Learning to Segment Every Thing

Ronghang Hu[1,2,*] Piotr Dollár[2] Kaiming He[2] Trevor Darrell[1] Ross Girshick[2]

[1]BAIR, UC Berkeley [2]Facebook AI Research (FAIR)

Abstract

Most methods for object instance segmentation require all training examples to be labeled with segmentation masks. This requirement makes it expensive to annotate new categories and has restricted instance segmentation models to ~100 well-annotated classes. The goal of this paper is to propose a new partially supervised training paradigm, together with a novel weight transfer function, that enables training instance segmentation models on a large set of categories all of which have box annotations, but only a small fraction of which have mask annotations. These contributions allow us to train Mask R-CNN to detect and segment 3000 visual concepts using box annotations from the Visual Genome dataset and mask annotations from the 80 classes in the COCO dataset. We evaluate our approach in a controlled study on the COCO dataset. This work is a first step towards instance segmentation models that have broad comprehension of the visual world.

1. Introduction

Object detectors have become significantly more accurate (*e.g.*, [10, 34]) and gained important new capabilities. One of the most exciting is the ability to predict a foreground segmentation mask for each detected object (*e.g.*, [15]), a task called *instance segmentation*. In practice, typical instance segmentation systems are restricted to a narrow slice of the vast visual world that includes only around 100 object categories.

A principle reason for this limitation is that state-of-the-art instance segmentation algorithms require *strong supervision* and such supervision may be limited and expensive to collect for new categories [23]. By comparison, bounding box annotations are more abundant and less expensive [4]. This fact raises a question: Is it possible to train high-quality instance segmentation models without complete instance segmentation annotations for all categories? With this motivation, our paper introduces a new *partially supervised* instance segmentation task and proposes a novel transfer learning method to address it.

*Work done during an internship at FAIR.

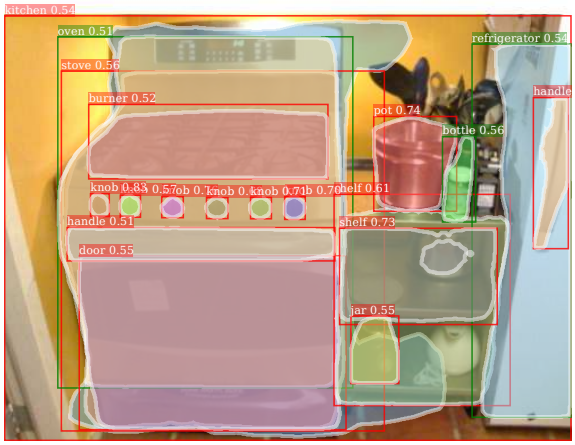

Figure 1. **We explore training instance segmentation models with partial supervision**: a subset of classes (green boxes) have instance mask annotations during training; the remaining classes (red boxes) have only bounding box annotations. This image shows output from our model trained for 3000 classes from Visual Genome, using mask annotations from only 80 classes in COCO.

We formulate the partially supervised instance segmentation task as follows: (1) given a set of categories of interest, a small *subset* has instance mask annotations, while the other categories have only bounding box annotations; (2) the instance segmentation algorithm should utilize this data to fit a model that can segment instances of *all* object categories in the set of interest. Since the training data is a mixture of strongly annotated examples (those with masks) and weakly annotated examples (those with only boxes), we refer to the task as *partially supervised*.

The main benefit of partially supervised *vs*. weakly-supervised training (*c.f*. [18]) is it allows us to build a large-scale instance segmentation model by exploiting both types of existing datasets: those with bounding box annotations over a large number of classes, such as Visual Genome [20], and those with instance mask annotations over a small number of classes, such as COCO [23]. As we will show, this enables us to scale state-of-the-art instance segmentation methods to thousands of categories, a capability that is critical for their deployment in real world uses.

To address partially supervised instance segmentation, we propose a novel *transfer learning* approach built on Mask R-CNN [15]. Mask R-CNN is well-suited to our

task because it decomposes the instance segmentation problem into the subtasks of bounding box object detection and mask prediction. These subtasks are handled by dedicated network 'heads' that are trained jointly. The intuition behind our approach is that once trained, the parameters of the bounding box head encode an embedding of each object category that enables the transfer of visual information for that category to the partially supervised mask head.

We materialize this intuition by designing a parameterized *weight transfer function* that is trained to predict a category's instance segmentation parameters as a function of its bounding box detection parameters. The weight transfer function can be trained end-to-end in Mask R-CNN using classes with mask annotations as supervision. At inference time, the weight transfer function is used to predict the instance segmentation parameters for *every* category, thus enabling the model to segment all object categories, including those without mask annotations at training time.

We explore our approach in two settings. First, we use the COCO dataset [23] to simulate the partially supervised instance segmentation task as a means of establishing quantitative results on a dataset with high-quality annotations and evaluation metrics. Specifically, we split the full set of COCO categories into a subset with mask annotations and a complementary subset for which the system has access to only bounding box annotations. Because the COCO dataset involves only a small number (80) of semantically well-separated classes, quantitative evaluation is precise and reliable. Experimental results show that our method improves results over a strong baseline with up to a 40% relative increase in mask AP on categories without training masks.

In our second setting, we train a *large-scale* instance segmentation model on 3000 categories using the Visual Genome (VG) dataset [20]. VG contains bounding box annotations for a large number of object categories, however quantitative evaluation is challenging as many categories are semantically overlapping (*e.g.*, near synonyms) and the annotations are not exhaustive, making precision and recall difficult to measure. Moreover, VG is not annotated with instance masks. Instead, we use VG to provide *qualitative* output of a large-scale instance segmentation model. Output of our model is illustrated in Figure 1 and 5.

2. Related Work

Instance segmentation. Instance segmentation is a highly active research area [12, 13, 5, 32, 33, 6, 14, 21, 19, 2], with Mask R-CNN [15] representing the current state-of-the-art. These methods assume a fully supervised training scenario in which *all* categories of interest have instance mask annotations during training. Fully supervised training, however, makes it difficult to scale these systems to thousands of categories. The focus of our work is to relax this assumption and enable training models even when masks are available for only a small subset of categories. To do this, we develop a novel transfer learning approach built on Mask R-CNN.

Weight prediction and task transfer learning. Instead of directly learning model parameters, prior work has explored *predicting* them from other sources (*e.g.*, [11]). In [8], image classifiers are predicted from the natural language description of a zero-shot category. In [38], a model regression network is used to construct the classifier weights from few-shot examples, and similarly in [27], a small neural network is used to predict the classifier weights of the composition of two concepts from the classifier weights of each individual concept. Here, we design a model that predicts the class-specific instance segmentation weights used in Mask R-CNN, instead of training them directly, which is not possible in our partially supervised training scenario.

Our approach is also a type of transfer learning [28] where knowledge gained from one task helps with another task. Most related to our work, LSDA [17] transforms whole-image classification parameters into object detection parameters through a domain adaptation procedure. LSDA can be seen as transferring knowledge learned on an image classification task to an object detection task, whereas we consider transferring knowledge learned from bounding box detection to instance segmentation.

Weakly supervised semantic segmentation. Prior work trains semantic segmentation models from weak supervision. Image-level labels and object size constraints are used in [30], while other methods use boxes as supervision for expectation-maximization [29] or iterating between proposals generation and training [4]. Point supervision and objectness potentials are used in [3]. Most work in this area addresses only semantic segmentation (not *instance* segmentation), treats each class independently, and relies on hand-crafted bottom-up proposals that generalize poorly.

Weakly supervised instance segmentation is addressed in [18] by training an instance segmentation model over the bottom-up GrabCut [35] foreground segmentation results from the bounding boxes. Unlike [18], we aim to exploit all existing labeled data rather than artificially limiting it. Our work is also complementary in the sense that bottom-up segmentation methods may be used to infer training masks for our weakly-labeled examples. We leave this extension to future work.

Visual embeddings. Object categories may be modeled by continuous 'embedding' vectors in a visual-semantic space, where nearby vectors are often close in appearance or semantic ontology. Class embedding vectors may be obtained via natural language processing techniques (*e.g.* word2vec [26] and GloVe [31]), from visual appearance information (*e.g.* [7]), or both (*e.g.* [37]). In our work, the parameters of Mask R-CNN's box head contain class-specific appearance information and can be seen as embedding vectors

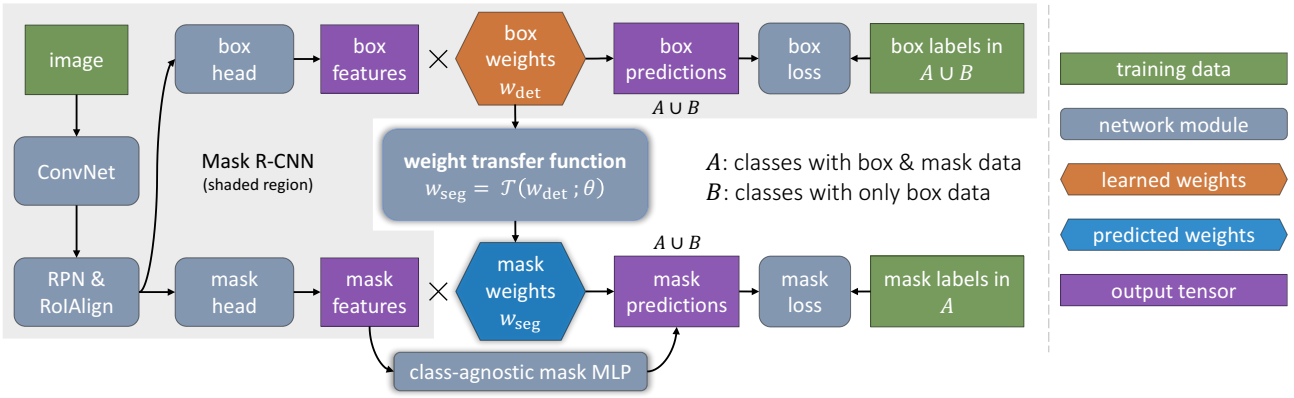

Figure 2. **Detailed illustration of our Mask^X R-CNN method.** Instead of directly learning the mask prediction parameters w_{seg}, Mask^X R-CNN predicts a category's segmentation parameters w_{seg} from its corresponding box detection parameters w_{det}, using a learned weight transfer function $\mathcal{T}$. For training, $\mathcal{T}$ only needs mask data for the classes in set A, yet it can be applied to all classes in set $A \cup B$ at test time. We also augment the mask head with a complementary fully connected multi-layer perceptron (MLP).

learned by training for the bounding box object detection task. The class embedding vectors enable transfer learning in our model by sharing appearance information between visually related classes. We also compare with the NLP-based GloVe embeddings [31] in our experiments.

3. Learning to Segment Every Thing

Let C be the set of object categories (*i.e.*, 'things' [1]) for which we would like to train an instance segmentation model. Most existing approaches assume that *all* training examples in C are annotated with instance masks. We relax this requirement and instead assume that $C = A \cup B$ where examples from the categories in A have masks, while those in B have only bounding boxes. Since the examples of the B categories are weakly labeled w.r.t. the target task (instance segmentation), we refer to training on the combination of strong and weak labels as a *partially supervised* learning problem. Noting that one can easily convert instance masks to bounding boxes, we assume that bounding box annotations are also available for classes in A.

Given an instance segmentation model like Mask R-CNN that has a bounding box detection component and a mask prediction component, we propose the **Mask^X R-CNN** method that transfers category-specific information from the model's bounding box detectors to its instance mask predictors.

3.1. Mask Prediction Using Weight Transfer

Our method is built on Mask R-CNN [15], because it is a simple instance segmentation model that also achieves state-of-the-art results. In brief, Mask R-CNN can be seen as augmenting a Faster R-CNN [34] bounding box detection model with an additional mask branch that is a small fully convolutional network (FCN) [24]. At inference time, the mask branch is applied to each detected object in order to predict an instance-level foreground segmentation mask. During training, the mask branch is trained jointly and in parallel with the standard bounding box head found in Faster R-CNN.

In Mask R-CNN, the last layer in the bounding box branch and the last layer in the mask branch both contain *category-specific* parameters that are used to perform bounding box classification and instance mask prediction, respectively, for each category. Instead of learning the category-specific bounding box parameters and mask parameters independently, we propose to predict a category's mask parameters from its bounding box parameters using a generic, category-agnostic weight transfer function that can be jointly trained as part of the whole model.

For a given category c, let w^c_{det} be the class-specific object detection weights in the last layer of the bounding box head, and w^c_{seg} be the class-specific mask weights in the mask branch. Instead of treating w^c_{seg} as model parameters, w^c_{seg} is parameterized using a generic weight prediction function $\mathcal{T}(\cdot)$:

$$w^c_{\text{seg}} = \mathcal{T}(w^c_{\text{det}}; \theta), \tag{1}$$

where θ are class-agnostic, learned parameters.

The *same* transfer function $\mathcal{T}(\cdot)$ may be applied to any category c and, thus, θ should be set such that $\mathcal{T}$ generalizes to classes whose masks are not observed during training. We expect that generalization is possible because the class-specific detection weights w^c_{det} can be seen as an appearance-based visual embedding of the class.

$\mathcal{T}(\cdot)$ can be implemented as a small fully connected neural network. Figure 2 illustrates how the weight transfer function fits into Mask R-CNN to form Mask^X R-CNN. As a detail, note that the bounding box head contains two types of detection weights: the RoI classification weights w^c_{cls} and the bounding box regression weights w^c_{box}. We experiment with using either only a single type of detection weights (*i.e.* $w^c_{\text{det}} = w^c_{\text{cls}}$ or $w^c_{\text{det}} = w^c_{\text{box}}$) or using the concatenation of the two types of weights (*i.e.* $w^c_{\text{det}} = [w^c_{\text{cls}}, w^c_{\text{box}}]$).

3.2. Training

During training, we assume that for the two sets of classes A and B, instance mask annotations are available

only for classes in A but not for classes in B, while all classes in A and B have bounding box annotations available. As shown in Figure 2, we train the bounding box head using the standard box detection losses on all classes in $A \cup B$, but only train the mask head and the weight transfer function $\mathcal{T}(\cdot)$ using a mask loss on the classes in A. Given these losses, we explore two different training procedures: stage-wise training and end-to-end training.

Stage-wise training. As Mask R-CNN can be seen as augmenting Faster R-CNN with a mask head, a possible training strategy is to separate the training procedure into detection training (first stage) and segmentation training (second stage). In the first stage, we train a Faster R-CNN using only the bounding box annotations of the classes in $A \cup B$, and then in the second stage the additional mask head is trained while keeping the convolutional features and the bounding box head fixed. In this way, the class-specific detection weights w^c_{det} of each class c can be treated as fixed class embedding vectors that do not need to be updated when training the second stage. This approach has the practical benefit of allowing us to train the box detection model once and then rapidly evaluate design choices for the weight transfer function. It also has disadvantages, which we discuss next.

End-to-end joint training. It was shown that for Mask R-CNN, multi-task training can lead to better performance than training on each task separately. The aforementioned stage-wise training mechanism separates detection training and segmentation training, and may result in inferior performance. Therefore, we would also like to jointly train the bounding box head and the mask head in an end-to-end manner. In principle, one can directly train with back-propagation using the box losses on classes in $A \cup B$ and the mask loss on classes in A. However, this may cause a discrepancy in the class-specific detection weights w^c_{det} between set A and B, since only w^c_{det} for $c \in A$ will receive gradients from the mask loss through the weight transfer function $\mathcal{T}(\cdot)$. We would like w^c_{det} to be homogeneous between A and B so that the predicted $w^c_{\text{seg}} = \mathcal{T}(w^c_{\text{det}}; \theta)$ trained on A can better generalize to B.

To address this discrepancy, we take a simple approach: when back-propagating the mask loss, we stop the gradient with respect to w^c_{det}, that is, we only compute the gradient of the predicted mask weights $\mathcal{T}(w^c_{\text{det}}; \theta)$ with respect to transfer function parameter θ but not bounding box weight w^c_{det}. This can be implemented as $w^c_{\text{seg}} = \mathcal{T}(\texttt{stop\_grad}(w^c_{\text{det}}); \theta)$ in most neural network toolkits.

3.3. Baseline: Class-Agnostic Mask Prediction

DeepMask [32] established that it is possible to train a deep model to perform *class-agnostic* mask prediction where an object mask is predicted regardless of the category. A similar result was also shown for Mask R-CNN with only a small loss in mask quality [15]. In additional experiments, [32] demonstrated if the class-agnostic model is trained to predict masks on a *subset* of the COCO categories (specifically the 20 from PASCAL VOC [9]) it can *generalize* to the other 60 COCO categories at inference time. Based on these results, we use Mask R-CNN with a class-agnostic FCN mask prediction head as a baseline. Evidence from [32] and [15] suggest that this is a strong baseline. Next, we introduce an extension that can improve both the baseline and our proposed weight transfer function.

We also compare with a few other baselines for unsupervised or weakly supervised instance segmentation in §4.3.

3.4. Extension: Fused FCN+MLP Mask Heads

Two types of mask heads are considered for Mask R-CNN in [15]: (1) an FCN head, where the $M \times M$ mask is predicted with a fully convolutional network, and (2) an MLP head, where the mask is predicted with a multi-layer perceptron consisting of fully connected layers, more similar to DeepMask. In Mask R-CNN, the FCN head yields higher mask AP. However, the two designs may be complementary. Intuitively, the MLP mask predictor may better capture the 'gist' of an object while the FCN mask predictor may better capture the details (such as the object boundary). Based on this observation, we propose to improve both the baseline class-agnostic FCN and our weight transfer function (which uses an FCN) by fusing them with predictions from a class-agnostic MLP mask predictor. Our experiments will show that this extension brings improvements to both the baseline and our transfer approach.

When fusing class-agnostic and class-specific mask predictions for K classes, the two scores are added into a final $K \times M \times M$ output, where the class-agnostic mask scores (with shape $1 \times M \times M$) are tiled K times and added to every class. Then, the $K \times M \times M$ mask scores are turned into per-class mask probabilities through a sigmoid unit, and resized to the actual bounding box size as final instance mask for that bounding box. During training, binary cross-entropy loss is applied on the $K \times M \times M$ mask probabilities.

4. Experiments on COCO

We evaluate our method on the COCO dataset [23], which is small scale w.r.t. the number of categories but contains exhaustive mask annotations for 80 categories. This property enables rigorous quantitative evaluation using standard detection metrics, like average precision (AP).

4.1. Evaluation Protocol and Baselines

We simulate the partially supervised training scenario on COCO by partitioning the 80 classes into sets A and B, as described in §3. We consider two split types: (1) The 20/60 split used by DeepMask [32] that divides the COCO categories based on the 20 contained in PASCAL VOC [9] and the 60 that are not. We refer to these as the 'voc' and 'non-voc' category sets from here on. (2) We also conduct exper-

method	voc → non-voc AP on B	AP on A	non-voc → voc AP on B	AP on A
transfer w/ randn	15.4	35.2	19.9	31.1
transfer w/ GloVe [31]	17.3	35.2	21.9	31.1
transfer w/ cls	18.1	35.1	25.2	31.1
transfer w/ box	19.8	35.2	25.7	31.1
transfer w/ cls+box	**20.2**	35.2	**26.0**	31.2
class-agnostic (baseline)	14.2	34.4	21.5	30.7
fully supervised (oracle)	30.7	35.0	35.0	30.7

(a) **Ablation on input to $\mathcal{T}$.** 'cls' is RoI classification weights, 'box' is box regression weights, and'cls+box' is both weights. We also compare with the NLP-based GloVe vectors [31]. Our transfer function $\mathcal{T}$ improves the AP on B while remaining on par with the oracle on A.

method	voc → non-voc AP on B	AP on A	non-voc → voc AP on B	AP on A
transfer w/ 1-layer, none	19.2	35.2	25.3	31.2
transfer w/ 2-layer, ReLU	19.7	35.3	25.1	31.1
transfer w/ 2-layer, LeakyReLU	**20.2**	35.2	**26.0**	31.2
transfer w/ 3-layer, ReLU	19.3	35.2	**26.0**	31.1
transfer w/ 3-layer, LeakyReLU	18.9	35.2	25.5	31.1

(b) **Ablation on the structure of $\mathcal{T}$.** We vary the number of fully connected layers in the weight transfer function $\mathcal{T}$, and try both ReLU and LeakyReLU as activation function in the hidden layers. The results show that '2-layer, LeakyReLU' works best, but in general $\mathcal{T}$ is robust to specific implementation choices.

method	voc → non-voc AP on B	AP on A	non-voc → voc AP on B	AP on A
class-agnostic	14.2	34.4	21.5	30.7
class-agnostic+MLP	17.1	35.1	22.8	31.3
transfer	20.2	35.2	26.0	31.2
transfer+MLP	**21.3**	35.4	**26.6**	31.4

(c) **Impact of the MLP mask branch.** Adding the class-agnostic MLP mask branch (see §3.4) improves the performance of classes in set B for both the class-agnostic baseline and our weight transfer approach.

method	training	stop grad on w_{det}	voc → non-voc AP on B	AP on A	non-voc → voc AP on B	AP on A
class-agnostic	sw	n/a	14.2	34.4	21.5	30.7
transfer	sw	n/a	20.2	35.2	26.0	31.2
class-agnostic	e2e	n/a	19.2	36.8	23.9	32.5
transfer	e2e		20.2	37.7	24.8	33.2
transfer	e2e	✓	**22.2**	37.6	**27.6**	33.1

(d) **Ablation on the training strategy.** We try both stage-wise ('sw') and end-to-end ('e2e') training (see §3.2), and whether to stop gradient from $\mathcal{T}$ to w_{det}. End-to-end training improves the results and it is crucial to stop gradient on w_{det}.

Table 1. **Ablation study of our method.** We use ResNet-50-FPN as our backbone network, and 'cls+box' and '2-layer, LeakyReLU' as the default input and structure of $\mathcal{T}$. Results in (a,b,c) are based on stage-wise training, and we study the impact of end-to-end training in (d). Mask AP is evaluated on the COCO dataset `val2017` split between the 20 PASCAL VOC categories ('voc') and the 60 remaining categories ('non-voc'), as in [32]. Performance on the strongly supervised set A is shown in gray.

iments using multiple trials with random splits of different sizes. These experiments allow us to characterize any bias in the voc/non-voc split and also understand what factors in the training data lead to better mask generalization.

Implementation details. We train our model on the COCO `train2017` split and test on `val2017`.[1] Each class has a 1024-d RoI classification parameter vector w^c_{cls} and a 4096-d bounding box regression parameter vector w^c_{box} in the detection head, and a 256-d segmentation parameter vector w^c_{seg} in the mask head. The output mask resolution is $M \times M = 28 \times 28$. In all our experimental analysis below, we use either ResNet-50-FPN or ResNet-101-FPN [22] as the backbone architecture for Mask R-CNN, initialized from a ResNet-50 or a ResNet-101 [16] model pretrained on the ImageNet-1k image classification dataset [36].

We follow the training hyper-parameters suggested for Mask R-CNN in [15]. Each minibatch has 16 images $\times$ 512 RoIs-per-images, and the network is trained for 90k iterations on 8 GPUs. We use 1e-4 weight decay and 0.9 momentum, and an initial learning rate of 0.02, which is multiplied by 0.1 after 60k and 80k iterations. We evaluate instance segmentation performance using average precision (AP), which is the standard COCO metric and equal to the mean of average precision from 0.5 to 0.95 IoU threshold of all classes.

Baseline and oracle. We compare our method to class-agnostic mask prediction using either an FCN or fused FCN+MLP structure. In these approaches, instead of predicting each class c's segmentation parameters w^c_{seg} from its bounding box classification parameters w^c_{det}, all the categories share the *same* learned segmentation parameters w^c_{seg} (no weight transfer function is involved). Evidence from DeepMask and Mask R-CNN, as discussed in §3.3, suggests that this approach is a strong baseline. In addition, we compare our approach with unsupervised or weakly supervised instance segmentation approaches in §4.3.

We also evaluate an 'oracle' model: Mask R-CNN trained on all classes in $A \cup B$ with access to instance mask annotations for *all* classes in A and B at training time. This fully supervised model is a performance upper bound for our partially supervised task (unless the weight transfer function can improve over directly learning w^c_{seg}).

4.2. Ablation Experiments

Input to $\mathcal{T}$. In Table 1a we study the impact of the input to the weight transfer function $\mathcal{T}$. For transfer learning to work, we expect that the input should capture information about how the visual appearance of classes relate to each other. To see if this is the case, we designed several inputs to $\mathcal{T}$: a random Gaussian vector ('randn') assigned to each class, an NLP-based word embedding using pretrained GloVe vectors [31] for each class, the weights from the Mask R-CNN box head classifier ('cls'), the weights from the box regression ('box'), and the concatenation of both weights ('cls+box'). We compare the performance of our transfer approach with these different embeddings to the strong baseline: class-agnostic Mask R-CNN.

First, Table 1a shows that the random control ('randn')

[1]The COCO `train2017` and `val2017` splits are the same as the `trainval35k` and `minival` splits used in prior work, such as [15].

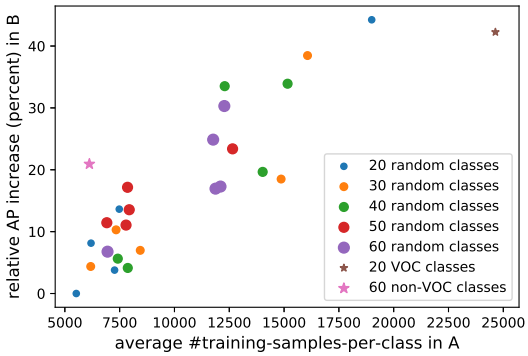

Figure 3. Each point corresponds to our method on a random A/B split of COCO classes. We vary $|A|$ from 20 to 60 classes and plot the relative change in mask AP on the classes in set B (those classes without mask annotations) *vs*. the average number of mask annotations per class in set A.

yields results on par with the baseline; they are slightly better on voc→non-voc and worse in the other direction, which may be attributed to noise. Next, the GloVe embedding shows a consistent improvement over the baseline, which indicates that these embeddings may capture some visual information as suggested in prior work [39]. However, inputs 'cls', 'box' and 'cls+box' all strongly outperform the NLP-based embedding (with 'cls+box' giving the best results), which matches our expectation since they encode visual information by design.

We note that all methods compare well to the fully supervised Mask R-CNN oracle on the classes in set A. In particular, our transfer approach slightly outperforms the oracle for all input types. This results indicates that our approach does not sacrifice anything on classes with strong supervision, which is an important property.

Structure of $\mathcal{T}$. In Table 1b we compare different implementations of $\mathcal{T}$: as a simple affine transformation, or as a neural network with 2 or 3 layers. Since LeakyReLU [25] is used for weight prediction in [27], we try both ReLU and LeakyReLU as activation function in the hidden layers. The results show that a 2-layer MLP with LeakyReLU gives the best mask AP on set B. Given this, we select the 'cls+box, 2-layer, LeakyReLU' implementation of $\mathcal{T}$ for all subsequent experiments.

Comparison of random A/B splits. Besides splitting datasets into voc and non-voc, we also experiment with random splits of the 80 classes in COCO, and vary the number of training classes. We randomly select 20, 30, 40, 50 or 60 classes to include in set A (the complement forms set B), perform 5 trials for each split size, and compare the performance of our weight transfer function $\mathcal{T}$ on classes in B to the class-agnostic baseline. The results are shown in Figure 3, where it can be seen that our method yields to up to over 40% relative increase in mask AP. This plot reveals a correlation between relative AP increase and the average number of training samples per class in set A. This indicates that to maximize transfer performance to classes in set B it may be more effective to collect a larger number of instance mask samples for each object category in set A.

Impact of the MLP mask branch. As discussed in §3.4, a class-agnostic MLP mask branch can be fused with either the baseline or our transfer approach. In Table 1c we see that either mask head fused with the MLP mask branch consistently outperforms the corresponding unfused version. This confirms our intuition that FCN-based mask heads and MLP-based mask heads are complementary in nature.

Effect of end-to-end training. Up to now, all ablation experiments use stage-wise training, because it is significantly faster (the same Faster R-CNN detection model can be reused for all experiments). However, as noted in §3.2, stage-wise training may be suboptimal. Thus, Table 1d compares stage-wise training to end-to-end training. In the case of end-to-end training, we investigate if it is necessary to stop gradients from $\mathcal{T}$ to w_{det}, as discussed. Indeed, results match our expectation that end-to-end training can bring improved results, *however only when back-propagation from $\mathcal{T}$ to w_{det} is disabled*. We believe this modification is necessary in order to make the embedding of classes in A homogeneous with those in B; a property that is destroyed when only the embeddings for classes in A are modified by back-propagation from $\mathcal{T}$.

4.3. Results and Comparison of Our Full Method

Table 2 compares our full Mask^X R-CNN method (*i.e.*, Mask R-CNN with 'transfer+MLP' and $\mathcal{T}$ implemented as 'cls+box, 2-layer, LeakyReLU') and the class-agnostic baseline using end-to-end training. In addition, we also compare with the following baseline approaches: a) unsupervised mask prediction using GrabCut [35] foreground segmentation over the Faster R-CNN detected object boxes (**Faster R-CNN tested w/ GrabCut**) and b) weakly supervised *instance segmentation* similar to [18], which trains an instance segmentation method (here we use Mask R-CNN) on the GrabCut segmentation of the ground-truth boxes (**Mask R-CNN trained w/ GrabCut**).

Mask^X R-CNN outperforms these approaches by a large margin (over 20% relative increase in mask AP). We also experiment with ResNet-101-FPN as the backbone network in the bottom half of Table 2. The trends observed with ResNet-50-FPN generalize to ResNet-101-FPN, demonstrating independence of the particular backbone used thus far. Figure 4 shows example mask predictions from the class-agnostic baseline and our approach.

5. Large-Scale Instance Segmentation

Thus far, we have experimented with a simulated version of our true objective: training large-scale instance segmentation models with broad visual comprehension. We

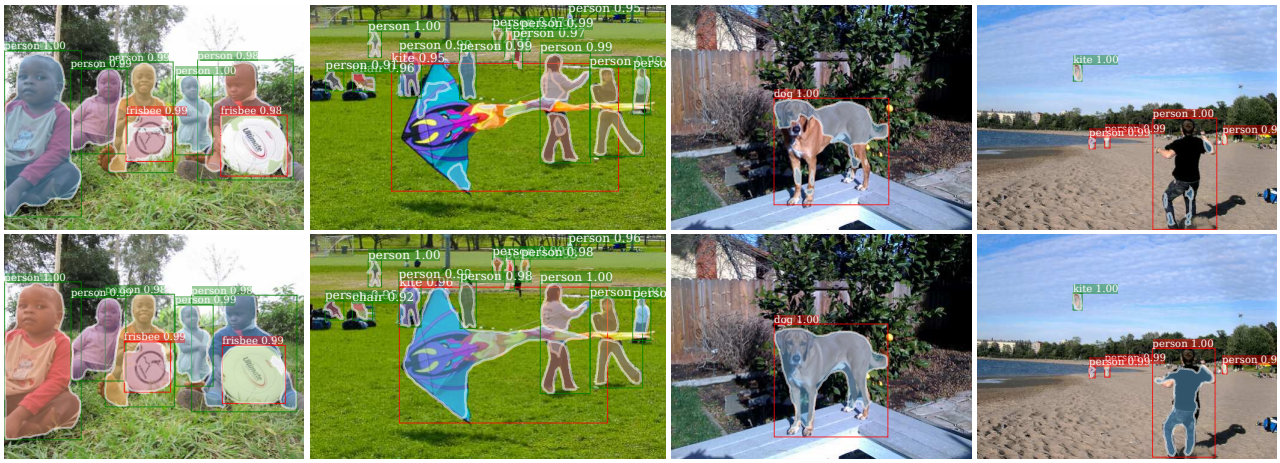

Figure 4. **Mask predictions from the class-agnostic baseline (top row) *vs*. our Mask^X R-CNN approach (bottom row)**. Green boxes are classes in set A while the red boxes are classes in set B. The left 2 columns are $A = \{\text{voc}\}$ and the right 2 columns are $A = \{\text{non-voc}\}$.

		voc → non-voc: test on $B = \{\text{non-voc}\}$						non-voc → voc: test on $B = \{\text{voc}\}$					
backbone	method	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
R-50-FPN	class-agnostic	19.2	36.4	18.4	11.5	23.3	24.4	23.9	42.9	23.5	11.6	24.3	33.7
	Faster R-CNN tested w/ GrabCut	12.6	24.0	11.9	4.3	12.0	23.5	12.1	27.7	8.9	4.3	12.0	23.5
	Mask R-CNN trained w/ GrabCut	19.5	39.2	17.0	6.5	20.9	**34.3**	19.5	46.2	14.2	4.7	15.9	32.0
	Mask^X R-CNN (ours)	**23.7**	**43.1**	**23.5**	**12.4**	**27.6**	32.9	**28.9**	**52.2**	**28.6**	**12.1**	**29.0**	**40.6**
	fully supervised (oracle)	33.0	53.7	35.0	15.1	37.0	49.9	37.5	63.1	38.9	15.1	36.0	53.1
R-101-FPN	class-agnostic	18.5	34.8	18.1	11.3	23.4	21.7	24.7	43.5	24.9	11.4	25.7	35.1
	Faster R-CNN tested w/ GrabCut	13.0	24.6	12.1	4.5	12.3	24.4	12.3	27.6	9.5	4.5	12.3	24.4
	Mask R-CNN trained w/ GrabCut	19.7	39.7	17.0	6.4	21.2	**35.8**	19.6	46.1	14.3	5.1	16.0	32.4
	Mask^X R-CNN (ours)	**23.8**	**42.9**	**23.5**	**12.7**	**28.1**	33.5	**29.5**	**52.4**	**29.7**	**13.4**	**30.2**	**41.0**
	fully supervised (oracle)	34.4	55.2	36.3	15.5	39.0	52.6	39.1	64.5	41.4	16.3	38.1	55.1

Table 2. **End-to-end training of Mask^X R-CNN.** As in Table 1, we use 'cls+box, 2-layer, LeakyReLU' implementation of $\mathcal{T}$ and add the MLP mask branch ('transfer+MLP'), and follow the same evaluation protocol. We also report AP_{50} and AP_{75} (average precision evaluated at 0.5 and 0.75 IoU threshold respectively), and AP over small (AP_S), medium (AP_M), and large (AP_L) objects. Our method significantly outperforms the baseline approaches in §4.3 on set B without mask training data for both ResNet-50-FPN and ResNet-101-FPN backbones.

believe this goal represents an exciting new direction for visual recognition research and that to accomplish it some form of learning from partial supervision may be required.

To take a step towards this goal, we train a large-scale Mask^X R-CNN model following the partially supervised task, using bounding boxes from the Visual Genome (VG) dataset [20] and instance masks from the COCO dataset [23]. The VG dataset contains 108077 images, and over 7000 category synsets annotated with object bounding boxes (but not masks). To train our model, we select the 3000 most frequent synsets as our set of classes $A \cup B$ for instance segmentation, which covers all the 80 classes in COCO. Since the VG dataset images have a large overlap with COCO, when training on VG we take all the images that are not in COCO `val2017` split as our training set, and validate our model on the rest of VG images. We treat all the 80 VG classes that overlap with COCO as our set A with mask data, and the remaining 2920 classes in VG as our set B with only bounding boxes.

Training. We train our large-scale Mask^X R-CNN model using the stage-wise training strategy. Specifically, we train a Faster R-CNN model to detect the 3000 classes in VG using ResNet-101-FPN as our backbone network following the hyper-parameters in §4.1. Then, in the second stage, we add the mask head using our weight transfer function $\mathcal{T}$ and the class-agnostic MLP mask prediction (*i.e.*, 'transfer+MLP'), with the 'cls+box, 2-layer, LeakyReLU' implementation of $\mathcal{T}$. The mask head is trained on subset of 80 COCO classes (set A) using the mask annotations in the `train2017` split of the COCO dataset.

Qualitative results. Mask AP is difficult to compute on VG because it contains only box annotations. Therefore we visualize results to understand the performance of our model trained on all the 3000 classes in $A \cup B$ using our weight transfer function. Figure 5 shows mask prediction examples on validation images, where it can be seen that our model predicts reasonable masks on those VG classes not overlapping with COCO (set B, shown in red boxes).

This visualization shows several interesting properties of our large-scale instance segmentation model. First, it has learned to detect abstract concepts, such as shadows and paths. These are often difficult to segment. Second, by simply taking the first 3000 synsets from VG, some of the concepts are more 'stuff' like than 'thing' like. For example,

Figure 5. **Example mask predictions from our Mask^X R-CNN on 3000 classes in Visual Genome.** The green boxes are the 80 classes that overlap with COCO (set A with mask training data) while the red boxes are the remaining 2920 classes not in COCO (set B without mask training data). It can be seen that our model generates reasonable mask predictions on many classes in set B. See §5 for details.

the model does a reasonable job segmenting isolated trees, but tends to fail at segmentation when the detected 'tree' is more like a forest. Finally, the detector does a reasonable job at segmenting whole objects and parts of those objects, such as windows of a trolley car or handles of a refrigerator. Compared to a detector trained on 80 COCO categories, these results illustrate the exciting potential of systems that can recognize and segment thousands of concepts.

6. Conclusion

This paper addresses the problem of large-scale instance segmentation by formulating a partially supervised learning paradigm in which only a subset of classes have instance masks during training while the rest have box annotations. We propose a novel transfer learning approach, where a learned *weight transfer function* predicts how each class should be segmented based on parameters learned for detecting bounding boxes. Experimental results on the COCO dataset demonstrate that our method greatly improves the generalization of mask prediction to categories without mask training data. Using our approach, we build a large-scale instance segmentation model over 3000 classes in the Visual Genome dataset. The qualitative results are encouraging and illustrate an exciting new research direction into large-scale instance segmentation. They also reveal that scaling instance segmentation to thousands of categories, without full supervision, is an extremely challenging problem with ample opportunity for improved methods.

References

[1] E. H. Adelson. On seeing stuff: the perception of materials by humans and machines. In *Human Vision and Electronic Imaging*, 2001. 3

[2] M. Bai and R. Urtasun. Deep watershed transform for instance segmentation. In *CVPR*, 2017. 2

[3] A. Bearman, O. Russakovsky, V. Ferrari, and L. Fei-Fei. Whats the point: Semantic segmentation with point supervision. In *ECCV*, 2016. 2

[4] J. Dai, K. He, and J. Sun. Boxsup: Exploiting bounding boxes to supervise convolutional networks for semantic segmentation. In *ICCV*, 2015. 1, 2

[5] J. Dai, K. He, and J. Sun. Convolutional feature masking for joint object and stuff segmentation. In *CVPR*, 2015. 2

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

[7] V. Dumoulin, J. Shlens, M. Kudlur, A. Behboodi, F. Lemic, A. Wolisz, M. Molinaro, C. Hirche, M. Hayashi, E. Bagan, et al. A learned representation for artistic style. *arXiv preprint arXiv:1610.07629*, 2016. 2

[8] M. Elhoseiny, B. Saleh, and A. Elgammal. Write a classifier: Zero-shot learning using purely textual descriptions. In *ICCV*, 2013. 2

[9] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes (VOC) Challenge. *IJCV*, 2010. 4

[10] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, 2014. 1

[11] D. Ha, A. Dai, and Q. V. Le. HyperNetworks. *arXiv preprint arXiv:1609.09106*, 2016. 2

[12] B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik. Simultaneous detection and segmentation. In *ECCV*. 2014. 2

[13] B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik. Hypercolumns for object segmentation and fine-grained localization. In *CVPR*, 2015. 2

[14] Z. Hayder, X. He, and M. Salzmann. Boundary-aware instance segmentation. In *CVPR*, 2017. 2

[15] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask R-CNN. In *ICCV*, 2017. 1, 2, 3, 4, 5

[16] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 5

[17] J. Hoffman, S. Guadarrama, E. S. Tzeng, R. Hu, J. Donahue, R. Girshick, T. Darrell, and K. Saenko. LSDA: Large scale detection through adaptation. In *NIPS*, 2014. 2

[18] A. Khoreva, R. Benenson, J. Hosang, M. Hein, and B. Schiele. Simple does it: Weakly supervised instance and semantic segmentation. In *CVPR*, 2017. 1, 2, 6

[19] A. Kirillov, E. Levinkov, B. Andres, B. Savchynskyy, and C. Rother. InstanceCut: from edges to instances with multicut. In *CVPR*, 2017. 2

[20] R. Krishna, Y. Zhu, O. Groth, J. Johnson, K. Hata, J. Kravitz, S. Chen, Y. Kalantidis, L.-J. Li, D. A. Shamma, M. Bernstein, and L. Fei-Fei. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *IJCV*, 2017. 1, 2, 7

[21] Y. Li, H. Qi, J. Dai, X. Ji, and Y. Wei. Fully convolutional instance-aware semantic segmentation. In *CVPR*, 2017. 2

[22] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie. Feature pyramid networks for object detection. In *CVPR*, 2017. 5

[23] 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, 4, 7

[24] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, 2015. 3

[25] A. L. Maas, A. Y. Hannun, and A. Y. Ng. Rectifier nonlinearities improve neural network acoustic models. In *ICML*, volume 30, 2013. 6

[26] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013. 2

[27] I. Misra, A. Gupta, and M. Hebert. From red wine to red tomato: Composition with context. In *CVPR*, 2017. 2, 6

[28] S. J. Pan and Q. Yang. A survey on transfer learning. *IEEE Trans. on Knowl. and Data Eng.*, 2010. 2

[29] G. Papandreou, L.-C. Chen, K. Murphy, and A. L. Yuille. Weakly-and semi-supervised learning of a DCNN for semantic image segmentation. In *ICCV*, 2015. 2

[30] D. Pathak, P. Krahenbuhl, and T. Darrell. Constrained convolutional neural networks for weakly supervised segmentation. In *ICCV*, 2015. 2

[31] J. Pennington, R. Socher, and C. Manning. GloVe: Global vectors for word representation. In *EMNLP*, 2014. 2, 3, 5

[32] P. O. Pinheiro, R. Collobert, and P. Dollar. Learning to segment object candidates. In *NIPS*, 2015. 2, 4, 5

[33] P. O. Pinheiro, T.-Y. Lin, R. Collobert, and P. Dollár. Learning to refine object segments. In *ECCV*, 2016. 2

[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. 1, 3

[35] C. Rother, V. Kolmogorov, and A. Blake. GrabCut: Interactive foreground extraction using iterated graph cuts. In *ACM ToG*, 2004. 2, 6

[36] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *IJCV*, 2015. 5

[37] Y.-H. H. Tsai, L.-K. Huang, and R. Salakhutdinov. Learning robust visual-semantic embeddings. *arXiv preprint arXiv:1703.05908*, 2017. 2

[38] Y.-X. Wang and M. Hebert. Learning to learn: Model regression networks for easy small sample learning. In *ECCV*, 2016. 2

[39] Y. Xian, Z. Akata, G. Sharma, Q. Nguyen, M. Hein, and B. Schiele. Latent embeddings for zero-shot classification. In *CVPR*, 2016. 6