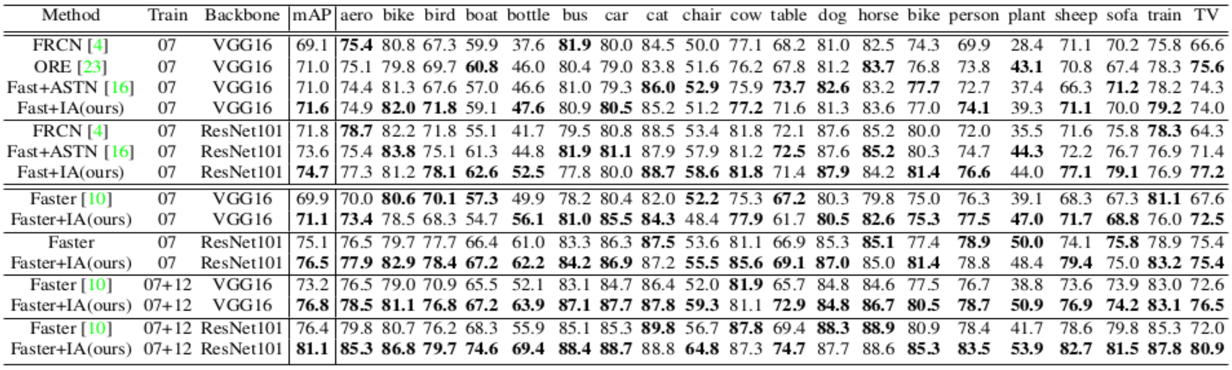
Boost the performance: IAN tends to boost the performance of the medium and large objects for Faster-RCNN. ResNet101 Faster-RCNN, ResNet101 SSD300 all of them sees the significant increase in the performance.

DISADVANTAGES

For SSD512, when Inverted Attention conducted on full features, IAN favors small and medium objects.

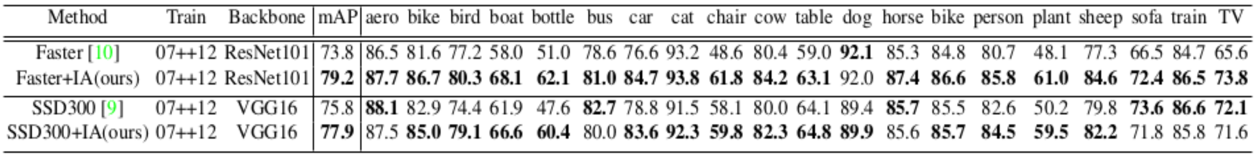
COMPARISON TO THE STATE OF ART

Extensive performance comparison on the PASCAL VOC 2007 with Fast-RCNN , Faster-RCNN and the state-of-the-art hard-sample generation approaches ORE and Fast-RCNN with ASTN. These approaches only provided results onFast-RCNN.



Method	Train	Backbone	mAP	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	bike	person	plant	sheep	sofa	train	TV
FRCN [4]	07	VGG16	69.1	**75.4**	80.8	67.3	59.9	37.6	**81.9**	80.0	84.5	50.0	77.1	68.2	81.0	82.5	74.3	69.9	28.4	71.1	70.2	75.8	66.6
ORE [23]	07	VGG16	71.0	75.1	79.8	69.7	**60.8**	46.0	80.4	79.0	83.8	51.6	76.2	67.8	81.2	**83.7**	76.8	73.8	**43.1**	70.8	67.4	78.3	**75.6**
Fast+ASTN [16]	07	VGG16	71.0	74.4	81.3	67.6	57.0	46.6	81.0	79.3	**86.0**	**52.9**	75.9	**73.7**	**82.6**	83.2	**77.7**	72.7	37.4	66.3	**71.2**	78.2	74.3
Fast+IA(ours)	07	VGG16	**71.6**	74.9	**82.0**	**71.8**	59.1	**47.6**	80.9	**80.5**	85.2	51.2	**77.2**	71.6	81.3	83.6	77.0	**74.1**	39.3	**71.1**	70.0	**79.2**	74.0
FRCN [4]	07	ResNet101	71.8	**78.7**	82.2	71.8	55.1	41.7	79.5	80.8	88.5	53.4	81.8	72.1	87.6	85.2	80.0	72.0	35.5	71.6	75.8	**78.3**	64.3
Fast+ASTN [16]	07	ResNet101	73.6	75.4	**83.8**	75.1	61.3	44.8	**81.9**	**81.1**	87.9	57.9	81.2	**72.5**	87.6	**85.2**	80.3	74.7	**44.3**	72.2	76.7	76.9	71.4
Fast+IA(ours)	07	ResNet101	**74.7**	77.3	81.2	**78.1**	**62.6**	**52.5**	77.8	80.0	**88.7**	**58.6**	**81.8**	71.4	**87.9**	84.2	**81.4**	**76.6**	44.0	**77.1**	**79.1**	76.9	**77.2**
Faster [10]	07	VGG16	69.9	70.0	**80.6**	**70.1**	**57.3**	49.9	78.2	80.4	82.0	**52.2**	75.3	**67.2**	80.3	79.8	75.0	76.3	39.1	68.3	67.3	**81.1**	67.6
Faster+IA(ours)	07	VGG16	**71.1**	**73.4**	78.5	68.3	54.7	**56.1**	**81.0**	**85.5**	**84.3**	48.4	**77.9**	61.7	**80.5**	**82.6**	**75.3**	**77.5**	**47.0**	**71.7**	**68.8**	76.0	**72.5**
Faster	07	ResNet101	75.1	76.5	79.7	77.7	66.4	61.0	83.3	86.3	**87.5**	53.6	81.1	66.9	85.3	**85.1**	77.4	**78.9**	**50.0**	74.1	**75.8**	78.9	75.4
Faster+IA(ours)	07	ResNet101	**76.5**	**77.9**	**82.9**	**78.4**	**67.2**	**62.2**	**84.2**	**86.9**	87.2	**55.5**	**85.6**	**69.1**	**87.0**	85.0	**81.4**	78.8	48.4	**79.4**	75.0	**83.2**	**75.4**
Faster [10]	07+12	VGG16	73.2	76.5	79.0	70.9	65.5	52.1	83.1	84.7	86.4	52.0	**81.9**	65.7	84.8	84.6	77.5	76.7	38.8	73.6	73.9	83.0	72.6
Faster+IA(ours)	07+12	VGG16	**76.8**	**78.5**	**81.1**	**76.8**	**67.2**	**63.9**	**87.1**	**87.7**	**87.8**	**59.3**	81.1	**72.9**	**84.8**	**86.7**	**80.5**	**78.7**	**50.9**	**76.9**	**74.2**	**83.1**	**76.5**
Faster [10]	07+12	ResNet101	76.4	79.8	80.7	76.2	68.3	55.9	85.1	85.3	**89.8**	56.7	**87.8**	69.4	**88.3**	**88.9**	80.9	78.4	41.7	78.6	79.8	85.3	72.0
Faster+IA(ours)	07+12	ResNet101	**81.1**	**85.3**	**86.8**	**79.7**	**74.6**	**69.4**	**88.4**	**88.7**	88.8	**64.8**	87.3	**74.7**	87.7	88.6	**85.3**	**83.5**	**53.9**	**82.7**	**81.5**	**87.8**	**80.9**

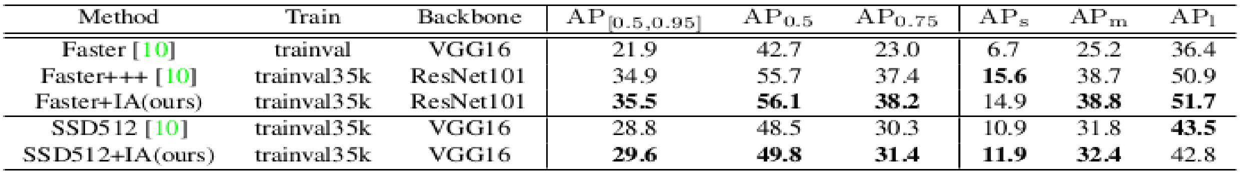
Table 4: Object detection Average Precision (AP) tested on VOC2007.



Method	Train	Backbone	mAP	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	bike	person	plant	sheep	sofa	train	TV
Faster [10]	07++12	ResNet101	73.8	86.5	81.6	77.2	58.0	51.0	78.6	76.6	93.2	48.6	80.4	59.0	**92.1**	85.3	84.8	80.7	48.1	77.3	66.5	84.7	65.6
Faster+IA(ours)	07++12	ResNet101	**79.2**	**87.7**	**86.7**	**80.3**	**68.1**	**62.1**	**81.0**	**84.7**	**93.8**	**61.8**	**84.2**	**63.1**	92.0	**87.4**	**86.6**	**85.8**	**61.0**	**84.6**	**72.4**	**86.5**	**73.8**
SSD300 [9]	07++12	VGG16	75.8	**88.1**	82.9	74.4	61.9	47.6	**82.7**	78.8	91.5	58.1	80.0	64.1	89.4	**85.7**	85.5	82.6	50.2	79.8	**73.6**	**86.6**	**72.1**
SSD300+IA(ours)	07++12	VGG16	**77.9**	87.5	**85.0**	**79.1**	**66.6**	**60.4**	80.0	**83.6**	**92.3**	**59.8**	**82.3**	**64.8**	**89.9**	85.6	**85.7**	**84.5**	**59.5**	**82.2**	71.8	85.8	71.6

Table 5: Object detection Average Precision (AP) tested on VOC2012.

For the PASCAL VOC2012 and COCO2017 datasets,used Faster-RCNN to construct Inverted Attention Network.The detection performance of PASCAL VOC2012 is shown in Table5. For ResNet101 Faster-RCNN, IA increased mAP from 73.8% of the baseline to 79.2%, which is 5.4% improvement. The AP on 19 categories achieved consistent performance gain. For ResNet101 SSD300, IA increased mAP from 75.8% of baseline to 77.9%, which is2.1% improvement. The AP on 14 categories achieved consistent performance gain.



Method	Train	Backbone	$AP_{[0.5,0.95]}$	$AP_{0.5}$	$AP_{0.75}$	AP_s	AP_m	AP_l
Faster [10]	trainval	VGG16	21.9	42.7	23.0	6.7	25.2	36.4
Faster+++ [10]	trainval35k	ResNet101	34.9	55.7	37.4	**15.6**	38.7	50.9
Faster+IA(ours)	trainval35k	ResNet101	**35.5**	**56.1**	**38.2**	14.9	**38.8**	**51.7**
SSD512 [10]	trainval35k	VGG16	28.8	48.5	30.3	10.9	31.8	**43.5**
SSD512+IA(ours)	trainval35k	VGG16	**29.6**	**49.8**	**31.4**	**11.9**	**32.4**	42.8

Table 6: Object detection Average Precision (AP) tested on COCO test-dev 2017.

Faster-RCNN baseline with VGG16 trained on the train and validation set of COCO produced 21.9% for AP [0.5,0.95]. Using ResNet101, AP [0.5,0.95] reached 35.5% even when the training data were reduced to the train and val35k subsets of COCO. SSD512 with VGG16 produced AP [0.5,0.95] 28.8. IAN tends to boost the performance of the medium and large objects for Faster-RCNN. While for SSD512, when IA was conducted on full features, IAN favors small and medium objects.

CONCLUSION

Inverted Attention computes attention using gradients of feature maps during training, and iteratively inverts attention along both spatial and channel dimension of the feature maps. The object detection network trained with IA spreads its attention to the whole objects. As a result, IA effectively improves diversity of features in training, and makes the network robust to image defects.