Data Distillation: Towards Omni-Supervised Learning

Ilija Radosavovic Piotr Dollár Ross Girshick Georgia Gkioxari Kaiming He
Facebook AI Research (FAIR)

Abstract

We investigate omni-supervised learning*, a special regime of semi-supervised learning in which the learner exploits all available labeled data plus internet-scale sources of unlabeled data. Omni-supervised learning is lower-bounded by performance on existing labeled datasets, offering the potential to surpass state-of-the-art fully supervised methods. To exploit the omni-supervised setting, we propose* data distillation*, a method that ensembles predictions from multiple transformations of unlabeled data, using a single model, to automatically generate new training annotations. We argue that visual recognition models have recently become accurate enough that it is now possible to apply classic ideas about self-training to challenging real-world data. Our experimental results show that in the cases of human keypoint detection and general object detection, state-of-the-art models trained with data distillation surpass the performance of using labeled data from the COCO dataset alone.*

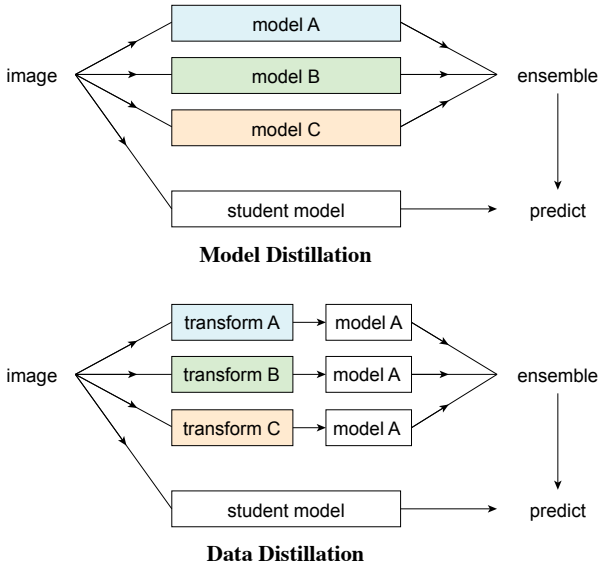

Figure 1. **Model Distillation [18] *vs*. Data Distillation.** In data distillation, ensembled predictions from a single model applied to multiple transformations of an unlabeled image are used as automatically annotated data for training a student model.

1. Introduction

This paper investigates *omni-supervised learning*, a paradigm in which the learner exploits as much well-annotated data as possible (*e.g.*, ImageNet [6], COCO [24]) and is also provided with potentially unlimited unlabeled data (*e.g.*, from internet-scale sources). It is a special regime of semi-supervised learning. However, most research on semi-supervised learning has *simulated* labeled/unlabeled data by splitting a fully annotated dataset and is therefore likely to be *upper-bounded* by fully supervised learning with all annotations. On the contrary, omni-supervised learning is *lower-bounded* by the accuracy of training on all annotated data, and its success can be evaluated by how much it surpasses the fully supervised baseline.

To tackle omni-supervised learning, we propose to perform knowledge distillation *from data*, inspired by [3, 18] which performed knowledge distillation *from models*. Our idea is to generate annotations on unlabeled data using a model trained on large amounts of labeled data, and then retrain the model using the extra generated annotations. However, training a model on its own predictions often provides no meaningful information. We address this problem by ensembling the results of a single model run on different transformations (*e.g.*, flipping and scaling) of an unlabeled image. Such transformations are widely known to improve single-model accuracy [20] when applied at test time, indicating that they can provide nontrivial knowledge that is not captured by a single prediction. In other words, in comparison with [18], which distills knowledge from the predictions of multiple models, we distill the knowledge of a single model run on multiple transformed copies of unlabeled data (see Figure 1).

Data distillation is a simple and natural approach based on "self-training" (*i.e.*, making predictions on unlabeled data and using them to update the model), related to which there have been continuous efforts [36, 48, 43, 33, 22, 46, 5, 21] dating back to the 1960s, if not earlier. However, our simple data distillation approach can become realistic largely thanks to the rapid improvement of fully-supervised models [20, 39, 41, 16, 12, 11, 30, 28, 25, 15] in the past few years. In particular, we are now equipped with accurate models that may make fewer errors than correct predictions. This allows us to trust their predictions on unseen

data and reduces the requirement for developing data cleaning heuristics. As a result, data distillation does not require one to change the underlying recognition model (*e.g.*, no modification on the loss definitions), and is a scalable solution for processing large-scale unlabeled data sources.

To test data distillation for omni-supervised learning, we evaluate it on the human keypoint detection task of the COCO dataset [24]. We demonstrate promising signals on this real-world, large-scale application. Specifically, we train a Mask R-CNN model [15] using data distillation applied on the original labeled COCO set and another large unlabeled set (*e.g.*, static frames from Sports-1M [19]). Using the distilled annotations on the unlabeled set, we have observed improvement of accuracy on the held-out validation set: *e.g.*, we show an up to 2 points AP improvement over the strong Mask R-CNN baseline. As a reference, this improvement compares favorably to the ~3 points AP improvement gained from training on a similar amount of extra *manually labeled* data in [27] (using private annotations). We further explore our method on COCO object detection and show gains over fully-supervised baselines.

2. Related Work

Ensembling [14] multiple models has been a successful method for improving accuracy. Model compression [3] is proposed to improve test-time efficiency of ensembling by compressing an ensemble of models into a single student model. This method is extended in knowledge distillation [18], which uses soft predictions as the student's target.

The idea of distillation has been adopted in various scenarios. FitNet [32] adopts a shallow and wide teacher models to train a deep and thin student model. Cross modal distillation [13] is proposed to address the problem of limited labels in a certain modality. In [26] distillation is unified with privileged information [44]. To avoid explicitly training multiple models, Laine and Aila [21] exploit multiple checkpoints during training to generate the ensemble predictions. Following the success of these existing works, our approach distills knowledge from a lightweight ensemble formed by multiple data transformations.

There is a great volume of work on semi-supervised learning, and comprehensive surveys can be found in [49, 4, 50]. Among semi-supervised methods, our method is most related to self-training, a strategy in which a model's predictions on unlabeled data are used to train itself [36, 48, 43, 33, 22, 46, 5, 21]. Closely related to our work on keypoint/object detection, Rosenberg *et al*. [33] demonstrate that self-training can be used for training object detectors. Compared to prior efforts, our method is substantially simpler. Once the predicted annotations are generated, our method leverages them as if they were true labels; it does not require any modifications to the optimization problem or model structure.

Multiple views or perturbations of the data can provide useful signal for semi-supervised learning. In the co-training framework [2], different views of the data are used to learn two distinct classifiers that are then used to train one another over unlabeled data. Reed *et al*. [29] use a reconstruction consistency term for training classification and detection models. Bachman *et al*. [1] employ the pseudo-ensemble regularization term to train models robust on input perturbations. Sajjadi *et al*. [35] enforce consistency between outputs computed for different transformations of input examples. Simon *et al*. [38] utilize multi-view geometry to generate hand keypoint labels from multiple cameras and retrain the detector. In an auto-encoder scenario, Hinton *et al*. [17] propose to use multiple "capsules" to model multiple geometric transformations. Our method is also based on multiple geometric transformations, but it does not require to modify network structures or impose consistency by adding any extra loss terms.

Regarding the large-scale regime, Fergus *et al*. [9] investigate semi-supervised learning on 80 millions tiny images. A Never Ending Image Learner (NEIL) [5] employs self-training to perform semi-supervised learning from web-scale image data. These methods were developed before the recent renaissance of deep learning. In contrast, our method is evaluated with strong deep neural network baselines, and can be applied to structured prediction problems beyond image-level classification (*e.g.*, keypoints and boxes).

3. Data Distillation

We propose *data distillation*, a general method for omni-supervised learning that distills knowledge from unlabeled data without the requirement of training a large set of models. Data distillation involves four steps: (1) training a model on manually labeled data (just as in normal supervised learning); (2) applying the trained model to multiple transformations of unlabeled data; (3) converting the predictions on the unlabeled data into labels by ensembling the multiple predictions; and (4) retraining the model on the union of the manually labeled data and automatically labeled data. We describe steps 2-4 in more detail below.

Multi-transform inference. A common strategy for boosting the accuracy of a visual recognition model is to apply the same model to multiple transformations of the input and then to aggregate the results. Examples of this strategy include using multiple crops of an input image (*e.g.*, [20, 42]) or applying a detection model to multiple image scales and merging the detections (*e.g.*, [45, 8, 7, 37]). We refer to the general application of inference to multiple transformations of a data point with a single model as *multi-transform inference*. In data distillation, we apply multi-transform inference to a potentially massive set of unlabeled data.

Figure 2. **Ensembling keypoint predictions from multiple data transformations can yield a single superior (automatic) annotation.** For visualization purposes all images and keypoint predictions are transformed back to their original coordinate frame.

Generating labels on unlabeled data. By aggregating the results of multi-transform inference, it is often possible to obtain a single prediction that is superior to any of the model's predictions under a single transform (*e.g.*, see Figure 2). Our observation is that the aggregated prediction generates *new knowledge* and in principle the model can use this information to learn from itself by generating labels.

Given an unlabeled image and a set of predictions from multi-transform inference, there are multiple ways one could automatically generate labels on the image. For example, in the case of a classification problem the image could be labeled with the average of the class probabilities [18]. This strategy, however, has two problems. First, it generates a "soft" label (a probability vector, not a categorical label) that may not be straightforward to use when retraining the model. The training loss, for example, may need to be altered such that its compatible with soft labels. Second, for problems with structured output spaces, like object detection or human pose estimation, it does not make sense to average the output as care must be taken to respect the structure of the output space.

Given these considerations, we simply ensemble (or aggregate) the predictions from multi-transform inference in a way that generates "hard" labels of the same structure and type of those found in the manually annotated data. Generating hard labels typically requires a small amount of task-specific logic that addresses the structure of the problem (*e.g.*, merging multiple sets of boxes by non-maximum suppression). Once such labels are generated, they can be used to retrain the model in a simple plug-and-play fashion, as if they were authentic ground-truth labels.

Finally, we note that while this procedure requires running inference multiple times, it is actually *efficient* because it is generally substantially less expensive than training multiple models from scratch, as is required by model distillation [3, 18].

Knowledge distillation. The new knowledge generated from unlabeled data can be used to improve the model. To do this, a student model (which can be the same as the original model or different) is trained on the *union* set of the original supervised data and the unlabeled data with automatically generated labels.

Training on the union set is straightforward and requires no change to the loss function. However, we do take two factors into considerations. First, we ensure that each training minibatch contains a mixture of manually labeled data and automatically labeled data. This ensures that every minibatch has a certain percentage of ground-truth labels, which results in better gradient estimates. Second, since more data is available, the training schedule must be lengthened to take full advantage of it. We discuss these issues in more detail in the context of the experiments.

4. Data Distillation for Keypoint Detection

This section describes an instantiation of data distillation for the application of multi-person keypoint detection.

Mask R-CNN. Our teacher and student models are the Mask R-CNN [15] keypoint detection variant. Mask R-CNN is a two-stage model. The first stage is a Region Proposal Network (RPN) [30]. The second stage consists of three heads for bounding box classification, regression, and keypoint prediction on each Region of Interest (RoI). The keypoint head outputs a heatmap that is trained to predict a one-hot mask for each keypoint type. We use ResNet [16] and ResNeXt [47] with Feature Pyramid Networks (FPN) [23] as backbones for Mask R-CNN. All implementations follow [15], unless specified.

Data transformations. This paper opts for *geometric* transformations for multi-transform inference, though other transformations such as color jittering [20] are possible. The only requirement is that it must be possible to ensemble the resulting predictions. For geometric transformations, if the prediction is a geometric quantity (*e.g.*, coordinates of a keypoint), then the inverse transformation must be applied to each prediction before they are merged.

We use two popular transformations: scaling and horizontal flipping. We resize the unlabeled image to a predefined set of scales (denoted by the shorter side of an image): [400, 1200] pixels with a stepsize of 100, which was selected by measuring the keypoint AP for the teacher model when applying these transformations on the validation set. The selected transformations can improve the model by a good margin, *e.g.* for ResNet-50 from 65.1 to 67.8 AP, which is then used as the teacher. Note that unless stated, we do *not* apply these transformation at test time for all baseline/distilled models.

Figure 3. **Random examples of annotations generated on static Sports-1M [19] frames using a ResNet-50 teacher.** The generated annotations have reasonably high quality, though as expected there are mistakes like inverted keypoints (top right).

Figure 3 shows some examples of the generated annotations on Sport-1M. They have reasonably high quality.

Ensembling. One could ensemble the multi-transform inference results from each stage and each head of Mask R-CNN. In our experiments, however, for simplicity we only apply multi-transform inference to the keypoint head; the outputs from the other stage (*i.e.*, RPN) and heads (*i.e.*, bounding box classification and regression) are from a single-scale without any transformations.

Thanks to the above simplification, it is easy for us to have a consistent set of detection boxes serving as the RoIs for all transformations (scales/flipping). On a single RoI, we extract the keypoint heatmaps from all transformations, and although they are from different geometric transformations, these heatmaps are with reference to the local coordinate system of the same RoI. So we can directly average the output (probability) of these heatmaps for ensembling. We take the argmax position in this ensembling result and generate the predicted keypoint location.

Selecting predictions. We expect the predicted boxes and keypoints to be reliable enough for generating good training labels. Nevertheless, the predictions will contain *false positives* that we hope to identify and discard. We use the predicted detection score as a proxy for prediction quality and generate annotations only from the predictions that are above a certain score threshold. In practice, we found that a score threshold works well if it makes "the average number of annotated instances per unlabeled image" roughly equal to "the average number of instances per labeled image". Although this heuristic assumes that the unlabeled and labeled images follow similar distributions, we found that it is robust and works well even in cases where the assumption does not hold.

As a dual consideration to false positives above, there may be *false negatives* (*i.e.*, missing detections) in the extra data, and the annotations generated should not necessarily be viewed as *complete* (*i.e.*, absence of an annotation does not imply true background). However, in our practice we have tried either to sample or not sample background regions from the extra data for training detectors, and have observed no difference in accuracy. For simplicity, in all experiments we view the generated data as complete, so the extra data are simply treated as if all annotations are correct.

Generating keypoint annotations. Each of the selected predictions consists of K individual keypoints (*e.g.*, left ear, nose, *etc.*). Since many of the object views do not show all of the keypoint types, the predicted keypoints are likely to contain false positives as well. As above, we choose a threshold such that the average numbers of keypoints are approximately equal in the supervised and generated sets.

Retraining. We train a student model on the union set of the original supervised images and the images with automatically generated annotations. To maintain supervision quality at the minibatch level, we use a fixed sampling ratio for the two kinds of data. Specifically, we randomly sample images for each minibatch such that the expected ratio of original images to generated labeled images is 6:4, unless stated otherwise.

We adopt the learning rate schedule similar to [15] and increase the total number of iterations to account for extra images. The learning rate starts from 0.02 and is divided by 10 after 70% and 90% of the total number of iterations. The impact of the total number of iterations will be discussed in the next section in context of Table 2.

We use a student model with the same architecture as the teacher. The student can either be fine-tuned starting from the teacher model or retrained from the initial weights (*i.e.*, those pre-trained on ImageNet [34]). We found that *retraining* consistently results in better performance, suggesting that the teacher model could have been in a poor local optimum. We opt for retraining in all experiments.

5. Experiments on Keypoint Detection

We evaluate data distillation on the keypoint detection task of the COCO dataset [24]. We report keypoint Average Precision following the COCO definitions, including AP (COCO's default, averaged over different IoU thresh-

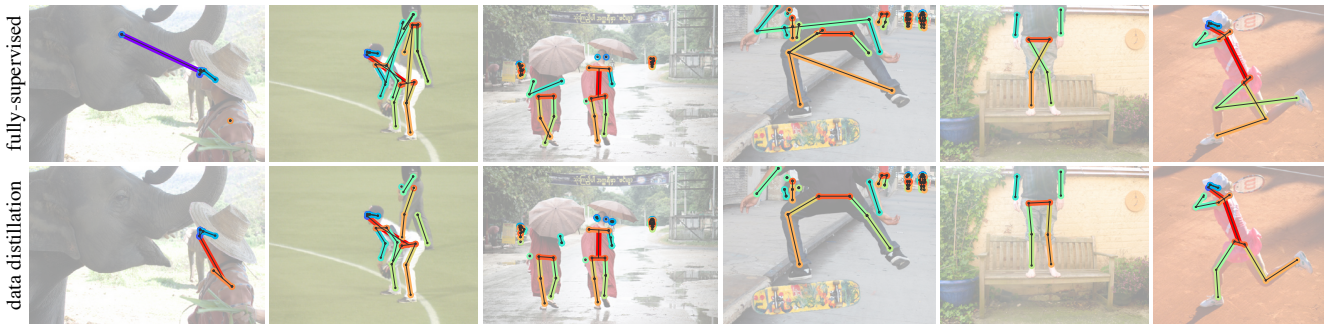

Figure 4. **Selected results of fully-supervised learning in the original `co-115` set (top) *vs*. data distillation in the `co-115` plus `s1m-180` sets (bottom).** The results are on the held-out data from COCO `test-dev`.

olds), AP_{50}, AP_{75}, AP_M (medium), and AP_L (large). In all experiments we report results on the 2017 validation set that contains 5k images (called `val2017`, formerly known as `minival`).

5.1. Data Splits

Our experiments involve several splits of data:

COCO labeled images. These are the original labeled COCO images that contain ground-truth person and keypoint annotations. In this paper, we refer to the 80k training images as **`co-80`**, a 35k subset of the 2014 validation images as **`co-35`**, and their union as **`co-115`** (in the 2017 version of COCO, `co-115` is the `train2017` set). We do not use the original train/val nomenclature because their roles may change in different experiments.

COCO unlabeled images. The 2017 version of COCO provides a collection of 120k unlabeled images, which we call **`un-120`**. These images are expected to have a *similar distribution* as the labeled COCO images.

Sports-1M static frames. We will show that our method can be robust to a *dissimilar distribution* of unlabeled data. We collect these images by using *static frames* from the Sports-1M [19] video dataset. We randomly sample 180k videos from this dataset. Then we randomly sample 1 frame from each video, noting that we do *not* exploit any temporal information even if it is possible. This strategy gives us 180k static images. We call this set **`s1m-180`**. We do not use any available labels from this static image set.

5.2. Main Results

We investigate data distillation in three cases:
(i) Small-scale data as a sanity check: we use `co-35` as the labeled data and treat `co-80` as unlabeled.
(ii) Large-scale data with similar distribution: we use `co-115` as the labeled data and `un-120` as unlabeled.
(iii) Large-scale data with dissimilar distribution: we use `co-115` as the labeled data and `s1m-180` as unlabeled.
The results are in Table 1, discussed as follows:

Small-scale data. As a sanity-check, we evaluate our approach in the classic semi-supervised setting by simulating labeled and unlabeled splits from all labeled images.

In Table 1a, we show results of data distillation performed on `co-35` as the labeled data and `co-80` treated as unlabeled data. As a comparison, we report supervised learning results using either `co-35` or `co-115`. This comparison shows that data distillation is a successful semi-supervised learning method: it surpasses the `co-35`-only counterpart by 5.3 points of AP by using unlabeled data (60.2 *vs*. 54.9). On the other hand, as expected, the semi-supervised learning result is lower than fully-supervised learning on `co-115` (60.2 *vs*. 65.1).

This phenomenon on small-scale data has been widely observed for many semi-supervised learning methods and datasets: if labels *were* available for all training data, then the accuracy of semi-supervised learning would be *upper-bounded* by using all labels. In addition, as the *simulated* splits are often at smaller scales, there is a relatively large gap for the semi-supervised method to improve in (*e.g*., from 54.9 to 65.1).

We argue that omni-supervised learning is a real-world scenario unlike the above simulated semi-supervised setting. Even though one could label many images, there are always more unlabeled data available (*e.g*., at internet-scale). We can thus pursue an accuracy that is *lower-bounded*. In addition, when trained with a larger dataset, the supervised baseline would be much *higher* (*e.g*., 65.1), leaving *less room* for models to gain from the unlabeled data.

Therefore, we argue that the large-scale, high-accuracy regime is more challenging and of more interest in practice. We investigate it in the following experiments.

Large-scale, similar-distribution data. Table 1b shows the scenario of a real-world omni-supervised learning application: we have a large-scale source of 120k COCO (`un-120`) images on hand, but *we do not have labels for them*. Can we improve over our best baseline results using these unlabeled data?

Table 1b shows the data distillation results on `co-115` plus `un-120`, comparing with the fully-supervised coun-

labeled	unlabeled	AP	AP_{50}	AP_{75}	AP_M	AP_L
co-35		54.9	80.5	59.0	50.1	62.8
co-35	co-80	**60.2**	**83.8**	**65.4**	**55.2**	**68.4**
co-115		65.1	86.6	70.9	59.9	73.6

(a) **Small-scale data.** Data distillation is performed on co-35 with labels and co-80 without labels, *vs*. fully-supervised learning performed on co-35 and co-115 respectively. The backbone is ResNet-50.

backbone	DD	AP	AP_{50}	AP_{75}	AP_M	AP_L
ResNet-50		65.1	86.6	70.9	59.9	73.6
ResNet-50	✓	**67.1**	**87.9**	**73.4**	**62.2**	**75.1**
ResNet-101		66.1	87.7	71.7	60.5	75.0
ResNet-101	✓	**67.8**	**88.2**	**73.8**	**62.8**	**76.0**
ResNeXt-101-32×4		66.8	87.5	73.0	61.6	75.2
ResNeXt-101-32×4	✓	**68.7**	**88.9**	**75.1**	**63.9**	**76.7**
ResNeXt-101-64×4		67.3	88.0	73.3	62.2	75.6
ResNeXt-101-64×4	✓	**69.1**	**88.9**	**75.3**	**64.1**	**77.1**

(b) **Large-scale, similar-distribution data.** Data distillation (DD) is performed on co-115 with labels and un-120 without labels, comparing with the supervised counterparts trained on co-115.

backbone	DD	AP	AP_{50}	AP_{75}	AP_M	AP_L
ResNet-50		65.1	86.6	70.9	59.9	73.6
ResNet-50	✓	**66.6**	**87.3**	**72.6**	**61.6**	**75.0**
ResNet-101		66.1	87.7	71.7	60.5	75.0
ResNet-101	✓	**67.5**	**87.9**	**73.9**	**62.4**	**75.9**
ResNeXt-101-32×4		66.8	87.5	73.0	61.6	75.2
ResNeXt-101-32×4	✓	**68.0**	**88.1**	**74.2**	**63.1**	**76.2**
ResNeXt-101-64×4		67.3	88.0	73.3	62.2	75.6
ResNeXt-101-64×4	✓	**68.5**	**88.8**	**74.9**	**63.7**	**76.5**

(c) **Large-scale, dissimilar-distribution data.** Data distillation (DD) is performed on co-115 with labels and s1m-180 without labels, comparing with the supervised counterparts trained on co-115.

Table 1. Data distillation for COCO keypoint detection. Keypoint AP is reported on COCO val2017.

terpart on co-115, the largest available annotated set on hand. Our method is able to improve over the strong baselines by 1.7 to 2.0 points AP. Our improvement is observed regardless of the depth/capacity of the backbone models, including ResNet-50/101 and ResNeXt-101.

We argue that these are non-trivial results. Because the baselines are very high due to using large amounts of supervised data (115k images in co-115), they might leave less room for further improvement, in contrast to the simulated semi-supervised setting. Actually, in recent work [27] that exploited an *extra* 1.5× *fully-annotated* human keypoint skeletons (contributed by in-house annotators), the improvement is ~3 points AP over their baseline. Given this context, our increase of ~2 points AP, contributed by a similar amount of extra *unlabeled* data, is very promising.

Large-scale, dissimilar-distribution data. Even though COCO data are images "in the wild", the co-115 and un-120 sets are subject to similar data distributions. As one further step toward omni-supervision in real cases, we investigate a scenario where the unlabeled images are from a *different distribution*.

	#iter	AP	AP_{50}	AP_{75}	AP_M	AP_L
fully-supervised	90k	64.2	86.4	69.2	59.1	72.6
	130k	**65.1**	**86.6**	**70.9**	**59.9**	**73.6**
	270k	64.7	**86.6**	70.4	59.7	73.0
data distillation	90k	63.6	85.9	69.2	58.8	71.7
	180k	65.8	87.3	71.6	60.8	74.2
	270k	66.5	**88.0**	72.2	61.5	74.6
	360k	**66.6**	87.3	**72.6**	**61.6**	**75.0**

Table 2. Ablation on **numbers of training iterations.** The models are trained on co-115 (and plus s1m-180 for data distillation). The backbone is ResNet-50. In all case, the learning rate is reduced by 10 at 70% and 90% of the total number of iterations.

Table 1c shows data distillation results on co-115 plus s1m-180. Comparing with the supervised baselines trained on co-115, our method shows consistent improvement with different backbones, achieving 1.2 to 1.5 points of AP increase. Moreover, the improvements in this case are reasonably close to those in Table 1b, even though the data distribution in Sport-1M is different. This experiment shows that our method, in the application of keypoint detection, is robust to the misaligned distribution of data. This is a promising signal for real-world omni-supervised learning.

Figure 4 shows some examples of the fully-supervised results trained in co-115 and the data distillation results trained in co-115 plus s1m-180.

5.3. Ablation Experiments

In addition to the above main results, we conduct several ablation experiments as analyzed in the following:

Number of iterations. It is necessary to train for more iterations when given more (labeled or distilled) data. To show that our method does ***not*** simply take advantage of longer training, we conduct a careful ablation experiment on the number of iterations in Table 2.

For the fully-supervised baseline, we investigated a total number of iterations of 90k (as done in [15]), 130k (~1.5× longer), and 270k (3× longer). Table 2 (top) shows that an appropriately long training indeed leads to better results, and the original schedule of 90k in [15] is suboptimal. However, without increasing the dataset size, training longer gives diminishing return and becomes *prone to overfitting*. The optimal number of 130k iterations is chosen and used in Tables 1 for the fully-supervised baselines.

In contrast, our data distillation method continuously improves when the number of iterations is increased from 90k to 360k as shown in Table 2 (bottom). With a short training of 90k, our method is inferior to its fully-supervised counterpart (63.6 *vs*. 64.2), which is understandable: the generated labels in the extra data have lower quality than the ground-truth labels, and the model may not benefit from them unless ground-truth labels have been sufficiently exploited. On the other hand, our method starts to show a healthy gain with sufficient training and surpasses its

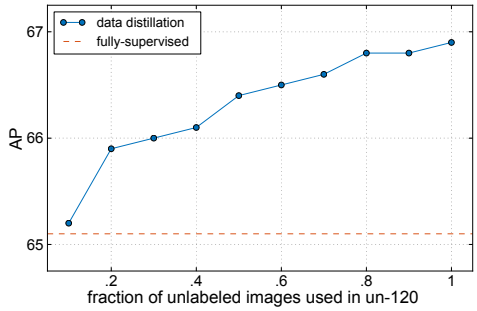

Figure 5. Data distillation applied to `coco-115` with labels and **different fractions of `un-120` images without labels**, comparing with the `coco-115` fully-supervised baseline, using ResNet-50.

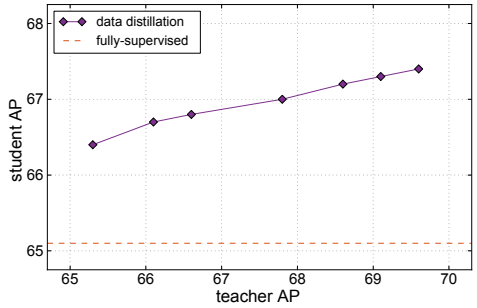

Figure 6. **Impact of teacher quality on data distillation.** Student ResNet-50 model is trained using data distillation on `coco-115` plus `un-120` and compared to its fully-supervised counterpart trained on `coco-115`. Each data point is from a different teacher whose AP can vary because of different backbones, number of supervised training iterations, and transform settings (*e.g.*, scales used).

fully-supervised counterpart. Actually, our method's performance has not saturated and is likely to improve when using more iterations. To have manageable experiments, for all other data distillation results in the paper, our method uses 360k iterations.

Amount of unlabeled data. To better understand the importance of the amount of unlabeled data, in Figure 5 we investigate using a subset of the `un-120` unlabeled data for data distillation (the labeled data is `coco-115`).

To have a simpler unified rule for handling the various sizes of the unlabeled set, for this ablation, we adopt a mini-batching and iteration strategy different from the above sections. Given a fraction ρ of `un-120` images used, we sample each minibatch with on average 1:ρ examples from the labeled and unlabeled data. The iteration number is adaptively set as 1+ρ times of that of the supervised baseline (130k in this figure). As such, the total number of sampled images from the labeled set is roughly the same regardless of the fraction ρ. We note that this strategy is *suboptimal* comparing with the setting in other tables, but it is a simplified setting that can apply to all fractions investigated.

Figure 5 shows that for all fractions of unlabeled data, our method is able to improve over the supervised baseline.

backbone	*test aug.?*	AP	AP_{50}	AP_{75}	AP_M	AP_L
ResNet-50		67.1	87.9	73.4	62.2	75.1
ResNet-50	✓	**68.9**	**88.8**	**75.8**	**64.4**	**76.4**
ResNet-101		67.8	88.2	73.8	62.8	76.0
ResNet-101	✓	**69.8**	**89.1**	**76.0**	**65.3**	**77.5**
ResNeXt-101-32×4		68.7	88.9	75.1	63.9	76.7
ResNeXt-101-32×4	✓	**70.6**	**89.3**	**77.2**	**65.7**	**78.4**
ResNeXt-101-64×4		69.1	88.9	75.3	64.1	77.1
ResNeXt-101-64×4	✓	**70.4**	**89.3**	**76.8**	**65.8**	**78.1**

Table 3. Ablation on **test-time augmentation.** A data distillation model is trained on `coco-115` + `un-120`, tested without and with test-time augmentations.

Actually, as can be expected, the supervised baseline becomes a *lower-bound* of accuracy in omni-supervised learning: the extra *unlabeled* data, when exploited appropriately such as in data distillation, should always provide extra information. Moreover, Figure 5 shows that there is a general trend of better results when using more unlabeled data. A similar trend, in the context of *fully-annotated* data, has been observed recently in [40]. However, our trend is observed in *unlabeled* data and can be more encouraging for the future study in computer vision.

Impact of teacher quality. To understand the impact of the teacher quality on data distillation, we produce different teacher models with different AP (see Figure 6 caption). Then we train the same student model on each teacher. Figure 6 shows the student AP *vs*. the teacher AP.

As expected, all student models trained by data distillation surpass the fully-supervised baseline. In addition, a higher-quality teacher in general results in a better student. This demonstrates a nice property of the data distillation method: one could expect a bigger improvement if a better teacher will be developed.

Test-time augmentations. Our data distillation method exploits multi-transform inference to generate labels. Multi-transform inference can also be applied at test-time to further improve results, a strategy typically called test-time augmentation. Table 3 shows the results of applying test-time augmentations on a data distillation model. The augmentations are the same as those used to generate distillation labels. It shows that test-time augmentations can still improve the results over our data distillation model.

Interestingly, the student model's 68.9 AP (ResNet-50, in Table 3) is higher than its corresponding (test-time augmented) teacher's 67.8 AP. We believe that this is *a signal of our approach being able to learn new knowledge from the extra unlabeled data*, instead of simply learning to be robust to the transforms. Even though we use multiple data-agnostic transforms, the distilled labels are *data-dependent* and may convey knowledge from the extra data.

This result also suggests that performing data distillation in an iterative fashion may improve the results further. We leave this direction for future work.

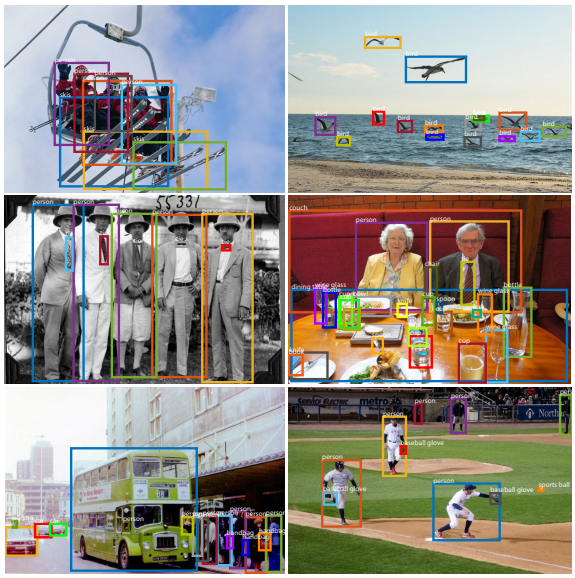

Figure 7. **Object detection annotations generated on `un-120`.**

6. Experiments on Object Detection

We investigate the generality of our approach by applying it to another task with minimal modification. We perform data distillation for object detection on the COCO dataset [24]. Here our data splits involve `co-35/80/115` as defined above. We test on `minival`.

6.1. Implementation

Our object detector is Faster R-CNN [30] with the FPN backbone [23] and the RoIAlign improvement [15]. We adopt the joint end-to-end training as described in [31]. Note that this is unlike in our keypoint experiments where we froze the RPN stage (which created the same set of boxes for keypoint ensembling). To produce the ensemble results, we simply take the union set of the boxes predicted under different transformations, and combine them using bounding box voting [10] (a process similar to non-maximum suppression that merges the suppressed boxes). This ensembling strategy on the union set of boxes shows the flexibility of our method: it is agnostic to how the results from multiple transformations are aggregated.

The object detection task involves multiple categories. A single threshold of score for generating labels may lead to strong biases. To address this issue, we set a per-category threshold of score confidence for annotating objects in the unlabeled data. We choose a threshold for each category such that its average number of annotated instances per image in the unlabeled dataset matches the average number of instances in the labeled dataset. Figure 7 shows some examples of the generated annotations on `un-120`.

6.2. Object Detection Results

We investigate data distillation in two cases (Table 4):

labeled	unlabeled	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
`co-35`		30.5	51.9	31.9	15.2	33.0	40.6
`co-35`	`co-80`	**32.3**	**53.8**	**33.9**	**16.8**	**35.5**	**43.7**
`co-115`		37.1	59.1	39.6	20.0	40.0	49.4

(a) **Small-scale data.** Data distillation is performed on `co-35` with labels and `co-80` without labels, *vs.* fully-supervised learning performed on `co-35` and `co-115`. The backbone is ResNet-50.

backbone	DD	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
ResNet-50		37.1	59.1	39.6	20.0	40.0	49.4
ResNet-50	✓	**37.9**	**60.1**	**40.8**	**20.3**	**41.6**	**50.8**
ResNet-101		39.2	61.0	42.3	21.7	42.9	52.3
ResNet-101	✓	**40.1**	**62.1**	**43.5**	**21.7**	**44.3**	**53.7**
ResNeXt-101-32×4		40.1	62.4	43.2	22.6	43.7	53.7
ResNeXt-101-32×4	✓	**41.0**	**63.3**	**44.4**	**22.9**	**45.5**	**54.8**

(b) **Large-scale data.** Data distillation (DD) is performed on `co-115` with labels and `un-120` without labels, comparing with the supervised counterparts trained on `co-115`.

Table 4. Data distillation for COCO object detection. Box AP is reported on COCO `val2017`.

(i) Small-scale data: we use `co-35` as the labeled data and treat `co-80` as unlabeled.
(ii) Large-scale data: we use `co-115` as the labeled data and `un-120` as unlabeled.

Small-scale data. Similar to the keypoint case, the semi-supervised learning result of data distillation (Table 4a) is higher than that of fully-supervised training in `co-35`, but upper-bounded by that in `co-115`. However, in this case, the data distillation is closer to the lower bound (32.3 *vs.* 30.5) and farther away from the upper bound. This result requires further exploration, which we leave to future work.

Large-scale data. Table 4b shows the data distillation result using `co-115` as labeled and `un-120` as unlabeled data, comparing with the fully-supervised result in `co-115`. Our method is able to improve over the fully-supervised baselines. Although the gains may appear small (0.8-0.9 points in AP and 0.9-1.1 points in AP_{50}), the signal is consistently observed for *all network backbones and for all metrics*. The biggest improvement is seen in the AP_M metric, with an increase of up to 1.8 points (from 43.7 to 45.5 in ResNeXt-101-32×4).

The results in Table 4a and 4b suggest that object detection with unlabeled data is a more challenging task, but unlabeled data with data distillation can still help.

7. Conclusion

We show that it is possible to surpass large-scale supervised learning with *omni-supervised learning*, *i.e.*, using all available supervised data together with large amounts of unlabeled data. We achieve this by applying *data distillation* to the challenging problems of COCO object and keypoint detection. We hope our work will attract more attention to this practical, large-scale setting.

Acknowledgements

We would like to thank Daniel Rueckert for his support and guidance during the initial stages of the project.

References

[1] P. Bachman, O. Alsharif, and D. Precup. Learning with pseudo-ensembles. In *Neural Information Processing Systems (NIPS)*, 2014. 2

[2] A. Blum and T. Mitchell. Combining labeled and unlabeled data with co-training. In *Proceedings of the eleventh annual conference on Computational learning theory*, 1998. 2

[3] C. Bucilu, R. Caruana, and A. Niculescu-Mizil. Model compression. In *SIGKDD*, 2006. 1, 2, 3

[4] O. Chapelle, B. Scholkopf, and A. Zien. *Semi-supervised learning*. MIT Press, 2006. 2

[5] X. Chen, A. Shrivastava, and A. Gupta. Neil: Extracting visual knowledge from web data. In *International Conference on Computer Vision (ICCV)*, 2013. 1, 2

[6] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition (CVPR)*, 2009. 1

[7] P. Dollár, R. Appel, S. Belongie, and P. Perona. Fast feature pyramids for object detection. *Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2014. 2

[8] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2010. 2

[9] R. Fergus, Y. Weiss, and A. Torralba. Semi-supervised learning in gigantic image collections. In *Neural Information Processing Systems (NIPS)*, 2009. 2

[10] S. Gidaris and N. Komodakis. Object detection via a multi-region & semantic segmentation-aware cnn model. In *International Conference on Computer Vision (ICCV)*, 2015. 8

[11] R. Girshick. Fast R-CNN. In *International Conference on Computer Vision (ICCV)*, 2015. 1

[12] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Computer Vision and Pattern Recognition (CVPR)*, 2014. 1

[13] S. Gupta, J. Hoffman, and J. Malik. Cross modal distillation for supervision transfer. In *Computer Vision and Pattern Recognition (CVPR)*, 2016. 2

[14] L. K. Hansen and P. Salamon. Neural network ensembles. *Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 1990. 2

[15] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask R-CNN. In *International Conference on Computer Vision (ICCV)*, 2017. 1, 2, 3, 5, 6, 8

[16] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Computer Vision and Pattern Recognition (CVPR)*, 2016. 1, 3

[17] G. Hinton, A. Krizhevsky, and S. Wang. Transforming auto-encoders. Technical report, 2006. 2

[18] G. Hinton, O. Vinyals, and J. Dean. Distilling the knowledge in a neural network. *arXiv:1503.02531*, 2015. 1, 2, 3

[19] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei. Large-scale video classification with convolutional neural networks. In *Computer Vision and Pattern Recognition (CVPR)*. Sports 1M / CC INTL 3.0 / https://github.com/gtoderici/sports-1m-dataset, 2014. 2, 4, 5

[20] A. Krizhevsky, I. Sutskever, and G. Hinton. Imagenet classification with deep convolutional neural networks. In *Neural Information Processing Systems (NIPS)*, 2012. 1, 2, 3

[21] S. Laine and T. Aila. Temporal ensembling for semi-supervised learning. In *International Conference on Learning Representations (ICLR)*, 2017. 1, 2

[22] L.-j. Li, G. Wang, and L. Fei-fei. Optimol: automatic online picture collection via incremental model learning. In *Computer Vision and Pattern Recognition (CVPR)*, 2007. 1, 2

[23] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie. Feature pyramid networks for object detection. In *Computer Vision and Pattern Recognition (CVPR)*, 2017. 3, 8

[24] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft COCO: Common objects in context. In *European Conference on Computer Vision (ECCV)*. COCO / CC INTL 4.0 / http://cocodataset.org/#home, 2014. 1, 2, 5, 8

[25] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *Computer Vision and Pattern Recognition (CVPR)*, 2015. 1

[26] D. Lopez-Paz, L. Bottou, B. Schölkopf, and V. Vapnik. Unifying distillation and privileged information. In *International Conference on Learning Representations (ICLR)*, 2016. 2

[27] G. Papandreou, T. Zhu, N. Kanazawa, A. Toshev, J. Tompson, C. Bregler, and K. Murphy. Towards accurate multi-person pose estimation in the wild. In *Computer Vision and Pattern Recognition (CVPR)*, 2017. 2, 6

[28] P. O. Pinheiro, R. Collobert, and P. Dollar. Learning to segment object candidates. In *Neural Information Processing Systems (NIPS)*, 2015. 1

[29] S. Reed, H. Lee, D. Anguelov, C. Szegedy, D. Erhan, and A. Rabinovich. Training deep neural networks on noisy labels with bootstrapping. *arXiv:1412.6596*, 2014. 2

[30] S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *Neural Information Processing Systems (NIPS)*, 2015. 1, 3, 8

[31] S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. *Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2017. 8

[32] A. Romero, N. Ballas, S. E. Kahou, A. Chassang, C. Gatta, and Y. Bengio. Fitnets: Hints for thin deep nets. In *International Conference on Learning Representations (ICLR)*, 2015. 2

[33] C. Rosenberg, M. Hebert, and H. Schneiderman. Semi-supervised self-training of object detection models. 2005. 1, 2

[34] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein,

A. C. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 2015. 5

[35] M. Sajjadi, M. Javanmardi, and T. Tasdizen. Regularization with stochastic transformations and perturbations for deep semi-supervised learning. In *Neural Information Processing Systems (NIPS)*, 2016. 2

[36] H. Scudder. Probability of error of some adaptive pattern-recognition machines. *IEEE Transactions on Information Theory*, 1965. 1, 2

[37] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun. Overfeat: Integrated recognition, localization and detection using convolutional networks. In *International Conference on Learning Representations (ICLR)*, 2014. 2

[38] T. Simon, H. Joo, I. Matthews, and Y. Sheikh. Hand keypoint detection in single images using multiview bootstrapping. In *Computer Vision and Pattern Recognition (CVPR)*, 2017. 2

[39] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations (ICLR)*, 2015. 1

[40] C. Sun, A. Shrivastava, S. Singh, and A. Gupta. Revisiting unreasonable effectiveness of data in deep learning era. In *International Conference on Computer Vision (ICCV)*, 2017. 7

[41] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *Computer Vision and Pattern Recognition (CVPR)*, 2015. 1

[42] C. Szegedy, S. Reed, D. Erhan, D. Anguelov, and S. Ioffe. Scalable, high-quality object detection. *arXiv:1412.1441*, 2014. 2

[43] V. Vapnik. *Statistical learning theory*. Wiley New York, 1998. 1, 2

[44] V. Vapnik and R. Izmailov. Learning using privileged information: similarity control and knowledge transfer. *JMLR*, 2015. 2

[45] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Computer Vision and Pattern Recognition (CVPR)*, 2001. 2

[46] J. Weston. Large-scale semi-supervised learning. In *Proceedings of NATO Advanced Study Institute on Mining Massive Data Sets for Security*, 2008. 1, 2

[47] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He. Aggregated residual transformations for deep neural networks. In *Computer Vision and Pattern Recognition (CVPR)*, 2017. 3

[48] D. Yarowsky. Unsupervised word sense disambiguation rivaling supervised methods. In *Proceedings of the 33rd annual meeting on Association for Computational Linguistics*, 1995. 1, 2

[49] X. Zhu. Semi-supervised learning literature survey. Technical report, Computer Sciences, University of Wisconsin-Madison, 2005. 2

[50] X. Zhu and A. B. Goldberg. Introduction to semi-supervised learning. *Synthesis lectures on artificial intelligence and machine learning*, 2009. 2