The surprising impact of mask-head architecture on novel class segmentation

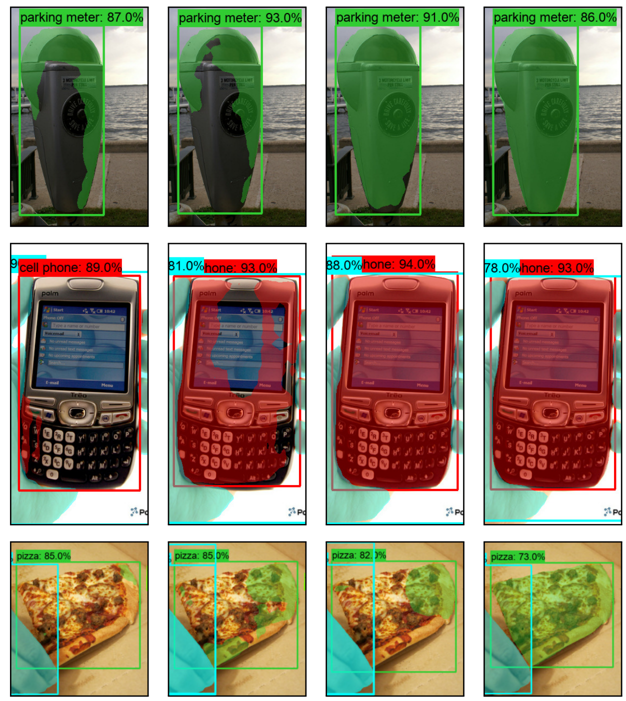
Vighnesh Birodkar, Zhichao Lu, Siyang Li, Vivek Rathod, Jonathan Huang
Google
{vighneshb,lzc,siyang,rathodv,jonathanhuang}@google.com

Abstract

Instance segmentation models today are very accurate when trained on large annotated datasets, but collecting mask annotations at scale is prohibitively expensive. We address the partially supervised instance segmentation problem in which one can train on (significantly cheaper) bounding boxes for all categories but use masks only for a subset of categories. In this work, we focus on a popular family of models which apply differentiable cropping to a feature map and predict a mask based on the resulting crop. Under this family, we study Mask R-CNN and discover that instead of its default strategy of training the mask-head with a combination of proposals and groundtruth boxes, training the mask-head with only groundtruth boxes dramatically improves its performance on novel classes. This training strategy also allows us to take advantage of alternative mask-head architectures, which we exploit by replacing the typical mask-head of 2-4 layers with significantly deeper off-the-shelf architectures (e.g. ResNet, Hourglass models). While many of these architectures perform similarly when trained in fully supervised mode, our main finding is that they can generalize to novel classes in dramatically different ways. We call this ability of mask-heads to generalize to unseen classes the strong mask generalization *effect and show that without any specialty modules or losses, we can achieve state-of-the-art results in the partially supervised COCO instance segmentation benchmark. Finally, we demonstrate that our effect is general, holding across underlying detection methodologies (including anchor-based, anchor-free or no detector at all) and across different backbone networks. Code and pre-trained models are available at* `https://git.io/deepmac`.

1. Introduction

Large labeled datasets like COCO [32] are crucial for deep neural network based instance segmentation methods [16, 3, 38]. However, collecting groundtruth masks can take $> 10\times$ more time than bounding box annotations. In COCO [32], mask annotations required $\approx$ 80 seconds on average whereas methods such as Extreme Clicking [37] yield bounding boxes in 7 seconds.



(a) ResNet-4 (b) ResNet-12 (c) ResNet-20 (d) Hourglass-20

Figure 1: The effect of mask-head architecture on mask predictions for unseen classes. *Despite never having seen masks from the 'parking meter', 'pizza' or 'mobile phone' class*, the rightmost mask-head architecture can segment these classes correctly. From left to right, we show better mask-head architectures predicting better masks. Moreover, this difference is *only* apparent when evaluating on unseen classes — if we evaluate on seen classes, all four architectures exhibit similar performance.

Given that boxes are much cheaper to annotate than masks, we address the "partially supervised" instance segmentation problem [20], where all classes have bounding box annotations but only a subset of classes have mask annotations. We will refer to classes with mask annotations as "seen" categories and classes without as "unseen". Doing well on this task requires the model to generalize in a strong sense, producing correct masks on unseen classes.

We consider a general family of *crop-then-segment* instance segmentation models where one extracts a feature map over an image, then given a tight bounding box around an instance, performs a differentiable crop (e.g.

ROIAlign [16]). The cropped feature map is then fed to a mask-head subnetwork to yield a final mask prediction. This mask prediction is performed in a class-agnostic manner so that a model trained from a subset of classes can be applied unchanged to novel classes.

One "naïve" baseline in this family is to adapt Mask R-CNN [16] to produce class-agnostic masks. But this approach is known to perform abysmally on unseen classes (e.g. on the standard partially supervised COCO benchmark, it achieves $< 20\%$ mask mAP on unseen classes vs $> 40\%$ on seen, [20]). Thus previous approaches have used, e.g., offline-trained shape priors [27] or specialty losses [10] yielding significantly improved results.

As a starting point, we revisit "naïve" Mask R-CNN to better understand the reasons for its poor performance. Our first finding is that the typical strategy of training the Mask R-CNN mask-head with a combination of groundtruth and proposed (typically noisy) boxes is a major culprit that inhibits its performance on novel classes. While training with noisy proposals gives slightly better results when fully supervised, we show that simply training the mask-head with *only* groundtruth boxes has a surprising impact on its performance on unseen classes (+9 mAP) (note that we follow the usual procedure of using predicted boxes at test time).

We next zoom out beyond Mask R-CNN to the more general family of crop-then-segment models. Our second major finding is that in the context of using the above slightly modified training regime, the architecture of the mask-head takes on a disproportionately impactful role in generalization to unseen classes. More specifically, we find that mask-heads that might perform similarly under full supervision can behave differently under partial supervision, generalizing to unseen classes in strikingly different ways.

While it is natural to experiment with different mask-head architectures, we note that their role in generalization has not been studied extensively in prior literature likely for the following reasons: (1) the choice of mask-head architecture has limited impact in the fully supervised setting, (2) heavier mask-heads adversely impact running time. and (3) as noted above, in architectures like Mask R-CNN, the benefits of using better mask-heads are not necessarily realized in the default training regimen. Thus most prior works in instance segmentation have settled on using shallow (2-4 layer) fully connected or convolution based mask-heads.

In our COCO experiments, we find that the difference between worst and best architectures is only 1% (absolute mAP) on seen classes but can be 7% on unseen classes (examples in Figure 1). This difference is visually palpable and subsequently changes the calculus for deciding whether it's worth using a heavier mask-head.

We refer to this effect of certain mask-head architectures on unseen classes as the *"strong mask generalization effect"* and illustrate it with 3 representative model classes: an anchor-free and anchor-based model, and one that discards detection altogether. We show that our effect is general, holding across underlying detection methodologies (or no detector at all) and across different backbone networks. We also identify architectural characteristics (such as depth and encoder-decoder arrangements) that empirically yield strong mask generalization properties.

One main finding is that deeper mask-heads generalize better despite being counter-intuitively more over-parameterized than shallower ones. Our anchor-based model, based on Mask R-CNN [16], employs mask-heads that are 20+ layers deep and we thus refer to this model as Deep-MARC (for Deep Mask-heads Above R-CNN). Similarly, our anchor-free model, which we use for most ablations, is based on CenterNet [55] and is called Deep-MAC (for Deep Mask-heads Above CenterNet). Using out-of-the-box mask-head architectures, we show that both Deep-MAC and Deep-MARC surpass the state-of-the-art [10] in the COCO partially supervised instance segmentation setting with 35.5 % and 38.7 % mAP respectively.

Due to space limitations, we have relegated a number of auxiliary findings to the Appendix. Among them, we show that: (1) two-stage training (i.e. self-distillation) helps, allowing us to achieve 40.4% mask mAP on unseen categories (Section B.2); (2) our models have reached a likely saturation point in terms of mask quality on COCO (Section B.1) — the implication is that future improvements on this particular benchmark are far more likely to come from detection; and (3) we demonstrate that we can achieve surprisingly strong mask generalization results with just 1 seen class (depending on the class, Section C).

We summarize our main contributions as follows:

- We identify the strong mask generalization effect in partially supervised instance segmentation and show that it is general, holding across underlying detectors like Mask R-CNN [16] and CenterNet [55] or without a detector, and across different backbones (Section 6).
- In order to unlock strong mask generalization, we show that it is necessary to train using tight groundtruth boxes instead of a combination of groundtruth and noisy proposals. We revisit vanilla Mask R-CNN with this insight and show that this change alone dramatically improves the performance on unseen classes (Section 5).
- We identify characteristics of mask-head architectures that lead to strong mask generalization (Section 7). Among other things, we find that Hourglass [36] architectures offer excellent performance. We use these findings to achieve state-of-the-art results on the COCO partially supervised instance segmentation task (Section 8) with our CenterNet and Mask R-CNN based models, Deep-MAC and Deep-MARC .

2. Related work

Object detection and instance segmentation. There has been a significant progress over the last decade in detection with successful convolutional models like OverFeat [44], YOLO [39, 40, 41, 2], Multibox [45, 8], SSD [35], RetinaNet [31], R-CNN and Fast/Faster versions [12, 11, 43, 17], EfficientDet [46], etc. While many of these works initially focused on box detection, more recently, many benchmarks have focused on the more detailed problem of instance segmentation (COCO [32], OID v5 [28, 1], LVIS [14]) and panoptic segmentation (COCO-Panoptic, [25]) which are arguably more useful tasks in certain applications. A major milestone in this literature was Mask R-CNN [16] which influenced many SOTA approaches today (e.g., [38, 34]) and by itself continues to serve as a strong baseline.

Anchor-free methods. State-of-the-art methods today are predominantly built on anchor-based approaches which predict classification/box offsets relative to a collection of fixed boxes arranged in sliding window fashion (called "anchors"). While effective, the performance of anchor-based methods often depend on manually-specified design decisions, e.g. anchor layouts and target assignment heuristics, a complex space to navigate for practitioners.

In recent years, however, this monopoly has been broken with the introduction of competitive "anchor-free" approaches [29, 7, 47, 56, 55, 26, 57, 4]. These newer anchor-free methods are simpler, more amenable to extension, offer competitive performance and consequently are beginning to be popular. Our anchor-free model (Section 3, Deep-MAC) in particular builds on the "CenterNet" architecture [55].

Due to the recency of competitive anchor-free methods there are fewer anchor-free instance segmentation approaches in literature. [30, 51, 52, 10] all add mask prediction capabilities on top of the (anchor-free) FCOS [47] framework. While the primary focus of our work is partial supervision, the fully supervised version of our model adds to this growing body of work, offering strong performance among anchor-free instance segmentation approaches.

Box-only supervision for instance segmentation. The above methods rely on access to massive labeled datasets which are costly to develop, with mask annotations especially so compared to box annotations. Researchers have thus begun to develop methods that are less reliant on mask annotations. In one formulation of this problem (which we might call *strictly box-supervised*) we ask to learn an instance segmentation model given only box annotations and no masks [24, 42, 19, 23, 48]. However this is intuitively a difficult approach and the performance of all of these methods is still a far cry from fully supervised performance of a strong baseline particularly at high IOU thresholds for mAP.

Partial supervision for instance segmentation Instead of going to the extreme end of discarding all mask annotations, Hu et al. [20] introduced the *partial supervision* formulation which allows for mask annotations from a small subset of classes to be used along with all box annotations. [20] observed that the "obvious" baseline of using a class-agnostic version of Mask R-CNN yielded poor results and proposed a method (Mask^X) for learning to predict mask-head weights given box-head weights hoping that this learned function will generalize to classes whose masks are not observed at training time.

Later papers [27, 10] however revisited the approach of attaching a class-agnostic mask-head on top of a detector, in both cases introducing novel architectures and additional losses to significantly improve generalization to novel classes. ShapeMask [27] builds on RetinaNet, learning a low dimensional shape space from observed masks and uses projections to this space to guide mask estimation; they also introduce a simple method to "condition" features cropped from the backbone on the instance that is being segmented. CP-Net [10], which is the current state of the art on this problem builds on FCOS [47], adding boundary prediction and attention-based aggregation in the mask branch.

We take a similar approach of using a class-agnostic mask-head, but while the ideas explored in these prior works are clearly beneficial, our objective is to demonstrate that mask-head architecture itself plays an underappreciated but significant role in generalization. Notably, by exploiting out-of-the-box architectures with strong mask generalization properties, we show that with only minor tweaks to the training procedure (Sections 5, 6) both of our models, Deep-MAC and Deep-MARC have state of the art performance in the partial supervision task.

3. Crop-then-segment instance segmentation

In this paper we consider a general family of "crop-then-segment" models that apply a per-instance crop (RoIAlign, [16]) operation after a feature extractor and pass the cropped features to a class-agnostic mask-head. For example, in our experiments, we use two detection-based instances of this family building on Mask R-CNN (anchor based) and Centernet (anchor-free), as well as a model that does not perform detection (and is simply provided with bounding boxes as input at test time). A schematic representation of this model family is drawn in Figure 2.

We focus specifically on two choices that one can make for models within the crop-then-segment family: (1) whether to crop to groundtruth boxes or both groundtruth boxes and proposals when training the mask heads (of the detector based models), and (2), which mask-head architecture to use. As we show, in order to achieve strong mask generalization, it is critical to (1) train with only groundtruth and (2) use significantly deeper mask head architectures

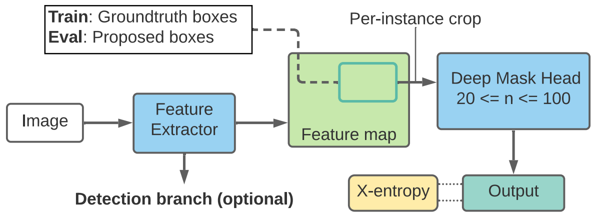


Figure 2: Diagram of the crop-then-segment instance segmentation model family. We identify the two crucial features that lead to strong mask generalization: Using a deeper class-agnostic mask-head and training it with *only* groundtruth boxes. At test-time, we use predicted boxes. The detection branch is marked optional because we show in Section 6 that it's not required for strong mask generalization.

than what is commonly used. To emphasize these aspects, we refer to our modified detection based models as *Deep-MARC (Deep Mask-heads Above R-CNN)* and *Deep-MAC (Deep Mask-heads Above CenterNet)*.

In both cases we keep the detection part of our models unchanged from the standard implementation and make only minimal changes where required to be compatible with our mask head architectures. Below we discuss our modifications to Mask R-CNN and CenterNet more in detail.

Deep-MARC: a Mask R-CNN based model. Deep-MARC is based on a class-agnostic version of Mask R-CNN [16] where we crop to only groundtruth boxes at training time (as mentioned above) and make minor changes to the mask prediction branch of Mask R-CNN, leaving the detection branch untouched.

Mask R-CNN by default crops its feature maps (using RoIAlign) to 14×14 resolution and upsamples to 28×28 before predicting per-instance masks. At test-time these are re-aligned with respect to the original box and resized to the resolution of the original image. When evaluating Mask R-CNN with its default mask-head, we keep this pathway untouched. Our implementation of Hourglass (HG) networks however requires its input size to be of the form 2^n due to its successive down-sampling and up-sampling layers. For our HG-20 mask-head we crop feature maps to 16×16 and upsample to 32×32 before predicting a class-agnostic mask. For our HG-52 mask-head, the crop and output size is doubled to 32×32 and 64×64 respectively. For Deep-MARC, we do not use any additional inputs to the mask-head.

Deep-MAC : an anchor-free model. Our Deep-MAC architecture builds instance segmentation capabilities on top of CenterNet [55],[1] a popular anchor-free detection approach, which models objects relative to their centers. For predicting bounding boxes, CenterNet outputs 3 tensors: (1) a class-specific heatmap which indicates the probability of the center of a bounding box being present at each location, (2) a class-agnostic 2-channel tensor indicating the height and width of the bounding box at each center pixel, and (3) since the output feature map is typically smaller than the image (stride 4 or 8), CenterNet also predicts an x and y direction offset to recover this discretization error at each center pixel.

Predicting instance masks with CenterNet (Deep-MAC). In parallel to the box-related prediction heads, we add a fourth *pixel embedding* branch P. For each bounding box b, we crop a region P_b from P corresponding to b via ROIAlign [16] which results in a 32×32 tensor. We then feed each P_b to a mask-head whose architecture is discussed in Section 6. Our final prediction at the end is a class-agnostic 32×32 tensor which we pass through a sigmoid to get per-pixel probabilities. We train this mask-head via a per-pixel cross-entropy loss averaged over all pixels and instances. During post-processing, the predicted mask is re-aligned according to the predicted box and resized to the resolution of the image.

In addition to this 32×32 cropped feature map, we add two inputs for improved stability of some mask-heads (but note that our main findings *do not depend* on having these additional inputs; see Appendix A.2.1): **(1) Instance embedding:** We add an additional head to the backbone that predicts a per-pixel embedding. For each bounding box b we extract its embedding from the center pixel. This embedding is tiled to a size of 32×32 and concatenated to the pixel embedding crop. This helps condition the mask-head on a particular instance and disambiguate it from others. **(2) Coordinate Embedding:** Inspired by CoordConv [33], we add a $32 \times 32 \times 2$ tensor holding normalized (x, y) coordinates relative to the bounding box b.

4. Experimental Setup

For all experiments in this paper we follow the typical partially supervised experimental setup for the COCO dataset with the 20 Pascal VOC [9] categories having instance masks at training time (as the *seen* categories) and the remaining 60 non-VOC classes not having instance masks at training time (as the *unseen* categories). In this case, we mainly care about performance on the 60 unseen (Non-VOC) categories since it is more challenging than the opposite variant and use this setting by default. We denote this training setting as `VOC-Masks-Only`. The only exception is Table 8 where we evaluate both variants to compare with other methods. All evaluations are performed on the `coco-val2017` set.

We train all mask heads with sigmoid cross entropy, and to handle the partially annotated training data, mask loss for each instance is only considered if a groundtruth mask is available. Below, we discuss experimental details specific to Deep-MARC and Deep-MAC. For reference, the fully supervised performance on COCO for Deep-MAC is 39.4 mAP and for Deep-MARC is 42.8 mAP on

[1] Not to be confused for the CenterNet from Duan et al. [7].

M.H. Train	Resnet	Mask mAP		
		Overall	VOC	Non-VOC
Prop.+GT	50	23.5	39.5	18.2
GT-Only	50	29.4	39.7	25.9
Prop.+GT	101	24.9	40.9	19.6
GT-Only	101	32.2	41.1	29.3

Table 1: Impact of Mask R-CNN mask-head training (M.H. Train) strategies on generalization to unseen classes with Resnet-50-FPN and Resnet-101-FPN backbones. All results are reported with the `VOC-Masks-Only` setting. There is a dramatic improvement in the performance on unseen classes (Non-VOC) when we train the mask-head with only groundtruth boxes. When evaluating, we use predicted boxes.

`coco-testdev2017` (see Appendix for details, Table 9), which is competitive with ShapeMask [27] and CP-Mask [10].

Deep-MARC. We train Deep-MARC with ResNet [18] backbones at 1024×1024 resolution with the $3\times$ schedule from Detectron2 [50]. When using the SpineNet [6] backbone we train at 1280×1280 resolution and use "Protocol C" [6]. The ResNet backbones are initialized from an ImageNet checkpoint whereas the SpineNet models are trained from scratch. All our models use synchronized batch-normalization [22, 15]. Deep-MARC is implemented in the TF Vision API [5]. We only alter the implementation of Mask R-CNN to support training with groundtruth boxes. All other detection and optimization hyperparameters are kept unchanged from their defaults.

Deep-MAC. We use a pixel embedding layer with 16 channels and an instance embedding layer with 32 channels. For all of our models, we use a mask loss weight of 5.0 and train with synchronized batch-normalization. We use the Hourglass-104 [36] backbone for experiments, unless noted otherwise. Our best models which beat state-of-the-art (Section 8) are trained at 1024×1024 resolution, with weights initialized from a COCO detection checkpoint. All other models are trained at 512×512 and initialized from an ExtremeNet [56] checkpoint inline with the original implementation of CenterNet [55]. For our best results we use CenterNet with the Hourglass [36] backbone. Deep-MAC is built on top of the open-source CenterNet implementation in the TF Object Detection API [21]. All other detection and optimization hyperparameters are kept unchanged from their defaults.

5. Cropping to *only* groundtruth boxes

Mask R-CNN is typically trained by performing ROIAlign on a combination of groundtruth boxes and proposals — this is a natural approach as it allows for the training distribution to be statistically more similar to the test time distribution and can even be thought of as a form of data augmentation. And for full supervision setups indeed it is slightly better to train with both groundtruth boxes and proposals (e.g. Table 12, Appendix). Our first surprising finding is that this situation is dramatically reversed on unseen classes in partially supervised setups, where we find that it is *far* better to train with only groundtruth boxes.

This effect is illustrated in Table 1, where we see that "naive" Mask R-CNN (**Prop+GT** rows) achieves extremely low mAP on unseen classes (relative, e.g, to seen classes), which is consistent with previous literature [20]. Training with groundtruth only (**GT-only** rows) on the other hand, dramatically improves performance for the Non-VOC (unseen) classes for which we do not provide masks at training time (+7.7mAP and + 9.7 mAP). Note that even with **GT-only** training, evaluation is always done with proposed boxes, as with all other methods we compare with.

Thus for the remainder of the paper we train only with groundtruth boxes unless otherwise specified. Why would it help so much to train with groundtruth boxes only? Our hypothesis (in Section 7) hinges on the finding of the next Section where we see that when training with groundtruth boxes only, mask-head architectures take on a new and significant role in generalization.

6. Going deeper with mask heads

In this section we pull our second lever by varying mask-head architectures. Our main finding is that *mask-heads affect generalization on unseen classes to a surprising extent*. In our experiments, we set our mask-heads to be Hourglass [36] (HG) and Resnets [18] (basic and bottleneck variants), with varying depth. We also use ResNet bottleneck [1/4$^{\text{th}}$], a variant of the ResNet (bottleneck) mask-head with $4\times$ fewer channels. We set the number of channels in the first layer of all mask-heads to 64, increasing this gradually in deeper layers (see Appendix, Section F). We also set the number of convolutions of each dimensionality roughly similar between mask-heads of similar depth.

Deep-MARC. To begin, let's continue with our Mask R-CNN based models (henceforth, Deep-MARC). In Table 2 we train ResNet-101-FPN based Deep-MARC models comparing the default mask-head (comprising 4 convolution layers) against the above out-of-the-box architectures (ResNet-4, HG-20, HG-52) and report mask mAP on seen and unseen classes. We first observe that when training with groundtruth boxes, the mAP on seen classes depends a little bit on the specific mask-head architecture, but the difference between worst and best case is relatively small ($40.3 \rightarrow 41.9$). However, for the same settings, the mAP on unseen classes varies much more significantly ($27.4 \rightarrow 34.4$). This indicates that mask-head architectures play a critical role in generalization to unseen classes, and not just by virtue of fitting the training classes better. In fact,

Mask-Head	VOC mAP		Non-VOC mAP	
	Prop. + GT.	**GT.**	**Prop. + GT.**	**GT.**
Default	40.9	41.1	19.6	29.3
ResNet-4	39.2	40.3	21.0	27.4
HG-20	41.6	41.4	20.6	33.8
HG-52	42.0	41.9	20.6	**34.4**

Table 2: Performance of Deep-MARC with different mask-heads under the `VOC-Masks-Only` setting with a ResNet-101-FPN backbone, comparing the performance when training the proposed boxes and groundtruth boxes (**Prop.+GT.**) and only groundtruth boxes (**GT.**). We see that performance on unseen classes depends significantly on the mask-head, but the benefit of better mask-heads is only apparent when training with groundtruth boxes. With the Hourglass (HG-52) mask-head and no other bells or whistles, Deep-MARC surpasses the previous state-of-the-art [10].

using an HG-52 mask-head without proposals is enough for Deep-MARC to surpass the previous SOTA [10].

To circle back to the previous section, we also see that this effect is tied to our choice to train with only groundtruth boxes — if we include proposals at training time, our models fare significantly worse on unseen classes and there is no clear signal on what mask-head architecture is best.

Deep-MAC. In Figure 3 we plot the results of a similar study for our anchor-free Deep-MAC model, this time cropping only to groundtruth but evaluating even more mask-head variants. And again we see a similar trend — while mAP on the seen classes depends a little bit on the specific mask-head architecture, the effect is small (38.8 $\rightarrow$ 40.0). However, for the same settings, the mAP on unseen classes varies significantly (25.0 $\rightarrow$ 32.5). We also see here that depth plays a role: empirically, it is important to go significantly beyond 4 layers to achieve the best performance. From a classical perspective, this is counterintuitive given the over-parameterization of very deep mask-heads, but perhaps is not so surprising in light of recent ways of rethinking generalization for deep learning [54, 53].

However, depth is not the only factor that drives generalization; Among the alternatives, the Hourglass mask-heads provide the best generalization performance to unseen classes for both Deep-MAC and Deep-MARC. And this is fortunate since it is also the most memory-efficient mask-head due to successive downsampling layers.

Finally, in Table 3, we show that our findings are not tied to our choice of Hourglass backbone. While comparing mask-heads when using ResNet-FPN and Hourglass backbones, we observe that performance on unseen classes is lower with ResNet backbones, but that mask-head architecture still strongly impacts generalization to unseen classes.

Strong mask generalization without a detector. To further make the point that the detection architecture does not play a critical role in our story, we consider a "detection-free" incarnation of our model family, in which we do not even require the model to produce detections. In this most basic of settings, we assume that a groundtruth box for each instance is provided as input and the task is to simply produce the correct segmentation mask. For this setting, we use the Deep-MAC architecture with an Hourglass backbone, cropping to each groundtruth box and passing the result to the mask-head. Since detection is no longer a task of interest, we drop all detection related losses and train only with sigmoid cross entropy loss for the masks. We also evaluate using the mean IOU metric instead of mask mAP.

Table 4 shows the results of this experiment using 3 different mask-head architectures. We observe that all archi-

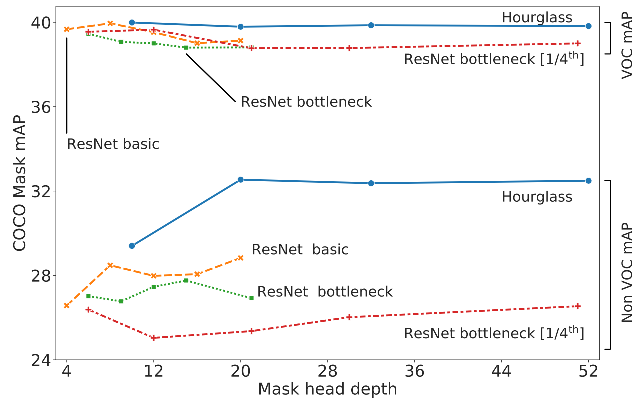


Figure 3: Effect of mask-head architecture and depth on instance segmentation performance over seen (VOC) and unseen (Non-VOC) classes. Although the performance on seen classes does not vary much across different architectures, there is significant variation in the performance on unseen classes. We report results with the `VOC-Masks-Only` setting.

Mask-Head	ResNet-101-FPN		Hourglass-104	
	Box	**Mask**	**Box**	**Mask**
ResNet-4	32.6	22.6	39.7	26.6
Hourglass-10	32.2	24.8	39.9	29.4
Hourglass-20	32.5	26.7	39.7	32.5

Table 3: Effect of Deep-MAC backbones on the performance of various mask-heads. Note that the box mAP is relatively unchanged as we train with all boxes. We train with the `VOC-Masks-Only` setting and report mask mAP.

Mask-Head	mIOU		
	Overall	**VOC**	**Non-VOC**
ResNet-4	67.0	78.6	62.1
Hourglass-20	78.6	81.0	77.8
Hourglass-52	78.9	81.1	79.2

Table 4: mIOU of Deep-MAC trained without any detection losses under the `VOC-Masks-Only` setting. Because we cannot compute mask mAP without a detector, we report mIOU computed over the full validation dataset and over VOC/non-VOC class splits. Hourglass mask-heads continue to show strong mask generalization on non-VOC classes, even when they are not coupled with a detector.

Mask-Head	Variant	Mask mAP		
		Overall	VOC	Non-VOC
ResNet-20	Default	31.4	39.1	28.8
Hourglass-20	Default	34.1	39.8	32.2
	No LRS	33.6	39.2	31.7
	No ED	31.7	39.1	29.2

Table 5: Isolating what makes Hourglass architectures achieve strong mask generalization. No LRS = No long range skip connections. No ED = No encoder-decoder structure, i.e, no downsampling or upsampling layers.

# Dilated conv layers replaced	Mask mAP		
	Overall	VOC	non-VOC
0	32.2	39.4	29.9
10	32.7	39.1	30.6
20	32.8	39.3	30.7

Table 6: Replacing regular convolutional layers with dilated convolutions (rate=2) in a ResNet-20 mask-head to isolate the effect of receptive field.

Backbone	Mask-Head	mIOU		
		Overall	VOC	Non-VOC
HG-52	HG-52	78.4	80.4	77.8
HG-104	ResNet-4	71.4	79.2	68.8

Table 7: Can we reproduce strong mask generalization by adding an hourglass network to the shared backbone instead of using it in the per-proposal mask-head? We compare two networks of similar depth where the first network has a deeper mask-head. For fair comparison, we use groundtruth boxes as input at evaluation time and report mIOU.

tectures have similar performance on the seen categories ($\sim$2.5% spread) whereas on unseen categories, the best mask-head (Hourglass-52) outperforms the worst (ResNet-4) by $>$16%. This confirms the strong mask generalization effect occurs in the detection-free setting and together with our results for Deep-MAC and Deep-MARC provide strong evidence that we would find similar effects using other detection architectures.

7. A closer look at mask-head architectures

Having established that mask-head architecture significantly affects strong mask generalization in crop-then-segment models, we now study the mask-heads in detail and identify the components that are most crucial. All ablations in this section are done on Deep-MAC and use the `VOC-Masks-Only` setting for training and evaluation.

What makes Hourglass mask-heads so good? We first address the question of which architectural elements are most responsible for the superior generalization of Hourglass networks. To investigate, we focus on 20-layer Hourglass and ResNet basic mask-heads. The Hourglass architecture differs in two main ways from ResNet, having a) an encoder-decoder structure in which the encoder downsamples the input and the decoder upsamples the result of the encoder, and b) long range skip connections connecting feature maps of the same size in the encoder and decoder.

To understand each difference in isolation, we explore the effect of (a) replacing the downsampling/upsampling layers with layers that do not change the feature map resolution, and (b) severing long range skip connections.

Table 5 shows the corresponding ablation results. We see that removing the long range skip connections (No LRS) has a small negative impact on the performance. More importantly, we find that the majority of the gap between ResNet and Hourglass is closed by getting rid of the encoder-decoder structure in the Hourglass mask-head (No ED). Given these results, we conclude that this style of downsampling followed by upsampling likely captures particularly appropriate inductive biases for segmentation.

What's so special about the mask-head? Next, given that an hourglass mask-head offers generalization advantages, we ask: could we reproduce these advantages by adding an hourglass network to the shared backbone instead of using it in the per-proposal mask-head? In other words, what is so special about the mask-head? Here we show that the answer is negative and that it is indeed the mask-head's architecture that impacts strong mask generalization.

Consider an HG-104 network which is a stack of two hourglass modules each with 52 layers. We compare (a) a model where all 104 layers are situated in the backbone and we use a simple ResNet-4 mask-head versus (b) a model with an HG-52 backbone and HG-52 mask-head. In both cases, inputs undergo roughly 100 layers contained within two hourglass modules but in the second case, the 52 layer mask-head is applied on a per-proposal basis.

Since using 52 layers in the backbone in general yields inferior detection quality compared to the 104 layer backbone, we use groundtruth boxes as input so that both models are on equal footing and we evaluate mIOU.

Our finding (Table 7) is that despite having slightly fewer total layers, our model with the 52 layer mask-head outperforms the model with the 4 layer mask-head by 9% mIOU on unseen classes (both models have similar performance on seen classes). More generally, this supports our hypothesis that within the entire architecture the mask-head plays a disproportionately significant role with respect to generalization to unseen classes.

Is it sufficient to have a large receptive field? Finally, given that depth and encoder/decoder structures do so well, it seems natural to conjecture that increased receptive field in these architectures may play a significant role.

To evaluate this hypothesis, we explore two additional families of mask-heads: (a) we replace the vanilla convolutions in a ResNet mask-head with dilated convolutions (w/

Model	b-box.	VOC → Non-VOC (mask)						Non-VOC → VOC (mask)						ms./im
	AP	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L	
Mask R-CNN [20]	38.6	18.5	24.8	18.1	11.3	23.4	21.7	24.7	43.5	24.9	11.4	25.7	35.1	56
Mask GrabCut [20]	38.6	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 [20]	38.6	23.8	42.9	23.5	12.7	28.1	33.5	29.5	52.4	29.7	13.4	30.2	41.0	—
ShapeMask [27]	45.4	33.2	53.1	35.0	18.3	40.2	43.3	35.7	60.3	36.6	18.3	40.5	47.3	224
CPMask [10]	41.5	34.0	53.7	36.5	18.5	38.9	47.4	36.8	60.5	38.6	17.6	37.1	51.5	—
Deep-MAC (ours)	44.5	35.5	54.6	38.2	19.4	40.3	50.6	39.1	62.6	41.9	17.6	38.7	54.0	232
Deep-MARC (ours)	48.6	**38.7**	**62.5**	**41.0**	**22.3**	**43.0**	**55.9**	**41.0**	**68.2**	**43.1**	**22.0**	**40.0**	**55.9**	170

Table 8: Partially supervised performance of Deep-MAC (CenterNet based) and Deep-MARC (Mask R-CNN based) compared to other models. We measure mask mAP on the `coco-val2017` set. The top row with label A → B indicates that we train on masks from set A and evaluate our masks on set B. Bounding box (b-box.) AP is an average over all classes. We use report inference time as milliseconds / image (ms./im) on a V100 GPU and compare with Detectron2 [50] and ShapeMask[27]. CPMask[10], Mask^X [20] R-CNN have not reported inference time.

rate 2), which has the effect of expanding receptive field without changing the depth or number of parameters, and (b) we use fully connected (MLP) mask-heads which have full receptive field. See Table 6 and Appendix A.2.2 for dilated and FC results, respectively.

Our experiments using both families of models show, first, that none of these models are able to reach the performance of Hourglass mask-heads, so there must be further factors at play beyond receptive field. On the other hand, growing the receptive field early seems to benefit generalization to some extent (e.g., shallow FC mask-heads outperform shallow convolutional mask-heads).

And this raises an interesting question which we leave for further study: what about receptive field would help unseen classes without simultaneously helping seen classes? Here we offer one conjecture based on our Mask R-CNN finding (Section 5), that it is important to train using groundtruth boxes instead of proposals. A groundtruth box, when tight on an instance, acts as a cue, indicating the object that is meant to be segmented. When trained on noisy proposals, we conjecture that Mask R-CNN tries to memorize the types of foreground classes seen at training time and thus struggles to generalize to unseen classes. With a precise cue, however, perhaps the model learns to compare interior pixels to boundary pixels to make this decision, a strategy that is more generalizable across categories and requires a large enough receptive field so that boundary pixels can interact with interior pixels.

8. Comparison with the state-of-the-art

We now train models at higher resolutions that previous sections with Deep-MAC being trained at 1024×1024 and Deep-MARC at 1280×1280. Deep-MAC uses an Hourglass-104 backbone and an Hourglass-100 mask-head, whereas Deep-MARC uses a SpineNet-143 [6] backbone and an Hourglass-52 mask-head. With these settings, Deep-MAC and Deep-MARC beat previous state-of-the-art approaches as seen in Table 8. Deep-MARC produces our best result which exceeds CPMask[10] on VOC to Non-VOC transfer by 4.7% and Non-VOC to VOC transfer by 4.2%. Compared to prior approaches, our method is end-to-end trainable and does not require auxiliary losses or specialty modules. Although Deep-MAC surpasses the state of the art by itself, we show in the Appendix (Section B.2) that we can do even better using distillation based training (achieving a Non-VOC mAP of 40.4% on the same problem).

9. Conclusions

In this work, we have identified and studied the surprising extent to which the mask-head architecture impacts generalization to unseen categories as well as the connection between this effect and the protocol of cropping to only groundtruth boxes at training time. Through extensive experiments, we demonstrated the generality of this effect across detection methodologies and backbone networks. And by exploiting this strong mask generalization effect, we established a new state of the art on this problem by a significant margin using a conceptually simple model.

While we have taken initial steps in understanding strong mask generalization, how to better understand the inductive biases encoded within mask-head architectures and how to explain our results theoretically remain important directions. Along these lines, we leave readers with pointers to two papers which have noted similar empirical phenomena where certain architectures generalize effectively to data outside of the training distribution. The Deep Image Priors work [49] similarly observed that Hourglass-style networks seem to automatically capture image level statistics in a natural way without being trained on data. [53] showed that sufficiently deep networks unlock a certain strong generalization behavior. We conjecture that there may be a common denominator at play and that exploring these synergies further would be a fruitful area of further research potentially yielding insights useful beyond segmentation.

Acknowledgements We would like to thank David Ross for thoughtful feedback and Pengchong Jin, Abdullah Rashwan and Xianzhi Du for helping with Mask R-CNN code.

References

[1] Rodrigo Benenson, Stefan Popov, and Vittorio Ferrari. Large-scale interactive object segmentation with human annotators. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 11700–11709, 2019. 3

[2] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*, 2020. 3

[3] Zhaowei Cai and Nuno Vasconcelos. Cascade r-cnn: high quality object detection and instance segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019. 1

[4] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. *arXiv preprint arXiv:2005.12872*, 2020. 3

[5] Chen Chen, Xianzhi Du, Le Hou, Jaeyoun Kim, Pengchong Jin, Jing Li, Yeqing Li, Abdullah Rashwan, and Hongkun Yu. Tensorflow official model garden, 2020. 5

[6] Xianzhi Du, Tsung-Yi Lin, Pengchong Jin, Golnaz Ghiasi, Mingxing Tan, Yin Cui, Quoc V Le, and Xiaodan Song. Spinenet: Learning scale-permuted backbone for recognition and localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11592–11601, 2020. 5, 8, 13

[7] Kaiwen Duan, Song Bai, Lingxi Xie, Honggang Qi, Qingming Huang, and Qi Tian. Centernet: Keypoint triplets for object detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 6569–6578, 2019. 3, 4

[8] Dumitru Erhan, Christian Szegedy, Alexander Toshev, and Dragomir Anguelov. Scalable object detection using deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2147–2154, 2014. 3

[9] Mark Everingham, SM Ali Eslami, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes challenge: A retrospective. *International journal of computer vision*, 111(1):98–136, 2015. 4

[10] Qi Fan, Lei Ke, Wenjie Pei, Chi-Keung Tang, and Yu-Wing Tai. Commonality-parsing network across shape and appearance for partially supervised instance segmentation. *arXiv preprint arXiv:2007.12387*, 2020. 2, 3, 5, 6, 8, 14

[11] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015. 3

[12] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Region-based convolutional networks for accurate object detection and segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 38(1):142–158, 2015. 3

[13] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256. JMLR Workshop and Conference Proceedings, 2010. 12

[14] Agrim Gupta, Piotr Dollar, and Ross Girshick. Lvis: A dataset for large vocabulary instance segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5356–5364, 2019. 3, 13

[15] Kaiming He, Ross Girshick, and Piotr Dollár. Rethinking imagenet pre-training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4918–4927, 2019. 5

[16] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017. 1, 2, 3, 4

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 3

[18] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 5

[19] Cheng-Chun Hsu, Kuang-Jui Hsu, Chung-Chi Tsai, Yen-Yu Lin, and Yung-Yu Chuang. Weakly supervised instance segmentation using the bounding box tightness prior. In *Advances in Neural Information Processing Systems*, pages 6586–6597, 2019. 3

[20] Ronghang Hu, Piotr Dollár, Kaiming He, Trevor Darrell, and Ross Girshick. Learning to segment every thing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4233–4241, 2018. 1, 2, 3, 5, 8

[21] Jonathan Huang, Vivek Rathod, Chen Sun, Menglong Zhu, Anoop Korattikara, Alireza Fathi, Ian Fischer, Zbigniew Wojna, Yang Song, Sergio Guadarrama, et al. Speed/accuracy trade-offs for modern convolutional object detectors. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7310–7311, 2017. 5

[22] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015. 5

[23] Hoel Kervadec, Jose Dolz, Shanshan Wang, Eric Granger, and Ismail Ben Ayed. Bounding boxes for weakly supervised segmentation: Global constraints get close to full supervision. *arXiv preprint arXiv:2004.06816*, 2020. 3

[24] Anna Khoreva, Rodrigo Benenson, Jan Hosang, Matthias Hein, and Bernt Schiele. Simple does it: Weakly supervised instance and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 876–885, 2017. 3

[25] Alexander Kirillov, Kaiming He, Ross Girshick, Carsten Rother, and Piotr Dollár. Panoptic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 9404–9413, 2019. 3

[26] Tao Kong, Fuchun Sun, Huaping Liu, Yuning Jiang, Lei Li, and Jianbo Shi. Foveabox: Beyound anchor-based object detection. *IEEE Transactions on Image Processing*, 29:7389–7398, 2020. 3

[27] Weicheng Kuo, Anelia Angelova, Jitendra Malik, and Tsung-Yi Lin. Shapemask: Learning to segment novel objects by refining shape priors. In *Proceedings of the IEEE*

International Conference on Computer Vision, pages 9207–9216, 2019. 2, 3, 5, 8, 14
[28] Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Tom Duerig, et al. The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale. *arXiv preprint arXiv:1811.00982*, 2018. 3
[29] Hei Law and Jia Deng. Cornernet: Detecting objects as paired keypoints. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 734–750, 2018. 3
[30] Youngwan Lee and Jongyoul Park. Centermask: Real-time anchor-free instance segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13906–13915, 2020. 3
[31] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017. 3
[32] 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 *European conference on computer vision*, pages 740–755. Springer, 2014. 1, 3
[33] Rosanne Liu, Joel Lehman, Piero Molino, Felipe Petroski Such, Eric Frank, Alex Sergeev, and Jason Yosinski. An intriguing failing of convolutional neural networks and the coordconv solution. In *Advances in Neural Information Processing Systems*, pages 9605–9616, 2018. 4
[34] Shu Liu, Lu Qi, Haifang Qin, Jianping Shi, and Jiaya Jia. Path aggregation network for instance segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8759–8768, 2018. 3
[35] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer, 2016. 3
[36] Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation. In *European conference on computer vision*, pages 483–499. Springer, 2016. 2, 5
[37] Dim P Papadopoulos, Jasper RR Uijlings, Frank Keller, and Vittorio Ferrari. Extreme clicking for efficient object annotation. In *Proceedings of the IEEE international conference on computer vision*, pages 4930–4939, 2017. 1
[38] Siyuan Qiao, Liang-Chieh Chen, and Alan Yuille. Detectors: Detecting objects with recursive feature pyramid and switchable atrous convolution. *arXiv preprint arXiv:2006.02334*, 2020. 1, 3
[39] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016. 3
[40] Joseph Redmon and Ali Farhadi. Yolo9000: better, faster, stronger. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7263–7271, 2017. 3
[41] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018. 3
[42] Tal Remez, Jonathan Huang, and Matthew Brown. Learning to segment via cut-and-paste. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 37–52, 2018. 3
[43] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015. 3
[44] Pierre Sermanet, David Eigen, Xiang Zhang, Michaël Mathieu, Rob Fergus, and Yann LeCun. Overfeat: Integrated recognition, localization and detection using convolutional networks. *arXiv preprint arXiv:1312.6229*, 2013. 3
[45] Christian Szegedy, Alexander Toshev, and Dumitru Erhan. Deep neural networks for object detection. In *Advances in neural information processing systems*, pages 2553–2561, 2013. 3
[46] Mingxing Tan, Ruoming Pang, and Quoc V Le. Efficientdet: Scalable and efficient object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10781–10790, 2020. 3
[47] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. Fcos: Fully convolutional one-stage object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 9627–9636, 2019. 3
[48] Zhi Tian, Chunhua Shen, Xinlong Wang, and Hao Chen. Boxinst: High-performance instance segmentation with box annotations. *arXiv preprint arXiv:2012.02310*, 2020. 3
[49] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Deep image prior. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9446–9454, 2018. 8
[50] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. `https://github.com/facebookresearch/detectron2`, 2019. 5, 8
[51] Enze Xie, Peize Sun, Xiaoge Song, Wenhai Wang, Xuebo Liu, Ding Liang, Chunhua Shen, and Ping Luo. Polarmask: Single shot instance segmentation with polar representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12193–12202, 2020. 3
[52] Hui Ying, Zhaojin Huang, Shu Liu, Tianjia Shao, and Kun Zhou. Embedmask: Embedding coupling for one-stage instance segmentation. *arXiv preprint arXiv:1912.01954*, 2019. 3
[53] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Michael C Mozer, and Yoram Singer. Identity crisis: Memorization and generalization under extreme overparameterization. *arXiv preprint arXiv:1902.04698*, 2019. 6, 8
[54] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. *arXiv preprint arXiv:1611.03530*, 2016. 6
[55] Xingyi Zhou, Dequan Wang, and Philipp Krähenbühl. Objects as points. *arXiv preprint arXiv:1904.07850*, 2019. 2, 3, 4, 5, 13
[56] Xingyi Zhou, Jiacheng Zhuo, and Philipp Krahenbuhl. Bottom-up object detection by grouping extreme and center

points. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 850–859, 2019. 3, 5

[57] Chenchen Zhu, Yihui He, and Marios Savvides. Feature selective anchor-free module for single-shot object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 840–849, 2019. 3