# Fully Convolutional Instance-aware Semantic Segmentation

Yi Li[1,2*] Haozhi Qi[2*] Jifeng Dai[2] Xiangyang Ji[1] Yichen Wei[2]
[1]Tsinghua University [2]Microsoft Research Asia
{liyi14,xyji}@tsinghua.edu.cn, {v-haoq,jifdai,yichenw}@microsoft.com

## Abstract

*We present the first fully convolutional end-to-end solution for instance-aware semantic segmentation task. It inherits all the merits of FCNs for semantic segmentation [29] and instance mask proposal [5]. It detects and segments the object instances jointly and simultanoulsy. By the introduction of position-senstive inside/outside score maps, the underlying convolutional representation is fully shared between the two sub-tasks, as well as between all regions of interest. The proposed network is highly integrated and achieves state-of-the-art performance in both accuracy and efficiency. It wins the COCO 2016 segmentation competition by a large margin. Code would be released at https://github.com/daijifeng001/TA-FCN.*

## 1. Introduction

Fully convolutional networks (FCNs) [29] have recently dominated the field of semantic image segmentation. An FCN takes an input image of arbitrary size, applies a series of convolutional layers, and produces per-pixel likelihood score maps for all semantic categories, as illustrated in Figure 1(a). Thanks to the simplicity, efficiency, and the local weight sharing property of convolution, FCNs provide an accurate, fast, and end-to-end solution for semantic segmentation.

However, conventional FCNs do not work for the instance-aware semantic segmentation task, which requires the detection and segmentation of individual object instances. The limitation is inherent. Because convolution is translation invariant, the same image pixel receives the same responses (thus classification scores) irrespective to its relative position in the context. However, instance-aware semantic segmentation needs to operate on region level, and the same pixel can have different semantics in different regions. This behavior cannot be modeled by a single FCN on the whole image. The problem is exemplified in Figure 2.

*Equal contribution. This work is done when Yi Li and Haozhi Qi are interns at Microsoft Research.

Certain translation-variant property is required to solve the problem. In a prevalent family of instance-aware semantic segmentation approaches [7, 16, 8], it is achieved by adopting different types of sub-networks in three stages: 1) an FCN is applied on the whole image to generate intermediate and shared feature maps; 2) from the shared feature maps, a pooling layer warps each region of interest (ROI) into fixed-size per-ROI feature maps [17, 12]; 3) one or more fully-connected (fc) layer(s) in the last network convert the per-ROI feature maps to per-ROI masks. Note that the translation-variant property is introduced in the fc layer(s) in the last step.

Such methods have several drawbacks. First, the ROI pooling step losses spatial details due to feature warping and resizing, which however, is necessary to obtain a fixed-size representation (*e.g.*, $14 \times 14$ in [8]) for fc layers. Such distortion and fixed-size representation degrades the segmentation accuracy, especially for large objects. Second, the fc layers over-parametrize the task, without using regularization of local weight sharing. For example, the last fc layer has high dimensional 784-way output to estimate a $28 \times 28$ mask. Last, the per-ROI network computation in the last step is not shared among ROIs. As observed empirically, a considerably complex sub-network in the last step is necessary to obtain good accuracy [36, 9]. It is therefore slow for a large number of ROIs (typically hundreds or thousands of region proposals). For example, in the MNC method [8], which won the 1st place in COCO segmentation challenge 2015 [25], 10 layers in the ResNet-101 model [18] are kept in the per-ROI sub-network. The approach takes 1.4 seconds per image, where more than $80\%$ of the time is spent on the last per-ROI step. These drawbacks motivate us to ask the question that, *can we exploit the merits of FCNs for end-to-end instance-aware semantic segmentation?*

Recently, a fully convolutional approach has been proposed for instance mask proposal generation [5]. It extends the translation invariant score maps in conventional FCNs to *position-sensitive* score maps, which are somewhat translation-variant. This is illustrated in Figure 1(b). The approach is only used for mask proposal generation and presents several drawbacks. It is blind to semantic cat-

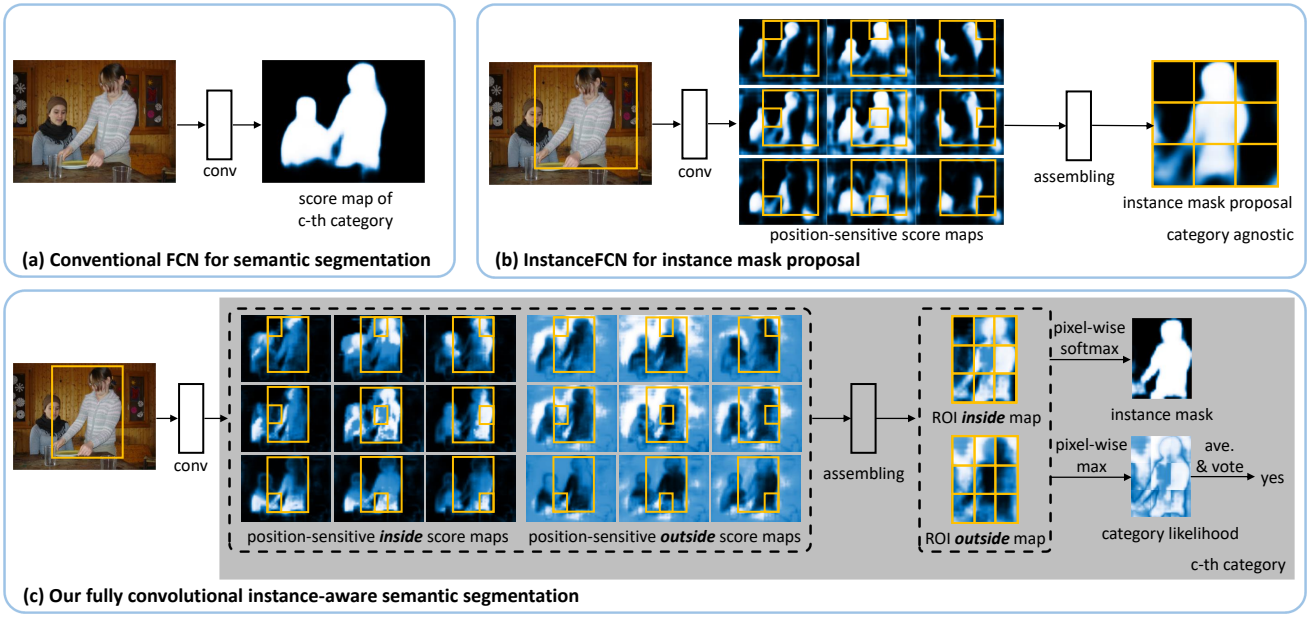


Figure 1. Illustration of our idea. (a) Conventional fully convolutional network (FCN) [29] for semantic segmentation. A single score map is used for each category, which is unaware of individual object instances. (b) InstanceFCN [5] for instance segment proposal, where $3 \times 3$ position-sensitive score maps are used to encode relative position information. A downstream network is used for segment proposal classification. (c) Our fully convolutional instance-aware semantic segmentation method (FCIS), where position-sensitive inside/outside score maps are used to perform object segmentation and detection jointly and simultanously.

egories and requires a downstream network for detection. The object segmentation and detection sub-tasks are separated and the solution is not end-to-end. It operates on square, fixed-size sliding windows ($224 \times 224$ pixels) and adopts a time-consuming image pyramid scanning to find instances at different scales.

In this work, we propose the first end-to-end fully convolutional approach for instance-aware semantic segmentation. Dubbed FCIS, it extends the approach in [5]. The underlying convolutional representation and the score maps are fully shared for the object segmentation and detection sub-tasks, via a novel joint formulation with no extra parameters. The network structure is highly integrated and efficient. The per-ROI computation is simple, fast, and does not involve any warping or resizing operations. The approach is briefly illustrated in Figure 1(c). It operates on box proposals instead of sliding windows, enjoying the recent advances in object detection [34].

Extensive experiments verify that the proposed approach is state-of-the-art in both accuracy and efficiency. It achieves significantly higher accuracy than the previous challenge winning method MNC [8] on the large-scale COCO dataset [25]. It wins the 1st place in COCO 2016 segmentation competition, outperforming the 2nd place entry by 12% in accuracy relatively. It is fast. The inference in COCO competition takes 0.24 seconds per image using ResNet-101 model [18] (Nvidia K40), which is $6\times$ faster than MNC [8]. Code would be released at https://github.com/daijifeng001/TA-FCN.

## 2. Our Approach

### 2.1. Position-sensitive Score Map Parameterization

In FCNs [29], a classifier is trained to predict each pixel's likelihood score of "*the pixel belongs to some object category*". It is translation invariant and unaware of individual object instances. For example, the same pixel can be foreground on one object but background on another (adjacent) object. A single score map per-category is insufficient to distinguish these two cases.

To introduce translation-variant property, a fully convolutional solution is firstly proposed in [5] for instance mask proposal. It uses $k^2$ position-sensitive score maps that correspond to $k \times k$ evenly partitioned cells of objects. This is illustrated in Figure 1(b) ($k = 3$). Each score map has the same spatial extent of the original image (in a lower resolution, *e.g.*, $16\times$ smaller). Each score represents the likelihood of "*the pixel belongs to some object instance at a relative position*". For example, the first map is for "at top left position" in Figure 1(b).

During training and inference, for a fixed-size square sliding window ($224\times224$ pixels), its pixel-wise foreground likelihood map is produced by assembling (copy-paste) its $k\times k$ cells from the corresponding score maps. In this way, a


Figure 2. Instance segmentation and classification results (of "person" category) of different ROIs. The score maps are shared by different ROIs and both sub-tasks. The red dot indicates one pixel having different semantics in different ROIs.

pixel can have different scores in different instances as long as the pixel is at different relative positions in the instances.

As shown in [5], the approach is state-of-the-art for the object mask proposal task. However, it is also limited by the task. Only a fixed-size square sliding window is used. The network is applied on multi-scale images to find object instances of different sizes. The approach is blind to the object categories. Only a separate "objectness" classification sub-network is used to categorize the window as object or background. For the instance-aware semantic segmentation task, a separate downstream network is used to further classify the mask proposals into object categories [5].

### 2.2. Joint Mask Prediction and Classification

For the instance-aware semantic segmentation task, not only [5], but also many other state-of-the-art approaches, such as SDS [15], Hypercolumn [16], CFM [7], MNC [8], and MultiPathNet [42], share a similar structure: two sub-networks are used for object segmentation and detection sub-tasks, *separately and sequentially*.

Apparently, the design choices in such a setting, *e.g.*, the two networks' structure, parameters and execution order, are kind of arbitrary. They can be easily made for convenience other than for fundamental considerations. We conjecture that the separated sub-network design may not fully exploit the tight correlation between the two tasks.

We enhance the "position-sensitive score map" idea to perform the object segmentation and detection sub-tasks *jointly and simultaneously*. The same set of score maps are shared for the two sub-tasks, as well as the underlying convolutional representation. Our approach brings no extra parameters and eliminates non essential design choices. We believe it can better exploit the strong correlation between the two sub-tasks.

Our approach is illustrated in Figure 1(c) and Figure 2. Given a region-of-interest (ROI), its pixel-wise score maps are produced by the assembling operation within the ROI. For each pixel in a ROI, there are two tasks: 1) detection:

whether it belongs to an object bounding box at a relative position (detection+) or not (detection-); 2) segmentation: whether it is inside an object instance's boundary (segmentation+) or not (segmentation-). A simple solution is to train two classifiers, separately. That's exactly our baseline *FCIS (separate score maps)* in Table 1. In this case, the two classifiers are two $1 \times 1$ conv layers, each using just one task's supervision.

Our joint formulation fuses the two answers into two scores: inside and outside. There are three cases: 1) high inside score and low outside score: detection+, segmentation+; 2) low inside score and high outside score: detection+, segmentation-; 3) both scores are low: detection-, segmentation-. The two scores answer the two questions jointly via softmax and max operations. For detection, we use max to differentiate cases 1)-2) (detection+) from case 3) (detection-). The detection score of the whole ROI is then obtained via average pooling over all pixels' likelihoods (followed by a softmax operator across all the categories). For segmentation, we use softmax to differentiate cases 1) (segmentation+) from 2) (segmentation-), at each pixel. The foreground mask (in probabilities) of the ROI is the union of the per-pixel segmentation scores (for each category). Similarly, the two sets of scores are from two $1 \times 1$ conv layer. The inside/outside classifiers are trained jointly as they receive the back-propagated gradients from both segmentation and detection losses.

The approach has many desirable properties. All the per-ROI components (as in Figure 1(c)) do not have free parameters. The score maps are produced by a single FCN, without involving any feature warping, resizing or fc layers. All the features and score maps respect the aspect ratio of the original image. The local weight sharing property of FCNs is preserved and serves as a regularization mechanism. All per-ROI computation is simple ($k^2$ cell division, score map copying, softmax, max, average pooling) and fast, giving rise to a negligible per-ROI computation cost.

### 2.3. An End-to-End Solution

Figure 3 shows the architecture of our end-to-end solution. While any convolutional network architecture can be used [39, 40], in this work we adopt the ResNet model [18]. The last fully-connected layer for $1000-$way classification is discarded. Only the previous convolutional layers are retained. The resulting feature maps have $2048$ channels. On top of it, a $1 \times 1$ convolutional layer is added to reduce the dimension to $1024$.

In the original ResNet, the effective feature stride (the decrease in feature map resolution) at the top of the network is 32. This is too coarse for instance-aware semantic segmentation. To reduce the feature stride and maintain the field of view, the "hole algorithm" [3, 29] (*Algorithme à trous* [30]) is applied. The stride in the first block of conv5 convolutional layers is decreased from 2 to 1. The effective feature stride is thus reduced to 16. To maintain the field of view, the "hole algorithm" is applied on all the convolutional layers of conv5 by setting the dilation as 2.

We use region proposal network (RPN) [34] to generate ROIs. For fair comparison with the MNC method [8], it is added on top of the conv4 layers in the same way. Note that RPN is also fully convolutional.

From the conv5 feature maps, $2k^2 \times (C+1)$ score maps are produced ($C$ object categories, one background category, two sets of $k^2$ score maps per category, $k = 7$ by default in experiments) using a $1 \times 1$ convolutional layer. Over the score maps, each ROI is projected into a $16\times$ smaller region. Its segmentation probability maps and classification scores over all the categories are computed as described in Section 2.2.

Following the modern object detection systems, bounding box (bbox) regression [13, 12] is used to refine the initial input ROIs. A sibling $1 \times 1$ convolutional layer with $4k^2$ channels is added on the conv5 feature maps to estimate the bounding box shift in location and size.

Below we discuss more details in inference and training.

**Inference** For an input image, $300$ ROIs with highest scores are generated from RPN. They pass through the bbox regression branch and give rise to another $300$ ROIs. For each ROI, we get its classification scores and foreground mask (in probability) for all categories. Figure 2 shows an example. Non-maximum suppression (NMS) with an intersection-over-union (IoU) threshold $0.3$ is used to filter out highly overlapping ROIs. The remaining ROIs are classified as the categories with highest classification scores. Their foreground masks are obtained by mask voting [8] as follows. For an ROI under consideration, we find all the ROIs (from the $600$) with IoU scores higher than $0.5$. Their foreground masks of the category are averaged on a per-pixel basis, weighted by their classification scores. The averaged mask is binarized as the output.

**Training** An ROI is positive if its box IoU with respect to the nearest ground truth object is larger than $0.5$, otherwise it is negative. Each ROI has three loss terms in equal weights: a softmax detection loss over $C + 1$ categories, a softmax segmentation loss [1] *over the foreground mask of the ground-truth category only*, and a bbox regression loss as in [12]. The latter two loss terms are effective only on the positive ROIs.

During training, the model is initialized from the pre-trained model on ImageNet classification [18]. Layers absent in the pre-trained model are randomly initialized. The training images are resized to have a shorter side of 600 pixels. We use SGD optimization. We train the model using 8 GPUs, each holding one image mini batch, giving rise to

[1] The term sums per-pixel losses over the ROI and normalizes the sum by the ROI's size.

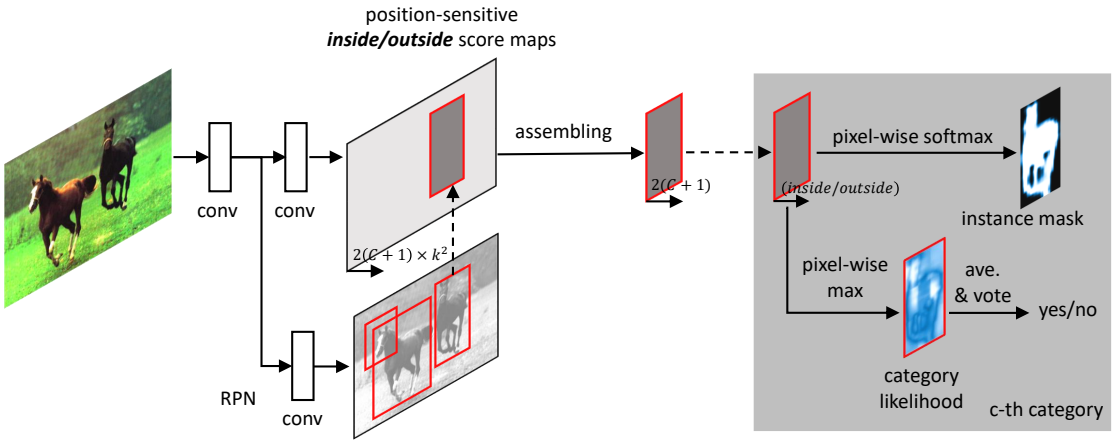


Figure 3. Overall architecture of FCIS. A region proposal network (RPN) [34] shares the convolutional feature maps with FCIS. The proposed region-of-interests (ROIs) are applied on the score maps for joint object segmentation and detection. The learnable weight layers are fully convolutional and computed on the whole image. The per-ROI computation cost is negligible.

an effective batch size $\times 8$. For experiments on PASCAL VOC [11], 30k iterations are performed, where the learning rates are $10^{-3}$ and $10^{-4}$ in the first 20k and the last 10k iterations respectively. The iteration number is $\times 8$ for experiments on COCO [25].

As the per-ROI computation is negligible, the training benefits from inspecting more ROIs at small training cost. Specifically, we apply online hard example mining (OHEM) [38]. In each mini batch, forward propagation is performed on all the 300 proposed ROIs on one image. Among them, 128 ROIs with the highest losses are selected to back-propagate their error gradients.

For the RPN proposals, 9 anchors (3 scales $\times$ 3 aspect ratios) are used by default. 3 additional anchors at a finer scale are used for experiments on the COCO dataset [25]. To enable feature sharing between FCIS and RPN, joint training is performed [8, 35].

## 3. Related Work

**Semantic Image Segmentation** The task is to assign every pixel in the image a semantic category label. It does not distinguish object instances. Recently, this field has been dominated by a prevalent family of approaches based on FCNs [29]. The FCNs are extended with global context [28], multi-scale feature fusion [4], and deconvolution [31]. Recent works in [3, 43, 37, 24] integrated FCNs with conditional random fields (CRFs). The expensive CRFs are replaced by more efficient domain transform in [2]. As the per-pixel category labeling is expensive, the supervision signals in FCNs have been relaxed to boxes [6], scribbles [23], or weakly supervised image class labels [19, 20].

**Object Segment Proposal** The task is to generate category-agnostic object segments. Traditional approaches, *e.g.*, MCG [1] and Selective Search [41], use low level image features. Recently, the task is achieved by deep learning approaches, such as DeepMask [32] and SharpMask [33]. Recently, a fully convolutional approach is proposed in [5], which inspires this work.

**Instance-aware Semantic Segmentation** The task requires both classification and segmentation of object instances. Typically, the two sub-tasks are accomplished separately. Usually, the segmentation task relies on a segment proposal method and the classification task is built on the region-based methods [13, 12, 34]. This paradigm includes most state-of-the-art approaches, such as SDS [15], Hypercolumn [16], CFM [7], MNC [8], MultiPathNet [42], and iterative approach [21]. Such approaches have certain drawbacks, as discussed in Section 1 and Section 2.2. In this work, we propose a fully convolutional approach with an integrated joint formulation for the two sub-tasks.

There are some endeavors [22, 26] trying to extend FCNs for instance-aware semantic segmentation, by grouping/clustering the FCN's output. However, all these methods rely on complex hand-crafted post processing, and are not end-to-end. The performance is also not satisfactory.

**FCNs for Object Detection** The idea of "position sensitive score maps" in [5] is adapted in R-FCN [9], resulting in a fully convolutional approach for object detection. The score maps are re-purposed from foreground-background segmentation likelihood to object category likelihood. R-FCN [9] only performs object classification. It is unaware of the instance segmentation task. Yet, it can be combined with [5] for instance-aware semantic segmentation task, in a straightforward manner. This is investigated in our experiments (Section 4.1).

# 4. Experiments

## 4.1. Ablation Study on PASCAL VOC

Ablation experiments are performed to study the proposed FCIS method on the PASCAL VOC dataset [11]. Following the protocol in [15, 7, 16, 8], model training is performed on the VOC 2012 train set, and evaluation is performed on the VOC 2012 validation set, with the additional instance mask annotations from [14]. Accuracy is evaluated by mean average precision, mAP$^r$ [15], at mask-level IoU (intersection-over-union) thresholds at 0.5 and 0.7.

The proposed **FCIS** approach is compared with alternative (almost) fully convolutional baseline methods, as well as variants of FCIS with different design choices. For fair comparison, ImageNet [10] pre-trained ResNet-101 model [18] is used for all the methods. OHEM is not applied.

**naïve MNC**. This baseline is similar to MNC [8] except that all convolutional layers of ResNet-101 are applied on the whole image to obtain feature maps, followed by ROI pooling on top of the last block of conv5 layers. A 784-way fc layer is applied on the ROI pooled features for mask prediction (of resolution $28 \times 28$), together with a 21-way fc layer for classification. The à trous trick is also applied for fair comparison. It is almost fully convolutional, with only single layer fc sub-networks in per-ROI computation.

**InstFCN + R-FCN**. The class-agnostic mask proposals are firstly generated by InstFCN [5], and then classified by R-FCN [9]. It is a straightforward combination of InstFCN and R-FCN. The two FCNs are separately trained and applied for mask prediction and classification, respectively.

**FCIS (translation invariant)**. To verify the importance of the translation-variant property introduced by the position sensitive score maps, this baseline sets $k = 1$ in the FCIS method to make it translation invariant.

**FCIS (separate score maps)**. To validate the joint formulation for mask prediction and classification, this baseline uses the two sets of score maps separately for the two sub-tasks. The first set of $k^2$ score maps are only for segmentation, in the similar way as in [5]. The second set is only for classification, in the same way as in R-FCN [9]. Therefore, the preceding convolutional classifiers for the two sets of score maps are not related, while the shallower convolutional feature maps are still shared.

Table 1 shows the results. The mAP$^r$ scores of the naïve MNC baseline are 59.1% and 36.0% at IoU thresholds of 0.5 and 0.7 respectively. They are 5.5% and 12.9% lower than those of the original MNC [8], which keeps 10 layers in ResNet-101 in the per-ROI sub-networks. This verifies the importance of respecting the translation-variant property for instance-aware semantic segmentation.

The result of "InstFCN + R-FCN" is reasonably good, but is still inferior than that of FCIS. The inference speed is also slow (1.27 seconds per image on a Nvidia K40 GPU).

| method | mAP$^r$@0.5 (%) | mAP$^r$@0.7 (%) |
|---|---|---|
| naïve MNC | 59.1 | 36.0 |
| InstFCN + R-FCN | 62.7 | 41.5 |
| FCIS (translation invariant) | 52.5 | 38.5 |
| FCIS (separate score maps) | 63.9 | 49.7 |
| FCIS | 65.7 | 52.1 |

Table 1. Ablation study of (almost) fully convolutional methods on PASCAL VOC 2012 validation set.

The proposed FCIS method achieves the best result. This verifies the effectiveness of our end-to-end solution. Its degenerated version "FCIS (translation invariant)" is much worse, indicating the position sensitive score map parameterization is vital. Its degenerated version "FCIS (separate score maps)" is also worse, indicating that the joint formulation is effective.

## 4.2. Experiments on COCO

Following the COCO [25] experiment guideline, training is performed on the 80k+40k trainval images, and results are reported on the test-dev set. We evaluate the performance using the standard COCO evaluation metric, mAP$^r$@[0.5:0.95], as well as the traditional mAP$^r$@0.5 metric.

**Comparison with MNC** We compare the proposed FCIS method with MNC [8], the 1st place entry in COCO segmentation challenge 2015. Both methods perform mask prediction and classification in ROIs, and share similar training/inference procedures. For fair comparison, we keep their common implementation details the same.

Table 2 presents the results using ResNet-101 model. When OHEM is not used, FCIS achieves an mAP$^r$@[0.5:0.95] score of 28.8% on COCO test-dev set, which is 4.2% absolutely (17% relatively) higher than that of MNC. According to the COCO standard split of object sizes, the accuracy improvement is more significant for larger objects, indicating that FCIS can capture the detailed spatial information better. FCIS is also much faster than MNC. In inference, FCIS spends 0.24 seconds per image on a Nvidia K40 GPU (0.19 seconds for network forward, and 0.05 seconds for mask voting), which is $\sim 6\times$ faster than MNC. FCIS is also $\sim 4\times$ faster in training. In addition, FCIS easily benefits from OHEM due to its almost free per-ROI cost, achieving an mAP$^r$@[0.5:0.95] score of 29.2%. Meanwhile, OHEM is unaffordable for MNC, because considerable computational overhead would be added during training.

| method | sampling strategy in training | train time/img | test time/img | mAP$^r$@[0.5:0.95] (%) | mAP$^r$@0.5 (%) | mAP$^r$@[0.5:0.95] (%) (small) | mAP$^r$@[0.5:0.95] (%) (mid) | mAP$^r$@[0.5:0.95] (%) (large) |
|---|---|---|---|---|---|---|---|---|
| MNC | random | 2.05s | 1.37s | 24.6 | 44.3 | 4.7 | 25.9 | 43.6 |
| **FCIS** | random | 0.53s | 0.24s | 28.8 | 48.7 | 6.8 | 30.8 | 49.5 |
| MNC | OHEM | 3.22s | 1.37s | N/A | N/A | N/A | N/A | N/A |
| **FCIS** | OHEM | 0.54s | 0.24s | **29.2** | **49.5** | **7.1** | **31.3** | **50.0** |

Table 2. Comparison with MNC [8] on COCO test-dev set, using ResNet-101 model. Timing is evaluated on a Nvidia K40 GPU.

| network architecture | mAP$^r$@[0.5:0.95] (%) | mAP$^r$@0.5 (%) | test time/img |
|---|---|---|---|
| ResNet-50 | 27.1 | 46.7 | 0.16s |
| ResNet-101 | 29.2 | 49.5 | 0.24s |
| ResNet-152 | 29.5 | 49.8 | 0.27s |

Table 3. Results of using networks of different depths in FCIS.

| | mAP$^r$@[0.5:0.95] (%) | mAP$^r$@0.5 (%) |
|---|---|---|
| FAIRCNN (2015) | 25.0 | 45.6 |
| MNC+++ (2015) | 28.4 | 51.6 |
| G-RMI (2016) | 33.8 | 56.9 |
| FCIS baseline | 29.2 | 49.5 |
| +multi-scale testing | 32.0 | 51.9 |
| +horizontal flip | 32.7 | 52.7 |
| +multi-scale training | 33.6 | 54.5 |
| +ensemble | **37.6** | **59.9** |

Table 4. Instance-aware semantic segmentation results of different entries for the COCO segmentation challenge (2015 and 2016) on COCO test-dev set.

**Networks of Different Depths** Table 3 presents the results of using ResNet of different depths in FCIS method. The accuracy is improved when the network depth is increased from 50 to 101, and gets saturated when the depth reaches 152.

**COCO Segmentation Challenge 2016 Entry** Based on the FCIS method, we participated in COCO segmentation challenge 2016 and won the 1st place.

Table 4 presents the results of our entry and other entries in COCO segmentation challenge 2015 and 2016. Our entry is based on FCIS, with some simple bells and whistles.

*FCIS Baseline*. The baseline FCIS method achieves a competitive mAP$^r$@[0.5:0.95] score of 29.2%, which is already higher than MNC+++ [8], the winning entry in 2015.

*Multi-scale testing*. Following [17, 18], the position-sensitive score maps are computed on a pyramid of testing images, where the shorter sides are of $\{480, 576, 688, 864, 1200, 1400\}$ pixels. For each ROI, we obtain its result from the scale where the ROI has a number of pixels closest to $224 \times 224$. Note that RPN proposals are still computed from a single scale (shorter side 600). Multi-scale testing improves the accuracy by 2.8%.

*Horizontal flip*. Similar to [42], the FCIS method is applied on the original and the flipped images, and the results in the corresponding ROIs are averaged. This helps increase the accuracy by 0.7%.

*Multi-scale training*. We further apply multi-scale training at the same scales as in multi-scale inference. For the finer scales, a random $600 \times 600$ image patch is cropped for training due to memory issues, as in [27]. This increases the accuracy by 0.9%.

*Ensemble*. Following [18], region proposals are generated using an ensemble, and the union of the proposals are processed by an ensemble for mask prediction and classification. We utilize an ensemble of 6 networks. The final result is 37.6%, which is 3.8% (11% relatively) higher than G-RMI, the 2nd place entry in 2016, and 9.2% (32% relatively) higher than MNC+++, the 1st place entry in 2015. Some example results are visualized in Figure 4.

**COCO Detection** The proposed FCIS method also performs well on box-level object detection. By taking the enclosing boxes of the instance masks as detected bounding boxes, it achieves an object detection accuracy of 39.7% on COCO test-dev set, measured by the standard mAP$^b$@[0.5:0.95] score. The result ranks 2nd in the COCO object detection leaderboard.

## 5. Conclusion

We present the first fully convolutional method for instance-aware semantic segmentation. It extends the existing FCN-based approaches and significantly pushes forward the state-of-the-art in both accuracy and efficiency for the task. The high performance benefits from the highly integrated and efficient network architecture, especially a novel joint formulation.

## References


[1] P. Arbelaez, J. Pont-Tuset, J. T. Barron, F. Marques, and J. Malik. Multiscale combinatorial grouping. In *CVPR*, 2014. 5

[2] L.-C. Chen, J. T. Barron, G. Papandreou, K. Murphy, and A. L. Yuille. Semantic image segmentation with task-specific

Figure 4. Example instance-aware semantic segmentation results of the proposed FCIS method on COCO test set. Check https://github.com/daijifeng001/TA-FCN for example results on the first 5k images on COCO test set.

edge detection using cnns and a discriminatively trained domain transform. In *CVPR*, 2016. 5

[3] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Semantic image segmentation with deep convolutional nets and fully connected crfs. In *ICLR*, 2015. 4, 5

[4] L.-C. Chen, Y. Yang, J. Wang, W. Xu, and A. L. Yuille. Attention to scale: Scale-aware semantic image segmentation. In *CVPR*, 2016. 5

[5] J. Dai, K. He, Y. Li, S. Ren, and J. Sun. Instance-sensitive fully convolutional networks. In *ECCV*, 2016. 1, 2, 3, 5, 6

[6] J. Dai, K. He, and J. Sun. Boxsup: Exploiting bounding boxes to supervise convolutional networks for semantic segmentation. In *ICCV*, 2015. 5

[7] J. Dai, K. He, and J. Sun. Convolutional feature masking for joint object and stuff segmentation. In *CVPR*, 2015. 1, 3, 5, 6

[8] J. Dai, K. He, and J. Sun. Instance-aware semantic segmentation via multi-task network cascades. In *CVPR*, 2016. 1, 2, 3, 4, 5, 6, 7

[9] J. Dai, Y. Li, K. He, and J. Sun. R-fcn: Object detection via region-based fully convolutional networks. In *NIPS*, 2016. 1, 5, 6

[10] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009. 6

[11] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes (VOC) Challenge. *IJCV*, 2010. 5, 6

[12] R. Girshick. Fast R-CNN. In *ICCV*, 2015. 1, 4, 5

[13] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, 2014. 4, 5

[14] B. Hariharan, P. Arbeláez, L. Bourdev, S. Maji, and J. Malik. Semantic contours from inverse detectors. In *ICCV*, 2011. 6

[15] B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik. Simultaneous detection and segmentation. In *ECCV*. 2014. 3, 5, 6

[16] B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik. Hypercolumns for object segmentation and fine-grained localization. In *CVPR*, 2015. 1, 3, 5, 6

[17] K. He, X. Zhang, S. Ren, and J. Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. In *ECCV*, 2014. 1, 7

[18] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 1, 2, 4, 6, 7

[19] S. Hong, H. Noh, and B. Han. Decoupled deep neural network for semi-supervised semantic segmentation. In *NIPS*, 2015. 5

[20] S. Hong, J. Oh, B. Han, and H. Lee. Learning transferrable knowledge for semantic segmentation with deep convolutional neural network. In *CVPR*, 2016. 5

[21] K. Li, B. Hariharan, and J. Malik. Iterative instance segmentation. In *CVPR*, 2016. 5

[22] X. Liang, Y. Wei, X. Shen, J. Yang, L. Lin, and S. Yan. Proposal-free network for instance-level object segmentation. *arXiv preprint*, 2015. 5

[23] D. Lin, J. Dai, J. Jia, K. He, and J. Sun. Scribblesup: Scribble-supervised convolutional networks for semantic segmentation. In *CVPR*, 2016. 5

[24] G. Lin, C. Shen, A. van den Hengel, and I. Reid. Efficient piecewise training of deep structured models for semantic segmentation. In *CVPR*, 2016. 5

[25] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft COCO: Common objects in context. In *ECCV*. 2014. 1, 2, 5, 6

[26] S. Liu, X. Qi, J. Shi, H. Zhang, and J. Jia. Multi-scale patch aggregation (mpa) for simultaneous detection and segmentation. In *CVPR*, 2016. 5

[27] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, and S. Reed. Ssd: Single shot multibox detector. In *ECCV*, 2016. 7

[28] W. Liu, A. Rabinovich, and A. C. Berg. Parsenet: Looking wider to see better. In *ICLR workshop*, 2016. 5

[29] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, 2015. 1, 2, 4, 5

[30] S. Mallat. *A wavelet tour of signal processing*. Academic press, 1999. 4

[31] H. Noh, S. Hong, and B. Han. Learning deconvolution network for semantic segmentation. In *ICCV*, 2015. 5

[32] P. O. Pinheiro, R. Collobert, and P. Dollar. Learning to segment object candidates. In *NIPS*, 2015. 5

[33] P. O. Pinheiro, T.-Y. Lin, R. Collobert, and P. Dollar. Learning to refine object segments. In *ECCV*, 2016. 5

[34] S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *NIPS*, 2015. 2, 4, 5

[35] S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. *PAMI*, 2016. 5

[36] S. Ren, K. He, R. Girshick, X. Zhang, and J. Sun. Object detection networks on convolutional feature maps. *arXiv preprint*, 2015. 1

[37] A. G. Schwing and R. Urtasun. Fully connected deep structured networks. *arXiv preprint*, 2015. 5

[38] A. Shrivastava, A. Gupta, and R. Girshick. Training region-based object detectors with online hard example mining. In *CVPR*, 2016. 5

[39] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015. 4

[40] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *CVPR*, 2015. 4

[41] K. E. A. van de Sande, J. R. R. Uijlings, T. Gevers, and A. W. M. Smeulders. Segmentation as selective search for object recognition. In *ICCV*, 2011. 5

[42] S. Zagoruyko, A. Lerer, T.-Y. Lin, P. O. Pinheiro, S. Gross, S. Chintala, and P. Dollár. A multipath network for object detection. In *ECCV*, 2016. 3, 5, 7

[43] S. Zheng, S. Jayasumana, B. Romera-Paredes, V. Vineet, Z. Su, D. Du, C. Huang, and P. Torr. Conditional random fields as recurrent neural networks. In *ICCV*, 2015. 5