Multi-Task Self-Training for Learning General Representations

Golnaz Ghiasi*, Barret Zoph*, Ekin D. Cubuk*, Quoc V. Le, Tsung-Yi Lin
Google Research, Brain Team
{golnazg,barretzoph,cubuk,qvl,tsungyi}@google.com

Abstract

Despite the fast progress in training specialized models for various tasks, learning a single general model that works well for many tasks is still challenging for computer vision. Here we introduce multi-task self-training (MuST), which harnesses the knowledge in independent specialized teacher models (e.g., ImageNet model on classification) to train a single general student model. Our approach has three steps. First, we train specialized teachers independently on labeled datasets. We then use the specialized teachers to label an unlabeled dataset to create a multi-task pseudo labeled dataset. Finally, the dataset, which now contains pseudo labels from teacher models trained on different datasets/tasks, is then used to train a student model with multi-task learning. We evaluate the feature representations of the student model on 6 vision tasks including image recognition (classification, detection, segmentation) and 3D geometry estimation (depth and surface normal estimation). MuST is scalable with unlabeled or partially labeled datasets and outperforms both specialized supervised models and self-supervised models when training on large scale datasets. Lastly, we show MuST can improve upon already strong checkpoints [24] trained with billions of examples. The results suggest self-training is a promising direction to aggregate labeled and unlabeled training data for learning general feature representations.

1. Introduction

Visual representation learning is a core problem in computer vision. Supervised and self-supervised pre-training have shown promising results in transferring the learned feature representations to downstream tasks. Typically, a model is pre-trained with a supervised [30, 11] or a self-supervised objective [5, 17, 18]. Despite the wide adoption of transfer learning from supervised training, the features may not necessarily be useful for downstream tasks. For example, He *et al*. found that ImageNet pre-training fails to improve COCO instance segmentation [19]. In contrast, Shao *et al*. showed features learned from Objects365 detection dataset improve COCO instance segmentation by a large margin [49]. Pre-training with a specialized task that aligns with the downstream target task still yields the best performance in object detection [34, 49] and semantic segmentation [4].

Intuitively, it is possible to learn general features by training a model to simultaneously do well on multiple tasks. Recent work in NLP started to show promising results on learning a generalist model with multi-task learning [60, 9]. In computer vision, the biggest challenge of training a multi-task model is in the data collection and annotation. Despite datasets like COCO [37], collecting a wide variety of annotations (*e.g*., instance segmentation, person keypoints, image caption) for the same image dataset is quite challenging. Due to the time consuming nature of annotating images with labels, it is hard to scale such efforts with the number of images and the number of tasks. The lack of large scale multi-task datasets impedes the progress in multi-task learning for computer vision.

In this work, we study using self-training to remedy the issue. We propose to use pseudo labeling to enable large scale multi-task feature learning for computer vision. Zoph

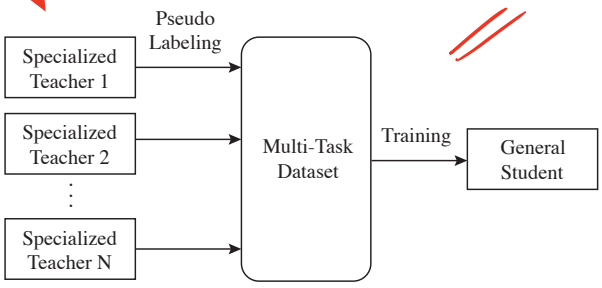

Figure 1. **An overview of Multi-Task Self-Training (MuST).** Specialized Teacher represents a supervised model trained on a single task and dataset (*e.g*., classification model trained on ImageNet). Specialized Teacher models are trained independently on their own tasks and datasets. They then generate pseudo labels on a shared dataset. Finally, a single General Student model is trained jointly using the pseudo (and supervised) labels on the shared dataset.

*Authors contributed equally.

Task to solve

Proposed solution

Problems related to transfer learning

Why multi-task learning is hard in computer vision? What are the challenges?

Can we leverage self-training for multi-task learning?

Why self-training?

et al. [67] observed that self-training further improves pre-training for transfer learning, and that self-training works even when pre-training fails to outperform a randomly initialized model. The gap between pre-training and self-training suggests that self-training can learn better features from pseudo labels. Inspired by this observation, we first investigate whether good features can be learned by only using pseudo labels. We train teacher models using datasets such as COCO or Objects365 to generate pseudo labels on unlabeled images. Figure 2 shows example pseudo labels on ImageNet. Surprisingly, we find a student model trained with only these pseudo labels preserves most of the transfer learning performance of its specialized teacher model. This finding suggests pseudo labels are effective at distilling the knowledge in a supervised dataset. Therefore, we can use pseudo labels to transfer knowledge from multiple teacher models to a single student model for representation learning.

We propose Multi-Task Self-Training (MuST) to train a generalist student model on the information distilled from teacher models trained on different tasks and datasets. Figure 1 shows the overview of the algorithm. MuST has three steps. First, it trains specialized teachers independently on labeled datasets. For example, one teacher can be trained with depth prediction and another teacher can be trained with object detection. The specialized teachers are then used to label a larger unlabeled dataset to create a multi-task pseudo labeled dataset. For example, these teachers can generate depth estimations and object detections on the ImageNet dataset. Finally, the dataset, which now contains pseudo labels from teacher models trained on different datasets/tasks, is used to train a student model with multi-task learning. Hence the student, for example, can do depth prediction and object detection at the same time.

In our experiments, we have four teacher models: classification, semantic segmentation, object box detection, and depth estimation. We design a simple model architecture (Figure 3) based on ResNet [21] and feature pyramid networks (FPN) [36]. The parameters in the ResNet-FPN backbone are shared across different tasks. For each individual task, it has a small task-specific head consisting of a few convolution layers followed by a linear prediction layer. Our experiments show that this simple model architecture is able to absorb the knowledge of different tasks in the shared backbone. The generalist student model is on par with/outperforms its specialist teacher models for all transfer learning tasks.

The recent self-supervised algorithms like SimCLR [5], MoCo [18] are shown to create representations that are on par or better than its supervised counterpart. In our experiments, MuST also outperforms SimCLR [5] by a large margin on segmentation and depth estimation tasks. We also observe that the representations learned by SimCLR is on par with those of supervised learning on ImageNet (1.3M images) but does not scale as well on JFT (300M images). On the contrary, MuST outperforms SimCLR [5] on both ImageNet and JFT. Moreover, MuST also outperforms supervised JFT pre-training for 5 out of 6 tasks except the image classification task. The results indicate the potential of MuST in learning general feature representations that improve with more unlabeled data.

Figure 2. **Examples of pseudo labels on ImageNet. Left:** bounding boxes labeled with an Objects365 teacher model. **Middle:** semantic segmentation labeled with a COCO teacher model. **Right:** depth labeled with a MiDaS teacher model.

Lastly, we show MuST can improve upon already strong checkpoints such as ALIGN [24]. We fine-tune ALIGN checkpoints, previously trained with billions of supervised examples, with MuST pseudo labels and find improvements on a suite of downstream tasks: detection, segmentation, and depth estimation tasks.

We summarize our contributions below:

- We propose Multi-Task Self-Training (MuST), a simple algorithm for creating general visual representations by multi-task learning with pseudo labels.
- We conduct experiments by jointly training across several datasets (*e.g.*, ImageNet, Objects365, COCO, JFT) to learn general feature representations that outperforms representations learned by supervised and self-supervised methods.
- We perform experiments to compare supervised, self-supervised, and MuST on 6 computer vision tasks including tasks in image recognition (classification, detection, segmentation) and 3D geometry estimation (depth and surface normal estimation).
- MuST can be used to improve upon already strong checkpoints and achieve competitive results on a variety of tasks compared to task-specific state-of-the-art models.

2. Related Work

Multi-Task learning: Multi-task learning has a rich history in deep learning [46]. A common strategy for multi-task learning is to share the hidden layers of a "backbone" model for different tasks [2]. More recently, multi-task

Multi-task isn't always the best choice

learning has led to improved accuracy in NLP [9, 38]. Although, Raffel *et al*. found that multi-task learning generally underperformed compared to pre-training followed by fine-tuning [42].

In the vision domain, Zamir *et al*. studied the transfer learning dependencies across 26 tasks with an indoor dataset [64]. Instead of exploring the task dependencies, we are interested in pushing a single model that can absorb knowledge of all tasks for learning general representations. Kokkinos *et al*. [29] and Xiao*et al*. [57] trained models across multiple datasets by simply zeroing losses for examples that don't have labels for a particular task. We propose to apply pseudo labels so every image is annotated with all tasks. Girshick *et al*. used a multi-task loss for classification and bounding-box regression to improve the training of object detectors [15]. We follow the similar approach of using one large backbone model and smaller heads for multiple tasks.

Self-training: Self-training is a popular technique to incorporate unlabeled data into supervised learning [62, 48, 45, 33]. The method works by using a supervised model to generate pseudo labels on unlabeled data. Then a student model is trained on the pseudo labeled data. Yalniz *et al*. [61] showed a model "pre-trained" with pseudo labels on a large unlabeled dataset (at hundreds millions scale) can improve classification accuracy. Noisy Student [58] used self-training to push state-of-the-art performance on ImageNet by training jointly with 130M pseudo labeled images. Chen *et al*. [3] obtained state-of-the-art panoptic segmentation results on Cityscapes with self-training. Zoph *et al*. [67] improved the state-of-the-art on object detection and semantic segmentation with self-training. All the above works focused on a single task. On the contrary, our work focuses on using self-training for multi-task learning to learn general representations.

Representation learning: Transfer learning from ImageNet pre-training has been the most widely used method in computer vision. BiT [30] and ViT [11] pre-trained the model on JFT-300M dataset [51] and obtained strong performance when fine-tuned on downstream vision tasks. In particular, Mahajan *et al*. showed model pre-trained with Instagram benefits other classification tasks but possibly harms localization performance [39]. Li *et al*. found that OpenImagesV4 pre-training [32] outperforms ImageNet pre-training when transferring to object detection and semantic segmentation [34]. Shao *et al*. showed similar findings using the Objects365 dataset [49]. This finding indicates supervised pre-training on a single classification task may not create representations general enough for many kinds of downstream applications.

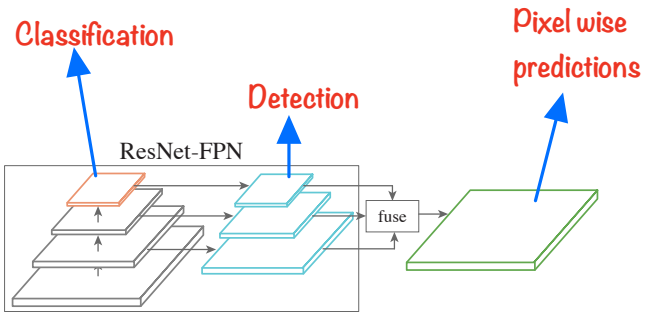

Figure 3. **The ResNet-FPN backbone architecture for multi-task learning.** **Orange:** the top-level features for classification. **Cyan:** multi-scale features for box detection and instance segmentation. **Green:** the high resolution features for pixel-wise tasks (*e.g*., segmentation, depth, and surface normal estimation.)

Self-supervised training is a popular method for representation learning without supervised data [25, 5, 17, 18, 22, 53]. By forcing the representations of an image to agree with each other under data augmentation [1], SimCLR and MoCo trained representations useful for downstream classification tasks [5, 18]. Grill *et al*. proposed the use of online and target neural networks for learning representations, which they evaluated on classification tasks as well as semantic segmentation, object detection, and depth estimation [17]. On the other hand, recent work has demonstrated the limitations of current self-supervised learning methods [41]. They found that aggressive cropping, commonly used in self-supervised learning (such as those used in MoCo [18], PIRL [40], SimCLR [5] etc.), leads to representations that are occlusion invariant, which can be effective for downstream classification tasks. However, these representations are not necessarily invariant to other symmetries of natural images (such as viewpoint invariance), which might be necessary for other downstream tasks such as semantic segmentation [41].

3. Method

3.1. Specialized Teacher Models

We want to learn from a set of teachers that provide rich training signals with their pseudo labels. We adopt four teacher models including four important tasks in computer vision: classification, detection, segmentation, and depth estimation. These tasks require visual understanding of objects and 3D geometry. Examples of the pseudo labels can be found in Figure 2. We train the classification, detection, and segmentation teacher models from scratch on medium/large scale datasets (*e.g*., ImageNet[47], Objects365 [49], COCO [28]). For depth teacher model, we download the pre-trained checkpoint from the open-source repository [44] [1].

Pseudo labeling: We transfer the knowledge in specialized teacher models to unlabeled or partially labeled datasets by pseudo labeling. We follow the practice in [67]

[1] https://github.com/intel-isl/MiDaS

to generate pseudo labels for detection and segmentation. For detection, we use a hard score threshold of 0.5 to generate pseudo box labels. For segmentation, we use a hard score threshold of 0.5 to generate semantic segmentation masks whereas pixels with a smaller prediction score are set to the *ignore* label. For classification, we use soft labels, which contain the probability distribution of all classes, because we find the performance is better than hard labels. For depth, we simply use the predicted depth as pseudo labels without further processing.

3.2. Multi-Task Student Model

Model architecture: Our goal is to train the student with multiple tasks to learn general visual representations. The first thing to design is a model architecture that can share most of the parameters across tasks. We define three task categories: (1) classification, (2) object detection, (3) pixel-wise prediction. The pixel-wise prediction task includes semantic segmentation, depth estimation, and surface normal prediction. Each category of task shares the same feature representations in the backbone model.

We design the backbone model based on ResNet [21] and feature pyramid networks (FPN) [36]. Figure 3 shows the overview of our architecture. We follow the common practice to design the feature representations for classification and detection tasks. We use *C5* feature map (orange) for classification and {*P3, P4, P5, P6, P7*} feature pyramid (cyan) for detection. We follow the practice in [67] to fuse {*P3, P4, P5, P6, P7*} into *P2* feature map (green) for pixel-wise prediction. The fuse operation simply rescales all feature maps into level 2 and sums them (which does not introduce any new parameters).

Each task category shares the same head architecture. The classification head follows the ResNet design. It is a linear prediction layer followed by average pooled *C5* features. The object detection task follows the head architecture in the Mask R-CNN [20]. We use 2 hidden convolution layers for RPN and 4 hidden convolution layers and 1 fully connected layers for Fast R-CNN. The pixel-wise prediction head has 3 convolution layers followed by the $C2$ features before the final linear prediction layer. If the student model learns from multiple tasks in the same task category (*e.g.*, semantic segmentation and depth prediction), each task owns its task specific head without sharing their parameters.

Teacher-student training: We want to study the effectiveness of learning from pseudo labels. Therefore, we design the training of teacher and student models such that the main differences between them are in the dataset and the labels. Unlike model distillation [23] and noisy student [58], we use the same model capacity and data augmentation in both the teacher and the student training. Despite that a teacher can be trained with a more specialized architecture for its own task, we train the teacher and student models using the same architecture shown in Figure 3.

Learning From Multiple Teachers: We propose Multi-Task Self-Training (MuST) to train a student model with multiple teachers. Prior multi-task learning works, which harnessed the information in multiple datasets, mainly focused on the scenario where each example is only labeled with one task or a few tasks [9, 29]. In MuST, every image has supervision for all tasks. The labels may come from supervised or pseudo labels. For example, when training on ImageNet, we can use supervised labels for classification and pseudo labels for detection, segmentation, and depth.

Balancing the loss contribution for each task is an open research area in multi-task learning [26, 6, 7, 63]. The loss of multi-task learning is the weighted sum of the losses from all tasks $L = \sum_i w_i L_i$. The weight w_i decides the loss contribution for the task i. In ImageNet experiments, we adopt $w_i = \frac{b_s lr_{it}}{b_{it} lr_s}$, where b denotes the batch size, lr denotes the learning rate, and the subscript denotes student or teacher model. The equation is derived from the scaling rule in [16], which scales the learning rate propotionally with batch size. The only exception is the depth loss, of which we choose its weight by a parameter sweep. In our experiments on JFT-300M, we use the algorithm in [26] to learn w_i for each task over the course of training.

Cross Dataset Training: MuST has the flexibility to leverage both labeled and unlabeled data. It can scale up the number of images by generating pseudo labels on the unlabeled data. Or it can use images which are partially labeled with one or more tasks. In our experiments, we show an example training across ImageNet, objects365, and COCO datasets. We use supervised labels whenever they are available and generate labels for all absent tasks using pseudo labels.

One challenge in cross dataset training is to balance the data coming from datasets of different sizes. Instead of designing sampling heuristics [9], we uniformly sample from the union of datasets. This works because every task is labeled on every image in MuST, thus we do not need to worry about under/over-sampling a task due to the imbalanced dataset size.

The second main difference compared to other self-training algorithm is that the supervised and pseudo labels are treated equally. We do not batch the examples of supervised and pseudo labels independently and assign them different weights like in [67, 58]. The images are uniformly sampled from the union of datasets and put into one mini-batch. Each example shares the same weight on its loss regardless if the loss is computed against a supervised or a

Training Datasets			Evaluation Datasets		
Name	Task	Num Images	Name	Task	Num Images
ImageNet [47]	Classification	1.2M	CIFAR-100 [31]	Classification	50k
Objects365 [49]	Detection	600k	Pascal [13]	Detection	16.5k
COCO [37]	Segmentation	118k	Pascal [13]	Segmentation	1.5k
MiDaS [44]	Depth	1.9M	NYU V2 [50]	Depth	47k
JFT [51]	Classification	300M	ADE [66]	Segmentation	20k
			DIODE [54]	Surface Normal	17k

Table 1. **Datasets using for MuST and for downstream fine-tuning evaluation**.

pseudo label. This makes MuST significantly simpler to use and to scale with multiple tasks.

3.3. Transfer Learning

To evaluate the representational quality of MuST and other baseline representations, we fine-tune them on a suite of downstream computer visions tasks. We adopt *end-to-end* fine-tuning instead of linear probe to the performance of each fine-tuning task. We fine-tune on CIFAR-100 classification, Pascal detection, Pascal semantic segmentation, NYU depth, ADE semantic segmentation and DIODE surface normal. Also note that all downstream datasets are different than the ones the specialized teacher models are trained on. Furthermore, surface normal prediction is a task that no specialized teacher model was trained for, testing the robustness of representations to held out tasks.

When fine-tuning a representation on a downstream task we sweep over the learning rate and number of training steps (See Appendix for full details). This allows for fair comparison between different representations.

4. Experiments

4.1. Experimental Settings

Training Datasets: Table 1 provides an overview of the datasets we use in the experiments. We experiment with four different datasets and tasks for training our supervised teacher models. These supervised models will then be the ones to generate pseudo labels on unlabeled/partially labeled images.

Evaluation Datasets: Next we describe the datasets that all of our representations will be fine-tuned on. Table 1 provides the list. We have different datasets with a total of five different tasks. Note the Surface Normal task is never used as a training task to test the task generality of the representations.

4.2. Learning with Multi-Task Self-Training

We run experiments to compare our MuST representation learning algorithm to state-of-the-art self-supervised and supervised learning methods.

MuST Improves Pre-training on ImageNet: Table 2 compares the MuST algorithm to self-supervised and supervised learning on ImageNet. On a suite of 6 downstream tasks MuST representations improves over state-of-the-art self-supervised learning and supervised learning on 4 and 5 tasks, respectively. MuST makes use of not only the ImageNet classification labels, but also pseudo labels generated from supervised models trained on Objects365 detection, COCO semantic segmentation, and MiDas depth. This additional information being trained on for ImageNet images leads to much more generalizable feature representations. We observe that self-supervised and supervised pre-training on ImageNet does not learn features that generalize nearly as well to tasks other than image classification.

MuST Improves With More Tasks/Datasets For Learning General Features: The MuST algorithm makes use of pseudo labels generated from independent supervised models trained on different datasets. We next study the importance of having pseudo labels being generated from multiple different teacher models trained on different tasks/datasets. Table 3 shows the representational quality improvement starting from using only supervised ImageNet labels and then adding three different types of pseudo labels obtained from three different datasets. As we continue to add pseudo labels from different tasks/datasets our representations improve in quality. For each new task added we obtain strong improvement across all 6 of our downstream tasks.

Pseudo Labels Preserve Transfer Learning Performance of Teacher Model: We next study how effectively pseudo labels preserve the transfer learning performance of teacher models trained on supervised datasets. To test this we train two supervised teacher models: object detection model on Objects365 and semantic segmentation model on COCO. The first two rows in Table 4 shows their supervised learning performance and their transfer learning performance on 6 downstream tasks. Next we generate pseudo labels on two datasets without labels: ImageNet (1.2M images) and JFT (300M images) . Now we train models from scratch on the pseudo labels on ImageNet and JFT. The next 4 rows

Settings		Transfer Learning Performance					
Method	Epochs	CIFAR-100 Classification	Pascal Detection	Pascal Segm.	NYU Depth	ADE Segm.	DIODE Normal
Self-supervised (SimCLR [5])	800	**87.1**	83.3	72.2	83.7	41.0	**52.8**
ImageNet Supervised	90	85.4	79.3	70.6	81.0	39.8	48.9
+ Multi-task Pseudo Labels	90	86.3 (+0.9)	**85.1** (+5.8)	**80.6** (+10.0)	**87.8** (+6.8)	**43.5** (+3.7)	**52.7** (+3.8)

Table 2. **Multi-Task Self-Training (MuST) outperforms supervised and self-supervised representations on ImageNet**. We compare MuST to state-of-the-art self-supervised and supervised learning using the same pre-training dataset (ImageNet). MuST learns more general features and achieves the best performance on $4/6$ downstream fine-tuning tasks. The performance differences show the impact of different training objectives.

Settings	Transfer Learning Performance					
Method	CIFAR-100 Classification	Pascal Detection	Pascal Segm.	NYU Depth	ADE Segm.	DIODE Normal
ImageNet Supervised	85.4	79.3	70.6	81.0	39.8	48.9
+ Depth Pseudo Labels	84.4(-1.0)	79.3(+0.0)	71.0(+0.4)	86.0(+5.0)	39.5(-0.3)	51.3(+2.4)
+ Depth / Segm. Pseudo Labels	85.3(-0.1)	81.6(+2.3)	78.6(+8.0)	87.2(+6.2)	41.5(+1.7)	52.4(+3.5)
+ Depth / Segm. / Detection Pseudo Labels	**86.3**(+0.9)	**85.1**(+5.8)	**80.6**(+10.0)	**87.8**(+6.8)	**43.5**(+3.7)	**52.7**(+3.8)

Table 3. **Multi-Task Self-Training (MuST) benefits from increasing the number of different pseudo label tasks.** We add depth, segmentation, and detection pseudo labels in addition to supervised ImageNet classification labels and test the representational quality. The results reveal that adding pseudo labels from more tasks leads to more general pre-trained models. All models are trained for 90 epochs on ImageNet.

in Table 4 reveal these results. We observe for both object detection and segmentation pseudo labels we obtain a degradation in the supervised learning quality (e.g. 26.1 vs 20.6/20.7), but that when the representations are *transferred* they obtain similar or better transfer learning performance than the teacher model. Furthermore, the representations obtained by training on JFT over ImageNet typically lead to better transfer learning performance, which reveals the scalability of the MuST method. As we get more and more unlabeled data, our method can easily take advantage of it and the representational quality improves.

Multi-Task Self-Training Across Datasets: MuST utilized pseudo labels generated from teacher models trained on different supervised learning datasets. A natural comparison is then to see how MuST compared against supervised multi-task supervised training where a model is trained on the union of the datasets and labels [29]. Table 5 compares the representational quality of MuST versus supervised multi-task training on three datasets: ImageNet, COCO and Objects365. For multi-task training we sample examples from the datasets with equal probability. Sampling examples with probabilities proportional to the size of the datasets does not work well. Because ImageNet and Objects365 datasets are much larger than COCO dataset and as a result for a batch size of 256 only 15 examples have non zero loss values for segmentation. On the other hand, for MuST *every image has any type of label* and we can sample examples with probabilities proportional to the size of datasets. When comparing the representation qualities MuST obtains the best performance on $6/6$ downstream tasks.

4.3. Scaling Multi-Task Self-Training

One benefit of MuST is that it can scale to unbounded amounts of unlabeled images. To test this hypothesis we move from the ImageNet setup with 1.2M images to JFT with 300M images.

Scaling Dataset Size and Training Iterations: Now instead of generating pseudo labels on 1.2M images, we scale the MuST training to have all three supervised teacher models to generate pseudo labels on 300M images. This process is trivially parallelizable, which makes the overall runtime low compared to training of the models. Table 6 shows the comparison of MuST vs self-supervised learning and supervised learning on the JFT dataset. On $5/6$ downstream tasks MuST outperforms the self-supervised SimCLR algorithm when using the same unlabeled data. We also train a supervised baseline on the multi-class labels available on JFT and find that MuST, using only the unlabeled images, outperforms the representation on $5/6$ downstream tasks. This is quite impressive considering the total sum of supervised images that MuST indirectly makes use of from the pseudo labels is only about 3.7M images compared to the 300M labeled JFT images. Adding JFT supervised labels can fur-

Settings		Performance		Transfer Learning Performance					
Task	Train Dataset	Obj365 Detection	COCO Segm.	CIFAR-100 Classification	Pascal Detection	Pascal Segm.	NYU Depth	ADE Segm.	DIODE Normal
		Teacher Model							
Detection	Objects365	26.1	—	84.0	87.6	78.8	90.1	46.0	55.6
Segmentation	COCO	—	53.8	80.8	82.2	80.2	86.6	42.8	51.0
		Student Model							
Detection	ImageNet	20.6	—	83.2	86.0	78.5	88.5	44.7	55.2
Detection	JFT	20.7	—	85.2	87.7	79.5	89.6	45.4	55.0
Segmentation	ImageNet	—	55.5	82.3	80.5	79.2	86.3	41.8	51.2
Segmentation	JFT	—	49.0	83.1	82.8	78.2	86.6	41.9	51.6

Table 4. **Models trained on supervised data or pseudo labeled data have similar transfer learning performance.** Results comparing how representations transfer if they are trained on supervised data or on pseudo labels that are generated by the supervised model. Pseudo labels effectively compress the knowledge in a supervised dataset. The performance of student models increases with the size of the unlabeled dataset. As the unlabeled dataset size increased, the performance of student model increases. This reveals the scalability of MuST. All student models are trained for the same training iterations (90 ImageNet epochs and 0.36 JFT epochs).

Settings	Transfer Learning Performance					
Method	CIFAR-100 Classification	Pascal Detection	Pascal Segm.	NYU Depth	ADE Segm.	DIODE Normal
Supervised Multi-Task	85.3	85.1	82.1	87.6	43.9	53.4
Supervised Multi-Task + Pseudo Labels	**86.3** (+1.1)	**86.2** (+1.1)	**82.3** (+0.2)	**88.2** (+0.6)	**45.4** (+1.5)	**54.7** (+1.3)

Table 5. **Comparing Multi-Task Training versus Multi-Task Self-Training.** We compare MuST against a baseline of doing supervised multi-task training on the union of all teacher datasets. We use three datasets: ImageNet, COCO and Objects365. Supervised model is jointly trained on the supervised labels of these three datasets. MuST trains jointly on all three supervised and pseudo labels generated by the teacher models. The transfer learning performance gets strong improvements by incorporating pseudo labels into every image.

Settings		Transfer Learning Performance					
Method	Epochs	CIFAR-100 Classification	Pascal Detection	Pascal Segm.	NYU Depth	ADE Segm.	DIODE Normal
Self-Supervised with JFT images (SimCLR [5])	1	85.6	82.4	71.0	83.7	41.4	54.4
Self-Supervised with JFT images (SimCLR [5])	2	85.8	83.7	73.3	84.3	42.2	55.3
Self-Supervised with JFT images (SimCLR [5])	5	86.1	84.1	74.9	84.8	43.0	56.0
JFT supervised	3	87.7	84.6	78.2	86.0	43.4	50.7
JFT supervised	5	88.6	84.9	79.7	86.1	44.3	51.1
JFT supervised	10	**89.6**	85.2	80.4	86.5	45.7	53.1
Multi-Task Pseudo Labels	2.5	87.6	87.8	82.2	**89.8**	47.0	56.2
JFT supervised + Multi-Task Pseudo Labels	2	88.3(+0.5)	**87.9**(+0.1)	**82.9**(+0.7)	89.5(-0.3)	**47.2**(+0.2)	**56.4**(+0.2)

Table 6. **Scaling Multi-Task Self-Training to 300M images.** We repeat the experiments in Table 2 on the JFT dataset (300M images with classification labels). The supervised learning benefits more from the additional images and annotations compared to the self-supervised SimCLR algorithm.

ther improve the performance on image classification and segmentation, showing the flexibility of MuST in using labeled and unlabeled data. Lastly, the student model not only learns general features for transferring, it is also capable of generating high quality predictions for multiple tasks. Figure 4 shows the predictions made by our strongest model.

Bootstrapping from Pre-trained Models. Next we study if MuST can improve upon checkpoints trained with billions of training examples. We use ALIGN checkpoints [24], which are trained with 1.8B image-text pairs, to initialize parameters for training both teacher models and the student model. We use the same teacher model tasks as our previous experiments. The pseudo labels are generated on JFT-300M dataset and the MuST student model is trained

Figure 4. **Visualization of the predictions generated by a multi-task student model.** The MuST student model not only learns general feature representations, but also makes high quality visual predictions with a single model.

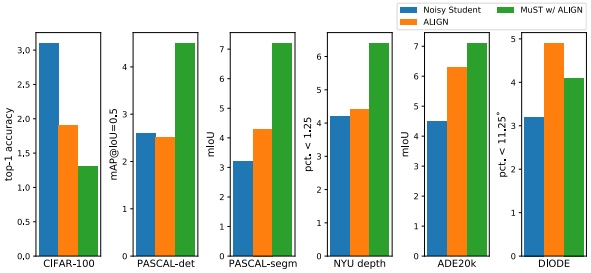

Figure 5. **Relative transfer learning performance gains over the ImageNet pre-trained model [52].** Checkpoints trained with more data or labels typically provide gains on transfer learning to downstream tasks. Fine-tuning the EfficientNet-B7 ALIGN checkpoint with MuST can further improve transfer learning performance for 4/6 downstream tasks.

on JFT for 1 epoch. Figure 5 shows relative transfer learning performance gains of Noisy Student [58], ALIGN [24], and MuST w/ ALIGN compared to the ImageNet checkpoint [52]. The figure shows MuST w/ ALIGN improves the ALIGN checkpoint by respectable margins for 4 out of the 6 downstream tasks. The performances are slightly worse for CIFAR-100 and DIODE surface normal prediction. We repeat the experiments with EfficientNet-L2 architecture and train the student model for 0.36 epoch on JFT. We report 4 downstream tasks showing improvements over the ALIGN checkpoint in Table 7. We find the student model trained with MuST improves the large ALIGN EfficientNet-L2 checkpoint and is competitive to the state-of-the-art models specialized for each dataset and task. Notably, MuST provides checkpoints ready to be fine-tuned for short iterations to achieve state-of-the-art performance while typical self-training methods [67] require pseudo labeling and long training iterations for each downstream task.

5. Discussion

Which pre-training method performs the best with large scaling training? Although self-supervised learning can outperform supervised learning on the ImageNet size dataset (1.3 million images/1k classes), supervised learning is still a better pre-training method on JFT size dataset (300 million images/18k classes). The gap may be compensated by training with more unlabeled data for self-supervised learning. However, self-training can also expand one or multiple supervised models by generating pseudo labels on unlabeled data. Overall, both self-supervised and self-training are able to scale, but at the moment self-training presents better performance in learning general features. A promising direction is to combine self-supervised and self-training for representation learning [65, 12, 59].

Settings	Transfer Learning Performance			
Method	Pascal Detection	Pascal Segm.	NYU Depth	ADE Segm.
Previous SoTA	**89.3** [14]	**90.0** [67]	90.4 [43]	**54.1** [8]
ALIGN [24]	86.2	86.6	91.1	54.0
MuST w/ [24]	88.2	**89.8**	**91.9**	**54.3**

Table 7. **MuST checkpoints are versatile and achieve competitive performance compared to state-of-the-art models.** MuST improves the transfer learning performance of the ALIGN EfficientNet-L2 checkpoint on these four downstream tasks.

Why use MuST over self-supervised learning? Both of the methods are scalable with the unlabeled training data, however, MuST can easily combine together all labeled and unlabeled data. However, self-supervised learning relies on the generalization from the pre-text task to downstream tasks, which does not always give good performance. It is easier to design pseudo labels if the downstream tasks of interest are known in advance. MuST also generalizes to unseen tasks (*e.g.*, surface normal prediction) given the tasks of the teacher model in this paper.

6. Conclusion

In this paper, we present MuST, a scalable multi-task self-training method for learning general representations. We compare with supervised and self-supervised learning approaches on ImageNet and JFT and evaluate on 6 datasets including visual recognition, localization, and 3D geometry prediction. We show that MuST outperforms or is on par with supervised and self-supervised learning on 5 out of 6 transfer learning tasks, except the classification task. Moreover, MuST can improve upon already strong checkpoints trained with billions of examples. The results show multi-task self-training is a scalable pre-training method and is able to learn general feature representations. We hope this work will encourage further research towards creating universal visual representations.

Acknowledgements

We would like to thank Yin Cui, Aravind Srinivas, Simon Kornblith, and Ting Chen for valuable feedback.

A. Details of Training and Evaluation Datasets

A.1. Training Datasets

In this section, we describe 5 datasets we use to train teacher models.

ImageNet: ImageNet [47] is a classification dataset with 1.2M training images and 1000 unique classes. All of its images are center cropped and have one primary object per image.

Objects365: Objects365 [49] is an object detection dataset that has 365 different classes and 600k training images.

COCO: The COCO dataset [37] contains 118k images that has a variety of different labels (e.g. object detection, instance segmentation, panoptic segmentation). For all experiments we use its pantopic segmentation labels.

MiDaS: The MiDaS depth model [44] that is used for generating our depth pseudo labels is trained on a diverse set of 5 depth datasets. The 5 depth datasets are DIML Indoor [27] (220k images), MegaDepth [35] (130k images), ReDWeb [56] (3600), WSVD [55] (1.5M), and 3D movies (75k). The model is trained to be invariant to the depth range and scale across all datasets, leading to a model that generates robust pseudo labels.

JFT: JFT [51] is a large-scale image multi-label classification dataset with 300M labeled images. This dataset is used to test the scale of MuST and various self-supervised learning algorithms.

A.2. Evaluation Datasets

Next we describe the datasets that all of our representations will be fine-tuned on. We have different datasets with a total of five different tasks. Note the Surface Normal task is never used as a training task to test the task generality of the representations.

CIFAR-100: CIFAR-100 is a classification dataset with 50k images and 100 unique classes.

PASCAL Detection: The Pascal Detection dataset [13] is an object detection dataset with 20 unique classes. We train the models on the `trainval` sets of PASCAL VOC 2007 and PASCAL VOC 2012 which include 16.5k images.

PASCAL Segmentation: The Pascal Segmentation dataset [13] is a semantic segmentation dataset with 20 unique classes. We train the models on the `train` set of the PASCAL VOC 2012 segmentation dataset which has 1.5k images.

NYU Depth V2: The NYU Depth v2 dataset [50] is a depth estimation dataset that contains 47584 train images and 654 validation images.

ADE Segmentation: ADE20k [66] is a segmentation dataset that contains 20k images with 150 object and stuff classes. The dataset contains a wide variety of different indoor and outdoor scenes along with object classes.

DIODE Surface Normal: The DIODE dataset [54] is a depth and surface normal dataset that contains 16884 images. The dataset contains a diverse set of both indoor and outdoor scenes for training and testing. We only make use of the surface normal labels.

B. Implementation Details

B.1. Training Teacher Models

In this section, we introduce the details of training teacher models, which are used to generate pseudo labels in MuST. All the models are trained with a ResNet-152 backbone model.

Objects365 Detection: We use batch size 256 and train for 140 epochs. The image size is 640. We apply scale jittering [0.5, 2.0] (i.e. randomly resample image between 320×320 to 1280×1280 and crop it to 512×512). The learning rate is 0.32 and the weight decay is set as 4e-5. The model is trained from random initialization. The final performance is 26.1 AP.

COCO Segmentation: We use the annotations in COCO panoptic segmentation dataset [28]. We train a semantic segmentation model that only predicts the semantic class for each pixel, instead of also predicting the object instance. We use batch size 256 and train for 384 epochs. The image size is 896. We apply scale jittering [0.5, 2.0]. The learning rate is 0.32 and the weight decay is set as 4e-5. The model is trained from random initialization. The final performance is 53.8 mIoU.

MiDaS Depth: We directly download the pre-trained MiDaS from the github repository and use it as a teacher model to generate pseudo labels.

ImageNet Classification: We use batch size 2048 and train for 400 epochs. The image size is 224. The learning rate is 0.8 and weight decay is 4e-5. We apply random augmentation [10] (2L-15M, 2 layers with magnitude 15) and label smoothing (0.1) to regularize the model training. The final performance is 81.6 top-1 accuracy.

B.2. Training Multi-Task Student Models

We use a batch size 256 for training student models in our experiments. The image size is 640. We apply scale jittering [0.5, 2.0] during training. The weight decay is 4e-5 in ImageNet experiments and 3e-6 in JFT experiments. No random augmentation [10] or label smoothing is applied.

B.3. Fine-tuning on Evaluation Datasets

For fine-tuning we initialize the parameters in the ResNet and FPN backbone with a pre-trained model and randomly initialize the rest of the layers. We perform *end-to-end* fine-tuning with an extensive grid search of the combinations of learning rate and training steps to ensure each pre-trained model achieves its best fine-tuning performance. We experiment with different weight decays but do not find it making a big difference and set it to 1e-4. All models are trained with cosine learning rate for simplicity. Below we describe the dataset, evaluation metric, model architecture, and training parameters for each task.

CIFAR-100: We use standard CIFAR-100 train and test sets and report the top-1 accuracy. We resize the image resolution to 256×256. We replace the classification head in the pre-trained model with a randomly initialized linear layer that predicts 101 classes, including background. We use a batch size of 512 and search the combination of training steps from 5000 to 20000 and learning rates from 0.005 to 0.32. We find the best learning rate for SimCLR (0.16) is much higher than the supervised model (0.005). This trend holds for the following tasks.

PASCAL Segmentation: We use PASCAL VOC 2012 train and validation sets and report the mIoU metric. The training images are re-sampled into 512×512 with scale jittering [0.5, 2.0]. We initialize the model from the pre-trained backbone and FPN [36] layers. We remove the pre-trained segmentation head and train from a randomly initialized head. We use a batch size of 64 and search the combination of training steps from 5000 to 20000 and learning rates from 0.005 to 0.32.

PASCAL Detection: We use PASCAL VOC 2007+2012 trainval set and VOC 2007 test set and report the AP_{50} with 11 recall points to compute average precision. The training images are resampled into 896 with scale jittering [0.5, 2.0]. we initialize the model from the pre-trained backbone and FPN [36] layers and randomly initialize the heads. We use a batch size of 32 and search the combination of training steps from 5000 to 20000 and learning rates from 0.005 to 0.32.

NYU Depth: We use NYU depth v2 dataset with 47584 train and 654 validation images. We report the percentage of predicted depth values within 1.25 relative ratio compared to the ground truth. The training images are resampled into 640 with scale jittering [0.5, 2.0]. we initialize the model from the pre-trained backbone and FPN [36] layers and randomly initialize the heads. We use a batch size of 64 and search the combination of training steps from 10000 to 40000 and learning rates from 0.005 to 0.32.

DIODE: We use DIODE outdoor dataset with 16884 train and 446 validation images. We report the percentage of the angle error less than 11.25°. We use the original image resolution 768 for training and evaluation. The training image is applied with scale jittering [0.5, 2.0]. we initialize the model from the pre-trained backbone and FPN [36] layers and randomly initialize the heads. We use a batch size of 32 and search the combination of training steps from 20000 to 80000 and learning rates from 0.01 to 0.16.

C. Visualization of Student Model Predictions

Figure 6 shows more visual examples of the predictions made by a single multi-task student model. The images are sampled from the validation set in ImageNet dataset.

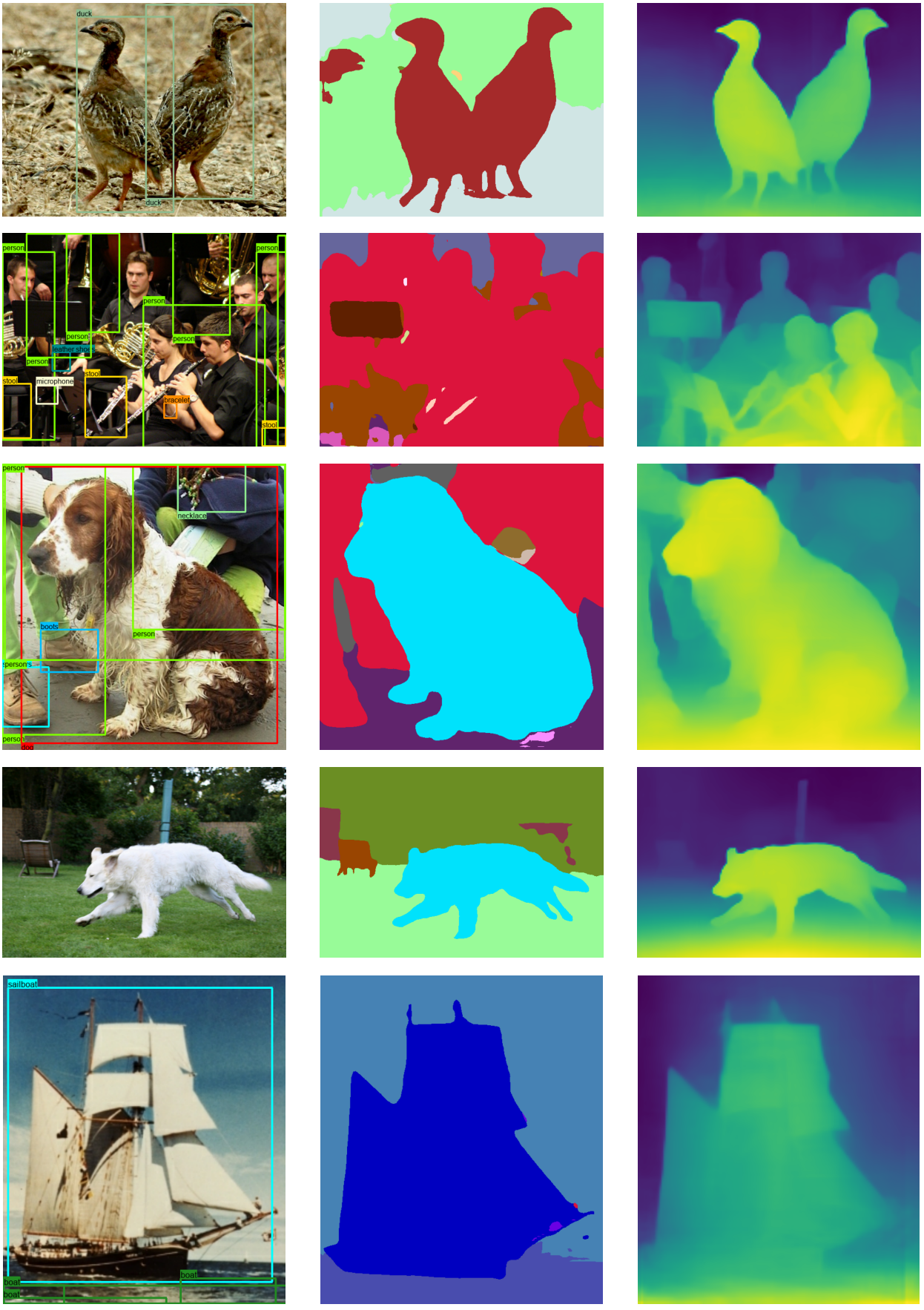

Figure 6. The visualization of inference on ImageNet dataset made by single MuST student model.

References

[1] Suzanna Becker and Geoffrey E Hinton. Self-organizing neural network that discovers surfaces in random-dot stereograms. *Nature*, 355(6356):161–163, 1992. 3

[2] Rich Caruana. Multitask learning. *Machine learning*, 28(1):41–75, 1997. 2

[3] Liang-Chieh Chen, Raphael Gontijo Lopes, Bowen Cheng, Maxwell D Collins, Ekin D Cubuk, Barret Zoph, Hartwig Adam, and Jonathon Shlens. Naive-student: Leveraging semi-supervised learning in video sequences for urban scene segmentation. In *ECCV*, 2020. 3

[4] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848, 2017. 1

[5] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *ICML*, 2020. 1, 2, 3, 6, 7

[6] Zhao Chen, Vijay Badrinarayanan, Chen-Yu Lee, and Andrew Rabinovich. GradNorm: Gradient normalization for adaptive loss balancing in deep multitask networks. *JMLR*, 2018. 4

[7] Zhao Chen, Jiquan Ngiam, Yanping Huang, Thang Luong, Henrik Kretzschmar, Yuning Chai, and Dragomir Anguelov. Just pick a sign: Optimizing deep multitask models with gradient sign dropout. In *NeurIPS*, 2020. 4

[8] Bowen Cheng, Alexander G. Schwing, and Alexander Kirillov. Per-pixel classification is not all you need for semantic segmentation. *CoRR*, abs/2107.06278, 2021. 8

[9] Kevin Clark, Minh-Thang Luong, Urvashi Khandelwal, Christopher D. Manning, and Quoc V. Le. Bam! born-again multi-task networks for natural language understanding. In *ACL*, 2019. 1, 3, 4

[10] Ekin D. Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V. Le. Randaugment: Practical automated data augmentation with a reduced search space, 2019. 10

[11] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale, 2020. 1, 3

[12] Jingfei Du, Edouard Grave, Beliz Gunel, Vishrav Chaudhary, Onur Celebi, Michael Auli, Ves Stoyanov, and Alexis Conneau. Self-training improves pre-training for natural language understanding, 2020. 8

[13] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *IJCV*, 2010. 5, 9

[14] Golnaz Ghiasi, Yin Cui, Aravind Srinivas, Rui Qian, Tsung-Yi Lin, Ekin D. Cubuk, Quoc V. Le, and Barret Zoph. Simple copy-paste is a strong data augmentation method for instance segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2918–2928, June 2021. 8

[15] Ross Girshick. Fast r-cnn. In *ICCV*, 2015. 3

[16] Priya Goyal, Piotr Dollár, Ross B. Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. Accurate, large minibatch SGD: training imagenet in 1 hour. *arXiv preprint arXiv:1706.02677*, 2017. 4

[17] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H. Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Koray Kavukcuoglu, Rémi Munos, and Michal Valko. Bootstrap your own latent: A new approach to self-supervised learning. In *NeurIPS*, 2020. 1, 3

[18] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *CVPR*, 2020. 1, 2, 3

[19] Kaiming He, Ross Girshick, and Piotr Dollár. Rethinking imagenet pre-training. In *ICCV*, 2019. 1

[20] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *ICCV*, 2017. 4

[21] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 2, 4

[22] Olivier J Hénaff, Aravind Srinivas, Jeffrey De Fauw, Ali Razavi, Carl Doersch, SM Eslami, and Aaron van den Oord. Data-efficient image recognition with contrastive predictive coding. *arXiv preprint arXiv:1905.09272*, 2019. 3

[23] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015. 4

[24] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc V. Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. *CoRR*, abs/2102.05918, 2021. 1, 2, 7, 8

[25] 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

[26] Alex Kendall, Yarin Gal, and Roberto Cipolla. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 4

[27] Youngjung Kim, Hyungjoo Jung, Dongbo Min, and Kwanghoon Sohn. Deep monocular depth estimation via integration of global and local predictions. *IEEE transactions on Image Processing*, 27(8):4131–4144, 2018. 9

[28] Alexander Kirillov, Kaiming He, Ross Girshick, Carsten Rother, and Piotr Dollar. Panoptic segmentation. In *CVPR*, June 2019. 3, 9

[29] Iasonas Kokkinos. Ubernet: Training a universal convolutional neural network for low-, mid-, and high-level vision using diverse datasets and limited memory. In *CVPR*, 2017. 3, 4, 6

[30] Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Joan Puigcerver, Jessica Yung, Sylvain Gelly, and Neil Houlsby. Big transfer (bit): General visual representation learning. In *ECCV*, 2020. 1, 3

[31] Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, 2009. 5

[32] Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Alexander Kolesnikov, Tom Duerig, and Vittorio Ferrari. The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale. *IJCV*, 2020. 3

[33] Dong-Hyun Lee et al. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on challenges in representation learning, ICML*, 2013. 3

[34] Hengduo Li, Bharat Singh, Mahyar Najibi, Zuxuan Wu, and Larry S. Davis. An analysis of pre-training on object detection. *CoRR*, abs/1904.05871, 2019. 1, 3

[35] Zhengqi Li and Noah Snavely. Megadepth: Learning single-view depth prediction from internet photos. In *CVPR*, 2018. 9

[36] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *CVPR*, 2017. 2, 4, 10

[37] 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 *ECCV*, 2014. 1, 5, 9

[38] Xiaodong Liu, Pengcheng He, Weizhu Chen, and Jianfeng Gao. Improving multi-task deep neural networks via knowledge distillation for natural language understanding. In *ACL*, 2019. 3

[39] Dhruv Mahajan, Ross Girshick, Vignesh Ramanathan, Kaiming He, Manohar Paluri, Yixuan Li, Ashwin Bharambe, and Laurens van der Maaten. Exploring the limits of weakly supervised pretraining. In *ECCV*, September 2018. 3

[40] Ishan Misra and Laurens van der Maaten. Self-supervised learning of pretext-invariant representations. In *CVPR*, 2020. 3

[41] Senthil Purushwalkam and Abhinav Gupta. Demystifying contrastive self-supervised learning: Invariances, augmentations and dataset biases. *arXiv preprint arXiv:2007.13916*, 2020. 3

[42] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *JMLR*, 2020. 3

[43] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. *ArXiv preprint*, 2021. 8

[44] René Ranftl, Katrin Lasinger, David Hafner, Konrad Schindler, and Vladlen Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2020. 3, 5, 9

[45] Chuck Rosenberg, Martial Hebert, and Henry Schneiderman. Semi-supervised self-training of object detection models. In *IEEE Workshops on Applications of Computer Vision (WACV/MOTION'05)*, 2005. 3

[46] Sebastian Ruder. An overview of multi-task learning in deep neural networks. *arXiv preprint arXiv:1706.05098*, 2017. 2

[47] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *IJCV*, 2015. 3, 5, 9

[48] H Scudder. Probability of error of some adaptive pattern-recognition machines. *IEEE Transactions on Information Theory*, 11(3):363–371, 1965. 3

[49] Shuai Shao, Zeming Li, Tianyuan Zhang, Chao Peng, Gang Yu, Xiangyu Zhang, Jing Li, and Jian Sun. Objects365: A large-scale, high-quality dataset for object detection. In *ICCV*, 2019. 1, 3, 5, 9

[50] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *ECCV*, 2012. 5, 9

[51] Chen Sun, Abhinav Shrivastava, Saurabh Singh, and Abhinav Gupta. Revisiting unreasonable effectiveness of data in deep learning era. In *ICCV*, 2017. 3, 5, 9

[52] Mingxing Tan and Quoc Le. EfficientNet: Rethinking model scaling for convolutional neural networks. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 6105–6114. PMLR, 09–15 Jun 2019. 8

[53] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive multiview coding. In *ECCV*, 2019. 3

[54] Igor Vasiljevic, Nick Kolkin, Shanyi Zhang, Ruotian Luo, Haochen Wang, Falcon Z Dai, Andrea F Daniele, Mohammadreza Mostajabi, Steven Basart, Matthew R Walter, et al. Diode: A dense indoor and outdoor depth dataset. *arXiv preprint arXiv:1908.00463*, 2019. 5, 9

[55] Chaoyang Wang, Simon Lucey, Federico Perazzi, and Oliver Wang. Web stereo video supervision for depth prediction from dynamic scenes. In *2019 International Conference on 3D Vision (3DV)*, pages 348–357. IEEE, 2019. 9

[56] Ke Xian, Chunhua Shen, Zhiguo Cao, Hao Lu, Yang Xiao, Ruibo Li, and Zhenbo Luo. Monocular relative depth perception with web stereo data supervision. In *CVPR*, 2018. 9

[57] Tete Xiao, Yingcheng Liu, Bolei Zhou, Yuning Jiang, and Jian Sun. Unified perceptual parsing for scene understanding. In *European Conference on Computer Vision*. Springer, 2018. 3

[58] Qizhe Xie, Eduard Hovy, Minh-Thang Luong, and Quoc V Le. Self-training with noisy student improves imagenet classification. In *CVPR*, 2020. 3, 4, 8

[59] Qiantong Xu, Alexei Baevski, Tatiana Likhomanenko, Paden Tomasello, Alexis Conneau, Ronan Collobert, Gabriel Synnaeve, and Michael Auli. Self-training and pre-training are complementary for speech recognition, 2020. 8

[60] Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. mt5: A massively multilingual pre-trained text-to-text transformer. In *NAACL*, 2020. 1

[61] I. Zeki Yalniz, Hervé Jégou, Kan Chen, Manohar Paluri, and Dhruv Mahajan. Billion-scale semi-supervised learning for image classification. *CoRR*, abs/1905.00546, 2019. 3

[62] David Yarowsky. Unsupervised word sense disambiguation rivaling supervised methods. In *ACL*, 1995. 3

[63] Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. Gradient surgery for multi-task learning. *arXiv preprint arXiv:2001.06782*, 2020. 4

[64] Amir R. Zamir, Alexander Sax, William Shen, Leonidas J. Guibas, Jitendra Malik, and Silvio Savarese. Taskonomy: Disentangling task transfer learning. In *CVPR*, 2018. 3

[65] Yu Zhang, James Qin, Daniel S. Park, Wei Han, Chung-Cheng Chiu, Ruoming Pang, Quoc V. Le, and Yonghui Wu. Pushing the limits of semi-supervised learning for automatic speech recognition, 2020. 8

[66] Bolei Zhou, Hang Zhao, Xavier Puig, Tete Xiao, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Semantic understanding of scenes through the ade20k dataset. *IJCV*, 127(3):302–321, 2019. 5, 9

[67] Barret Zoph, Golnaz Ghiasi, Tsung-Yi Lin, Yin Cui, Hanxiao Liu, Ekin D. Cubuk, and Quoc V. Le. Rethinking pre-training and self-training. In *NeurIPS*, 2020. 2, 3, 4, 8

SUMMARY

What is it about?

- One of the challenging problems in computer vision is learning a single model that works for a number of tasks. For example, developing a model that can do classification, detection, segmentation, etc at the same time
- To tackle the above problem, the authors proposed a new solution, multi-task self-training (MuST), a methodology to address the above limitation
- A natural question that comes to mind: What makes multi-task learning so challenging in Computer Vision? Annotations! Annotating a dataset that too for different tasks is utterly laborious and a time consuming task. This makes it hard to scale this effort with the number of images and the tasks, hence becoming a major blocker for multi-task learning in vision
- The method proposed by the authors to tackle this problem outperforms(max cases) supervised, self-supervised on six different tasks

Why self-training?

- Self-training is a popular technique to incorporate unlabelled data into supervised learning. First, a model is trained on the labeled data in a supervised fashion. Then the model is to used to generate pseudo labels for the unlabelled dataset. Then another model is trained by combining both the labeled dataset as well as the pseudo labeled dataset. This is very cheap to implement and provides huge gains in the final model
- One of the papers from Google Brain last year proved that self-training not only helps to improve pre-training and transfer learning but also works when pre-training fails to outperform a randomly initialized model.
- This gap between self-training and pre-training suggests that self-training can learn better features from pseudo labels.
- Inspired by this, the authors tried to do multi-task learning using only pseudo labels. The pseudo labels were generated by the teacher models trained on different labeled datasets. Surprisingly, a student model trained on these pseudo labels only preserves most of the performance of its counterpart teacher model.
- The above finding suggests pseudo labels help in knowledge distillation. Hence we can train multiple teachers using pseudo labels and try distilling those features into a single student model.

Algorithm Overview

The algorithm consists of three main steps:

- It trains specialized teacher models independently on labeled datasets, for example, one teacher model trained for classification, another for object detection, etc.
- The specialized teacher models are then used to generate pseudo labels to create a multi-task pseudo labeled dataset. For example, we can combine the pseudo labels for object detection and segmentation for the ImageNet dataset.
- A final student model is trained on the above curated multi-task pseudo labeled dataset with multi-task learning.

Specialized Teacher Models

- As stated earlier, each one of the teacher models is trained on a labeled dataset in a supervised fashion. The authors focus on four different tasks in this case: Classification, Detection, Segmentation, and Depth Estimation. Why these tasks? Because these are the four elements to the visual understanding of the objects and the 3D geometry.
- For generating the pseudo labels using these teacher models, the selection process is the same. For example, a threshold of 0.5 to generate pseudo box labels and pseudo semantic segmentation masks. For classification tasks, soft labels are used instead of hard labels. And for depth prediction, the labels generated are used as it is without any further processing

Multi-task Student Model

- Once we got the pseudo labels, all we have to do is to train a student model in a multi-task traning setting on the multi-task curated dataset
- The student model has the same architecture as the teacher models. The authors uses ResNet based backbone with a feature pyramid network. The features at different levels in this FPN is used for different tasks. C5 feature map is used for classification, P3-P7 for detection and P3-P7 fused into P2 for pixel wise predictions (segmentation as well as depth prediction)
- The only difference between the student and the teacher models is the dataset and the kind of labels used for training them

One of the best things about curating a dataset using pseudo labels in this way is that you don't have to worry about sampling the dataset in a multi-task training setting.

Also, the authors use end-to-end fine tuning during transfer learning settings for downstream tasks. For training details like hparams used, etc. please refer to the Experiments section above