Foreground-Specialized Model Imitation for Instance Segmentation

Dawei Li*†[1], Wenbo Li*[2], and Hongxia Jin[2]

[1] Amazon Lab126
daweili@amazon.com
[2] Samsung Research America AI Center
{wenbo.li1,hongxia.jin}@samsung.com

Abstract. Instance segmentation is formulated as a multi-task learning problem. However, knowledge distillation is not well-suited to all subtasks except the multi-class object classification. Based on such a competence, we introduce a lightweight foreground-specialized (FS) teacher model, which is trained with foreground-only images and highly optimized for object classification. Yet, this leads to discrepancy between inputs to the teacher and student models. Thus, we introduce a novel Foreground-Specialized model Imitation (FSI) method with two complementary components. First, a reciprocal anchor box selection method is introduced to distill from the most informative output of the FS teacher. Second, we embed the foreground-awareness into student's feature learning via either adding a co-learned foreground segmentation branch or applying a soft feature mask. We conducted an extensive evaluation against the others on COCO and Pascal VOC.

Keywords: Knowledge distillation · Instance segmentation.

1 Introduction

To deploy deep learning models on resource-constrained edge devices, researchers have been devoting efforts in four major directions: (1) model compression, i.e., quantization and pruning [10]; (2) better light-weight backbones such as MobileNet [14] and ShuffleNet [35]; (3) reduced model architecture such as YOLO [28] and SSD [24] for one-stage object detection and (4) model imitation which trains a compact and device-friendly model to imitate the behavior a more powerful yet more computationally expensive model. The focus of this work is to improve the model imitation for object instance segmentation.

A typical and widely used model imitation method is KD [13] which transfers the knowledge from a large model with stronger generalization capability to a lightweight model, while both models are trained on the same dataset. One major challenge faced by the existing KD methods is that the teacher models where the knowledge is transferred from normally have many layers and parameters, e.g.,

* Equal contributions. † This work was performed at Samsung Research America.

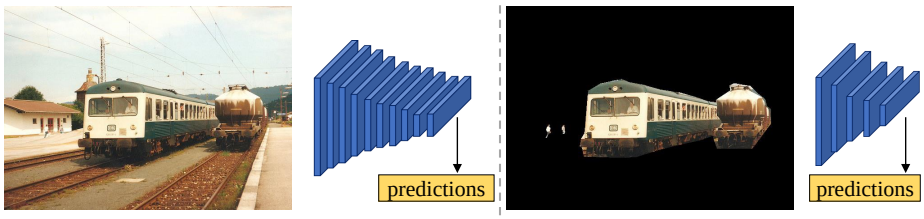


Fig. 1. The difference between teacher models in traditional knowledge distillation (KD) and our foreground-specialized teacher model. **(a)** In KD, the teacher model is normally much larger than the student model. **(b)** The foreground-specialized teacher model can be as small as the student model and thus can be efficiently trained and imitated.

using VGGNet [31] as the teacher backbone, so training and distilling from such models are time-consuming and would rely on high-end computing devices with large memories[3]. In addition, a recent work [6] shows that the final accuracy of the student model does not increase monotonically with the size of the teacher model. As the teacher model gets larger, the accuracy of the student model first increases and then decreases. This phenomenon means that substantial effort would be required to explore the optimal teacher model in order to achieve the most accurate student model.

The goal of the object instance segmentation is to detect and delineate each distinct object of interest from an input image. It is normally formulated as a multi-task learning problem including bounding box detection, object classification and mask prediction [11]. The multi-task learning formulation poses additional challenges for applying the KD to object instance segmentation, because the KD is designed exclusively for the multi-class classification problem, i.e., the object classification, which is only one out of three sub-tasks of the object instance segmentation. Therefore, it is counterproductive to apply KD to the other two sub-tasks of the object instance segmentation, i.e., bounding box detection and mask prediction, which KD is not designed for.

In order to resolve the above challenges, we focus on exploring a more effective and efficient way of model imitation for the object instance segmentation. Unlike the conventional KD in which the teacher model takes the same input as the student model, we simplify the input to the teacher model in order to relieve its learning burdens on the two sub-tasks, i.e., bounding box detection and mask prediction, but makes the teacher model dedicated to object classification which KD is designed exclusively for. Specifically, as shown in Figure 1, we simplify the input to the teacher model by removing the background stuff from the training images while leaving only pixels for the foreground objects. In this fashion, it will

[3] In distillation, the memory cost can be reduced when the outputs of teacher models are pre-computed. Yet, this disables on-the-fly data augmentation, a critical component for improving the model accuracy especially when the dataset is small.

Table 1. Model accuracy (mAP@[0.5,.95]) comparison on COCO dataset by transforming the input with different backbone architectures of YOLACT. **Standard:** the model trained and validated using the complete images. **FS-Teacher:** the model trained and validated using foreground-only images.

Model	ResNet-18	ResNet-50	ResNet-101
Standard	23.76	27.97	29.73
FS-Teacher	41.08	46.69	47.86

be tremendously simpler for a teacher model on the bounding box detection and the mask prediction, because the foreground objects are salient on the vacant background.

We name the teacher model trained on the simplified input as the foreground-specialized (FS) teacher model. As shown in Table 1, we observe a significant performance gain achieved by the teacher models on the simplified input, over that achieved by the standard teacher models on the original input. One may argue the "unfairness" of the comparison between the FS teacher models and the standard ones in Table 1 because of the presence/absence of the input simplification on the validation data, i.e., foreground-only images vs. the original images. Yet, the KD method is mainly concerned about the output of a teacher model while barely favoring preprocessing tricks, so the performance gain is still quite beneficial despite the negligible "unfairness". In Table 1, note that even the FS teacher model based on the backbone ResNet-18 performs much better than the standard teacher model based on ResNet-101. As such, it can enable a significantly more efficient teacher model.

As shown in Figure 1, compared to the standard teacher model in KD, the FS teacher model takes different input but shares the output format. To accommodate this new change, we introduce a novel Foreground-Specialized model Imitation (FSI) which includes two complementary modules that allow the student model to better imitate the teacher. First, instead of distilling knowledge from all three types of teacher output, i.e., classification, bounding boxes, and instance masks, we only distill knowledge from the teacher's classification output which is what the teacher model was designed exclusively for. In addition, to deal with the highly unbalanced positive and negative anchor boxes, a reciprocal anchor box selection method is introduced to distill knowledge based on those most informative teacher outputs. Second, though we could not filter out the background from the input of the student model with ease, the student model can be encouraged to better imitate the teacher by embedding the foreground-awareness into the feature learning. Particularly, we introduce two solutions by either applying a learned latent soft foreground mask to the intermediate convolution features (at the cost of reduced inference speed) or co-learning a foreground segmentation task by attaching a branch to the student's backbone (not affecting the inference speed). Both solutions are demonstrated to be effective in improving the student model's accuracy through extensive evaluations.

2 Related Works

It is a challenging task to deploy deep neural networks on pervasive mobile and edge computing devices that have limited computing resources. Towards this goal, model compression techniques [10] have been introduced to approximate a given deep learning model with a compact one that reduces the storage cost, and representative methods include quantization [16, 27], pruning [12], and low-rankness [17]. Model compression is effective in reducing the model size, however, the accuracy is generally bounded by the model before compression and may rely on specialized hardware and/or software support for speedup [9].

More complicated vision tasks like object detection and instance segmentation rely on a multi-stage architecture [11, 29] to achieve favorable accuracy at the cost of heavy computation. To enable on-device inference, researchers have come up with a more effective approach that uses a single-stage architecture [28, 24, 2] to substantially reduce the computational cost while still achieving satisfactory accuracy. Key technologies behind the success include multi-scale anchor boxes [24], feature pyramid networks [21], focal loss [22], etc.

On the other hand, model imitation technologies, mostly, KD [13] have been introduced to enforce additional guidance using a large teacher model's prediction in addition to the ground truth labels. The KD method was first introduced in the classification problem and the insight behind it is that a pre-trained large model on the same dataset has already learned the essential underlying relationship among different classes that can be generalized to new unseen samples, e.g., cars are close to trucks but are quite different from apples, and this essential information is reflected by the model's output. Later works have extended KD to other problems including object detection [3, 33], semantic segmentation [25], and sequence learning problems [18]. However, a recent thorough evaluation [6] on the efficacy of KD indicates that larger models do not often make better teachers due to mismatched capacity of teacher and student models.

Another research topic related to our work is learning using privileged information (LUPI) [32, 19], which assumes additional information or modality about the data is provided at training but may not be available at test time. Most existing LUPI works assume the extra data modalities, e.g., the depth modality, can be easily obtained or generated [20, 26, 8, 5].However, for many real application scenarios, the majority training data are large-scale crowd-sourced wild data which does not have such extra information. Instead, our approach generates specialized training data from the ground-truth labels as a resolution to simplify the task and train strong specialized models. In addition, only our work uses such models as the KD teachers.

3 Method

In this section, we present our method in detail. We first give a brief recap of the YOLACT one-stage object instance segmentation method. Then we describe how the FS teacher model is trained and the necessity of imitating the model.

Table 2. Misclassification errors on COCO dataset.

Model	Conf@0.5	Conf@0.7	Conf@0.9
Standard	49.71%	42.59%	37.67%
FS-Teacher	38.63%	30.87%	23.34%

After that, we delineate our model imitation approach targeting the peculiarities that are different from the existing KD method.

3.1 Recap of YOLACT

Two-stage object instance methods such as Mask-RCNN [11] are not suitable for on-device inference. Following the idea of single-stage object detection methods [28, 24], researchers have recently proposed one-stage object instance segmentation methods [34, 2] enabling real-time inference speed on mobile and edge devices. Among those methods, YOLACT [2] is the state-of-the-art solution considering both the accelerated inference speed and satisfactory segmentation accuracy. Compared to Mask-RCNN, it achieves about 4× speed-up with fairly close accuracy.

YOLACT is based on the RetinaNet [22] architecture which is composed of a feature pyramid network (FPN) [21] based backbone and a set of classification and box regression branches following each pyramid feature map to predict the class category and bounding box coordinates for each anchor box. On top of RetinaNet, YOLACT adds a protonet branch to infer a set of latent mask prototypes and further, for each anchor box, predicts a coefficient vector for composing the mask prototypes. The final output mask for each anchor box is the weighted sum of the latent mask prototypes given the predicted coefficients.

3.2 FS Teacher Training

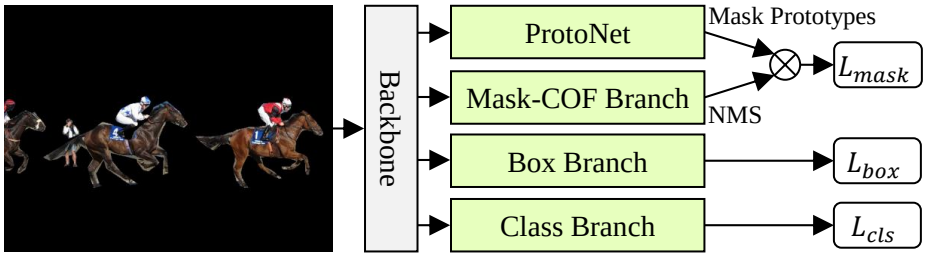


Fig. 2. The training of a teacher based on YOLACT.

The FS model is trained using exactly the same model architecture and outputs as introduced in § 3.1 and is illustrated in Figure 2. The only change is that the background of the input image is removed. This input transformation does not require additional annotation efforts since the foreground can be conveniently determined as the union of masks for each foreground object. Pixels not covered by the foreground mask is set to 0 (i.e., black) before feeding into the model.

The teacher model does not need to be larger than the student model and we demonstrate that the student model could achieve substantial accuracy improvement when both the teacher model and the student model use the same backbone network (i.e., ResNet-18). However, it must be noted that (1) the output feature map sizes from both models' backbone and (2) the number of anchor boxes are required to be consistent between the two models so that each anchor box from the student model can find a unique mapping from the teacher, and thus it knows where it can distill the knowledge.

Boosted classification accuracy In this subsection, we discuss and validate that the FS teacher model has significantly improved classification performance. For instance segmentation, there could be two major types of errors. (1) Object not detected, i.e., there's no positive object detected or a detected object with IoU below a given threshold. (2) Misclassification, i.e., an object is detected with enough IoU to a ground-truth but is wrongly classified as a different class. The FS teacher model is trained without background interference which means that the training burden is greatly reduced to mostly classifying objects and identifying the object boundaries (especially for overlapped objects).

We measure the misclassification errors of the standard model which is trained and validated using the complete images and the FS model and we show the comparison in Table 2. Specifically, we calculate the misclassification error as the percentage of objects (averaged over all classes) which (1) has been detected with an IoU above 0.75 to a ground-truth object and (2) is wrongly classified. In addition, we set different confidence thresholds on each measurement, e.g., Conf@0.5 means we use the detected objects that are classified to a object class with confidence score greater than 0.5. The measured result indicates that the FS teacher model can reduce the misclassification errors by more than 14%.

Challenges on distilling from FS model The challenges are mostly from the fact that the teacher and the student models have different input. For the student model, there's no trivial way of obtaining foreground masks in real-time inference. Applying foreground segmentation (using models like DeepLab [4]) as pre-processing not only incurs extra computational burden but also cannot guarantee a flawless foreground mask. Therefore, instead of manipulating on the input inference images, we propose a new distillation method to have the student model effectively imitate the FS teacher model as described in detail in the following subsection.

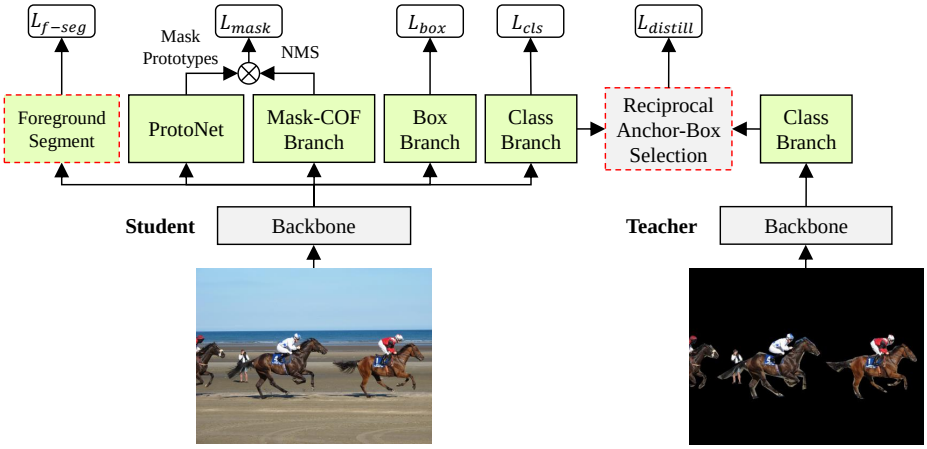


Fig. 3. Overview of the imitation training with FSI. The *Reciprocal Anchor-Box Selection* module identifies the critical knowledge from the teacher for the student to learn; the *Foreground Segment* module embeds the foreground-awareness into the student's feature learning.

3.3 FSI: Foreground-Specialized Model Imitation

As shown in Fig. 3, FSI is contains two novel modules: (1) reciprocal anchor box selection and (2) foreground segment.

Reciprocal anchor box selection The core idea is to have the student to learn from only what the teacher is specialized for. The FS teacher has been trained to concentrate on classifying the foreground objects while it hasn't been challenged much to predict the masks or to regress the bounding boxes, considering that the image background has been zeroed out. This means the distillation should only focus on teacher's classification output for each anchor box. Meanwhile, distilling the classification output for all anchor boxes is highly inefficient given the extremely unbalanced distribution of positive and negative boxes and would absorb teacher's adverse knowledge when it makes a wrong prediction.

Therefore, we propose a reciprocal anchor box selection method to effectively learn the most critical knowledge from the teacher. Concretely, the selected knowledge includes the teacher's "opinion" on two sets of anchors. The first set includes anchor boxes that the teacher gives positive opinions, i.e., correctly predicted as foreground objects. This part represents the essence of teacher's knowledge that the student should absorb. The second set includes anchor boxes that have been wrongly predicted as foreground objects by the student but correctly predicted as background by the teacher. This represents student's wrong knowledge. In Algorithm 1, we present the pseudo-code for the method and we tensorize the operations for fast training in our implementation.

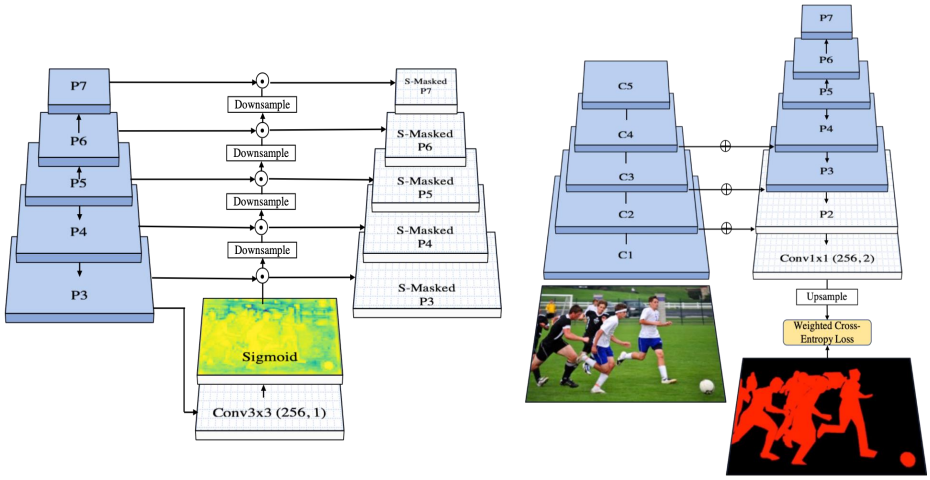
Algorithm 1: Reciprocal Anchor Box Selection

Input : $\hat{Y}^t_{cls}$ and $\hat{Y}^s_{cls}$ are classification output from the *teacher* model and the *student* model of all anchor boxes S, respectively.
Output: S_{recip} is a set of selected anchor boxes for distillation.

begin
  $S_{recip} \leftarrow \emptyset$ **for** $i = 1 \rightarrow size(S)$ **do**
    $\hat{y}^t_i = \hat{Y}^t_{cls}.\text{get}(i)$; $t_{cls} = argmax(\hat{y}^t_i)$; `// teacher prediction`
    $\hat{y}^s_i = \hat{Y}^s_{cls}.\text{get}(i)$; $s_{cls} = argmax(\hat{y}^s_i)$; `// student prediction`
    if $t_{cls} = gt_i$ *and* $(t_{cls} > 0$ *or* $s_{cls} > 0)$; `// 0 represents background`
    then
      S_{recip}.insert(i)

Foreground-aware feature learning FS teacher extracts features without the background interference. To imitate the teacher better, we propose two alternative approaches to embed the foreground awareness into student's learned features.

Soft feature mask Inspired by the Squeeze-and-Excitation Networks [15], as shown in Fig. 4 (a), we generate a latent soft feature mask for FPN features for an adaptive feature calibration based on teacher's guidance. Despite its effectiveness in improving the student's accuracy, it incurs 17% inference speed reduction with the ResNet18 backbone.



(a) Soft Feature Mask (b) Foreground Segmentation

Fig. 4. Foreground-aware feature learning approaches.

Foreground segmentation Instead of adding additional operations to the feature maps, we attach a binary segmentation branch to the FPN features of the student as shown in Fig. 4 (b), we apply a weighted pixel-wise binary cross-entropy loss [30] posing higher weight on the foreground pixels so that the foreground regions get more attention in the feature learning. The added layers (white layers in Fig. 4 (b)) will only be used in the training and will not be touched during inference, so the student's inference speed will not be affected. We prefer to this approach because of its balanced performance on accuracy and efficiency.

Loss function We introduce two additional loss terms (of which the weights are both set to 1) to the loss function of YOLACT. The first is KD loss $L_{distill}$ which measures the KL-Divergence between the classification outputs of the teacher and student on a set of selected anchor boxes S_{recip}:

$$L_{distill} = \frac{1}{|S_{recip}|} \sum_{i \in S_{recip}} KL_{div}(\sigma(\frac{\hat{y}_i^t}{T}), \sigma(\frac{\hat{y}_i^s}{T})), \tag{1}$$

where T is a temperature [13] and $\sigma(\cdot)$ denotes the softmax.

The second loss term is the binary foreground segmentation loss $L_{f\text{-}seg}$ which is a weighted pixel-wise binary cross-entropy loss [30] and the weight between foreground and background pixels is set as 2:1.

Our overall learning objective is:

$$L_{YOLACT} + \lambda_1 L_{distill} + \lambda_2 L_{f\text{-}seg} \tag{2}$$

where λ_1 and λ_2 are hyper-parameter to balance the loss terms and are both set to 1 in our experiments.

4 Experiments

Setup We conduct experiments on COCO [23] and Pascal VOC [7]. Teachers and students only differ in their backbones, and unless otherwise noted, students use ResNet-18 as their backbone. All teachers and students are first trained individually with an input size of 550×550. Then, in the model imitation training stage, teachers are frozen and students are fine-tuned with the losses defined in § 3. We perform all experiments on 4 NVIDIA Tesla V100 GPUs with a batch size of 32. Both the initial training stage and the imitation stage use the SGD optimizer following the schedule proposed in YOLACT [2] and all the training sessions finish after 54 epochs for COCO and 112 epochs for Pascal VOC. The distillation temperature T is set to 3 for COCO and 1 for Pascal VOC. The accuracy metric used are the mean average precision (mAP) at IoU = 0.5 (denoted as @0.5) and the mAP averaged for IoU $\in$ [0.5 : 0.05 : 0.95] (denoted as @[.5, .95]).

Table 3. Comparison on COCO. The first row shows the performance of the student without being taught by the teacher.

Teacher Backbone	Method	COCO		Pascal VOC	
		Mask@0.5	Mask@[0.5,.95]	Mask@0.5	Mask@[0.5,.95]
N/A	Student	40.44	23.76	68.98	43.74
ResNet-50	KD-Cls	42.96	25.29	70.61	45.90
	KD-Hint	43.15	25.68	70.08	45.78
	KD-FHint	43.67	25.94	70.08	45.78
	KD-All	43.17	25.65	70.12	45.56
ResNet-101	KD-Cls	42.49	25.26	69.07	44.67
	KD-Hint	42.52	25.04	69.39	44.69
	KD-FHint	43.11	25.37	69.70	44.84
	KD-All	42.72	25.12	69.81	44.98
ResNet-50	FSI-SFM	44.29	26.42	70.55	**46.27**
	FSI-FS	44.30	26.50	70.66	**46.27**
ResNet-101	FSI-SFM	43.97	26.23	69.62	44.65
	FSI-FS	43.77	26.11	68.89	44.30
ResNet-18	FSI-SFM	44.27	26.39	**71.04**	46.21
	FSI-FS	**44.43**	**26.53**	70.72	46.17

Compared baselines We compare our method against 4 KD baseline methods which use a large teacher that is trained on the same dataset and shares the same input/output format as the student. (1) *KD-Cls* distills exactly the same teacher knowledge as our method but uses a larger teacher model. (2) *KD-Hint [3]* distills CNN features from intermediate layers in addition to the classification output. Specifically, we add an adaptation layer (1x1 conv) to each FPN's output feature map, and the output of each adaptation layer is compared to the corresponding feature map of the teacher model by calculating the L2 distance as the hinted feature loss $L_{distill\text{-}hint}$. (3) *KD-FHint [33]* is similar to *KD-Hint*, but only distill CNN features from the foreground regions. (4) *KD-All* is built on top of *KD-Hint* via adding an additional mask prototype distillation loss $L_{distill\text{-}mask}$ (smoothed L1 loss) for the corresponding mask prototypes generated in YOLACT's protonet. The compared KD baseline methods use two different backbones for teachers, i.e., ResNet-50 and ResNet-101. Our methods are named as FSI-SFM and FSI-FS. Suffix "SFM" and "FS" represent alternative approaches to embed the foreground awareness, i.e., Soft Feature Mask (SFM) and Foreground Segmentation (FS).

Comparison with baselines We show results on COCO and Pascal VOC in Table 3. The proposed FSI method brings notable improvement: mAP has increased by 2.77 and 2.47 respectively for Mask@[0.5,.95]. For Pascal VOC, the result ResNet-18 student model almost achieves the accuracy of the ResNet-50 model (46.21 vs 46.87 referring to Table 7). FSI-SFM and FSI-FS have similar performance, and both outperforms the compared KD baselines. In addition, we

Table 4. Per-Class Result on COCO. **BT:** Accuracy before teaching. **AT:** Accuracy after teaching.

	person	bike	car	m-bile	plane	bus	train	truck	boat	t-light
BT	28.52	8.71	22.75	19.03	40.93	51.24	54.47	21.51	10.10	14.52
AT	31.56	10.64	25.57	21.97	42.95	54.23	56.37	24.65	12.15	15.82
	hydrant	s-sign	p-meter	bench	bird	cat	dog	horse	sheep	cow
BT	50.76	53.38	32.21	8.97	16.14	55.65	47.40	27.81	23.85	28.86
AT	53.72	55.67	35.39	11.32	15.53	60.57	50.14	29.91	25.40	30.85
	elephant	bear	zebra	giraffe	b-pack	umbrella	handbag	tie	suitcase	frisbee
BT	42.35	59.34	40.37	37.86	5.12	28.81	4.96	12.21	16.51	42.29
AT	46.33	63.39	41.56	39.25	6.98	32.68	6.47	9.12	19.50	47.40
	skis	snow-b	s-ball	kite	b-bat	b-glove	skate-b	surf-b	t-racket	bottle
BT	0.65	10.30	25.31	14.65	11.44	24.67	14.89	17.63	38.85	16.38
AT	0.91	12.87	26.31	16.38	13.10	29.01	19.05	20.07	41.51	19.42
	w-glass	cup	fork	knife	spoon	bowl	banana	apple	sandwich	orange
BT	15.23	23.43	2.60	1.64	1.50	24.83	9.51	8.74	25.22	18.04
AT	17.58	26.44	4.65	2.77	3.23	26.71	11.38	12.74	29.64	19.66
	broccoli	carrot	hotdog	pizza	donut	cake	chair	couch	p-plant	bed
BT	12.65	6.61	12.68	37.80	30.00	22.26	4.74	22.67	10.34	25.72
AT	15.38	10.03	14.89	38.56	35.40	26.25	6.73	26.60	13.01	27.87
	d-table	toilet	tv	laptop	mouse	remote	k-board	c-phone	m-wave	oven
BT	9.55	47.52	45.15	43.74	45.73	12.38	34.77	19.57	41.63	22.61
AT	11.34	51.78	50.09	45.52	48.37	15.60	38.33	23.37	48.03	26.25
	toaster	sink	fridge	book	clock	vase	scissor	t-bear	h-drier	t-brush
BT	14.73	22.71	36.19	1.36	40.44	19.82	8.68	28.73	2.93	4.88
AT	18.90	26.00	41.18	1.76	42.36	22.31	16.72	31.85	4.89	5.49

Table 5. Ablation study on COCO.

RECIP	SFM	FS	NegBox	Hint	Mask@0.5	Mask@[0.5,.95]
✗	✗	✓	✗	✗	43.86	26.15
✓	✗	✗	✗	✗	43.28	25.70
✓	✗	✓	✓	✗	44.23	26.35
✓	✗	✓	✗	✓	44.08	26.34
✓	✗	✓	✗	✗	44.43	26.53
✓	✓	✗	✗	✗	44.27	26.39

don't observe clear accuracy improvement by adding more loss terms to these baselines. This might be due to the fact that a large number of hyper-parameters need to be set and optimized for different datasets and teacher architectures. By contrast, our method only distills the knowledge from the teacher's classification output and thus avoids the complicated hyper-parameter tuning. Overall, we observe that a too heavy teacher (ResNet-101 vs. ResNet-50) is not necessarily better than the lightweight one (ResNet-18) as a consequence of greatly mismatched model capacity as addressed in [6]. In addition, we present the result

Table 6. Results of co-learned object detection on COCO and Pascal VOC. The first row shows the accuracy of the student.

T-Backbone	Method	COCO		Pascal VOC	
		Box@0.5	Box@[0.5,.95]	Box@0.5	Box@[0.5,.95]
ResNet-18	Student	44.61	25.01	72.23	44.88
ResNet-50	KD-Cls	46.80	26.86	72.71	46.83
	KD-Hint	47.33	27.14	72.50	46.75
	KD-FHint	47.67	27.24	72.60	46.71
	KD-All	47.34	27.16	72.87	46.69
ResNet-101	KD-Cls	46.83	26.67	72.76	44.25
	KD-Hint	46.45	25.86	72.44	44.12
	KD-FHint	47.21	27.07	72.46	44.19
	KD-All	46.57	25.96	72.74	44.40
ResNet-18	FSI-SFM	48.23	27.67	**73.07**	**47.61**
ResNet-18	FSI-FS	**48.54**	**28.01**	73.01	47.12

for each COCO class in Table 4. Among the 80 classes, 78 of them get obviously improved accuracy. The only exceptions are birds (slightly reduced accuracy) and ties (relatively small objects with a small number of validation images).

Ablation study To show the effect of the different components we have designed, we present the ablation study on COCO in Table 5. The abbreviated keywords in the table are explained as follows: *RECIP:* Reciprocal anchor box selection. If marked as ✗, we only select the anchor boxes that the teacher has correctly predicted as foreground objects and ignores those wrongly predicted by the student. *NegBox:* 300 random selected background boxes correctly predicted by teachers are added for distillation. *Hint:* Feature distillation loss [3]. *SFM:* Foreground awareness with soft feature mask. *FS:* Training with foreground segmentation branch. The first row of Table 5 reflects the effectiveness of the reciprocal anchor box selection. When the student only learns from teacher's essential knowledge but does not correct its own mistake, mAP drops by 0.57. Without learning the foreground-aware features, mAP drops by 1.15 as indicated at the second row.[4] Distilling on more anchor boxes or adding additional distillation loss actually reduces the performance.

Co-learned object detection task Object detection is a co-learned task for instance segmentation in YOLACT. We see the trend is similar to the instance segmentation in the evaluation. The experiment result and the comparison with baseline methods are presented in Table 6.

Large student models The proposed FSI method can be applied to improving large students, which is not feasible for the conventional KD. We conduct experiments on students with ResNet50 and ResNet101 as backbones where teachers

[4] Simply adding the the foreground awareness (FS) to the student without a teacher can improve mAP by 0.4.

share the same architecture. In addition to YOLACT, we also use the recently released YOLACT++ [1] which further improves the base model accuracy by incorporating deformable convolutions, optimizing anchor box scales and aspect ratios, and adding a mask re-scoring branch. We present the results in Table 7 where considerable improvements are observed in the majority of scenarios. The only exceptions are mAP values of Mask@0.5 on Pascal VOC where only a slight improvement is achieved indicating an upper bound of the YOLACT architecture. However, we still observe an increase of 2 for mAP of Mask@[0.5,.95] showing that the proposed FSI method outputs more higher-quality masks.

Larger teacher model In this experiment, we investigate whether using a larger teacher model, i.e., the model size of the teacher model is larger than the student, can further improve the model accuracy. We run experiments by replacing the small ResNet-18 backbone in the FSI teacher models with larger backbones of ResNet-50 and ResNet-101. The experiment results on MS COCO and Pascal VOC datasets are presented in Table 8. For both datasets, we observe that using a large backbone in the teacher network could not further improve the accuracy of the student models. This result can be explained as the mismatched capacity [6] that small students are unable to mimic large teachers on their classification capability. The performance even drops when ResNet-101 backbone is used due to the huge difference between the model sizes of the teacher model and the student model.

Working with model compression KD and model compression are two complimentary directions. KD is to improve an originally lightweight model but compression is to reduce the model size while trying to maintain the accuracy. Therefore, a common practise is to first apply KD to improve the model's accuracy and then apply the compression to reduce the model size. We present the result of applying quantization [16] to a model distilled by our FSI method and an original base model in Table 9. Both models have similar accuracy drop after compression but our distilled model still performs better.

Training efficiency Because a small teacher model is used, FSI has advantages in greatly reduced training overhead compared to traditional KD with a large teacher model.

Training speed The proposed method also substantially reduces the training time in 3 aspects (1) *Simpler task:* Training the teacher model is much faster since the transformed input images make the training task simpler and we find that the model converges with around 40% less epochs; (2) *Smaller teacher model:* The savings come from the reduced model forward time and further speedup could come from using a larger batch size. (3) *Easier hyper-parameter tuning:* Determining the optimal hyper-parameter setting (i.e., weights among different loss terms) for the compared baseline methods requires significant amount of training/engineering efforts. Compared to our method, existing solutions have more loss terms and also need to find the optimal model size for the teacher.

Table 7. Large students'results.

Dataset	Model	Training State	Mask@0.5	Mask@[0.5,.95]
COCO	YOLACT	Before Teaching	45.92	27.97
	ResNet50	After Teaching	49.71	30.43
	YOLACT	Before Teaching	48.01	29.73
	ResNet101	After Teaching	52.06	31.95
	YOLACT++	Before Teaching	52.71	33.69
	ResNet50	After Teaching	54.49	35.03
	YOLACT++	Before Teaching	53.17	34.46
	ResNet101	After Teaching	54.84	35.55
Pascal	YOLACT	Before Teaching	72.34	46.87
	ResNet50	After Teaching	72.59	48.93
	YOLACT	Before Teaching	72.72	48.26
	ResNet101	After Teaching	73.11	50.21

Table 8. Large teachers on COCO and Pascal VOC.

T-Backbone	Method	MS COCO		Pascal VOC	
		Mask@0.5	Mask@[0.5,.95]	Mask@0.5	Mask@[0.5,.95]
ResNet-18	Student	40.44	23.76	68.98	43.74
ResNet-50	FSI-SFM	44.29	26.42	70.55	46.27
	FSI-FS	44.30	26.50	70.66	46.27
ResNet-101	FSI-SFM	43.97	26.23	69.62	44.65
	FSI-FS	43.77	26.11	68.89	44.30
ResNet-18	FSI-SFM	44.27	26.39	**71.04**	**46.21**
ResNet-18	FSI-FS	**44.43**	**26.53**	70.72	46.17

Table 9. Compressing model before and after distillation.

Model	mAP before compression	mAP after compression (4×)
ResNet-18 (Original)	23.76	22.89
ResNet-18 (Distilled)	26.53	25.62

5 Conclusion

In this paper, we introduce FSI, a foreground-specialized teacher model imitation framework for improving the accuracy of instance segmentation methods. Given that the teacher takes different input from the student, FSI incorporates two novel modules to have the student learn from the teacher better: (1) identifying the most essential teacher knowledge and (2) embedding the foreground-awareness into student's feature learning. We demonstrate the effectiveness of FSI on COCO and Pascal VOC by comparing them to KD baselines. The methodology presented in this work could have the potential to be applied to other multi-task learning problems.

References

1. Bolya, D., Zhou, C., Xiao, F., Lee, Y.J.: YOLACT++: better real-time instance segmentation. CoRR **abs/1912.06218** (2019)
2. Bolya, D., Zhou, C., Xiao, F., Lee, Y.J.: YOLACT: real-time instance segmentation. In: ICCV (2019)
3. Chen, G., Choi, W., Yu, X., Han, T.X., Chandraker, M.: Learning efficient object detection models with knowledge distillation. In: NIPS (2017)
4. Chen, L., Papandreou, G., Schroff, F., Adam, H.: Rethinking atrous convolution for semantic image segmentation. CoRR **abs/1706.05587** (2017)
5. Chen, S., Zhao, Q.: Attention-based autism spectrum disorder screening with privileged modality. In: ICCV (2019)
6. Cho, J.H., Hariharan, B.: On the efficacy of knowledge distillation. In: ICCV (2019)
7. Everingham, M., Eslami, S.M.A., Gool, L.V., Williams, C.K.I., Winn, J.M., Zisserman, A.: The pascal visual object classes challenge: A retrospective. IJCV (2015)
8. Garcia, N.C., Morerio, P., Murino, V.: Modality distillation with multiple stream networks for action recognition. In: ECCV (2018)
9. Han, S., Kang, J., Mao, H., Hu, Y., Li, X., Li, Y., Xie, D., Luo, H., Yao, S., Wang, Y., Yang, H., Dally, W.B.J.: ESE: efficient speech recognition engine with sparse LSTM on FPGA. In: FPGA (2017)
10. Han, S., Mao, H., Dally, W.J.: Deep compression: Compressing deep neural network with pruning, trained quantization and huffman coding. In: ICLR (2016)
11. He, K., Gkioxari, G., Dollár, P., Girshick, R.B.: Mask R-CNN. In: ICCV (2017)
12. He, Y., Liu, P., Wang, Z., Hu, Z., Yang, Y.: Filter pruning via geometric median for deep convolutional neural networks acceleration. In: CVPR (2019)
13. Hinton, G.E., Vinyals, O., Dean, J.: Distilling the knowledge in a neural network. CoRR **abs/1503.02531** (2015)
14. Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., Adam, H.: Mobilenets: Efficient convolutional neural networks for mobile vision applications. CoRR **abs/1704.04861** (2017)
15. Hu, J., Shen, L., Sun, G.: Squeeze-and-excitation networks. In: CVPR (2018)
16. Jacob, B., Kligys, S., Chen, B., Zhu, M., Tang, M., Howard, A.G., Adam, H., Kalenichenko, D.: Quantization and training of neural networks for efficient integer-arithmetic-only inference. In: CVPR (2018)
17. Kim, Y., Park, E., Yoo, S., Choi, T., Yang, L., Shin, D.: Compression of deep convolutional neural networks for fast and low power mobile applications. In: ICLR (2016)
18. Kim, Y., Rush, A.M.: Sequence-level knowledge distillation. In: EMNLP (2016)
19. Lambert, J., Sener, O., Savarese, S.: Deep learning under privileged information using heteroscedastic dropout. In: CVPR (2018)
20. Lee, K., Ros, G., Li, J., Gaidon, A.: SPIGAN: privileged adversarial learning from simulation. In: ICLR (2019)
21. Lin, T., Dollár, P., Girshick, R.B., He, K., Hariharan, B., Belongie, S.J.: Feature pyramid networks for object detection. In: CVPR (2017)
22. Lin, T., Goyal, P., Girshick, R.B., He, K., Dollár, P.: Focal loss for dense object detection. In: ICCV (2017)
23. Lin, T., Maire, M., Belongie, S.J., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft COCO: common objects in context. In: ECCV (2014)
24. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S.E., Fu, C., Berg, A.C.: SSD: single shot multibox detector. In: ECCV (2016)

25. Liu, Y., Chen, K., Liu, C., Qin, Z., Luo, Z., Wang, J.: Structured knowledge distillation for semantic segmentation. In: CVPR (2019)
26. Luo, Z., Hsieh, J., Jiang, L., Niebles, J.C., Fei-Fei, L.: Graph distillation for action detection with privileged modalities. In: ECCV (2018)
27. Rastegari, M., Ordonez, V., Redmon, J., Farhadi, A.: Xnor-net: Imagenet classification using binary convolutional neural networks. In: ECCV (2016)
28. Redmon, J., Divvala, S.K., Girshick, R.B., Farhadi, A.: You only look once: Unified, real-time object detection. In: CVPR (2016)
29. Ren, S., He, K., Girshick, R.B., Sun, J.: Faster R-CNN: towards real-time object detection with region proposal networks. In: NIPS (2015)
30. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: MICCAI (2015)
31. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: ICLR (2015)
32. Vapnik, V., Izmailov, R.: Learning using privileged information: similarity control and knowledge transfer. JMLR (2015)
33. Wang, T., Yuan, L., Zhang, X., Feng, J.: Distilling object detectors with fine-grained feature imitation. In: CVPR (2019)
34. Xie, E., Sun, P., Song, X., Wang, W., Liu, X., Liang, D., Shen, C., Luo, P.: Polarmask: Single shot instance segmentation with polar representation. In: CVPR (2020)
35. Zhang, X., Zhou, X., Lin, M., Sun, J.: Shufflenet: An extremely efficient convolutional neural network for mobile devices. In: CVPR (2018)