Introduction

1. This method provide the end-to-end training comparing to previous methods.

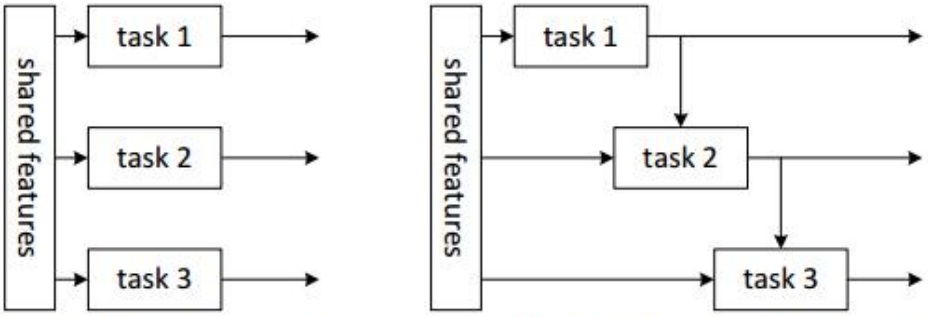

Figure 1. Illustrations of common multi-task learning (left) and our multi-task cascade (right).

2. The authors develop a differentiable ROI warping layer to account for the gradient of predicted box positions.

3.

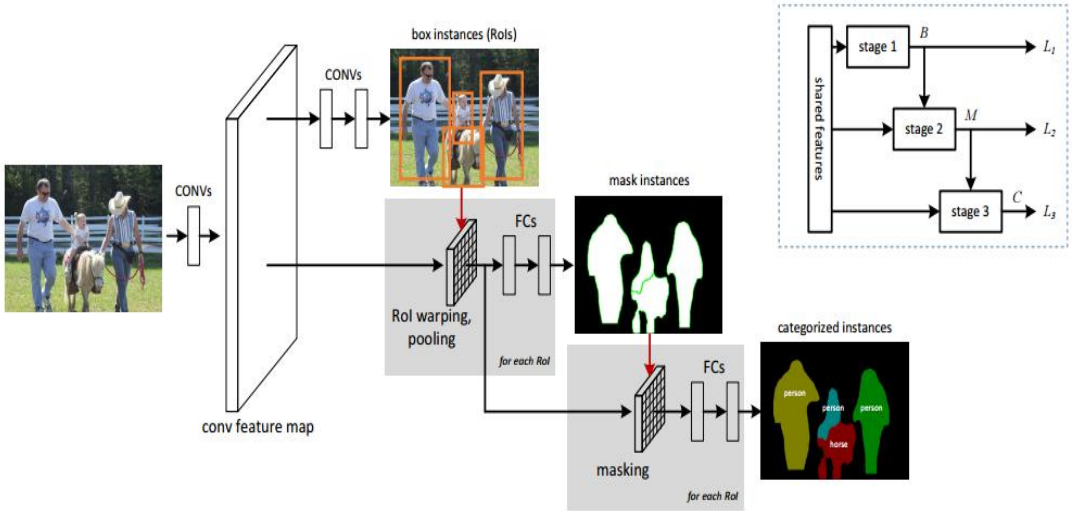

Figure 2. Multi-task Network Cascades for instance-aware semantic segmentation. At the top right corner is a simplified illustration.

Related Work

Roi Pooling

The purpose of RoI pooling is for producing a fixed-size feature from an arbitrary box.

Implementing Details

1.Regressing Box-level Instances

Just follow the work of Region Proposal Networks.

2.Regressing Mask-level Instances

1.The second stage takes the shared convolutional features and stage-1 boxes as input. It outputs a pixel-level segmentation mask for each box proposal. In this stage, a mask-level instance is still class-agnostic.

2.Our method only regresses masks from a few proposed boxes and so reduces computational cost.

3.We append two extra fully-connected (fc) layers to this feature for each box. The first fc layer (with ReLU) reduces the dimension to 256, followed by the second fc layer that

regresses a pixel-wise mask. The second fc layer has m^2 outputs, each performing binary logistic regression to the ground truth mask.

3.Categorizing Instances

The third stage takes the shared convolutional features, stage-1 boxes, and stage-2 masks as input. It outputs category scores for each instance.

Differentiable RoI Warping Layers.

Here F(Θ) is reshaped as an n-dimensional vector.

$$\mathcal{F}_i^{RoI}(\Theta) = G(B_i(\Theta))\mathcal{F}(\Theta).$$

$$\mathcal{F}_i^{RoI}{}_{(u',v')} = \sum_{(u,v)}^{W\times H} G(u,v;u',v'|B_i)\mathcal{F}_{(u,v)},$$

$$g(u,u'|x_i,w_i) = \kappa(x_i + \frac{u'}{W'}w_i - u),$$

$$\kappa(\cdot) = max(0,1 - |\cdot|)$$

(u',v') stands for the position in the pre-defined RoI warping output map. $x_i + \frac{u'}{W'}w_i - u$ is the position of (u',v') in features map with respect to the bounding box. So G describes

the related score from features map and bounding box to the new RoI warping output.And according the Chain Rule ,we can backpropagate on the bounding box information (xi,yi,wi,hi).

$$\frac{\partial L_2}{\partial B_i} = \frac{\partial L_2}{\partial \mathcal{F}_i^{RoI}} \frac{\partial G}{\partial B_i} \mathcal{F}$$

G is set as 28*28.A max pooling layer is then applied to produce a lower-resolution output, e.g., 7×7 for VGG-16.

Then we append a **Masking Layers**..

$$\mathcal{F}_i^{Mask}(\Theta) = \mathcal{F}_i^{RoI}(\Theta) \cdot M_i(\Theta).$$

4.Cascades with More Stages

On stage 3, we add a 4(N+1)-d fc layer for regression class-wise bounding boxes , which is a sibling layer with the classifier layer.So we can obtain the new region proposals.Stages 2 and 3 are performed for the second time on these proposals.The new stages 4 and 5 share the same structures as stages 2 and 3, except that they use the regressed boxes from stage 3 as the new proposals.And the backpropagation from Stage 5 can help us to better converge.

5.Others

1.Non-maximum suppression

Just to eliminate the redundant bounding boxes.

The threshold of the Intersection-over-Union (IoU) ratio for this NMS is 0.7. After that, the top-ranked 300 boxes [26] will be used for stage 2.

2.Positive/negative samples

On stage 2, for each proposed box we find its highest overlapping ground truth mask. If the overlapping ratio (IoU) is greater than 0.5, this proposed box is considered as positive and contributes to the mask regression loss; otherwise is ignored in the regression loss. The mask regression target is the intersection between the proposed box and the ground truth mask, resized to $m \times m$ pixels.

On stage 3, we consider two sets of positive/negative samples. In the first set, the positive samples are the instances that overlap with ground truth boxes by box-level IoU $\geqslant$ 0.5 (the negative samples are the rest). In the second set, the positive samples are the instances that overlap with ground truth instances by box-level IoU $\geqslant$ 0.5 and mask-level IoU $\geqslant$ 0.5.

The loss function of stage 3 involves two (N+1)-way classifiers, one for classifying mask-level instances and the other for classifying box-level instances (whose scores are not used for inference).

3.Hyper-parameters

We use the ImageNet pretrained models (e.g., VGG-16) to initialize the shared convolutional layers and the corresponding 4096-d fc layers. The extra layers are initialized randomly.

In our system, each mini-batch involves 1 image, 256 sampled anchors for stage 1 as in [26] 2 , and 64 sampled RoIs for stages 2 and 3.

We train the model using a learning rate of 0.001 for 32k iterations, and 0.0001 for the next 8k.

We train the model in 8 GPUs, each GPU holding 1 mini-batch. The images are resized such that the shorter side has 600 pixels.

4.Inference

We use 5-stage inference for both 3-stage and 5-stage trained structures. The inference process gives us a list of 600

instances with masks and category scores (300 from the stage 3 outputs, and 300 from the stage 5 outputs). We post-process this list to reduce similar predictions. We first apply NMS (using box-level IoU 0.3 [10]) on the list of 600 instances based on their category scores. The prediction masks of the not-suppressed instance and its similar instances are merged together by weighted averaging, pixel-by-pixel, using the classification scores as their averaging weights. The averaged masks, taking continuous values in [0, 1], are binarized to form the final output masks. The averaging step improves accuracy by ~ 1% over the NMS outcome. This postprocessing is performed for each category independently.

Results

Ablation Experiments

	ZF net				VGG-16 net			
training strategies	(a)	(b)	(c)	(d)	(a)	(b)	(c)	(d)
shared features?		✓	✓	✓		✓	✓	✓
end-to-end training?			✓	✓			✓	✓
training 5-stage cascades?				✓				✓
mAP^r@0.5 (%)	51.8	52.2	53.5	**54.0**	60.2	60.5	62.6	**63.5**

Comparisons with State-of-the-art Methods

method	mAP^r@0.5 (%)	mAP^r@0.7 (%)	time/img (s)
O^2P [2]	25.2	-	-
SDS (AlexNet) [13]	49.7	25.3	48
Hypercolumn [14]	60.0	40.4	>80
CFM [7]	60.7	39.6	32
MNC [ours]	**63.5**	**41.5**	**0.36**

On the PASCAL VOC 2012 validation set.

Object Detection Evaluations

Given mask-level instances generated by our model, we simply assign a tight bounding box to each instance.

Our method also has box-level outputs from the box regression layers in stage 3/5.

system	training data	mAP^b (%)
R-CNN [10]	VOC 12	62.4
Fast R-CNN [9]	VOC 12	65.7
Fast R-CNN [9]	VOC 07++12	68.4
Faster R-CNN [26]	VOC 12	67.0
Faster R-CNN [26]	VOC 07++12	70.4
MNC [ours]	VOC 12	70.9
MNC_{box} [ours]	VOC 12	73.5
MNC_{box} [ours]†	VOC 07++12	**75.9**

This give us the intuition that mask-level annotation can improve the performance on Object Detection.

Experiments on MS COCO Segmentation

network	mAP@[.5:.95] (%)	mAP@.5 (%)
VGG-16 [27]	19.5	39.7
ResNet-101 [16]	**24.6**	**44.3**

Table 5. Our baseline segmentation result (%) on the MS COCO *test-dev* set. The training set is the *trainval* set.