Knowledge Distillation for Low-Power Object Detection: A Simple Technique and Its Extensions for Training Compact Models Using Unlabeled Data

Amin Banitalebi-Dehkordi
Huawei Technologies Canada Co., Ltd.
amin.banitalebi@huawei.com

Abstract

The existing solutions for object detection distillation rely on the availability of both a teacher model and ground-truth labels. We propose a new perspective to relax this constraint. In our framework, a student is first trained with pseudo labels generated by the teacher, and then fine-tuned using labeled data, if any available. Extensive experiments demonstrate improvements over existing object detection distillation algorithms. In addition, decoupling the teacher and ground-truth distillation in this framework provides interesting properties such as: 1) using unlabeled data to further improve the student's performance, 2) combining multiple teacher models of different architectures, even with different object categories, and 3) reducing the need for labeled data (with only 20% of COCO labels, this method achieves the same performance as the model trained on the entire set of labels). Furthermore, a by-product of this approach is the potential usage for domain adaptation. We verify these properties through extensive experiments.

1. Introduction

Deployment of deep learning models to edge devices often imposes constraints on size, latency, and runtime memory. Knowledge distillation [3, 13] is one of the most promising ways of producing compact models. Knowledge distillation for image classification was introduced in [13], where an ensemble of large teacher models was distilled to a smaller student model without a considerable performance loss. The main idea was to extract the so-called 'dark knowledge' and teach that to the student model. This was achieved by introducing a loss term on the "soft-targets" (more details in Section 2.1). Since then, there has been a large amount of publications focused on improving the distillation method of [13] (a.k.a vanilla distillation).

The majority of the distillation related literature (e.g. [3, 13, 32, 38]) focus on the image classification task and in fact the formulation proposed in [13] holds only for classification networks. That being said, object detection [30, 29, 9, 20, 31, 19, 17, 15] is a much more practical task, and this motivated others to investigate distillation for object detection.

The existing works [18, 37, 5, 23, 6] on object detection distillation propose to use feature or detector outputs (box, confidence, class probabilities) matching between student and teacher models, so the student's activations follow those of the teacher's. These methods are often verified by choosing students of the same architecture as the teacher, but shallower and thinner. These solutions rely on the availability of both a teacher model and ground truth labels. In addition, for multi-teacher distillation, they assume that all teachers detect the same object classes. These assumptions limit the practicality of using knowledge distillation for commercial services where a user uploads a large teacher model to the cloud (but provides no labeled data or a reduced set only), and the goal is to train a compact student model from it. It also makes it difficult to consider the case where teacher models are experts in detecting different type and number of object classes, and potentially with different architectures.

This paper proposes a simple, yet powerful approach for knowledge distillation in object detector neural networks. We argue that knowledge distillation is intrinsically different between the image classification and object detection tasks. In other words, the so called dark knowledge does not lie anymore in some layers' features. Instead, the student model generalizes better when it is first presented with simpler object labels explicitly. Moreover, student object detectors can incrementally improve from teacher distillation and ground truth training. This idea is in some sense related to [26] where 'teacher assistants' were shown to be helpful for image classification distillation. Teacher assistants introduced in [26] are neural networks of sizes (thus capacities) between those of the student and the teacher. [26] argued that students learn better first from the teacher assistants, as it is easier for the students to learn logits of a more similar feature space. The fundamental difference between the methodology proposed in this paper (for object detection) and [26] (for image classification) is that we do not introduce any extra models. Instead, we argue that the student can learn better if it is first trained together with the teacher guidance and a subset of data that is carefully generated for it to learn. In other words, instead of letting the student network learn from the training data on its own, we utilize the teacher to label the data for the student, thereby providing a subset of the whole training dataset, with predictions that are easier to follow by the student.

The proposed framework decouples distilling from the ground truth data and the teacher model from each other. This provides several nice properties such as:

a Being able to use unlabeled data to further improve the student's performance.

b Reducing the need for labeled data.

c Distilling from (combining) multiple teacher models of different architectures, even with different object classes (with or without overlap).

d Performing domain adaptation as a by-product, where limited or no labels are available for the target domain.

These properties are often practical necessities for commercial cloud model compression services. We verify these properties through extensive experiments in Section 3.

The main contributions of this paper are summarized as:

- *A new object detection distillation strategy*: with properties such as being able to improve the distillation performance using unlabeled data, reducing the need for labeled data, or combining multiple detector models. To the best of our knowledge, there has been no previous work on object detection distillation that decouples learning from the teacher and ground truth data, nor one that combines generic object detectors of (non-)overlapping object classes like our method does.

- *Insights gained from an extensive set of experiments*: We designed a comprehensive set of experiments to evaluate the proposed object detection distillation algorithm. The experiments are broken down in a way that they provide insights on 1) whether or not using unlabeled data can help the distillation, 2) if so, what is the impact of the unlabeled data size on the performance improvements? 3) the role of techniques such as feature matching, imitation masking, or box matching, in object detection distillation.

The rest of this paper is organized as follows: Section 2 reviews the related works. Section 3 describes the proposed distillation framework in details. Section 4 discusses the experiments results. And, Section 5 concludes the paper.

2. Related works

This section provides a brief overview of the related works, and draws connections between them and our work.

2.1. Relations to distillation for image classification

Classification distillation exploits soft-targets matching between the teacher and student logits. It was argued in [13] that soft-targets provide a better discrimination between the classes with low likelihood values, thereby allowing the student to learn more than just the best target class label:

$$L = L^{GT} + \alpha L^{KD} \tag{1}$$

where L^{GT} denotes the ground truth loss term, L^{KD} is the distillation loss, and α is a weighting parameter.

There are other studies that tried to build upon the baseline distillation of [13]. Among them, FitNets [32] proposed the student to mimic the teacher feature maps (hints). This imposes a constraint on the student which can sometimes be too strict, since the capacities of teacher and student may differ considerably. Attention Transfer [41] relaxes the assumptions of FitNet. The student network is trained not only to have similar features to the teacher, but also to have similar attention maps. In [35] authors designed pruned student models and customized the distillation for target hardware. [22, 7] proposed data free distillation methods for the classification task where there are no training data available.

Even though many ideas can be borrowed from the classification literature, but there are still some intrinsic differences between the tasks of classification and detection that requires careful specialized designs for object detection distillation. One can still employ soft-logits matching on layers of a detector network (backbone, head, or intermediate layers), but how to distill bounding boxes (a regression problem) may not be best addressed through logits matching.

In addition, classification distillation requires identical target classes between the teacher(s) and student. Different or only partially overlapping classes result in catastrophic learning problems, since unlike detectors (that learn to detect only the target object of interest and ignore everything else), classifiers will have to pick a class anyway. In case a training example doesn't belong to a category known to a teacher, it will be assigned a wrong class. Subsequently, the student will be trained with a wrong class. For example two teachers: one a person/cat classifier, and one a car/bicycle classifier, can't be jointly distilled to a student model (not a problem for detection as we show in Section 4.4.2). This problem exists in supervised distillation, and becomes even more challenging in the case of distillation with unlabeled data. Although there have been attempts to address this problem [36], but generally its literature is very limited.

That said, supplementary materials [1] contain more details and numerical evaluations on the classification task.

2.2. Related works on object detection distillation

Among the recent works, [18] proposed a framework for distilling knowledge to object detector models where the distillation loss term is based on distance between detector features of the teacher and student. This enforces the student model to mimic the teacher. [37] built on this idea by introducing the concept of imitation (objects) masks. These masks highlight the location of objects. The distillation loss term is masked by the imitation mask, thereby the student is pushed only to learn the objects, without any constraint on the background within the distillation loss term. Note that this algorithm requires the ground truth object labels to be available. On the other hand, [5] argued that in the case of object detector models, the backbone CNN features provide a stronger discriminative ability than the detector head features. Authors in [5] suggested that students distilled from teacher backbone features generalize better than the ones trained with the detector head features.

While [18, 37, 5] defined knowledge distillation loss as a feature matching distance, [23] proposed to directly use the detector outputs (boxes, confidence scores, and class probabilities). To this end, the distillation loss in [23] is defined as three terms on objectness confidence score, class probabilities, and bounding boxes predicted. For each prediction, the two later terms are weighted by the objectness score, so the contribution of each predicted bounding box is according to its confidence level of being an object. It is worth noting that we found in our experiments that this method of knowledge distillation works better when candidate boxes predicted by the teacher model are filtered (e.g. with non-max suppression). Otherwise, if a teacher model is not trained well, its predictions might be too noisy for a student model to learn from.

A hybrid approach was taken in [6] where distillation loss has a feature matching term in addition to a box/probabilities matching term. The feature matching term is a L2 loss between the backbone features. The detection outputs term however includes a bounded regression loss for the bounding boxes and a cross entropy loss for the classification probabilities.

As mentioned in Section 1, the existing object detection distillation methods require labeled data to perform distillation, often pose constraints on models architectures, and can't handle non/partially overlapping object categories.

2.3. Relations to semi/self/un-supervised learning

There are several lines of works in semi/self/un-supervised learning related to our method. First are the generative models (in the context of distillation are sometimes called data-free). These methods learn to generate realistic examples to train the student with. The examples are generated either from unlabeled training data, or from noise [40, 2]. These methods are mostly trained and evaluated on classification datasets and at relatively low resolutions (32x32, 64x64, or highest ones at 256x256 [2]). On the other hand, state-of-the-art object detectors require high resolutions (e.g. as high as 1536 for EfficientDet-D7 [34]). Also, classification methods do not simply extend to detection (see 2.1). Moreover, generative methods need not only to generate image contents, but also bounding boxes.

Another line of related work is zero-shot distillation, where synthesized data impressions from the teacher are used as surrogates to train the student [24, 28]. The advantage of these methods is that they do not need any training data. However, similar to generative works, they are mostly in the context of classification (see 2.1). Moreover, the upper-bound of performance for both these works and generative works is often considered to be the student's performance with full data on supervised training, or on original knowledge distillation [24, 28, 40, 2]. We go well above these bounds, as shown in Section 4.

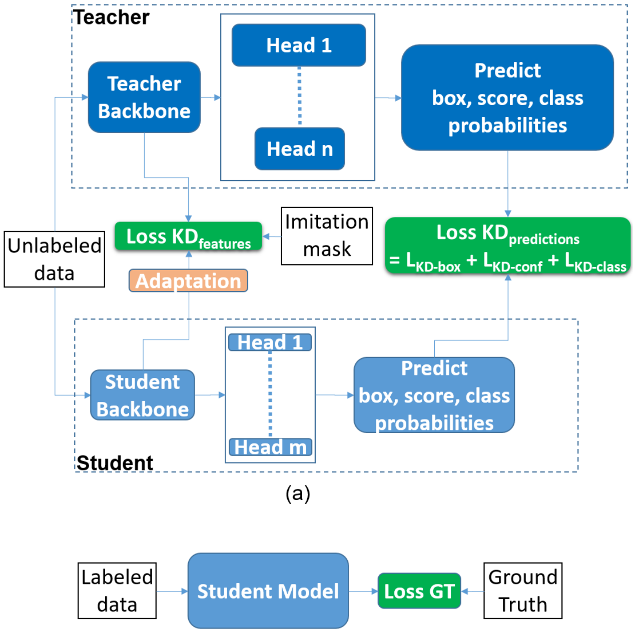
In the context of semi-supervised learning, [39] used a large set of unlabeled data for improving the accuracy of ImageNet classifiers. Note that using unlabeled data in weakly supervised classification (i.e. training with pseudo labels of teacher(s)) is only feasible if the unlabeled data is collected from the same classes of the teacher model. This is due to the fact that each and every training example is assigned to a class, and having images from outside classes results in bad label assignment (less problematic for ImageNet since it has a large number (1000) of classes). For example, one can't use arbitrary images for distilling a cat/dog classifier. This becomes less of an issue when relying on other sources of information, e.g. crawling data from the web, and using meta-data or file tags (but it's not always possible to do so). Our proposed method does not suffer from this phenomenon. It is shown in Section 4.4.1 that even a two category object detector can effectively be trained via distillation using the entire open images dataset (unlabeled), without a curated example selection procedure.

Self-supervised learning is another related area, that has gained momentum recently. The idea is to learn general representations from one or more auxiliary tasks [14]. Since the auxiliary tasks are not aware of the down-stream task, they rely on unsupervised representation learning only. Having a teacher model along side the unlabeled data is expected to perform better, as shown in Section 4.

3. The proposed method

This section elaborates on the proposed knowledge distillation strategy for object detection. Our hypothesis is that in a teacher-student distillation scenario, the student model learns better if it is first trained by a 'teaching source' that has a comparable capacity. This makes sense because the intermediate teaching source acts as a hint for the student model during distillation.

An observation motivated our hypothesis further: if we allow the student model to learn only the teacher's understanding of the data, it can later generalize better on the


Teacher
Teacher Backbone
Head 1
Head n
Predict box, score, class probabilities
Unlabeled data
Loss KD$_{\text{features}}$
Imitation mask
Loss KD$_{\text{predictions}}$ = L$_{\text{KD-box}}$ + L$_{\text{KD-conf}}$ + L$_{\text{KD-class}}$
Adaptation
Head 1
Student Backbone
Predict box, score, class probabilities
Head m
Student
Labeled data
Student Model
Loss GT
Ground Truth

(a)

(b)

Figure 1: Distillation for object detection: (**a**) Step 1: distill from only a teacher, and (**b**) Step 2: fine-tune using any available labeled data.

whole set of ground truth labels. To this end, we label the training data with a pre-trained teacher model. Since the teacher model does not have a perfect detection ability, it will only detect a limited number of object instances within a subset of training samples. These samples are the ones that are likely easier for the student to learn from and teacher detections (although they could be noisy) are the ones that the student can follow better than trying to explore the entire search space on its own. This was tested with a two class subset of the Microsoft COCO dataset [21] (Person & Bicycle). After learning from the teacher's pseudo labels, fine-tuning was done using the ground truth labels. The 2-stage training strategy achieved a higher mean Average Precision (mAP) value than the standard supervised training of the student (See Section 3 for more details and a comprehensive set of follow-up experiments).

The observation above further motivates the proposed distillation framework. In this framework, the student model is trained in two steps: 1) distillation using the teacher model only, without seeing the ground truth training labels, and 2) fine-tuning over the ground truth training data. Decoupling the distillation in this way allows the student model to first learn from a teacher-guided sub-space of the original parameter space. This is an easier job for the student and positions it to learn better when it sees the ground truth labels during the fine-tuning step. Fig. 1 illustrates the architecture of this approach.

Loss function for the first distillation step is defined as:

$$L^{KD} = L^{KD}_F + L^{KD}_P \tag{2}$$

where L^{KD} refers to the distillation loss. L^{KD}_F and L^{KD}_P denote the feature matching and object detection prediction components of the distillation loss. L^{KD}_F is define as:

$$L^{KD}_F = ||F^T \otimes M^T - F^S \otimes M^T||^2_2 \tag{3}$$

where F^T and F^S are features from the teacher and student, M^T denotes the imitation (objects) mask generated from the bounding box predictions of the teacher, and $\otimes$ is the element-wise multiplication operator. Our experiments were in agreement with [5] in that backbone features prove to be more useful. Also, the imitation (objects) masks [37] are being applied to the backbone features (not the detector heads). In order to ensure the shapes and dimensions are compatible for matching, an adaptation block needs to be added after the students feature response. We found out in our experiments that a minimal adaptation size achieves the best performance, so a one layer convolution was used for adaptation. This makes sense since only the main student network without the adaptation is later used for validation, so the knowledge of detection should stay in the actual student model and not the adaptation piece. It is also worth noting that Non-Max Suppression (NMS) is applied to the teacher model predictions to reduce the noisiness of the predicted bounding boxes. In the case of YOLO-based object detectors [29], the prediction loss component is defined as:

$$L^{KD}_P = L^{KD}_{box} + L^{KD}_{conf} + L^{KD}_{class} \tag{4}$$

The three components in (4) account for bounding box regression, objectness confidence, and class probability. We used a modified loss definition compared to the original YOLO model [29] for better convergence.

The loss components in our set up are defined as:

$$L^{KD}_{box} = L^{KD}_{xy} + L^{KD}_{wh} \tag{5}$$

$$L^{KD}_{xy} = \sum_{i=0}^{K^2} \sum_{j=0}^{B} M^T_{ij}[(x^T_{ij} - x^S_{ij})^2 + (y^T_{ij} - y^S_{ij})^2] \tag{6}$$

$$L^{KD}_{wh} = \sum_{i=0}^{K^2} \sum_{j=0}^{B} M^T_{ij}[(w^T_{ij} - w^S_{ij})^2 + (h^T_{ij} - h^S_{ij})^2] \tag{7}$$

$$\begin{aligned} L^{KD}_{conf} = &\sum_{i=0}^{K^2} \sum_{j=0}^{B} M^T_{ij} \times \sigma_E(M^T_{ij}, C^S_{ij}) \\ &+ (1 - M^T_{ij}) \times \mathbb{1}^{ign}_{ij} \times \sigma_E(M^T_{ij}, C^S_{ij}) \end{aligned} \tag{8}$$

$$L_{class}^{KD} = \sum_{i=0}^{K^2} \sum_{j=0}^{B} M_{ij}^{T} \times \sigma_E(M_{ij}^{T}, P_{ij}^{S}) \tag{9}$$

where L_{xy}^{KD} and L_{wh}^{KD} are regression losses for the center and size of the boxes, B is number of predicted boxes, K is number of YOLO grid cells in each direction, M_{ij}^{T} is object (imitation) mask defined by the teacher at coordinate location (i, j), (x, y) is center of a box, (w, h) are width and height of a box, $\mathbb{1}_{ij}^{ign}$ is an ignore mask (0 when IOU between the predicted box and the teacher's box is less than a threshold e.g. 0.5, and 1 otherwise), C_{ij}^{S} is confidence logit predicted by the student, σ_E is sigmoid cross entropy, and P_{ij}^{S} denotes class probability logits predicted by the student.

In the case of YOLO architectures with more than one scale (e.g. YOLOv3), the distillation loss defined in (2) is calculated per each scale and then summed up to form the overall loss. Also, in the case of RCNN like detectors [9] the formulation of (4) needs to be modified accordingly.

The second step of distillation leverages the ground truth labels. Loss is defined similar to (4), but between the student predictions and the ground truth:

$$L^{GT} = L_{box} + L_{conf} + L_{class} \tag{10}$$

where L^{GT} is the detection loss between the student predictions and the ground truth labels. The three loss components of (10) are calculated similar to (5)-(9). However, instead of the teacher model predictions, the ground truth labels are used. During the optimization, L^{KD} and L^{GT} are minimized independently, one after the other.

This way of knowledge distillation provides several interesting characteristics:

a **Usage of unlabeled data**: Since the first distillation step does not use any ground truth labels, it is possible to leverage an arbitrary large set of unlabeled images. Teacher's knowledge is distilled to student over a large dataset and this can results in a more accurate student.

b **Reducing the need for labeled data**: Data labeling is costly. A model that is trained through distillation by a teacher (and potentially large amounts of unlabeled data), may only need a limited amount of additional labels to achieve an acceptable performance (Section 3).

c **Combining (merging) pre-trained teacher models**: To this end, different teachers go over the data and provide their predictions. The predictions are then aggregated. The student model is trained with the resulting data/labels to achieve a fair performance on the union of all teachers object classes. Having labeled data for fine-tuning can boost the student model's performance.

d **Domain adaptation via distillation**, i.e. to distill knowledge of the teacher's domain to a student in another domain: Suppose teacher T is trained with data in domain D^T. The goal is to train a student model to work with data in domain D^S that is similar to D^T but not identical. An example application would be a surveillance camera system where day-time data from one camera are labeled and a model is well trained on those data (teacher). A second camera (student) that operates at night has no labeled data. It is possible to leverage the teacher-student framework proposed here to distill the knowledge from the day-time camera to the night-time one. To this end, the teacher model is used to make predictions on data collected from the student domain, so the student can be trained on these predictions. If any labeled data are available from the student domain they can be used for fine-tuning the student, otherwise, even fine-tuning on the teacher domain helps improving the student's model performance on the student domain data (More details in Section 3).

As mentioned earlier, the four properties above are important for supporting practical use-cases and applications of knowledge distillation. In the next section, we provide experiments results for each of these cases.

Algorithm 1 Knowledge distillation for object detection.

```
Inputs: Teacher model $\theta_T$; unlabeled data $\overline{\mathcal{D}}$
Optional: Labeled data $\mathcal{D}$
Output: Student model $\theta_S$
procedure OBJECT DETECTION DISTILLATION($\theta_T$,$\overline{\mathcal{D}}$,$\mathcal{D}$)
    $\overline{\mathbf{y}} \leftarrow GeneratePseudoLabels(\theta_T, \overline{\mathcal{D}})$
    $\theta_S \leftarrow$ Perform distillation according to (2) on $\overline{\mathcal{D}}$, $\overline{\mathbf{y}}$
    if $\mathcal{D} \neq \varnothing$ then
        $\theta_S \leftarrow FineTune(\theta_S, \mathcal{D})$
```

4. Experiment results and discussions

This section studies the performance of the proposed object detection distillation approach through a comprehensive set of experiments.

4.1. Model architectures

Without the loss of generality, we choose YOLOv3 architecture for most of experiments. Note that the distillation framework proposed in this paper fits well also with other detector models such as RCNN or RetinaNet based detectors. In most of the experiments, the original YOLOv3 architecture with DarkNet53 backbone is selected as the teacher. Later in Section 3 we also try a FasterRCNN teacher to study the effect of architecture change. For the student, we use a custom slim YOLOv3 with a 23 layer backbone. This model is shallower and thinner than the teacher, and its size on disk is around one tenth of the teacher. Training is done from scratch with ImageNet pretrained weights. Hyper-parameters such as non-max suppression parameters, score threshold, maximum number of detections, and other parameters are consistent between the teacher and student, and are similar to the ones used in the DarkNet implementation of YOLOv3 [30].

4.2. Tuning and training tricks

When evaluating the distillation performance, we need to ensure that the student, teacher, and distilled student models are highly tuned and have achieved a capacity that may not be improved further with common tuning and training tricks. To this end, we employed the following techniques to boost the validation accuracy for the trained models. With that, we tried to decouple improvements that can be made through distillation from teacher models and the ones from typical training tricks and tunings:

Augmentation:

- Random crop (with constraints on bounding box)
- Random color distortions in the HSV domain
- Random flip: horizontal or vertical
- Random expansion (place to a larger canvas)

Multi-scale input: Performance of object detectors is usually compared at a given input image resolution. Using various input sizes (in the range of 320 × 320 to 640 × 640) dynamically during the training consistently improved the performance in our experiments. The validation image size is set at 416 × 416.

Mix-up training: Mix-up was first introduced for classifiers and GANs [42] to alleviate issues such as memorization and sensitivity to adversarial examples. It involves with mixing training examples and their associated labels. This idea was modified here to be used for the detection task.

Label smoothing: Label smoothing did not help with the distillation performance and is not used in our experiments. This finding is in agreement with the state-of-the-art [27].

Focal loss: Focal loss [20] was initially proposed to improve the object detection performance on hard examples and to address the class imbalance between foreground and background. We include focal loss during the training.

Learning rate scheduling: For each training job, we tried separately a wide range of learning rate scheduling methods: exponential, piecewise, fixed, cosine decay, and cosine decay with restart. We also used a warm up [11] strategy as it prevents divergence at the start of training.

4.3. Evaluation metric

After each training job, the model with the best validation mAP is selected. The training process is repeated for 10 times to reduce the impact of random initialization on the performance measured (around 2-3K GPU-hours per each table entry reported here). The average mAP along with the range of mAP for best models are presented in this section. We utilized the mAP implementation available in the FAIR's Detectron repo [10]. The mAPs @50% performance are reported in this section.

4.4. Performance evaluations

4.4.1 Teacher-Student distillation experiments

The proposed teacher-student framework is evaluated on both small and large scale number of object classes to ensure the results do not behave differently at various scales. We first evaluate the models using a subset of COCO dataset with only two object classes, 'person' and 'bicycle'. Then we evaluate the models on the entire COCO 2017 object detection dataset (80 classes, 118K training and 5K validation examples). Moreover, later in this section we design experiments in which multiple two-class teacher object detectors are combined to form a multi-class student.

Table 1 shows the performance on two class object detection. For distilled models in this table, we used the COCO dataset, the subset images that include at least one of these two classes. Table 2 shows the results when OpenImagesV5 dataset [16] is used for teacher only distillation step (out of 1.7M images, we use whatever many the teacher can predict at least one bounding box on). No labels were available for this step. In all experiments, the validation set of COCO was used for mAP calculation. Also, the tables in this section incorporate abbreviations when referring to various methods, for the sake of brevity. To this end, SD, FM, IM, PM, UD, FT, SSL, and OID denote ”Supervised Distillation”, ”Feature Matching”, ”Imitation Masking”, ”Predictions (boxes, scores, and classes) Matching”, ”Unsupervised Distillation (only distill with teacher's pseudo labels on unlabeled data)”, ”Fine-Tuning”, ”Self-Supervised Learning”, and Open Images Dataset, respectively.

It is observed from Table 1 that the proposed distillation strategy performs well. Feature matching and imitation masks have also improved the performance. In addition, Table 2 shows that using unlabeled data has considerably improved the overall mAP. Supplementary material [1] contains examples were students trained with distillation can detect a higher number of bounding boxes than the students trained supervised without distillation.

To study the case that teacher and student are from different architectures and trained on different data, we change the teacher to a FasterRCNN model trained on Open Images dataset. Table 3 shows results for this experiment, where the FasterRCNN teacher is distilled to the custom YOLO-based student. It is observed from this table that changing the architecture slightly reduces the improvement gap, but the trend is consistent with Table 1 and 2.

In the case of complete (80 object classes) COCO dataset, similar conclusions can be drawn, as observed in Table 4. Moreover, Fig. 2 shows the mAP achieved when the size of labeled data for the second step of distillation

Model	Mean mAP over 10 runs (min, max)
Teacher	0.6009 (0.599,0.603)
Baseline: student supervised	0.5295 (0.527,0.532)
SD$_{\text{COCO-FM-IM}}$ [37]	0.5421 (0.541,0.544)
SD$_{\text{COCO-FM}}$ [18]	0.5365 (0.534,0.539)
SD$_{\text{COCO-PM}}$ [23]	0.5284 (0.527,0.531)
UD$_{\text{COCO}}$	0.4553 (0.453,0.458)
UD$_{\text{COCO}}$ + FT$_{\text{COCO}}$	**0.5533** (0.553,0.554)
UD$_{\text{COCO-FM-IM}}$	0.4673 (0.466,0.468)
Ours: UD$_{\text{COCO-FM-IM}}$ + FT$_{\text{COCO}}$	**0.5629** (0.562,0.564)

Table 1: Two class knowledge distillation. (63K training samples, 2.7K validation). Abbreviations defined in 4.4.1.

Model	Mean mAP over 10 runs (min, max)
UD$_{\text{OID}}$	0.4940 (0.492,0.496)
UD$_{\text{OID}}$ + FT$_{\text{COCO}}$	0.5815 (0.580,0.583)
UD$_{\text{OID-FM-IM}}$	0.4993 (0.498,0.501)
Ours: UD$_{\text{OID-FM-IM}}$ + FT$_{\text{COCO}}$	**0.5899** (0.589,0.591)

Table 2: Two-class object detection distillation using larger unlabeled set (63K labeled + 1.03M unlabeled training samples, 2.7K validation). Abbreviations are defined in 4.4.1.

Model	Mean mAP over 10 runs (min, max)
UD$_{\text{COCO}}$	0.4248 (0.422,0.426)
UD$_{\text{COCO}}$ + FT$_{\text{COCO}}$	0.5425 (0.540,0.545)
UD$_{\text{OID}}$	0.4592 (0.456,0.463)
Ours: UD$_{\text{OID}}$ + FT$_{\text{COCO}}$	**0.5647** (0.562,0.566)

Table 3: Using an entirely different teacher architecture (FasterRCNN teacher to a slim YOLO-based student) (63K labeled + 1.03M unlabeled training samples, 2.7K validation). Abbreviations are defined in 4.4.1.

(fine-tuning) is reduced. It is observed from Fig. 2 that even with 20% of labels, distilled student is on par with the student trained with all labels. Note that if the data size used for fine-tuning is too small and the student is fine-tuned for long, it will then over-fit to the small set of fine-tuning data, and yields lower validation accuracy.

An interesting observation in Fig. 2 is that in the 80-class case, the student trained only with the teacher without any fine-tuning can outperform the student that is trained supervised with all labels. This is not the case for the two-class scenario. The reason is that the 80-class teacher is much more accurate than the 80-class student trained from scratch. There is around 28% mAP gap between the two, while this gap for the two-class case is around 7%. That means the teacher's knowledge in the 80-class case becomes much more valuable for the student, and with the help of unlabeled data it can achieve a reasonable performance. In the two-class case however, the student does not rely on the teacher as much, and generally has an easier task to learn.

Model	Mean mAP over 10 runs (min, max)
Teacher	0.6206 (0.618,0.623)
Baseline: Student supervised	0.3449 (0.342,0.347)
SSL$_{\text{COCO-rotnet}}$ + FT$_{\text{COCO}}$ [8]	0.3303 (0.330,0.331)
SSL$_{\text{OID-rotnet}}$ + FT$_{\text{COCO}}$ [8]	0.3429 (0.342,0.343)
SD$_{\text{COCO-FM-IM}}$ [37]	0.3817 (0.381,0.383)
SD$_{\text{COCO-FM}}$ [18]	0.3755 (0.375,0.376)
SD$_{\text{COCO-PM}}$ [23]	0.3694 (0.369,0.370)
UD$_{\text{COCO}}$	0.3435 (0.341,0.346)
UD$_{\text{COCO}}$ + FT$_{\text{COCO}}$	0.4015 (0.400,0.403)
UD$_{\text{COCO-FM-IM}}$	0.3496 (0.348,0.351)
UD$_{\text{COCO-FM-IM}}$ + FT$_{\text{COCO}}$	0.4079 (0.406,0.409)
UD$_{\text{OID}}$	0.3567 (0.355,0.359)
UD$_{\text{OID}}$ + FT$_{\text{COCO}}$	0.4166 (0.415,0.418)
UD$_{\text{OID-FM-IM}}$	0.3608 (0.359,0.361)
Ours: UD$_{\text{OID-FM-IM}}$ + FT$_{\text{COCO}}$	**0.4220** (0.421.0.423)

Table 4: Object detection distillation on COCO dataset (80 classes) (118K labeled + 1.28M unlabeled training samples, 5K validation). Abbreviations are defined in 4.4.1.

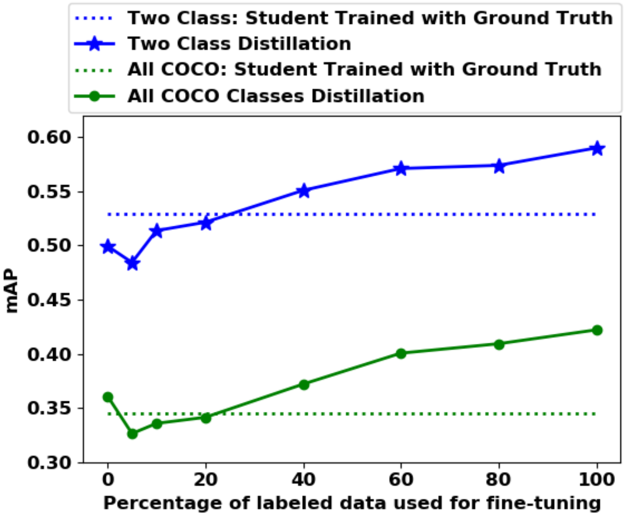


Figure 2: Distillation with limited amounts of labeled data

4.4.2 Combining object detectors

The proposed knowledge distillation framework allows for ability to merge multiple object detectors, even when they have overlapping object classes. To this end, all teachers perform a forward pass on a set of (unlabeled) examples and

Model	Mean mAP over 10 runs (min, max)
Teacher 1: Person-Car	0.6668 (0.665,0.669)
Teacher 2: Car-Cat	0.7299 (0.728,0.732)
Teacher 3: Person-Bicycle	0.6009 (0.599, 0.603)
Student: Person-Bicycle-Car-Cat	0.5844 (0.582,0.586)
$\mathrm{UD}_{\mathrm{COCO}}$ (3 teachers)	0.5250 (0.524,0.527)
$\mathrm{UD}_{\mathrm{COCO}}$ (3 teachers) + $\mathrm{FT}_{\mathrm{COCO}}$	0.6008 (0.598,0.602)
$\mathrm{UD}_{\mathrm{OID}}$ (3 teachers)	0.5386 (0.536,0.540)
Ours: $\mathrm{UD}_{\mathrm{OID}}$ (3 teachers) + $\mathrm{FT}_{\mathrm{COCO}}$	**0.6269** (0.624,0.629)

Table 5: Learning from multiple teachers (70K (labeled) + 1.31M (unlabeled) training samples, 3K validation). Abbreviations are defined in section 4.4.1.

their predictions are collected and aggregated. The student is then trained on this collection with the distillation formulation of (2). Note that the proposed approach doesn't assume any constraints on the type and number of object categories associated to different object detectors that are to be combined. Therefore, feature maps corresponding to different detectors (with potentially different object categories) highlight spatially different regions. As a result, it doesn't make sense anymore to apply feature matching or imitation masking to these detectors. Consequently, the feature matching loss term on equation (2) is discarded for this experiment. After unsupervised distillation, the student is fine-tuned with labeled data, if any are available. It is also worth noting that in general, aggregating the predictions of different models can be done in various ways such as a) affirmative: stacking all predictions and considering them all (even with duplicates), b) consensus: More than half of the teacher models must agree to consider that a region contains an object (based on IOU), and c) unanimous: All teacher models must agree to consider that a region contains an object [4]. In our case, where teachers can have non-overlapping object categories, only the affirmative strategy makes sense. In addition, after the aggregation we can optionally perform NMS to reduce the overlapping predictions. We noticed that applying NMS does not result in an improvement in the combined model's performance, likely due to the fact that noisy predictions are helping the student model to learn better. To verify this solution, we designed an experiment in which there are 3 teacher models, each detecting 2 object classes from the COCO dataset. Table 5 shows the results of this experiment, and confirms the effectiveness of the proposed solution. Supplementary materials [1] contain additional results.

4.4.3 Domain adaptation via knowledge distillation

The proposed distillation framework holds in case teacher and student models are intended for two slightly different domains. In this case, knowledge from the teacher's domain is transferred to the student domain. Suppose teacher is trained on dataset A, in our experiment a subset of COCO dataset that contains one or more humans captured during the day. The student is supposed to learn to detect humans in a different subset of COCO, dataset B that is non-overlapping with dataset A, and contains images of people captured at night.

Model	Mean mAP over 10 runs (min, max)
Baseline: teacher trained on day data	0.3325 (0.330,0.334)
$\mathrm{UD}_{\mathrm{night}}$	0.2618 (0.258,0.263)
$\mathrm{UD}_{\mathrm{night}} + \mathrm{FT}_{\mathrm{day}}$	0.3771 (0.375,0.379)
$\mathrm{UD}_{\mathrm{night}} + \mathrm{FT}_{\mathrm{night}}$	0.4318 (0.429,0.434)
$\mathrm{UD}_{\mathrm{night\text{-}FM\text{-}IM}}$	0.2779 (0.276,0.281)
$\mathrm{UD}_{\mathrm{night\text{-}FM\text{-}IM}} + \mathrm{FT}_{\mathrm{day}}$	0.3972 (0.394,0.399)
Ours: $\mathrm{UD}_{\mathrm{night\text{-}FM\text{-}IM}} + \mathrm{FT}_{\mathrm{night}}$	**0.4578** (0.455,0.459)

Table 6: Knowledge distillation for domain adaptation (2K day & 2K night images for training, 1K night images for validation). Abbreviations are defined in section 4.4.1.

Distillation process starts with teacher model to train on A, and then perform a forward pass on B to collect its labels. Teacher predictions are distilled to student according to the proposed distillation approach. If no labels are available from B, then the student is fine-tuned over A, otherwise it is fine-tuned on B. It is observed in Table 6 that distillation improves the adaptation performance.

We provide additional results on the task of robustness against corruptions (another form of domain shift) in the supplementary materials [1]. Future works include applying the proposed method to classical domain adaptation datasets such as SVHN ↔ MNIST or KITTI ↔ Cityscapes.

5. Conclusion

This paper proposes a new perspective on knowledge distillation for object detection. It proposes to decouple the distillation from teacher and labeled data. To this end, the teacher model uses a pool of unlabeled data to provide the student with a subset of the entire parameters space to search in. The teacher distillation uses feature matching with imitation masking and detection loss on bounding boxes, confidence scores, and class probabilities. If any labeled data are available it will be used for fine-tuning the student. This way of distillation allows for leveraging unlabeled data, combining several teacher models with different object classes, and seems promising for domain adaptation.

References

[1] Authors. Knowledge distillation for low-power object detection: A simple technique and its extensions for training

compact models using unlabeled data, 2021. Supplementary materials 02-supp.pdf. 2, 6, 8

[2] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*, 2018. 3

[3] Cristian Buciluă, Rich Caruana, and Alexandru Niculescu-Mizil. Model compression. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 535–541, 2006. 1

[4] Angela Casado-Garcıa and Jónathan Heras. Ensemble methods for object detection. In *European conference on artificial intelligence, ECAI*, 2020. 8

[5] Bor-Chun Chen, Larry S Davis, and Ser-Nam Lim. An analysis of object embeddings for image retrieval. *arXiv preprint arXiv:1905.11903*, 2019. 1, 3, 4

[6] Guobin Chen, Wongun Choi, Xiang Yu, Tony Han, and Manmohan Chandraker. Learning efficient object detection models with knowledge distillation. In *Advances in Neural Information Processing Systems*, pages 742–751, 2017. 1, 3

[7] Hanting Chen, Yunhe Wang, Chang Xu, Zhaohui Yang, Chuanjian Liu, Boxin Shi, Chunjing Xu, Chao Xu, and Qi Tian. Data-free learning of student networks. In *IEEE International Conference on Computer Vision*, 2019. 2

[8] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. *arXiv preprint arXiv:1803.07728*, 2018. 7

[9] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014. 1, 5

[10] Ross Girshick, Ilija Radosavovic, Georgia Gkioxari, Piotr Dollár, and Kaiming He. Detectron. https://github.com/facebookresearch/detectron, 2018. 6

[11] Akhilesh Gotmare, Nitish Shirish Keskar, Caiming Xiong, and Richard Socher. A closer look at deep learning heuristics: Learning rate restarts, warmup and distillation. *arXiv preprint arXiv:1810.13243*, 2018. 6

[12] Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, et al. The many faces of robustness: A critical analysis of out-of-distribution generalization. *arXiv preprint arXiv:2006.16241*, 2020. 12

[13] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015. 1, 2

[14] Longlong Jing and Yingli Tian. Self-supervised visual feature learning with deep neural networks: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020. 3

[15] George Jose, Aashish Kumar, Srinivas Kruthiventi SS, Sambuddha Saha, and Harikrishna Muralidhara. Real-time object detection on low power embedded platforms. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops, ICCVW*, pages 0–0, 2019. 1

[16] Ivan Krasin, Tom Duerig, Neil Alldrin, Vittorio Ferrari, Sami Abu-El-Haija, Alina Kuznetsova, Hassan Rom, Jasper Uijlings, Stefan Popov, Andreas Veit, et al. Openimages: A public dataset for large-scale multi-label and multi-class image classification. *Dataset available from https://github.com/openimages*, 2(3):18, 2017. 6

[17] Fanrong Li, Zitao Mo, Peisong Wang, Zejian Liu, Jiayun Zhang, Gang Li, Qinghao Hu, Xiangyu He, Cong Leng, Yang Zhang, et al. A system-level solution for low-power object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops, ICCVW*, pages 0–0, 2019. 1

[18] Quanquan Li, Shengying Jin, and Junjie Yan. Mimicking very efficient network for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6356–6364, 2017. 1, 2, 3, 7

[19] Yanghao Li, Yuntao Chen, Naiyan Wang, and Zhaoxiang Zhang. Scale-aware trident networks for object detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 6054–6063, 2019. 1

[20] 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. 1, 6

[21] 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. 4

[22] Raphael Gontijo Lopes, Stefano Fenu, and Thad Starner. Data-free knowledge distillation for deep neural networks. *arXiv preprint arXiv:1710.07535*, 2017. 2

[23] Rakesh Mehta and Cemalettin Ozturk. Object detection at 200 frames per second. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 0–0, 2018. 1, 3, 7

[24] Paul Micaelli and Amos J Storkey. Zero-shot knowledge transfer via adversarial belief matching. In *Advances in Neural Information Processing Systems*, pages 9551–9561, 2019. 3

[25] Claudio Michaelis, Benjamin Mitzkus, Robert Geirhos, Evgenia Rusak, Oliver Bringmann, Alexander S Ecker, Matthias Bethge, and Wieland Brendel. Benchmarking robustness in object detection: Autonomous driving when winter is coming. *arXiv preprint arXiv:1907.07484*, 2019. 12

[26] Seyed-Iman Mirzadeh, Mehrdad Farajtabar, Ang Li, and Hassan Ghasemzadeh. Improved knowledge distillation via teacher assistant: Bridging the gap between student and teacher. *arXiv preprint arXiv:1902.03393*, 2019. 1

[27] Rafael Müller, Simon Kornblith, and Geoffrey E Hinton. When does label smoothing help? In *Advances in Neural Information Processing Systems*, pages 4696–4705, 2019. 6

[28] Gaurav Kumar Nayak, Konda Reddy Mopuri, Vaisakh Shaj, R Venkatesh Babu, and Anirban Chakraborty. Zero-shot knowledge distillation in deep networks. *arXiv preprint arXiv:1905.08114*, 2019. 3

[29] 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. 1, 4

[30] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018. 1, 6

[31] 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. 1

[32] Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. Fitnets: Hints for thin deep nets. *arXiv preprint arXiv:1412.6550*, 2014. 1, 2

[33] Kihyuk Sohn, Zizhao Zhang, Chun-Liang Li, Han Zhang, Chen-Yu Lee, and Tomas Pfister. A simple semi-supervised learning framework for object detection. *arXiv preprint arXiv:2005.04757*, 2020. 12

[34] 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

[35] Jack Turner, Elliot J Crowley, Valentin Radu, José Cano, Amos Storkey, and Michael O'Boyle. Hakd: Hardware aware knowledge distillation. *stat*, 1050:24, 2018. 2

[36] Jayakorn Vongkulbhisal, Phongtharin Vinayavekhin, and Marco Visentini-Scarzanella. Unifying heterogeneous classifiers with distillation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3175–3184, 2019. 2

[37] Tao Wang, Li Yuan, Xiaopeng Zhang, and Jiashi Feng. Distilling object detectors with fine-grained feature imitation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4933–4942, 2019. 1, 3, 4, 7

[38] Tongzhou Wang, Jun-Yan Zhu, Antonio Torralba, and Alexei A Efros. Dataset distillation. *arXiv preprint arXiv:1811.10959*, 2018. 1

[39] Qizhe Xie, Minh-Thang Luong, Eduard Hovy, and Quoc V Le. Self-training with noisy student improves imagenet classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10687–10698, 2020. 3

[40] Hongxu Yin, Pavlo Molchanov, Jose M Alvarez, Zhizhong Li, Arun Mallya, Derek Hoiem, Niraj K Jha, and Jan Kautz. Dreaming to distill: Data-free knowledge transfer via deepinversion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8715–8724, 2020. 3

[41] Sergey Zagoruyko and Nikos Komodakis. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. *arXiv preprint arXiv:1612.03928*, 2016. 2

[42] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017. 6