TOPNet: Thinking Outside The Bounding Box

Shuqin Xie[1], Chao Xu[1], Shu Liu[2], Alan Yuille[3], Jiaya Jia[2]
[1]Shanghai Jiao Tong University, China [2] The Chinese University of Hong Kong
[3] Johns Hopkins University
{qweasdshu, xuchao.19962007}@sjtu.edu.cn, {sliu, leojia}@cse.cuhk.edu.hk
alan.l.yuille@gmail.com

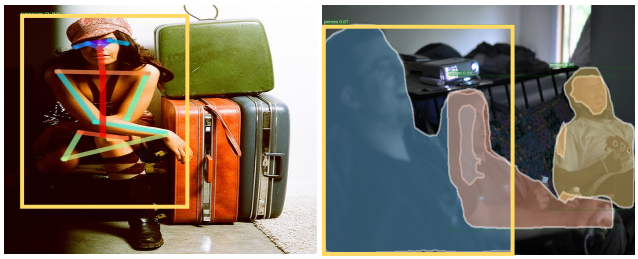
Abstract

Proposal-based methods have achieved great success in various tasks. A general principle is that predictions are made within proposals, which could introduce problems of incomplete prediction and wrong prediction. While expanding the proposal can alleviate these problems, we show that directly performing inference on the expanded proposal could be harmful and causes an ambiguity issue. This is due to the violation of the "one proposal-one instance" assumption. In this paper, we propose a new attention module to address the crucial ambiguity problem in a simple yet effective way. It is achieved by predicting an indicator, which is an attention map, to specify the target object in the expanded proposal. Moreover, this module can be readily incorporated into existing proposal-based methods, enabling them to fully leverage the benefit of expanding the proposal. We apply our module to a popular proposal-based framework Mask R-CNN and observe a significant performance boost. Experiments on the challenging COCO instance segmentation task and pose estimation task demonstrate the effectiveness and generality of the proposed method. Code will be made publicly available.

1. Introduction

Proposal-based methods have achieved great success in recent years, yielding impressive improvements on many computer vision tasks. For example, Mask R-CNN [16] and PANet [25] achieve state-of-art performances on the challenging COCO [23] instance segmentation task. In the field of multi-person pose estimation, new records are frequently being set by proposal-based methods [28, 10, 5] too.

The spirit of these methods is to separate the object detection and specific prediction task into two stages. The two-stage design greatly simplifies the problem at hand and allows an easy utilization of progress made in both fields. Moreover, the performance can be further improved by multi-task learning [16].



(a) Incomplete prediction (b) Wrong prediction

Figure 1: Illustration of the failure cases. (a) A case of incomplete prediction, where the proposal is imprecise and the resulted predictions are not complete; (b) An example of wrong prediction, where the mask network falsely activates some region of the wrong instance within the proposal.

Although the assumption of "one proposal is one instance" implied in these methods makes it reasonable to only perform inference within the proposal, two important problems arise. The first one is incomplete prediction caused by inaccurate bounding box detection, as illustrated in Figure 1(a). We observe that in many cases, the proposal manages to cover the majority of the object but fails to precisely locate its boundary. As a result, the prediction is not complete and misses out on some parts of the object. The problem becomes more severe when the predictions inside the bounding box are mutually influential. For example, in pose estimation, missing out on the ankle could influence the detection of the knee.

The second problem is the wrong prediction problem, as shown in Figure 1(b). The network falsely activates regions that belong to other instances within the proposal. This problem is caused by the existence of multiple instances in one bounding box, which violates the basic assumption of "one proposal-one instance". In these cases, it's difficult for the network to decide which one is the target object, so instead it adopts a more conservative strategy – activate them all. When occlusion happens, the problem becomes more serious.

Expanding the proposal is a promising approach to ad-

dress these problems. Firstly, expanding the proposal to a sufficiently large scale can ensure the coverage of the entire object, therefore solving the incompleteness issue. Additionally, it introduces more context information, which is helpful for addressing the wrong prediction problem. For example, in the case of Figure 1(b), if we can observe the entire body of the person on the right, the network might be able to realize that the falsely activated regions belong to that person and manage to correct the error.

Though it sounds appealing, performing inference on the expanded proposal is risky and possibly more harmful. As the proposal enlarges, the chance of containing multiple objects in one proposal also increases, therefore reintroducing the wrong prediction problem. Previous methods [10, 5] that adopt this schema have to carefully choose the expanding scale in order to balance the "incompleteness v.s. ambiguity" trade-off. Extra human engineering effort is usually required to obtain a reasonable result.

Rather than this naive expansion strategy, in this paper, we propose a rather simple yet effective method to address the challenging ambiguity issue. Our main contribution is to predict an *indicator* for the expanded proposal, which functions like an attention map to specify the target object within the expanded proposal.

Our *indicator* is generated by a novel attention module, named the TOP (Target Of the Proposal) module. It first predicts a target object from the local proposal, and then differentiates it from the context information of the expanded proposal. With this module, inference on the expanded proposal becomes feasible without causing extra ambiguity. Moreover, this module can be readily incorporated into any existing proposal-based frameworks to fully leverage the benefit of inference on larger-size proposals.

One example is to incorporate our module in the popular Mask R-CNN [16]. The new pipeline, which we call TOPNet, can effectively perform inference on the expanded proposal, thus greatly relieving the incomplete prediction and wrong prediction problems.

To demonstrate the effectiveness of our method, we conduct experiments on the challenging COCO [23] instance segmentation task and keypoint detection task. Without bells and whistles, we outperform the Mask R-CNN baseline by a large margin (1.5 mAP on instance segmentation and 1.9 mAP on keypoint detection). Moreover, our design is general and we expect to see improvements on other proposal-based methods too.

To summarize, our contributions are as follows:

- We propose a simple yet effective method which enables proposal-based framework to perform inference on the expanded proposal.
- We present a novel attention module, namely TOP module, to address the crucial ambiguity problem when expanding the bounding box.
- Our method significantly eases the incomplete and wrong prediction problems in proposal-based frameworks. Experiments on the COCO [23] instance segmentation and pose estimation sets demonstrate the effectiveness of our method.

2. Related Work

Proposal-based Framework in Object Detection The proposal-based framework was first proposed for object detection. R-CNN [14] utilized Selective Search [37] to enumerate object proposal regions and used convolutional neural network (CNN) [21, 35] to classify them. Fast R-CNN [13] and SPPNet [17] made the pipeline more efficient by extracting the entire feature map for the input image. Proposal-specific features were pooled from this global feature map. As a result, much computation was shared by proposals. Faster R-CNN [33] has the region proposal network (RPN) to generate object proposals in neural networks, not only leading to better accuracy, but also increasing the speed of entire framework. R-FCN [9] further shared computation by making nearly all computation fully convolutional. Each proposal just needs to pool the score from one global score map. Recently proposed FPN [22] modified the network structure for better feature representation and assigned proposals to appropriate scales for better performance. Cascade R-CNN [3] is a new method, which analyzed the property of localization quality of object proposals and enhanced the accuracy of the framework by sequentially refining proposals. The context information was not used in this framework.

Instance Segmentation The majority of instance segmentation methods follow the proposal-based framework. SDS [15], CFM [7] and MNC [8] are direct extensions of R-CNN, SPPNet and Faster R-CNN respectively. Instead of using box proposals, those methods took mask proposals as input for classification. Specifically, mask proposals were generated by fully-connected layers in MNC, inspired by DeepMask [29]. As a result, each output is with pixel-wise mask and corresponding class label. Mask R-CNN [16] extended FPN by introducing a parallel mask prediction branch, which is a small fully convolutional neural network (FCN). The decomposition of classification and segmentation helps the network produce decent segmentation results. More recently, PANet [25] enhanced the network structure to achieve better performance.

There are some other methods for instance segmentation. Most of them aimed at designing representations to decode instance masks. For example, DWT [1] learned the energy with respect to boundaries and applied watershed algorithm to generate the instance mask. SGN [24] and InstanceCut

[20] both learned instance boundaries and further decoded instance masks from them. Embedding is learned in [11, 26] to map pixels belonging to the same instance close to each other in the learned embedding space. Recurrent neural network was also used for instance segmentation [34, 32] where there is still much room to improve performance.

Human Pose Estimation The field of single human pose estimation is also reshaped by convolution neural networks. DeepPose [36] was the first work that used a ConvNet to estimate human pose, and achieved significant improvement over traditional method [30]. Many works [38, 2, 27, 6] further advanced the performance by using better network architectures or more reasonable loss function. The progress made in single person pose estimation had motivated researchers to develop new algorithms for multi-person pose estimation. DeepCut and DeeperCut [31, 18] used Faster R-CNN to detect human parts and then grouped them into individual person by integer programming. [4] proposed PAF to model the connection between joints and used it for grouping. [26] predicted a tag for each joint and assembled the joints with similar tag to be an instance. Different from these bottom-up methods, proposal-based frameworks operated in a top-down manner. [28, 10, 5] used an object detector to first detect human instances and then applied single person pose estimation algorithm to these proposals. [16] extended the Mask R-CNN framework by replacing masks with keypoint heatmaps and achieved competitive results. Our method is closer to the proposal-based methods, but we allow prediction to be made outside the proposal.

Larger Context Region For object detection, using larger regions can capture more context information and help network prediction. For example, methods of [12], [39] and [41] pooled the feature grid for each proposal from regions with different scales simultaneously. Multi-scale feature grids are directly concatenated or fused as the new representation to be fed to classification and regression subnetworks. Since only class label and coarse bounding boxes are needed in object detection, the misalignment of features in terms of spatial location is reduced and does not influence the performance much.

3. Method

In this paper, we aim to fully leverage the benefit of performing inference on the expanded proposal, which reduces to two subtasks: indicating the target object and making prediction on the expanded proposal. The first task is tackled by the TOP module. The second task is accomplished by using the output of the TOP module.

We organize this section as follows: first we introduce some notations in Section 3.1. Then we present the TOP module in detail in Section 3.2. In Section 3.3, we show how to augment existing proposal-based methods with the TOP module to form a TOPNet which allows inference on the expanded proposal. The inference and training algorithms for the TOPNet are also presented.

3.1. Notation

In the beginning, we introduce notation used in this paper. We represent the local proposal as $\mathcal{B}_l = (x_l^{tl}, y_l^{tl}, x_l^{br}, y_l^{br})$, where (x_l^{tl}, y_l^{tl}) and (x_l^{br}, y_l^{br}) are locations for top-left corner and bottom-right corner respectively. Its size is denoted by $H_l \times W_l$. The expanded proposal $\mathcal{B}_e$ is represented by $(x_e^{tl}, y_e^{tl}, x_e^{br}, y_e^{br})$ where its size is denoted as $H_e \times W_e$. It shares the same center as $\mathcal{B}_l$, but is k times larger. The indicator is denoted as $\mathcal{I}$, with resolution $H_\mathcal{I} \times W_\mathcal{I}$.

3.2. TOP Module

The TOP module is an attention module whose goal is to identify the target object within the expanded proposal. We notice that in many cases, the local proposal usually contains only one object; even if it includes multiple objects, a dominant one usually exists. In other words, although the target object is very ambiguous in the expanded proposal, it's actually quite clear in the local proposal. If we can take advantage of this knowledge, we can clearly specify the target object within the expanded proposal and thus address the ambiguity issue.

Therefore, we adopt the design illustrated in Figure 2 for the attention module. First of all, we predict the target object from the local proposal. Then, we differentiate it from the context information introduced by expanding the bounding box. The resulting output, which is an attention map, can clearly indicate the target object within the expanded proposal.

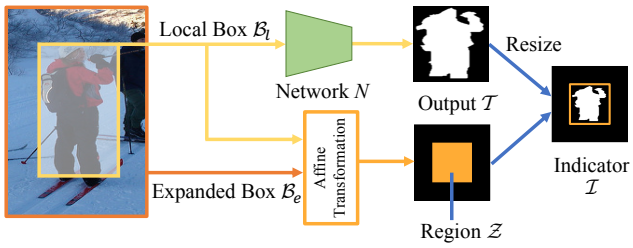


Figure 2: The TOP module work flow. First we predict the target object $\mathcal{T}$ within the local proposal. Then we compute the region $\mathcal{Z}$ in $\mathcal{I}$ that corresponds to the local proposal and resize $\mathcal{T}$ to fill $\mathcal{Z}$, padding the rest of $\mathcal{I}$ with 0.

Predicting the target object. We first predict the target object from the local proposal. The target object can be represented in different forms, depending on the specific task.

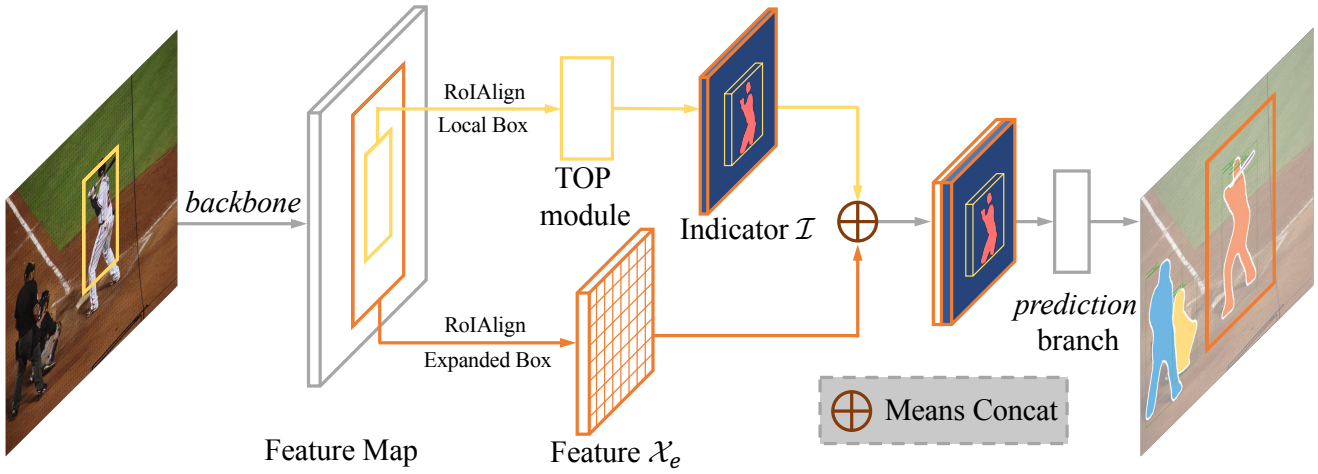


Figure 3: Pipeline for TOPNet. First, for each local proposal (yellow box), we use the TOP module to predict an *indicator* $\mathcal{I}$. Then we expand the proposal (orange box) and pool a feature $\mathcal{X}_e$ from the feature map. The pooled feature $\mathcal{X}_e$ is then concatenated with the *indicator* $\mathcal{I}$ and fed into the *prediction* branch to obtain the final prediction.

For example, we use object mask for instance segmentation, and keypoint score map for pose estimation. No matter the structure, for each local proposal $\mathcal{B}_l$, we use a network N to predict the target object within $\mathcal{B}_l$. The output $\mathcal{T}$ is assumed to be a map-like tensor with resolution $H_{\mathcal{T}} \times W_{\mathcal{T}}$.

Generating the indicator. As illustrated in Figure 2, we generate the *indicator* $\mathcal{I}$ as follows. First, we compute a region $\mathcal{Z}$ in $\mathcal{I}$ that corresponds to the local proposal. The boundary of $\mathcal{Z}$ can be obtained by an affine transformation and the affine matrix $\mathbf{M}$ is defined as

$$\mathbf{M} = \begin{bmatrix} s_x & 0 & -s_x \cdot x_e^{tl} \\ 0 & s_y & -s_y \cdot y_e^{tl} \end{bmatrix}, \tag{1}$$

where $s_x = W_{\mathcal{I}}/W_e$ and $s_y = H_{\mathcal{I}}/H_e$. And the boundary $\mathcal{B}_z$ is given by

$$\begin{bmatrix} x_z^{tl} \\ y_z^{tl} \end{bmatrix} = \mathbf{M} \begin{bmatrix} x_l^{tl} \\ y_l^{tl} \\ 1 \end{bmatrix}, \quad \begin{bmatrix} x_z^{br} \\ y_z^{br} \end{bmatrix} = \mathbf{M} \begin{bmatrix} x_l^{br} \\ y_l^{br} \\ 1 \end{bmatrix}. \tag{2}$$

Then, we resize $\mathcal{T}$ to the same size as $\mathcal{Z}$ by bilinear interpolation [19] and use it to fill $\mathcal{Z}$. For the rest of $\mathcal{I}$, we fill them with 0.

Note that the *indicator* $\mathcal{I}$ generated in this way can well distinguish the target object from the context information. Moreover, the spatial relationship between the local proposal and the expanded proposal is preserved, so even if the prediction on the local proposal is not correct, *e.g.* the wrong prediction cases, we can still know the target object as long as there is a dominant one. This is because the non-dominant object will have many zeros on the *indicator*, while the dominant one does not, so the target object is still clear.

3.3. TOPNet

In this section, we show how to incorporate the TOP module into existing proposal-based frameworks. The augmented framework, which we call TOPNet, allows inference on the expanded proposal. Here we only use Mask R-CNN [16] as an example, but the design is general and should be applicable to other methods too.

Mask R-CNN. We start by a brief introduction of the Mask R-CNN [16] framework. It's a conceptually simple two-stage framework, where the first stage generates a set of candidate proposals and the second stage performs inference on these proposals. The first stage is a region proposal network (RPN) proposed by [33]. The second stage consists of three branches, a *classification* branch for classifying object category, a *box* branch for regressing the bounding box boundary, and a *mask* branch to predict the object mask within the proposal.

TOPNet. Different from Mask R-CNN [16], TOPNet performs inference on the expanded proposal. An illustration of the TOPNet is shown in Figure 3. Apart from the *box* branch and *classification* branch, it has a TOP module to specify the target object and a *prediction* branch to perform inference on the expanded proposal.

TOPNet can be readily extended from the Mask R-CNN framework [16]. Specifically, we use the *mask* branch in Mask R-CNN as the network N in the TOP module, since the object mask can well represent the target object within the local proposal. We further add an extra *prediction* branch in parallel with the *box* branch and *classification* branch to perform inference on the expanded proposal.

Inference. The inference of TOPNet consists of two steps. First, for each proposal, the TOP module generates an *indicator* $\mathcal{I}$ to specify the target object. Then the *prediction* branch combines the *indicator* with the feature from the expanded proposal to perform inference. Since it's straightforward to obtain $\mathcal{I}$ by running the TOP module, we focus the discussion on the second step.

Specifically, we extract a feature $\mathcal{X}_e$ from the expanded proposal $\mathcal{B}_e$ using the *RoIAlign* operation. The resolution of $\mathcal{X}_e$ is set to $H_\mathcal{I} \times W_\mathcal{I}$ in order to match the resolution of the *indicator* $\mathcal{I}$. Then we concatenate $\mathcal{X}_e$ with $\mathcal{I}$ and feed the concatenated feature into the *prediction* branch to obtain the prediction on the expanded proposal.

We explain why a simple concatenation works here. From the input's perspective, the feature $\mathcal{X}_e$ contains all information of all objects within the expanded proposal $\mathcal{B}_e$, and the *indicator* $\mathcal{I}$ highlights the target object in $\mathcal{B}_e$, so a simple concatenation of both is sufficient to generate an input that has no ambiguity but covers all necessary information in order to make the correct prediction. Fusing them in a more complex way could possibly get better performance, but that's beyond the scope of this paper.

Training. Our framework can be trained in an end-to-end manner using standard back-propogation algorithms. Specifically, the loss L for each proposal is defined as $L = L_{cls} + L_{box} + L^e_{mask}$, similar to the multi-task loss in [16]. The classification loss L_{cls} and the box localization loss L_{box} are the same as those in [13, 33]. The mask loss L^e_{mask} is the average binary cross entropy loss between the predicted mask and the binary mask label. Note that L^e_{mask} is the mask loss on the expanded proposal and the e superscript denotes the expanded proposal.

In the above formulation, because the TOP module is fully differentiable, $\mathcal{T}$ can be automatically learned using gradients from the final mask loss L^e_{mask}. However, we empirically found applying an auxiliary loss for $\mathcal{T}$ can yield better performance. This is probably because the automatically learned $\mathcal{T}$ may not indicate the target object well, so the *indicator* becomes somehow ambiguous. By introducing the auxiliary loss, we impose constraints on $\mathcal{T}$ so that $\mathcal{I}$ becomes clearer and leads to better performance.

The precise form of the auxiliary loss depends on the specific task. For example, in the instance segmentation task, we require $\mathcal{T}$ to predict the object mask within the local proposal, so the auxiliary loss is L^l_{mask} and the l superscript denotes the local proposal. After introducing the auxiliary loss, the final loss for our framework is

$$L = L_{cls} + L_{box} + L^e_{mask} + L^l_{mask}$$

4. Experiments

We conduct experiments on the instance segmentation task and pose estimation task to demonstrate the effectiveness of our method. Both tasks can be formulated as problems that require bounding box detection, thus can be addressed by proposal-based methods. We start by introducing the instance segmentation experiment and then talk about the human pose estimation experiment.

4.1. Instance Segmentation

We apply our method to the challenging instance segmentation task. Experiments are carried out on the COCO [23] instance segmentation set and results are evaluated against the standard COCO metrics. We adopt Mask R-CNN [16] as our baseline, which is a strong and popular framework in instance segmentation.

Dataset. COCO [23] instance segmentation dataset is one of the most challenging datasets in instance segmentation. Complex scenes and frequently occurring occlusions make it extremely difficult to achieve good performance on this dataset. The training set (*train-2017*) consists of roughly 115k images and the validation set (*val-2017*) holds about 5k images. The annotations for these two sets are provided. The testing images are divided into two subsets, the *test-dev* for development purposes and the *test-challenge* for challenge purposes only. They both hold 20k images and their annotations are not available.

Implementation Details. All models are trained on *train-2017* and evaluated on *val-2017* and *test-dev*. The training settings are the same as in Mask R-CNN [16]. Specifically, we use 8 GPUs to train the network. The batch size for each GPU is 2, so the total batch size is 16. For each image, we sample 512 regions-of-interest (RoIs) from the RPN proposals and set the ratio between positive and negative proposals to 1:3. The models are trained with 90k iterations in total. The learning rate starts from 0.02 and decreases 10 times at iteration 60k and 80k.

The details for TOPNet are as follows. We set the expanding scale to 2, so the expanded proposal is twice as large as the local proposal. For the TOP module, the network N consists of a *RoIAlign* layer and four 3×3 convolutional layers with 256 channels and a deconvolutional layer. The output resolution of the *RoIAlign* layer is 14×14 and the resolution of the *indicator* is 28×28.

The *prediction* branch consists of a *RoIAlign* layer and eight 3×3 convolutional layers with channels 256 as well as a deconvolutional layer. The output resolution of *RoIAlign* is 28×28; the resolution of final prediction is 56×56. Note that the *RoIAlign* output resolution here is twice as large as the one in the TOP module due to the expansion ratio.

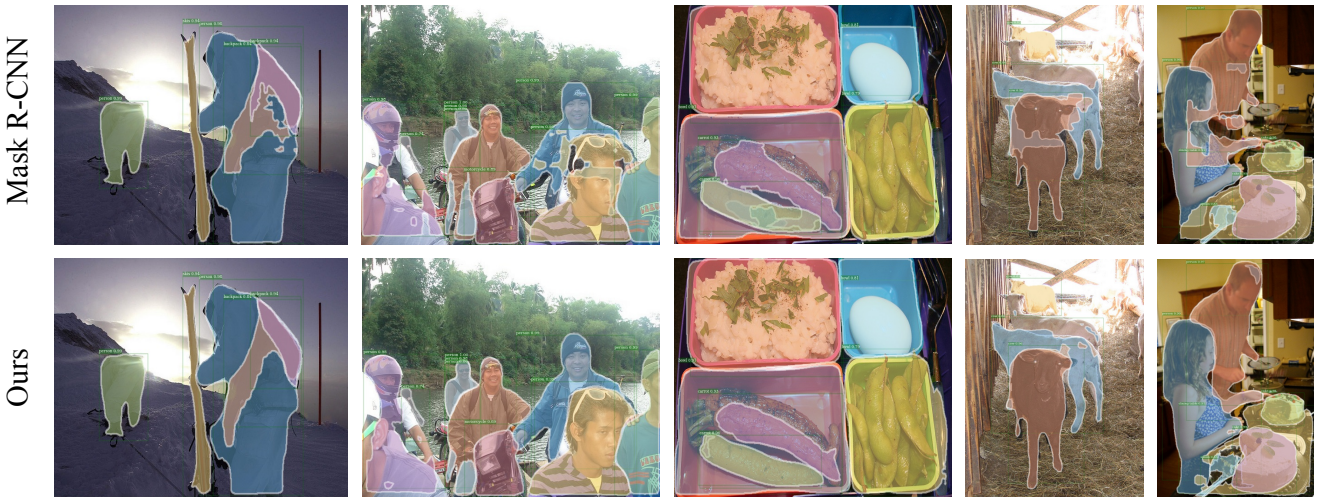


Figure 4: Qualitative comparisons between our method and Mask R-CNN baseline. Our method can effectively utilize context and corrects some cases where the original Mask R-CNN fails.

test-dev	backbone	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
Mask R-CNN	ResNet-50-FPN	34.2	56.4	36.0	14.8	36.0	49.7
Mask R-CNN	ResNet-101-FPN	35.9	58.5	38.0	15.9	38.2	51.8
Ours	ResNet-50-FPN	35.4	56.4	37.8	15.3	37.0	51.9
Ours	ResNet-101-FPN	**37.0**	**58.6**	**39.6**	**16.3**	**38.9**	**54.0**

(a) Results for *test-dev*

val-2017	backbone	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
Mask R-CNN	ResNet-50-FPN	33.9	55.8	35.8	14.9	36.3	50.9
Mask R-CNN	ResNet-101-FPN	35.9	58.3	38.0	15.9	38.9	53.2
Ours	ResNet-50-FPN	35.4	56.3	37.8	15.8	37.6	53.5
Ours	ResNet-101-FPN	**37.1**	**58.4**	**39.6**	**16.9**	**39.8**	**55.7**

(b) Results for *val-2017*

Table 1: Results on the COCO *test-dev* and *val-2017* subsets.

Results We compare our method with state-of-the-art algorithm Mask R-CNN [16] on the *test-dev* and *val-2017* subsets. Results under the standard COCO metrics are reported in Table 1. We observe consistent improvements over Mask R-CNN with different backbones (1.5 mAP on ResNet-50-FPN and 1.1 mAP on ResNet-101-FPN). The improvements mainly come from the benefit of inference on the expandped proposal. Note that our method is orthogonal to the development of backbone network and could achieve better performance when a stronger backbone network is employed.

Moreover, the improvement at a high IoU threshold is much larger than that at a low threshold (2.0 point for AP_{75} *vs*. 0.5 point for AP_{50}). This is because under a low threshold, even the wrong predictions could be treated as correct. With a relatively high threshold, contrarily, wrong predictions are more easily eliminated. We thus advocate that high IoU threshold benefits from our strategy.

Qualitative Results We provide qualitative comparisons between our method and the Mask R-CNN baseline in Figure 4. Mask R-CNN may fail when multiple instances appear in one proposal and the activation is on the wrong object. Our method effectively ameliorates this problem by utilizing the context information from the expanded proposal. The second, fourth and fifth columns in Figure 4 provide illustration of these cases.

4.2. Human Pose Estimation

In this section, we apply our method to another challenging task, human pose estimation. We show that our design generalizes well and can benefit different tasks that require bounding box detection.

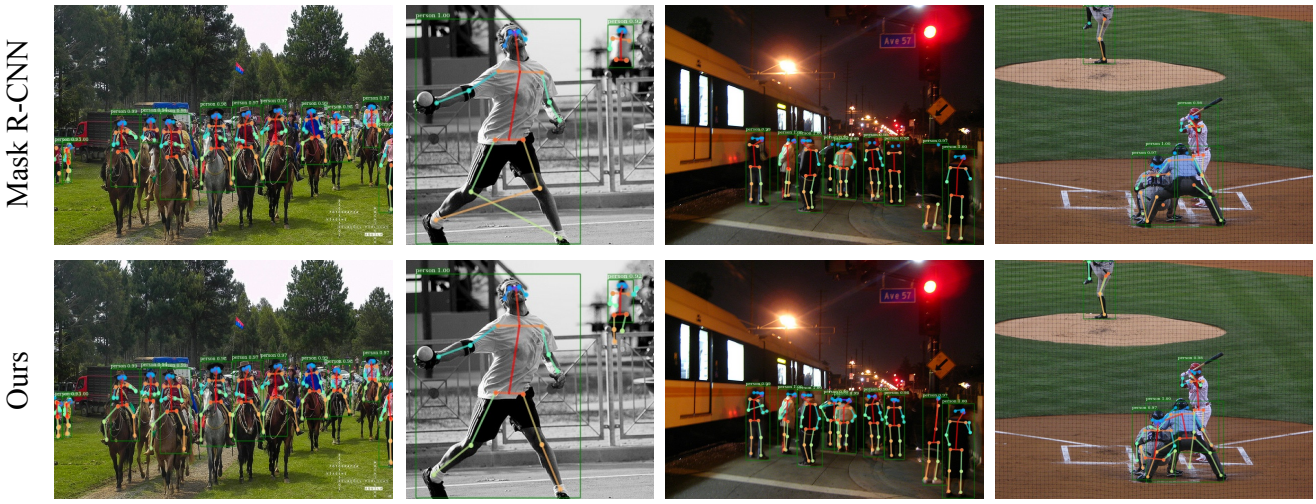


Figure 5: Keypoint detection results compared with Mask R-CNN.

test-dev	AP^{kp}	AP^{kp}_{50}	AP^{kp}_{75}	AP^{kp}_{M}	AP^{kp}_{L}
Mask R-CNN	62.7	**87.0**	68.4	57.4	71.1
Ours	**64.6**	**87.0**	**71.7**	**60.2**	**72.3**

(a) Results on COCO *test-dev*

val-2017	AP^{kp}	AP^{kp}_{50}	AP^{kp}_{75}	AP^{kp}_{M}	AP^{kp}_{L}	AP^{box}
Mask R-CNN	64.2	86.4	69.9	58.5	73.4	**53.6**
Ours	**66.1**	**86.5**	**72.6**	**61.8**	**74.1**	50.8

(b) Results on COCO *val-2017*

Table 2: Results of keypoint detection on COCO *test-dev* and *val-2017*. Both use ResNet-50-FPN as the backbone network. AP^{kp} is AP under keypoint metric. AP_{box} is the AP for box detection (person only).

Dataset We carry out experiments on the COCO keypoint dataset [23], which requires both accurate human instances detection and precise human keypoints localization. The training set contains over 100K human instances with over 1 million labeled keypoints. The testing set consists of roughly 80000 human instances, and is equally divided into four subsets, namely *test-challenge*, *test-dev*, *test-standard* and *test-preserved*.

Implementation details As in [16], we perform some minor modifications to adapt to the pose estimation task. Specifically, the keypoint location is represented by a one-hot mask, and we require the *prediction* branch to predict a K-channel mask, each channel for a kind of keypoints (*e.g.* left wrist, right knee). The network N within the TOP module follows the same changes in [16]. It now consists of eight 3x3 convolution layers with 512 channels and a deconv layer. An extra 2x bilinear upsampling layer is added to increase the resolution. The final resolution for $\mathcal{T}$ is 56×56. Note the resolution for the *indicator* is still 28×28. The architecture of *prediction* branch is similar to N but doubles the number of convolutional layers. The resolution for the final prediction is 112×112.

Results We report results on the "test-dev" and "val-2017" subsets in Table 2. Our method outperforms Mask R-CNN by a large margin (close to 2 mAP). We notice the improvement on a high threshold is much larger than the improvement on a low one (3.3 point on AP^{kp}_{75} *vs.* $0 \sim 0.1$ point on AP^{kp}_{50}). The difference again comes from correcting the failure cases, which were originally regarded as correct on the low threshold.

It is to our surprise that although the box quality of our method is worse than that of Mask R-CNN (2.8 mAP lower), our method still accomplishes better keypoint detection eventually. This again demonstrates the general ability of our method to produce high quality results.

Qualitative Results Results of our method and Mask R-CNN are shown in Figure 5. Our method manages to find

and locate keypoints that are missed by Mask R-CNN.

5. Ablation Studies

In this section, we perform exhaustive ablation studies to analyze the proposed method in detail. Note that the experiments are conducted on the instance segmentation task. For results on the pose estimation part, please refer to the supplementary file.

We use ResNet-50-FPN as the backbone network throughout all experiments and results on *val-2017* are reported. Note that we slightly change the training settings to save time cost. Specifically, the number of RoIs per image is reduced from 512 to 128 and the total iteration number is reduced from 90k to 45k. The learning rate decreases by a factor of 10 at the 30k and 40k iteration. Although the precise number is different from the complete version, the improvement tendency is consistent and thus is quite desirable.

TOP module: We first study the influence of the TOP module. We remove it from the pipeline and directly train a model on the expanded proposal. This experiment is denoted as *w/o TOP module* and the result is reported on Table 3(a). We observe a significant performance decrease (close to 2 mAP) compared to the Mask R-CNN baseline, due to the ambiguity issue introduced by expanding the proposal. Incorporating the TOP module can greatly ease this problem and leads to better performance.

Context: We then investigate the effect of context information. In this experiment, we keep the attention module but do not perform inference on the expanded proposal, which reduces our method to a simple refinement process. The experiment is denoted as *w/o context* and the result is reported on Table 3(b). As we can see, a small improvement (0.3 mAP) over the *baseline* is achieved, due to the refinement process. But still, it's not compatible with the *full-model* (0.8 mAP less), which benefits from employing the context information.

Auxiliary loss: We further analyze the impact of auxiliary loss. We remove the auxiliary loss and automatically learn the attention map without any specific requirement. This experiment is denoted as *w/o auxiliary loss* and result is reported on Table 3(c). Without the auxiliary loss, the performance drops about 0.7 mAP. But still, it performs better than the baseline (0.5 mAP improvement).

Inference on the expanded proposal: Finally, we study the influence of the precise method of performing inference on the expanded proposal. MultiPath [40] also uses expanded proposal for inference. Different from ours, it directly concatenates the feature from the local proposal and the feature from the expanded proposal to perform the final prediction. Although it's originally proposed for object detection, it's applicable to instance segmentation too. Therefore, we extend their method to instance segmentation and report result on Table 4. Our method obtains much better result than MultiPath [40], although they both perform inference on the expanded proposal. The main reason is that our *indicator* preserves the spatial relationship between the local proposal and the expanded proposal, while MultiPath loses that information. This also proves the effectiveness of the TOP module.

val-2017	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
baseline	31.3	52.1	32.9	12.8	33.6	47.5
full-model	**32.5**	**52.9**	**34.7**	**13.1**	**34.5**	**50.1**
(a) *w/o TOP module*	29.4	50.1	30.4	12.0	31.4	45.1
(b) *w/o context*	31.7	52.3	33.4	12.9	33.9	48.9
(c) *w/o auxiliary loss*	31.8	52.0	33.7	12.8	33.5	48.7

Table 3: Ablation studies results. baseline and full-model are the results of Mask R-CNN and TOPNet respectively. (a) *w/o TOP module* is the result of removing the TOP module. (b) *w/o context* is the result of removing the context information. (c) *w/o auxiliary loss* is the result of removing the auxiliary loss.

val-2017	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
MultiPath [40]	30.4	50.9	31.6	12.7	32.6	46.2
Ours	**32.2**	**52.3**	**34.0**	**13.2**	**34.0**	**49.8**

Table 4: Ablation studies results. MultiPath is using the method in [40] to perform inference on the expanded proposal. **Ours** is the result of TOPNet.

6. Conclusion

In this paper, we investigated two common problems in proposal-based frameworks, *i.e*. incomplete prediction and wrong prediction. We showed that expanding the proposal could greatly alleviate these problems, but cares must be taken to avoid breaking the 'one proposal-one instance' assumption. A novel attention module, namely TOP module, was proposed to address the crucial ambiguity issue. It could be readily incorporated into existing proposal-based frameworks to allow the inference on the expanded proposal. The augmented method, which we called TOPNet, achieved significant improvement over the original framework. Experiments on instance segmentation and human pose estimation demonstrated the effectiveness and generality of our method.

References

[1] Min Bai and Raquel Urtasun. Deep watershed transform for instance segmentation. In *CVPR*, 2017. 2

[2] Adrian Bulat and Georgios Tzimiropoulos. Human pose estimation via convolutional part heatmap regression. In *European Conference on Computer Vision*. Springer, 2016. 3

[3] Zhaowei Cai and Nuno Vasconcelos. Cascade R-CNN: delving into high quality object detection. In *CVPR*, 2018. 2

[4] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. *arXiv preprint arXiv:1611.08050*, 2016. 3

[5] Yilun Chen, Zhicheng Wang, Yuxiang Peng, Zhiqiang Zhang, Gang Yu, and Jian Sun. Cascaded pyramid network for multi-person pose estimation. *arXiv preprint arXiv:1711.07319*, 2017. 1, 2, 3

[6] Xiao Chu, Wei Yang, Wanli Ouyang, Cheng Ma, Alan L Yuille, and Xiaogang Wang. Multi-context attention for human pose estimation. *arXiv preprint arXiv:1702.07432*, 2017. 3

[7] Jifeng Dai, Kaiming He, and Jian Sun. Convolutional feature masking for joint object and stuff segmentation. In *CVPR*, 2015. 2

[8] Jifeng Dai, Kaiming He, and Jian Sun. Instance-aware semantic segmentation via multi-task network cascades. *CVPR*, 2016. 2

[9] Jifeng Dai, Yi Li, Kaiming He, and Jian Sun. R-FCN: object detection via region-based fully convolutional networks. *CoRR*, 2016. 2

[10] Hao-Shu Fang, Shuqin Xie, Yu-Wing Tai, and Cewu Lu. RMPE: Regional multi-person pose estimation. In *ICCV*, 2017. 1, 2, 3

[11] Alireza Fathi, Zbigniew Wojna, Vivek Rathod, Peng Wang, Hyun Oh Song, Sergio Guadarrama, and Kevin P. Murphy. Semantic instance segmentation via deep metric learning. *CoRR*, 2017. 2

[12] Spyros Gidaris and Nikos Komodakis. Object detection via a multi-region and semantic segmentation-aware CNN model. In *ICCV*, 2015. 3

[13] Ross Girshick. Fast R-CNN. In *ICCV*, 2015. 2, 5

[14] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, 2014. 2

[15] Bharath Hariharan, Pablo Arbeláez, Ross Girshick, and Jitendra Malik. Simultaneous detection and segmentation. In *ECCV*. 2014. 2

[16] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross B. Girshick. Mask R-CNN. In *ICCV*, 2017. 1, 2, 3, 4, 5, 6, 7

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. *PAMI*, 2015. 2

[18] Eldar Insafutdinov, Leonid Pishchulin, Bjoern Andres, Mykhaylo Andriluka, and Bernt Schiele. Deepercut: A deeper, stronger, and faster multi-person pose estimation model. In *European Conference on Computer Vision*. Springer, 2016. 3

[19] Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al. Spatial transformer networks. In *Advances in neural information processing systems*, 2015. 4

[20] Alexander Kirillov, Evgeny Levinkov, Bjoern Andres, Bogdan Savchynskyy, and Carsten Rother. Instancecut: From edges to instances with multicut. In *CVPR*, 2017. 2

[21] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. ImageNet classification with deep convolutional neural networks. In *NIPS*, 2012. 2

[22] Tsung-Yi Lin, Piotr Dollár, Ross B. Girshick, Kaiming He, Bharath Hariharan, and Serge J. Belongie. Feature pyramid networks for object detection. In *CVPR*, 2017. 2

[23] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft COCO: Common objects in context. In *ECCV*. 2014. 1, 2, 5, 7

[24] Shu Liu, Jiaya Jia, Sanja Fidler, and Raquel Urtasun. SGN: Sequential grouping networks for instance segmentation. In *ICCV*, 2017. 2

[25] Shu Liu, Lu Qi, Haifang Qin, Jianping Shi, and Jiaya Jia. Path aggregation network for instance segmentation. In *CVPR*, 2018. 1, 2

[26] Alejandro Newell and Jia Deng. Associative embedding: End-to-end learning for joint detection and grouping. *CoRR*, 2016. 2, 3

[27] Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation. In *European Conference on Computer Vision*. Springer, 2016. 3

[28] George Papandreou, Tyler Zhu, Nori Kanazawa, Alexander Toshev, Jonathan Tompson, Chris Bregler, and Kevin Murphy. Towards accurate multi-person pose estimation in the wild. *arXiv preprint arXiv:1701.01779*, 2017. 1, 3

[29] Pedro H. O. Pinheiro, Ronan Collobert, and Piotr Dollár. Learning to segment object candidates. In *NIPS*, 2015. 2

[30] Leonid Pishchulin, Mykhaylo Andriluka, Peter Gehler, and Bernt Schiele. Poselet conditioned pictorial structures. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013. 3

[31] Leonid Pishchulin, Eldar Insafutdinov, Siyu Tang, Bjoern Andres, Mykhaylo Andriluka, Peter V Gehler, and Bernt Schiele. Deepcut: Joint subset partition and labeling for multi person pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 3

[32] Mengye Ren and Richard S. Zemel. End-to-end instance segmentation with recurrent attention. In *CVPR*, 2017. 3

[33] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *NIPS*, 2015. 2, 4, 5

[34] Bernardino Romera-Paredes and Philip Hilaire Sean Torr. Recurrent instance segmentation. In *ECCV*, 2016. 3

[35] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2014. 2

[36] Alexander Toshev and Christian Szegedy. Deeppose: Human pose estimation via deep neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014. 3

[37] Jasper RR Uijlings, Koen EA van de Sande, Theo Gevers, and Arnold WM Smeulders. Selective search for object recognition. *IJCV*, 2013. 2

[38] Shih-En Wei, Varun Ramakrishna, Takeo Kanade, and Yaser Sheikh. Convolutional pose machines. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 3

[39] Sergey Zagoruyko, Adam Lerer, Tsung-Yi Lin, Pedro H. O. Pinheiro, Sam Gross, Soumith Chintala, and Piotr Dollár. A multipath network for object detection. In *BMVC*, 2016. 3

[40] Sergey Zagoruyko, Adam Lerer, Tsung-Yi Lin, Pedro O Pinheiro, Sam Gross, Soumith Chintala, and Piotr Dollár. A multipath network for object detection. *arXiv preprint arXiv:1604.02135*, 2016. 8

[41] Xingyu Zeng, Wanli Ouyang, Junjie Yan, Hongsheng Li, Tong Xiao, Kun Wang, Yu Liu, Yucong Zhou, Bin Yang, Zhe Wang, Hui Zhou, and Xiaogang Wang. Crafting GBD-Net for object detection. *arXiv:1610.02579*, 2016. 3