

# Factors of Influence for Transfer Learning across Diverse Appearance Domains and Task Types

Thomas Mensink\*, Jasper Uijlings\*, Alina Kuznetsova, Michael Gygli, and Vittorio Ferrari

**Abstract**—Transfer learning enables to re-use knowledge learned on a source task to help learning a target task. A simple form of transfer learning is common in current state-of-the-art computer vision models, i.e. pre-training a model for image classification on the ILSVRC dataset, and then fine-tune on any target task. However, previous systematic studies of transfer learning have been limited and the circumstances in which it is expected to work are not fully understood. In this paper we carry out an extensive experimental exploration of transfer learning across vastly different image domains (consumer photos, autonomous driving, aerial imagery, underwater, indoor scenes, synthetic, close-ups) and task types (semantic segmentation, object detection, depth estimation, keypoint detection). Importantly, these are all complex, structured output tasks types relevant to modern computer vision applications. In total we carry out over 2000 transfer learning experiments, including many where the source and target come from different image domains, task types, or both. We systematically analyze these experiments to understand the impact of image domain, task type, and dataset size on transfer learning performance. Our study leads to several insights and concrete recommendations: (1) for most tasks there exists a source which significantly outperforms ILSVRC’12 pre-training; (2) the image domain is the most important factor for achieving positive transfer; (3) the source dataset should *include* the image domain of the target dataset to achieve best results; (4) at the same time, we observe only small negative effects when the image domain of the source task is much broader than that of the target; (5) transfer across task types can be beneficial, but its success is heavily dependent on both the source and target task types.

**Index Terms**—Transfer Learning, Computer Vision.

## 1 INTRODUCTION

Transfer learning is omnipresent in computer vision. The common practice is transfer learning through ILSVRC’12 pre-training: train on the ILSVRC’12 image classification task [4], copy the resulting weights to a target model, then fine-tune for the target task at hand. This strategy was shown to be effective on a wide variety of datasets and task types, including image classification [7], [20], [42], [47], [60], object detection [39], semantic segmentation [95], human pose estimation [43], [128], [131], and depth estimation [19], [33]. Intuitively, the reason for this success is that the network learns a strong generic visual representation, providing a better starting point for learning a new task than training from scratch.

But can we do better than ILSVRC’12 pre-training? And what factors make a source task good for transferring to a given target task? Some previous works aim to demonstrate that a generic representation trained on a single large source dataset works well for a variety of classification tasks [59], [76], [127]. Others instead try to automatically find a subset of the source dataset that transfers best to a given target [37], [79], [82], [122]. By virtue of dataset collection suites such as VTAB [127] and VisDA [81], recently several works in transfer learning and domain adaptation experiment with target datasets spanning a variety of image domains [26], [59], [62], [76], [81], [127], [129]. However, most previous work focuses solely on image classification [26], [37], [62], [71], [82], [92],

[127] or a single structured prediction task [45], [67], [106], [108], [122].

In this paper, we go beyond previous works by providing a large-scale exploration of transfer learning across a wide variety of image domains and task types. In particular, we perform over 2000 transfer learning experiments across 20 datasets spanning seven diverse image domains (consumer, driving, aerial, underwater, indoor, synthetic, close-ups) and four task types (semantic segmentation, object detection, depth estimation, keypoint detection). In many of our experiments the source and target come from different image domains, task types, or both. For example, we find that semantic segmentation on COCO [70] is a good source for depth estimation on SUN RGB-D [99] (Tab. 4e); and even that keypoint detection on Stanford Dogs [8], [53] helps object detection on the Underwater Trash [34] dataset (Tab. 5b). We then do a systematic meta-analysis of these experiments, relating transfer learning performance to three underlying factors of variation: the difference in image domain between source and target tasks, their difference in task type, and the size of their training sets. This yields new insights into when transfer learning brings benefits and which source works best for a given target.

At a high level, our main conclusions are: (1) for most target tasks we are able to find sources that significantly outperform ILSVRC’12 pre-training; (2) the image domain is the most important factor for achieving positive transfer; (3) the source dataset should *include* the image domain of the target dataset to achieve best results; (4) at the same time, we observe only small negative effects when the image domain of the source task is much broader than that of the target; (5) transfer across task types can be beneficial, but its success is heavily dependent on both the source and target task types.

- Google Research
- Primary contacts: [mensink@google.com](mailto:mensink@google.com) and [jrru@google.com](mailto:jrru@google.com).

\* Equal contribution.

Manuscript submitted in March, 2021, revision submitted in September, 2021, and accepted in November 2021 for future publication in TPAMI. Copyright may be transferred without notice, after which this version may no longer be accessible.

The rest of our paper is organized as follows: [Sect. 2](#) discusses related work. [Sect. 3](#) discusses the three factors of variation. [Sect. 4](#) details and validates the network architectures that we use. [Sect. 5](#) presents our transfer learning experiments. [Sect. 6](#) presents a detailed analysis of our results. [Sect. 8](#) concludes our paper.

## 2 RELATED WORK

We review here related work on transfer learning in computer vision. We take a rather broad definition of this term, and include several families of works that transfer knowledge from a task to another, even though they are not explicitly positioned as ‘transfer learning’. We pay particular attention to their experimental setting and analyses, as this is the aspect most related to our work.

**Domain adaptation.** This family of works adapts image classifiers from a source domain to a target domain, typically containing the same classes but appearing in different kind of images. Some of the most common techniques are minimizing the discrepancy in feature distributions [36], [71], [92], [101], [107], [110], [129], embedding domain alignment layers into a network [13], [73], [89], directly transform images from the target to the source domain [9], [91], [93]. Earlier works consider relatively small domain shifts, e.g. where the source and target datasets are captured by different cameras [92]. Recent works explored larger domain shifts, e.g. across clipart, web-shopping products, consumer photos and artist paintings [110], or across simple synthetic renderings and real consumer photos [62], [81], and even fusing multiple datasets spanning several modalities such as hand-drawn sketches, synthetic renderings and consumer photos [129]. Despite this substantial variation in domain, most works consider only object classification tasks, with relatively few papers tackling semantic segmentation [45], [67], [81], [103], [106], [124].

**Few-shot learning.** Works in this area aim to learn to classify new target classes from very few examples, typically 1-10, by transferring knowledge from source classes with many training samples. Typically the source and target classes come from the same dataset, hence there is no domain shift. There are two main approaches: metric-based and optimization-based. Optimization-based methods employ meta-learning [32], [80], [84], [85], [117]. These methods train a model such that it can be later adapted to new classes with few update steps. MAML [32] is the most prominent example of this line of work. Despite its success, [84] later showed that its performance largely stems from learning a generic feature embedding, rather than a model that can adapt to new data faster. Metric-based methods [57], [74], [98], [104], [111] aim at learning an embedding space which allows to classify examples of any class using a distance-based classifier, e.g. nearest neighbor [98]. Overall, the community seems to be reaching a consensus [17], [38], [46], [83], [100], [105]: the key ingredient to high-performing few-shot classification is learning a general representation, rather than sophisticated algorithms for adapting to the new classes. In line with these works, we study what representation is suitable for solving a target task. In contrast with these works, our focus is on more complex structured prediction.

While most few-shot learning works focus on classification, there are a few exceptions, e.g. for object detection [113], [114], [118] and semantic segmentation [94]. These works follow a similar setup with source and targets coming from the same dataset. We also focus on structured prediction tasks, however, we tackle more realistic scenarios, where a model is adapted to new datasets, appearance domains, and task types.

**Transfer learning.** A common practice in computer vision is to start from a neural network trained for image classification on ILSVRC’12 [4] as a generic source ([Sect. 1](#)). Because of its success, this strategy is the starting point in all our experiments, and should be seen as the baseline to beat for any transfer learning method.

While many works simply apply this strategy as a practical ‘trick of the trade’, several papers explore transfer learning in more depth, attempting to understand when it works, to find even better source datasets than ILSVRC’12, or to propose more sophisticated transfer techniques. The typical setting considers a very large source dataset: besides ILSVRC’12 [26], [37], [59], [76], [79], also ImageNet21k with 9M images [26], [59], [76], [82], [127], Open Images with 1.7M images [122], Places-205 with 2.5M images [37], and even JFT-300M with 300M images [26], [59], [76], [79], [82]. Several works [26], [59], [76], [127] report experiments on target datasets spanning different image domains, especially since the advent of the VTAB suite [127] which assembles datasets captured using a standard camera, as well as remote sensing, medical, and synthetic ones. Yet, each paper considers little variation in the source dataset, experimenting with 1-3 sources overall, and sometimes picking just one for each target dataset [37]. More importantly, the vast majority of reported experiments are on image classification, for both source and target datasets. Note how VTAB downgrades some structured tasks to simpler versions that can be expressed as classification, e.g. predict the depth of the closest object to the camera, as opposed to a depth value for each object or pixel.

In terms of method, most works follow the classical protocol of learning a feature representation from the source dataset, then fine-tune with a new task head on each target dataset in turn. However, they differ in how they select source images for a given target. Some aim to demonstrate that a single generic representation trained on the *whole* large source dataset works well for all target tasks [59], [127]. Others instead try to automatically find a subset of the source dataset that transfers best to a given target [37], [79], [82], [122]. The subsets correspond either to subtrees of the source class hierarchy [79], [82], or are derived by clustering source images by appearance [122]. After selecting a subset, they retrieve its corresponding pre-trained source model and fine-tune only that one on the target dataset.

Taskonomy [126] explores a converse scenario. They perform transfer learning across many task types, including several dense prediction ones, e.g. surface normal estimation and semantic segmentation. However, there is only one image domain (indoor scenes) as all the tasks are defined on the same images (one dataset). This convenient setting enables them to study which source task type helps the most which target task type, by exhaustively trying out all pairs. As an powerful outcome, they derive a taxonomy of task





Fig. 1: We explore transfer learning across a wide variety of image domain and task types. We show here example images for the 20 datasets we consider, highlighting their visual diversity. We grouped them into manually defined image domains: consumer in orange, driving in green, synthetic in red, aerial in purple, underwater in blue, close-ups in yellow, and indoor in magenta.

types, with directed edges weighted by how much a task helps another.

Finally, several other papers perform transfer learning experiments on more complex tasks than image classification as a way to validate their proposed method, but typically only for a few source-target dataset pairs, and only within the same task type (e.g. from Open Images [63] to CityScapes [21] for instance segmentation [122]; and across pairs of ILSVRC’12 [4], COCO [70], Pascal VOC [2] for object detection [108]).

**Other related work.** Finally we discuss two other directions of related work. Experimenting over a collection of datasets is getting more common, e.g. for *robust vision* approaches [3], [65], [86]. However, the general aim for robust methods is to learn a *single* model which performs well across several datasets for one task type.

A difficulty for sequential learning of neural networks, is the tendency of *catastrophic forgetting* [24], [56], [96]. In

our transfer learning setup, the new target model might have forgotten the old source tasks. However, forgetting is not relevant in our study, since we are interested in performance on the target task only. Moreover, we analyse in which conditions the target task can successfully re-use the knowledge from the source task.

### 3 FACTORS OF INFLUENCE

We study transfer learning at scale across three factors of influence: the difference in image domain between source and target tasks (Sect. 3.2), their difference in task type (Sect. 3.3), and the size of the source and target training sets (Sect. 3.4). To study these factors of influence, we introduce a collection of 20 datasets in Sect. 3.5, which we have chosen to cover these factors well.

### 3.1 Transfer Learning through pre-training

This paper explores transfer learning from a source task to a target task. We define a task as the combination of a task type (e.g. object detection, semantic segmentation) and a dataset (e.g. COCO). We follow the widespread practice of initializing the backbones of our networks with the weights obtained from image classification pre-training on ILSVRC'12 [7], [20], [39], [43], [47], [60], [95], [131]. This leads to the following process:

- 1) we train a model on ILSVRC'12 classification.
- 2) we copy the weights of the backbone of the ILSVRC'12 classification model to the source model. We randomly initialize the head of the source model, which is specific to the task type.
- 3) we train on the source task.
- 4) we copy the weights of the backbone of the source model to the target model. Again, we randomly initialize its head.
- 5) we train on the target training set.
- 6) we evaluate on the target validation set.

This protocol essentially defines a *transfer chain*: **ILSVRC'12**  $\rightarrow$  **source task**  $\rightarrow$  **target task**. We compare these *transfer chains* to the default practice: **ILSVRC'12**  $\rightarrow$  **target task**. To ensure fair comparisons, we have one set of ILSVRC'12 pre-trained weights which are used throughout all experiments in this paper. Analogously, for each source task we create one set of pre-trained weights used throughout all our experiments.

### 3.2 Image Domain

We want to study transfer learning across a wide range of different image domains. For this we considered many publicly available datasets and manually selected the following domain types: *consumer photos*, *driving*, *indoor*, *aerial*, *underwater*, *close-ups*, and *synthetic*. We deliberately only use domains from RGB image sensors, and exclude imagery from other sensors, such as CT scans, multi-spectral imagery, or lidar, because it is unclear how knowledge could be re-used when transferring across datasets acquired by different sensors.

To measure the difference between source and target domains, one way is to use these manually defined domain types. This results in a simple binary *same* or *different* measure. We also use a continuous similarity metric obtained directly from the visual appearance of the source and target datasets, to offer a more fine grained metric. In particular, we extract image features from a backbone based on our multi-source semantic segmentation model (Sect. 5.1), where we attach a spatial average pooling layer to the backbone, resulting in an 720 dimensional image vector. The domain difference is then the average distance of a target image to its closest source image:

$$D(T|S) = \frac{1}{|T|} \sum_t \left( \min_s d(f_t, f_s) \right) \quad (1)$$

where,  $|\cdot|$  denotes the cardinality,  $f$  denotes an image feature vector, and for  $d(\cdot, \cdot)$  we use the Euclidean distance. From each dataset we sample 1000 images to compute  $D(T|S)$ . The same images are used irrespective of

whether the dataset is used as target or source. Due to the min operation in Eq. (1) this is an asymmetric measure, i.e.  $D(A|B) \neq D(B|A)$ . This measure enables more fine-grained analysis than using our manually defined domains.

### 3.3 Task type

In this study we focus on structured prediction tasks, which all involve some form of spatial localization. Since training on these tasks yields spatially sensitive features, we hypothesise that these tasks could benefit each other. We consider four task types:

- 1) *semantic segmentation*: predict the class label for each pixel in the image.
- 2) *object detection*: predict tight bounding boxes around objects and predict their class labels.
- 3) *keypoint detection*: detect the image location of body joints or parts, for the human and dog classes in the datasets we consider.
- 4) *depth estimation*: predict for each pixel the distance from the camera to the physical surface (from a single image).

These four task types significantly vary in nature: semantic segmentation and depth estimation are pixel-wise prediction tasks, whereas keypoint detection requires identifying sparse (but related) points in an image, and object detection requires predicting bounding boxes. Hence, while these tasks are all about spatial localisation, their different nature makes it interesting to study the influence of transferring from one task type to another.

### 3.4 Dataset Size

The size of the target training set is important (e.g. [59]). When a target dataset is very large, the effect of transfer learning is likely to be minimal: all the required visual knowledge can be gathered directly from this target dataset. Therefore we consider two transfer learning settings, one using only a small number of target images and one using the full target dataset. We believe that using a small target training set is most relevant in practice, given that we often want to train strong models from a small set of annotated images.

We also perform some experiments varying the size of the source training set. A source model trained on a larger dataset is likely to be more beneficial for transfer learning [47], [72], [102]. Hence, in some of our experiments we limit all source sets to have a maximum number of images, sampled uniformly from the dataset. This allows to study the influence of source domain versus source size.

### 3.5 Dataset collection

To study these factors of influence on transfer learning we selected 20 public available datasets annotated for one (or more) of our task types, ensuring to span various very different image domains and with a large variety in dataset size. An overview of the datasets and their task types which we use is given in Tab. 1. The variety in visual appearance is illustrated in Fig. 1.

While we consider the annotations listed in Tab. 1, some datasets have additional annotation types available. For example, CityScapes [21] also provides disparity and



Nr	Name	Reference	Domain	Semantic	Detection	Keypoints	Depth	# Train	# Val
1	Pascal Context	[75]	Consumer	60	-	-	-	5K	5K
2	Pascal VOC	[31]	Consumer	22	20	-	-	10K	1449
3	MPII Human Pose	[6]	Consumer	-	-	16	-	17K	7K
4	ADE20K	[130]	Consumer	150	-	-	-	20K	2K
5	COCO Panoptic	[12], [55], [70]	Consumer	134	80	17	-	118K	5K
6	KITTI	[5]	Driving	30	-	-	-	150	50
7	CamVid	[10]	Driving	23	-	-	-	367	101
8	CityScapes	[21]	Driving	33	-	-	-	3K	500
9	India Driving Dataset (IDD)	[109]	Driving	35	-	-	-	7K	981
10	Berkeley Deep Drive (BDD)	[123]	Driving	20	10	-	-	7K	3K
11	Mapillary Vista Dataset	[77]	Driving	66	-	-	-	18K	2K
12	ISPRS	[88]	Aerial	6	-	-	-	4K	437
13	iSAID	[115], [119]	Aerial	16	-	-	-	27K	9K
14	SUN RGB-D	[99]	Indoor	37	-	-	✓	5K	425
15	ScanNet	[23]	Indoor	41	-	-	-	19K	5K
16	SUIM	[49]	Underwater	8	-	-	-	1525	110
17	Underwater Trash (UWT)	[34]	Underwater	-	3	-	-	6K	1144
18	Stanford Dogs	[8], [53]	Close-ups	-	-	20	-	7K	5K
19	vKITTI2	[11], [35]	Synthetic (driving)	9	-	-	✓	43K	7K
20	vGallery	[116]	Synthetic (indoor)	8	-	-	✓	44K	12K
	ILSVRC'12/ImageNet	[25], [90]	Websearch	1,000 classes for classification				1.2M	50K

<sup>a</sup> From the Pascal VOC datasets we use VOC2012 for semantic segmentation and VOC2007 for object detection. <sup>b</sup> For object detection BDD100K is used. <sup>c</sup> iSAID and ISPRS consist of few very high resolution images. Using the procedure of the iSAID devkit [115], we split each high-res image into multiple partially overlapping low-res images ( $800 \times 800$ ), which we use for training and evaluation. Train and validation images originate from different high-res images and hence are not overlapping in content. <sup>d</sup> The SUN RGB-D dataset contains imagery from NYU depth v2 [97], Berkeley B3DO [50], and SUN3D [120].

TABLE 1: Overview of the 20 datasets used in this paper. They vary in size, task types (we consider semantic segmentation, object detection, keypoint detection, depth estimation), number of classes, and image domain. We included datasets from consumer imagery, driving, aerial, indoor, underwater, close-ups, and synthetic domains. Some datasets (such as KITTI [5]) have more annotation types available, but are not used in this study.

Berkeley Deep Drive [123] also includes road segmentation annotations.

In our experiments, each dataset plays both roles of source and target, in different experiments in turn. This allows us to extensively study the aforementioned factors of influence for transfer learning over a wide range of image domains, both within task types and across task types, and for training sets of various size.

## 4 SETTING UP NETWORK ARCHITECTURES FOR TRANSFER LEARNING

This section describes our network setup. Three things are important for our study: a common framework with shared data augmentation, a common backbone, and high quality models specific to each task type.

### 4.1 Data normalization and augmentation

During training various data normalization and augmentation techniques are typically used [22], [61] which have significant impact on model performance. We unify the data normalization and augmentation step, because it changes the input of the network, *i.e.* it influences the low-level statistics of the input imagery. This allows to prevent variations in data normalization and augmentation from affecting transfer learning performance.

Data augmentation also influences what the model learns. For example, applying large rotation transformations or anisotropic scaling makes the network invariant to

those aspects, which may be beneficial for some tasks but detrimental for others, *e.g.* full rotation invariant networks are not able to distinguish 6 from 9. In this paper we do not want data normalization and augmentation to be confounding factors in transfer learning experiments. Hence we aim to keep data normalization and augmentation as simple as possible without compromising on accuracy, and apply the same protocol to all experiments.

**Illumination normalization.** For each dataset we normalize the images so that each color channel has zero mean and standard deviation 1. This form of whitening minimizes illumination biases from the different datasets as much as possible.

**Data augmentation.** To unify the data augmentation across all four task types considered, we ran many experiments to estimate the importance of common data augmentation techniques. We found the following augmentations to always have a positive (or neutral) effect and thus use them in all experiments: (1) random horizontal flipping of the image; (2) random rescaling of the image; and (3) taking a random crop. For object detection and keypoint detection, we consider only random crops fully containing at least one object. As in [65], we found that the image scales that lead to the best performance are intrinsic to the dataset, and hence dataset-dependent. Since we see this (partly) as a property of the image domain, we optimized for each task the input resolution of the network. Overall, the input resolutions during training range from  $420 \times 420$  for Pascal Context to  $713 \times 713$  for SUN RGB-D to  $512 \times 1024$  for

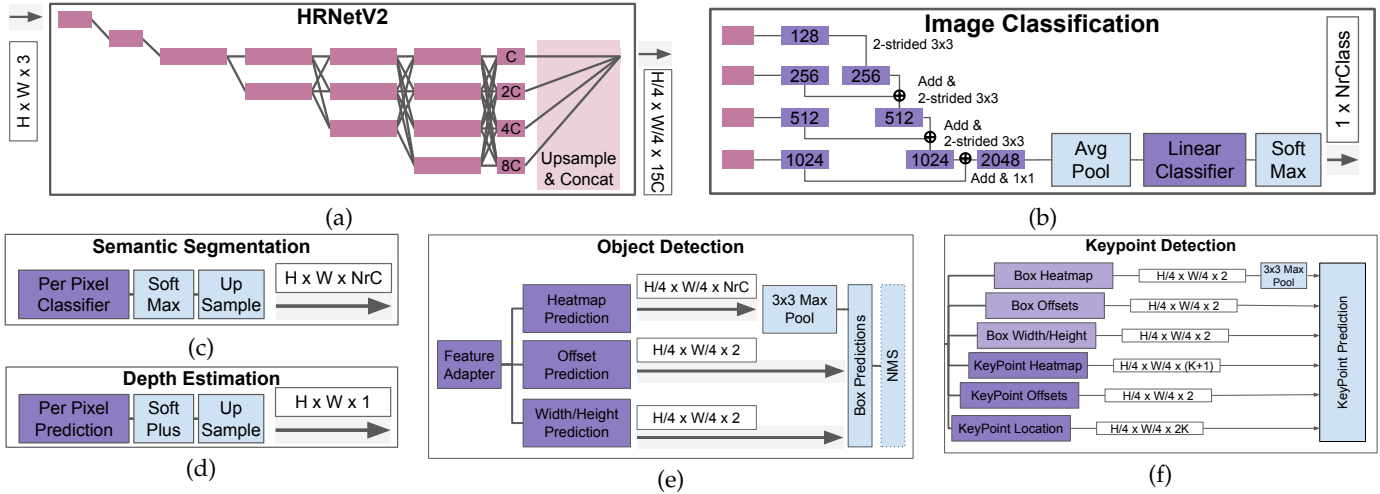


Fig. 2: Schematic illustrations of the backbone (a) and task type specific network architectures (b – f). Red and purple blocks are trainable (backbone and task type specific networks respectively), while blue blocks have no trainable parameters.

Backbone	Depth	Semseg 20	Detection 20	Keypoints 17
69M	721 < 1%	14K < 1%	2.4M 3.5%	10M 14%

TABLE 2: Number of parameters in the backbone and the respective task heads. The backbone forms the majority of the neural network and contains visual knowledge which could be shared to new target tasks.

CityScapes, where the latter is a landscape format common to most driving datasets. We always evaluate at a single scale and resize each image such that one side matches the input resolution used for that task.

After careful consideration, we decided not to use image rotation, varying image aspect ratio, or any form of color augmentation. *Image rotation* is incompatible with object detection which (generally) assumes axis-aligned ground-truth boxes. Moreover, we were able to reproduce the semantic segmentation performance of [65] even without rotation augmentation (see Sect. 4.3). *Varying image aspect ratio* did not yield positive effects on semantic segmentation, and also such augmentation is uncommon for object detection and keypoint detection. *Color augmentation* random changes in hue, contrast, saturation, and brightness did not yield positive effects for us on semantic segmentation and object detection, while substantially slowing down training.

**Our Setup.** To conclude, we consistently apply a limited number of data augmentation techniques across all experiments. The only difference across experiments is the input resolution, which is fixed per-dataset and is thus consistent with the image domain factor of variation we want to study (Sect. 3.2). This uniform protocol effectively cancels out varying data augmentation as a potential causal factor influencing the performance of transfer learning.

## 4.2 Network Architectures

Transfer learning through pre-training is only possible when the models for all task types share the same backbone ar-

chitecture. To have meaningful results, we need to choose a backbone which works well across all task types we explore. In this section we outline the used backbone architecture and the task type specific networks, these are illustrated in Fig. 2. More details are in Appendix A.

**Backbone Architecture.** For the backbone the recent high-resolution HRNetV2 [112] architecture was chosen. It extends the ResNet architecture to preserve high-resolution spatial features. Where a regular ResNet reduces the image resolution at every stage, HRNetV2 *also* keeps a parallel high-resolution branch, as illustrated in Fig. 2a. The resulting four output maps are combined into a single feature map by upscaling and concatenation. The backbone is pre-trained with supervised classification on ILSVRC’12 using the architecture shown in Fig. 2b.

HRNetV2 has two main advantages. First, it outperforms ResNet [44], ResNeXt [121], Wide ResNet [125], and Stacked Hourglass Networks [78] on three of the tasks we explore: semantic segmentation, keypoint detection, and object detection. Second, the high-resolution output feature maps allow to use relative shallow task type specific heads, see Tab. 2 for the number of trainable weights.

**Semantic Segmentation.** For semantic segmentation, the task type head is shown in Fig. 2c, it is an adoption of the network head proposed in [112]. It consists of a linear classifier, a softmax layer, and a bi-linear up-sampling layer to produce the final predicted segmentation map in the input image resolution.

**Object Detection.** For object detection we follow the CenterNet<sup>1</sup> approach of Zhou et al. [131] as shown in Fig. 2e. Each pixel is classified as being the center point of a bounding box of a specific class (*i.e.* the *heatmap*); additionally the bounding box size and offsets are predicted. Using center points to predict bounding boxes results in a simpler model than a two-stage architecture like Faster-RCNN [87], while being about as accurate [27], [131].

1. With CenterNet we denote a family of detection models which predicts boxes via their center points [27], [131].

Description	Top-1 Acc	Description	RMSE ( $\downarrow$ )	$\delta_1$ ( $\uparrow$ )	Description	AP50
ResNet-50 [112]	76.9	ResNet DORN [33]	.509	82.8	Hourglass-104 - CenterNet - 1 stage [131]	84.2
HRNetV2-W44 [112]	77.0	ResNet D-DFN [19]	.528	86.6	HRNetV1-W44 - 2 stage cascade [112]	90.8
<b>ours</b> HRNetV2-W48	79.5	<b>ours</b> HRNetV2-W48	.493	82.5	<b>ours</b> HRNetV2-W48 - CenterNet - 1 stage	81.1

(a)		(b)		(c)		
Description	MAP	Dataset	MSEG [65]	Ours		
ResNet-101 - CenterNet [131]	34.6	BDD	63.2	65.0		
Hourglass ResNet-104 - CenterNet [131]	40.3	CityScapes	77.6	75.7		
Hourglass ResNet-104 MS+flip - CenterNet [131]	45.1	COCO Panoptic	52.6	54.0		
HRNetV2-W48-pyramid - CenterNet [112]	43.4	Pascal Context	46.0	47.1		
<b>ours</b> HRNetV2-W48 - CenterNet	39.0	ScanNet	62.2	61.9		

TABLE 3: Comparison of our task type specific networks to recent networks on standard benchmarks (see text for details).

**Keypoint Detection.** For keypoint detection we follow (again) the CenterNet approach of [131] as illustrated in Fig. 2f. This essentially uses the object detection architecture to predict a bounding box for each person/dog, and then predicts the location of the keypoints within this bounding box.

**Depth Estimation.** For monocular depth estimation we mimic the architecture of the semantic segmentation head. We use a single regression layer, followed by a soft-plus layer and a bi-linear upsampling layer, see Fig. 2d. The soft-plus activation:  $\log(\exp(x) + 1)$  is a differential clipping to convert the logit values to depth ensuring that the predictions are positive, it is also used in [40], [132].

### 4.3 Single Task Performances

In order to validate our setup, we compare the performance of the used networks for each task type to a set of baselines. More details are in Appendix A.

**Backbone.** In order to validate our backbone architecture we use image classification on ILSVRC’12 and measure Top-1 accuracy. We compare our setup to a ResNet50 and HRNetV2-W44 [112] network. The results are in Tab. 3a. Our model performs best, reaching 79.5% Top-1 accuracy, validating our choice of architecture.

**Semantic Segmentation.** To validate our setup, we compare on a subset of the MSEG dataset collection [65]. For this experiment we use the annotation provided in MSEG, which uses fewer classes than our transfer setting setup. Training starts from a ILSVRC’12 pre-trained backbone.

Performance is evaluated by the Intersection-over-Union averaged over classes (mIoU) [30]. During evaluation we process complete images at a single resolution only. The results are shown in Tab. 3e. Our models perform on par with those of [65], which uses a similar architecture, but with different normalisation and data augmentation steps and a different implementation.

**Object Detection.** We validate our setup by training on COCO17 and evaluate performance on its 5K validation set, without using Non-Maximum Suppression (NMS). We compare to results reported in [131] and [112].

The evaluation is based on the COCO definition of mean Average Precision (mAP) [70]. The results are in Tab. 3d. We observe that the best results are obtained by using data augmentation during evaluation or by using a feature pyramid in the detection phase. Without these enhancements, our

implementation performs close to the Hourglass ResNet-104 and outperforms the ResNet-101. Hence we conclude that we can base our transfer learning experiments on a strong, modern object detection framework.

**Keypoint Detection.** We validate our setup on the keypoint detection task on the COCO dataset and report the mean Average Precision at 0.5 Object Keypoint Similarity (AP50) [1].

Results are in Tab. 3c, where we compare to the Hourglass-104 CenterNet [131] which yields 84.2 AP50<sup>2</sup> and to the HRNetV1 Cascade [112] which yields 90.8. Our setup uses a much simpler single-stage CenterNet head. With a performance of 81.1 AP50, it is near [131] and thus strong enough for our transfer learning exploration.

**Depth Estimation.** To validate our setup we compare monocular depth estimation on the NYUDepthV2 [97] dataset, using the root mean squared error (RMSE) metric and the  $\delta < 1.25$  accuracy, where  $\delta = \max(\frac{z}{z}, \frac{z}{z})$  is a measure of relative accuracy defined in [64].

The results are in Tab. 3b, where we compare to two recent ResNet based models: [33] uses depth-specific losses based on ordinal regression; and [19] uses a depth-specific network architecture, but with the same loss function as we do. From the results. Our proposed light-weight depth prediction model outperforms both depth specific models on RMSE and hence, it is well suited for our monocular depth estimation transfer learning experiments.

## 5 TRANSFER LEARNING EXPERIMENTS

In this section we describe our transfer learning experiments. We mainly conduct experiments in two settings: transfer learning with a small target training set and with the full target set. The analysis of the results across all experiments will be discussed in Section 6.

### 5.1 Setup

**Transfer Chains.** In our experimental setup we consider *transfer chains*: ILSVRC’12  $\rightarrow$  source  $\rightarrow$  target. Specifically, we first train a single classification model on ILSVRC’12, whose weights we reuse in all our experiments. To train a source model  $S$ , we first copy the ILSVRC’12 backbone weights into it. We randomly initialize the task head of  $S$ .

<sup>2</sup> Result obtained via correspondence with the authors: The public available model in the zoo (see [repo](#)) obtaining 64.0 AP50 on `coco-test`, obtains 84.2 AP50 on `coco-val`, without flipping.

Then we fine-tune until convergence on the source training set. This results in a single set of backbone weights per source task which we reuse in all our experiments. Analogously, we continue and copy the backbone weights of  $S$  into the backbone of  $T$ , randomly initialize the task head of  $T$ , and fine-tune until convergence on the target training set.

**Baseline.** As baseline we consider the default in the community, which is starting from ILSVRC'12 pretrained backbone weights: ILSVRC'12  $\rightarrow$  target.

**Evaluation.** Our core question is: can we get additional gains over our baseline by picking a good source? We therefore measure improvements w.r.t. the baseline, which we call *relative transfer gain*:

$$r(T|S) = \left( \frac{m(T|S)}{m(T|ILSVRC)} - 1 \right) * 100 \quad (2)$$

where  $m$  denotes a metric specific for the task type (e.g. mean-IoU for semantic segmentation), which is evaluated for all tasks on complete images at a single resolution only. Since for depth estimation lower values of  $m$  mean better performance, we multiply  $r$  by  $-1$  in that case.

This notion of gain is similar in spirit to the one defined in Taskonomy [126]. However their gain is the percentage of test images for which the transfer model outperforms a model trained from scratch. Instead, our *relative transfer gain* metric refers to a much stronger baseline: a model fine-tuned from ILSVRC'12. Moreover, we evaluate *how much better* the target model becomes in terms of a standard metric specific to each task type, averaged over all test images.

**Multi-Source Models.** Inspired by the success of using generic visual representations for various computer vision tasks [58], [59], [65], we include a multi-source model in our experiments.

We train a multi-source model for a specific task type based on several datasets. For each dataset a separate head is attached to a single, common backbone:

- semantic segmentation is trained across iSAID, COCO, Mapillary, ScanNet, SUIM, vGallery, and vKITTI2;
- depth estimation is trained across all three depth datasets we consider: SUN RGB-D, vGallery, and vKITTI2;
- object detection is trained across all four object detection datasets: COCO, BDD, Pascal VOC, and Underwater Trash.

The multi-source models are trained using dataset interleaving at the batch level, *i.e.* each batch is sampled from another dataset, and each head classifies only the images from that particular dataset. When doing transfer learning, only the weights of the common backbone are used as a source model.

**Training.** We determined the number of training steps per dataset for all source models in preliminary experiments. We use these for source model training and in the full target training setting. We lower the number of steps for the small target training setting. For each experiment in this paper, we selected the best model from three learning rates. More training details are in Appendix A.

	Aerial		Consumer			Driving				Indoor			Underwater		Synthetic		Scratch		
	iSAID 40K	iSAID 28K	PContext 5K	PVOC 10K	ADE20K 20K	COCO 120K	KITTI 150	CamVid 367	City 3.0K	IDD 7K	BDD 7K	Mapillary 18K	SUN 5K	ScanNet 19K	SUM 15K	vKITTI2 42K		vGallery 44K	
iSPRS	-	9	-1	-6	-2	0	-1	-5	0	-5	-1	-1	8	-7	-1	-5	5	-32	
iSAID	-16	-	-7	-12	-12	-8	-13	-20	-16	-9	-12	-9	-14	-12	-8	-20	-18	-52	
PContext	-13	-7	-	27	0	40	-7	-11	-11	-9	-12	-6	-19	-29	-12	-11	-8	32	-80
PVOC	-21	-8	29	-	-13	31	-9	-13	-12	-12	-12	-12	-26	-31	-12	-12	-11	26	-84
ADE20K	-5	1	-3	-8	-	13	-2	-3	-2	-3	-6	-1	-5	-11	-6	-1	-4	14	-67
COCO	-7	-3	-4	-5	-7	-	-3	-4	-6	-5	-7	-7	-10	-15	-6	-4	-3	-	-77
KITTI	-3	2	3	2	6	6	-	2	11	8	6	10	-6	-3	-1	1	2	9	-21
CamVid	-1	1	4	2	11	2	9	-	13	9	11	17	4	3	-1	-1	3	12	-18
City	-7	-2	0	-2	2	4	-1	-1	-	9	6	17	-6	-9	-2	-2	-3	15	-41
IDD	-6	-1	1	0	1	6	-1	-4	4	-	3	12	-5	-9	-2	0	-4	10	-44
BDD	-10	-2	1	1	7	12	1	2	10	13	-	29	-8	-12	-3	-2	-5	21	-41
Mapillary	-7	-1	1	-1	4	11	-3	-4	9	10	7	-	-10	-12	-3	-3	-4	-	-60
SUN	-13	-6	10	-1	55	60	-1	-11	-11	-12	-10	-11	-	61	0	-10	-7	73	-66
ScanNet	-18	-6	3	-2	31	56	-4	-5	-7	-6	-7	-5	71	-	-6	-6	-6	-	-62
SUIM	-11	1	-1	1	0	4	-1	-5	-4	-1	-2	-6	-8	-10	-	0	-3	-	-46
vKITTI2	-3	0	-1	-1	1	0	0	1	2	2	1	2	-1	-2	0	-	1	-	-16
vGallery	-2	0	0	-1	1	1	0	1	0	1	0	1	0	0	-1	0	-	-	-10

(a)

	Depth		Detection				Keypoints				Scratch
	vKITTI2 42K	vGallery 44K	COCO 120K	BDD 70K	UWUT 6K	Multi 12K	St. Dogs 12K	MPII 25K	COCO 120K	iSAID 28K	
iSAID	-13	-35	-21	-29	-40	-28	-21	-17	-13	-36	-52
PContext	-16	-40	-15	-4	-47	-39	-7	-14	-6	-37	-80
COCO	-8	-14	-6	-2	-44	-24	-5	-8	-5	-36	-77
KITTI	-3	-6	-4	-4	3	-6	-1	-3	-3	-8	-21
CamVid	-4	0	4	-1	10	-1	4	-13	-4	-11	-18
IDD	-4	-9	-5	-6	-3	-13	-2	-6	-3	-11	-44
SUN	8	-39	-12	-9	-34	-31	-3	-10	-4	-28	-66
ScanNet	13	-38	-13	-18	-36	-27	-11	-9	-3	-27	-62
SUIM	-8	-16	-4	-6	-19	-9	-3	-5	-1	-14	-46

(b)

	Detection		Depth		Keypoints		Segmentation		Scratch									
	VOC07 5K	UWUT 6K	COCO 120K	vKITTI2 42K	St. Dogs 12K	MPII 25K	PContext 5K	Multi 12K										
VOC07	-	-14	-8	207	-49	-74	-45	-15	-11	0	-21	4	-38	-28	-22	-26	25	8
UWUT	6	-	-12	52	-18	-29	-17	-5	-8	9	-9	-11	-6	-12	-14	-16	-8	-9
BDD	-15	-1	-	94	-16	-23	-2	-4	-3	19	-3	5	-12	14	23	-4	-1	15
COCO	30	5	-5	-	-26	-72	-14	-6	1	8	-6	-1	-11	-6	-4	-1	16	12

(c)

	Keypoints		Depth		Detection		Segmentation		Scratch
	St. Dogs 12K	MPII 25K	COCO 120K	vKITTI2 42K	COCO 120K	UWUT 6K	COCO 120K	Multi 12K	
St. Dogs	-	110	42	-25	-90	141	128	-33	-20
MPII	-8	-	166	-30	-81	36	37	-31	-46
COCO	-29	90	-	-54	-98	39	39	-51	-63

(d)

	Depth		Detection		Keypoints		Segmentation		Scratch								
	vKITTI2 42K	vGallery 44K	COCO 120K	BDD 70K	UWUT 6K	St. Dogs 12K	MPII 25K	vKITTI2 42K									
SUN	-	9	2	1	-7	-6	0	1	0	-3	9	-1	4	-7	-3	7	-33
vKITTI2	0	-	-1	-1	0	0	0	-2	-1	0	0	-1	0	0	-1	1	-16
vGallery	0	-6	-	-5	-3	-3	-5	-1	-3	-6	3	-1	-5	-2	1	3	-46

(e)

TABLE 4: Relative improvement gains for the small target training setting. In Tab. 4a both the source and target task types are semantic segmentation. See main text for details.



## 5.2 Transfer learning with a small target training set

The first setting we study is transfer learning with a small target training set, in which we limit the number of annotated examples are available for training. We deem this setting as the most challenging and practically relevant: obtain good performance for a structured prediction task by annotating only a few images. Transfer learning is particularly relevant in such a low-data regime.

Concretely, we limit the number of target training images to 150 per dataset (this is less than 3% of the available train data for 14 out of the 20 datasets). For COCO and ADE20k we make an exception and use 1000 images, which is still less than 1% of the available train data for COCO and 5% for ADE20K. The reason is that these datasets have a large number of classes following a long-tail distribution (COCO has 134 classes for segmentation, and ADE20K has 150). Just 150 images did not properly cover all classes: In all experiments we use a seeded selection, such that for each dataset all models are fine-tuned using the exact same training samples.

**Experiment.** Our results are shown in Tab. 4 (all relative transfer gains). Positive transfer gains are in green, negative transfer gains are in purple, no gains are represented by white. For ease of presentation, we group results by the target task type, which leads to multiple tables. Then we group by the source task type (two separate tables for semantic segmentation, blue vertical line in other experiments). Then we group by image domain (marked in blue above in Tab. 4a). Finally, within each image domain we order sources by their size. For completeness, absolute performance table are available in Appendix C. We tried to keep all experiments as homogenous as possible, while maintaining a wide scope of transfer across many diverse domains and task types. This leads to two peculiarities, which we clarify below.

First, in most experiments, the source training set and the target training set are disjoint (i.e. always when transferring within a task type, and most of the time across task types). However, when transferring across task types, a few of the source-target pairs use (partly) the same training set, e.g. COCO Object Detection as source for COCO Keypoint Detection in Tab. 4d or BDD Semantic Segmentation as source for BDD Object Detection in Tab. 4c. In these experiments the source models have been trained on *all* images of the training dataset for the source task type, while the target model is fine-tuned on only a small part of that training set, and for a different task type. This setting is relevant if one is interested in extending the annotation of a dataset from one task type to another (e.g. one has a lot of object bounding boxes, but very few depth masks for a dataset).

Second, when transferring within a task type, the multi-source models are only applied for target datasets which were not part of their training (hence the blank entries for this column in Tab. 4a). This is necessary for meaningful experiments, as otherwise some target training sets would simply be a subset of the images and annotations present in the multi-source.

	Aerial		Consumer				Driving				Indoor				Underwater		Synthetic		Search
	ISPRS 4.0K	ISAD 28K	PContext 5K	PVOC 10K	ADE20K 20K	COCO 120K	KITTI 150	CamVid 387	City 3.0K	IDO 7K	BDD 7K	Mapillary 18K	SUN 5K	ScanNet 19K	SUN 1.3K	vKITTI2 42K	vGallery 44K	Multi	
ISPRS	-	7	-3	-3	0	0	-4	-4	-4	-1	0	-2	-7	-4	-2	-5	0	35	
ISAD	0	-1	-1	-3	-3	0	-1	0	0	-1	0	-2	-5	-1	-1	0	1	-44	
PContext	-1	0	-	5	0	8	0	-1	0	-1	-1	-2	-4	-1	0	0	5	-46	
PVOC	-1	-1	-1	-1	-1	1	0	0	-2	-1	-1	-2	-2	-1	0	-1	0	-38	
ADE20K	-4	-2	-4	-5	-	-1	-2	-2	-2	-4	-2	-4	-5	-4	-3	-3	1	-39	
COCO	-6	-5	-6	-6	-6	-	-6	-5	-6	-6	-6	-6	-7	-6	-6	-6	-2	-37	
KITTI	1	3	5	3	6	7	-	2	11	6	6	11	-4	-2	3	-1	1	-24	
CamVid	-4	-8	-1	-9	2	-5	-5	-	4	3	2	12	-8	-4	-2	-1	1	-24	
City	-5	-5	-2	-6	-6	-6	-5	-4	-	-4	-5	5	-8	-10	-2	-4	-5	1	
IDO	0	-1	-1	-1	-1	-1	1	0	0	-	-1	2	-1	-3	-1	0	-1	-22	
BDD	-1	-2	-3	-2	-1	-1	0	1	2	2	-	0	-3	-6	0	0	-1	-33	
Mapillary	-4	-3	-4	-5	-6	-3	-2	-3	-2	-2	-4	-	-7	-8	-4	-3	-4	-30	
SUN	-3	-3	-4	-5	3	5	-3	-2	-1	-1	-1	-3	-	1	-3	-3	-2	-33	
ScanNet	-1	0	0	-1	2	5	1	2	1	1	0	1	1	-	-1	1	0	-33	
SUN	-4	-3	-7	-7	-3	-4	-2	-3	-4	-5	-2	-5	-3	-4	-	-4	-3	-36	
vKITTI2	-1	-1	-1	-1	-1	-1	0	0	0	4	7	7	7	7	7	-	0	-5	
vGallery	-1	0	0	-1	0	0	0	0	0	0	0	0	0	0	-1	0	-	-9	

	Detection				Depth		Keypoints		Segmentation							
	VOC07 5K	UWT 6K	BDD 70K	COCO 120K	Multi	SUN 5K	vKITTI2 42K	St. Dogs 12K	COCO 120K	SUN 1.3K	SUN 5K	BDD 7K	vKITTI2 42K	COCO 120K	Multi	
VOC07	-	-10	-16	18	11	-11	-41	-8	-3	-7	-4	-7	-3	-2	-17	-1
UWT	15	-	-1	16	-9	-12	-24	-10	7	10	11	-3	-5	0	-9	6
BDD	1	0	-	2	2	-1	-8	0	0	0	0	-1	0	0	-25	1
COCO	1	0	1	-	-2	-3	-9	-3	-2	0	-2	-2	-2	-1	-5	-3

	Keypoints		Depth		Detection		Segmentation		
	St. Dogs 12K	Multi	COCO 120K	SUN 5K	Multi	COCO 120K	Multi	Multi	
St. Dogs	-	0	-1	-2	-53	-2	-3	-8	-1
MPII	7	-	26	6	-66	7	9	-2	-1
COCO	0	0	-	0	-34	-2	-2	-1	0

	Depth	
	SUN 5K	Multi
SUN	-	-6
vKITTI2	3	-
vGallery	1	0

TABLE 5: Results for transfer learning in the full target training setting. Performance measured in *relative improvement gain* over a model fine-tuned from ILSVRC'12. In Table (a) both the source and target task types are semantic segmentation. See main text for details.

## 5.3 Transfer learning with full target training set

In this setting we use the full available target training set. We expect transfer learning to have a lesser effect, given that the target training set contains more information.

We perform a study similar to Section 5.2, but focus more on transfer within a task type. The results are shown in Tab. 5, again grouped by respectively target task type, source task type, image domain, and source size. Absolute numbers can be found in Appendix C.

## 5.4 Small source and small target training set

In this last setting we limit the size of the target training set as in Section 5.2, and also limit the size of each source training set to 1500 samples. This enables to study transfer learning effects where sources differ in appearance domain, but not in the amount of labeled images available. For all except two datasets, this implies a reduction to the

	Aerial		Consumer				Driving				Indoor				Underwater		Synthetic		Scratch
	ISPRS	ISID	PContext	PVOC	ADE20K	COCO	KITTI	CamVid	City	IDD	BDD	Mapillary	SUN	ScanNet	SUM	vKITTI2	vGallery		
ISPRS	-	6	3	2	-2	0	-1	-5	-4	-1	0	1	-6	-6	-4	1	-4	-32	
ISID	-12	-	-4	-2	-10	-12	-13	-20	-14	-11	-14	-12	-9	-13	-7	-19	-18	-52	
PContext	-8	-4	-	12	-4	12	-7	-11	-8	-8	-11	-5	-13	-18	-10	-8	-5	-80	
PVOC	-15	-7	18	-	-11	11	-9	-13	-12	-8	-9	-7	-17	-20	-11	-11	-9	-84	
ADE20K	-3	0	-3	-3	-	1	-2	-3	-2	-2	-4	-1	-3	-9	-6	0	-3	-67	
COCO	-4	-1	-5	-3	-6	-	-3	-4	-4	-5	-4	-4	-7	-9	-6	-3	-1	-77	
KITTI	-1	-3	1	-2	-1	2	-	2	8	2	3	1	-4	-11	0	-1	1	-21	
CamVid	0	3	3	6	4	3	9	-	11	9	11	7	-2	4	1	1	1	-18	
City	-4	-3	-2	-4	-1	0	-1	-1	-	3	3	6	-9	-12	-5	-4	-5	-41	
IDD	-4	-2	0	-1	-3	-2	-1	-4	3	-	1	6	-8	-8	-3	0	-4	-44	
BDD	-6	-3	0	2	1	-1	1	1	8	7	-	14	-11	-10	-4	-2	-3	-41	
Mapillary	-4	-2	0	-1	-2	-1	-3	-4	7	5	5	-	-9	-12	-4	-3	-3	-60	
SUN	-8	-1	3	-1	20	15	-1	-11	-7	-6	-10	-7	-	40	-4	-10	-5	-66	
ScanNet	-12	-3	1	-4	14	10	-4	-5	-8	-2	-9	-6	49	-	-4	-5	-5	-62	
SUM	-2	1	-5	2	-3	2	-1	-5	-6	-4	-3	-2	-2	-9	-	-4	-2	-46	
vKITTI2	-3	-2	-2	-4	-2	-2	0	1	0	-3	-1	-2	-2	-4	-3	-	-2	-16	
vGallery	-2	-1	-1	-1	-1	-1	0	1	-1	0	-1	0	-1	-1	-2	-1	-	-10	

TABLE 6: Results for transfer learning in the small source and small target setting. All results transferring within semantic segmentation. Performance measured in *relative improvement gain* over a model fine-tuned from ILSVRC’12. See main text for details.

number of images available (KITTI and CamVid have only respectively 150 and 367 training images overall).

We study transfer learning in this setting only for the semantic segmentation task type and only for transfer within-task. The results are shown in Tab. 6.

## 6 ANALYSIS

In this section we analyse our results across different settings, task types and domains. To facilitate the discussion, we distinguish four levels of relative transfer gains (as defined in Eq. (2)):

- VP Very positive transfer effect, when  $r > 10$ ;
- P Positive transfer effect, when  $r > 2$ ;
- I Insignificant transfer effect, when  $2 \leq r < -2$ ;
- N Negative transfer effect when  $r < -2$ .

We report the percentage of experiments for each level, but we do not report insignificant transfer (I). Tab. 7 shows the main results, split into the small target training set and full target training set settings, and filtered by whether we transfer within/through image domain and within/through task type.

**A1: Classic ILSVRC’12 transfer learning always outperforms training a model from scratch.** For *all* our experiments starting from ILSVRC’12 outperforms training from scratch. And by a large margin: even when using the full target training set, transferring from ILSVRC’12 improves performance by 5% – 46% (see Tab. 5a, right-most column). This confirms that ILSVRC’12 pre-training is a solid way of (starting) transfer learning, which explains why this practice is widespread.

**A2: For most target tasks there exists a source task which brings further benefits on top of ILSVRC’12 pre-training.**

This can be seen in Tab. 4 and Tab. 5, by noticing that for almost any row there are green entries, indicating positive relative transfer gain over ILSVRC’12 for that source model.

To quantify this observation, we have computed the number of target tasks for which there is a source leading to positive (P) or very positive (VP) transfer effect on top of ILSVRC’12 pre-training. To do this, for each target task we take the transfer gain brought by the best source (Tab. 8). In the small target training set regime, there is a positive effect for 85% of the target tasks, and very positive for 67%. Even in the full target training set regime, for 56% of the target tasks there is a positive effect, and very positive for 19%.

So although ILSVRC’12 pre-training is the de-facto standard way to do transfer learning, there is surprisingly much to gain from an additional transfer step, and this for all task types. Next, we analyze what are the factors that influence these benefits.

### A3: The image domain strongly affects transfer gains.

From Tab. 4a we see that most positive gains occur when the source and target tasks are in the same image domain (within-domain transfer). Conversely, often transfer across image domains yields negative gains. For example, for the consumer datasets as target, all other domains are bad sources. This makes the image domain an important factor. As can be seen, the size of the source dataset also plays a role: for example, for *driving* as a target domain, the larger driving and consumer datasets are generally better sources than the smaller ones. However, this effect is far less important than the image domain. Finally, Tab. 4b shows transfer from other task types to segmentation (cross-task-type transfer). Here, most effects are negative.

To quantify, we first compare the individual effects of image domain and task type in Tab. 7. Out of all experiments with small target training set, within-domain, cross-task-type setting, 43% yields positive transfer gains and 37% negative ones. Conversely, out of all experiments in the cross-domain, within-task-type setting, only 14% yields positive transfer gains while 55% negative ones. This pattern is repeated when using the full target training set. Hence we conclude that, given a target task, the image domain of the source is more important to achieve good transfer gains than its task type.

Above we considered image domains as manually defined, intuitive types. As presented in Section 3.2, we also consider a continuous measure of domain distance based on image appearance. Tab. 9 visualizes the domain distance for semantic segmentation datasets, with lighter colors indicating a smaller distance. This appearance distance correlates with the manually defined domains. One exception is the vKITTI2 dataset, which seems to be closer to other driving datasets than to the other synthetic dataset vGallery.

With such a continuous measure of domain distance we can compare its influence on transfer gains to the influence of source size. We first measure the rank correlation between domain distance and transfer gains using Kendall  $\tau$  [52]. For the small target training set experiments this yields a correlation of 0.40, while the correlation between source size and transfer gains is 0.12. Therefore the image domain is a more important factor of influence than source size.

**A4: For positive transfer, the source image domain should include the target domain.** In most of our experiments a

		Small target training set				Full target training set			
image domain transfer	task type transfer	P	VP	N	#	P	VP	N	#
		$r > 2\%$	$r > 10\%$	$r < -2\%$		$r > 2\%$	$r > 10\%$	$r < -2\%$	
all	all	21%	11%	56%	509	14%	3%	41%	382
within	within	69%	44%	17%	64	33%	9%	22%	78
within	cross	43%	22%	37%	49	22%	6%	39%	18
cross	within	14%	5%	55%	242	8%	1%	43%	242
cross	cross	5%	2%	79%	154	7%	2%	64%	44

TABLE 7: Percentage of experiments for which we measured positive/negative transfer effects of a certain magnitude. We show these for the small target training set setting (i.e. measured over all source-target pairs from all tables in Tab. 4) and full target training setting (measured over tables in Tab. 5). We aggregate these under different image domain and task type transfer constraints. # denotes the number of experiments considered according to the constraint.

		Small target training set				Full target training set			
image domain transfer	task type transfer	P	VP	N	#	P	VP	N	#
		$r > 2\%$	$r > 10\%$	$r < -2\%$		$r > 2\%$	$r > 10\%$	$r < -2\%$	
all	all	85%	67%	4%	27	56%	19%	7%	27
within	within	73%	64%	14%	22	50%	15%	12%	26
within	cross	65%	35%	18%	17	33%	17%	0%	6
cross	within	56%	30%	26%	27	30%	4%	19%	27
cross	cross	32%	11%	53%	19	29%	14%	14%	7

TABLE 8: Transfer effects induced by the best available source for each target task. We show these for the small target training setting (i.e. max over each row of the tables in Tab. 4) and full target training setting (i.e. max over rows in Tab. 5). We aggregated these under different image domain and task type transfer constraints. # denotes the number of target tasks considered according to the constraints.

source from a broader domain helps a target from a more specific domain contained within it. This can be qualitatively observed from Tab. 4a. For example, sources from the consumer domain achieve positive transfer not only on consumer targets, but also on driving or indoor. Indeed, the consumer datasets also contain images of street views and indoor scenes (Fig. 1). Conversely, if we look at the consumer domain as a target, *none* of the driving or indoor sources yield any positive transfer. The same pattern can also explain the success of the multi-source model: this model always yields positive transfer, and by design it contains images from each of the manually defined target domains.

To visualize domain inclusion, we created a t-SNE plot of the activations of 150 images from each semantic segmentation dataset (Fig. 3a; same features as Section 3.2). The visualization show that the consumer domain points are indeed scattered around the plot, covering almost all other domains. Other domains form more compact clusters, e.g. aerial in green, and driving in yellow/orange.

Finally, we also quantitatively demonstrate that domain inclusion matters. To do so, we change the function used in Eq. (1) to match target images to source images. We have four strategies: (1) Assign one-to-one each target image to a source image such that the overall Earth Mover’s Distance is minimized (i.e. matching is done through the Hungarian algorithm). This is a measure of overlap of the distributions of the two image domains. (2) Assign each target image to its closest source. This measures *inclusion* of the target dataset

in the source dataset and is identical to Eq. (1). (3) Assign each source image to its closest target. A large distance here means that many source images are far from the target. (4) Take the average of (2) and (3), yielding a symmetric measure.

We correlate each of the four measures to transfer gains using Kendall- $\tau$  rank correlation. To remove any influence of task type and dataset size (both source and target), we do this in the small source and small target set setting (Section 5.4). The results are shown in Tab. 10. The assignment measure capturing inclusion (2) has the highest correlation with transfer gains. Interestingly, measure (3) has (very) low correlation. This suggests that even if the source domain has images far from the target domain, this has only a small influence on transfer gains. This confirms that the most important aspect for a source to yield positive transfer is that it should *include* the target image domain.

**A5: Multi-source models yield good transfer, but are outperformed by the largest within-domain source.**

Tab. 4a shows that our multi-source semantic segmentation model yields positive transfer gains for all targets. This is in line with our previous observations that the source domain should include the target, while it is less important that it spans a much broader range. Indeed, by construction the source data of this multi-source model spans all domains. However, we also observe that the largest within-domain source almost always yields better transfer gains: In 7 out of these 10 experiments the largest within-domain source yields significantly better transfer (i.e.  $> 2\%$ ). Only



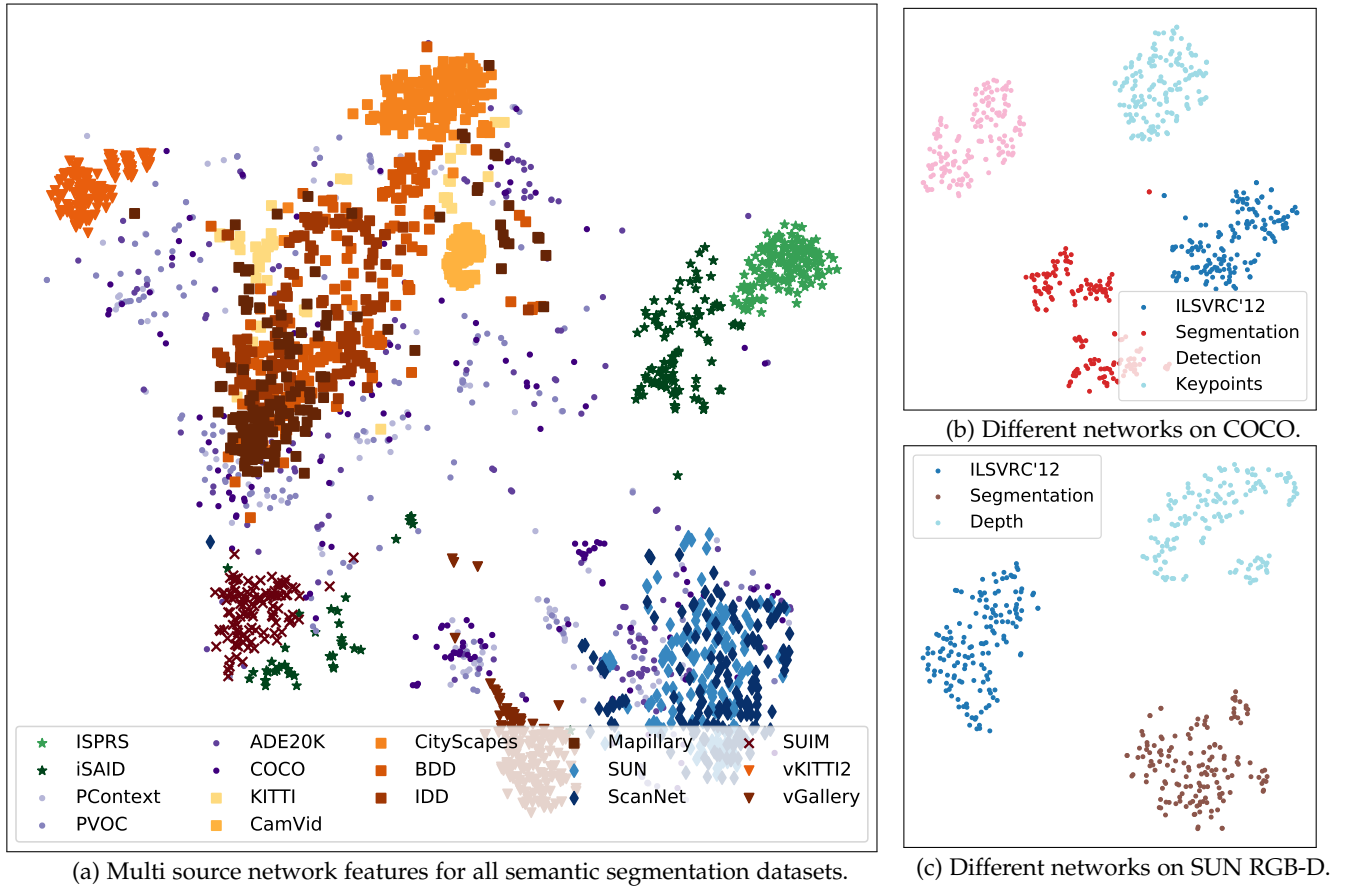


Fig. 3: Illustration of the appearance distribution of the datasets, according to different feature encoding networks, using t-SNE. Fig. 3a visualizes the semantic segmentation datasets using features extracted from the multi-source network. Fig. 3b and Fig. 3c visualizes the COCO and SUN RGB-D datasets using networks trained on different task types (ILSVRC'12 indicates classification).

for the target SUN RGB-D the multi-source model is better. This is explainable since the multi-source model includes both COCO (consumer) and ScanNet (indoor) which are both good sources for SUN RGB-D. Hence the multi-source model arguably covers the domain of this dataset better than the largest single source model.

#### A6: Transfer across task types can bring positive transfer gains.

Depending on the choice of task types for the source and target, cross-task-type transfer can be beneficial: for 65% of the targets within the same image domain as the source, cross-task-type transfer results in positive transfer gains (Tab. 8). To study this effect more precisely, we split the results over different cross-task-type pairs in Tab. 11. The left column looks at cross-task-type transfer *on the same dataset* (the specific setting studied in detail in Taskonomy [126]). Here we can see that object detection and keypoint detection help each other. Segmentation helps object detection but not vice versa. Semantic segmentation and keypoint detection hurt each other. When transferring to a different dataset, but still saying in the same image domain (right column), results show similar effects (except that keypoint detection stops helping object detection).

Our results confirm some of the observations in [126] as we also observe that some task types are more easily to

transfer from / to than others, and that transfer across task types is asymmetric, *i.e.* if task type A is beneficial for task type B, the reverse might not hold.

However, even when we observe gains through cross-task-type transfer within *the same dataset*, if there is another within-domain source with the same task type, it yields even better gains. Two examples are the object detection results for BDD and COCO as targets in Tab. 4c. This suggests that having a source with the same image domain and task type as the target is better than having annotations for another task type on the target dataset.

Tab. 7 shows that for cross-task-type transfer to work, the image domain of source and target should be the same: cross-domain, cross-task transfer yields negative transfer for 79% of the experiments, and positive for 5%. Moreover, out of those 5%, all except one experiment have consumer images as a source which includes the target domain of driving, indoor, and (arguably) underwater. Hence cross-task-type transfer can only be expected to work if the image domain from the source includes that of the target.

To understand how the task type influences the features learned by the source models, we visualize the features learned on the same dataset but for different task types using t-SNE (cosine distance, which removes arbitrary scaling effects). We do this for the exact same set of images

	Aerial		Consumer				Driving				Indoor				Underwater		Synthetic	
	ISPRS	ISAIID	PContext	PVOC	ADE20K	COCO	KITTI	CamVid	City	IDD	BDD	Mapillary	SUN	ScanNet	SUIM	vKITTI2	vGallery	
ISPRS	-	13	17	17	17	16	20	21	19	18	16	19	17	17	17	20	20	
ISAIID	16	-	19	19	18	18	21	22	20	20	18	20	19	19	18	21	20	
PContext	27	25	-	12	17	15	23	23	23	20	20	20	24	22	23	21	25	
PVOC	27	26	12	-	18	15	23	23	23	20	20	20	24	22	24	22	26	
ADE20K	23	22	15	16	-	15	20	20	20	18	17	17	20	19	21	19	22	
COCO	27	25	16	16	17	-	22	22	22	19	19	19	24	21	23	21	25	
KITTI	26	24	14	15	13	13	-	12	11	10	11	10	24	22	21	12	23	
CamVid	27	25	14	15	12	13	11	-	11	10	10	10	24	21	22	12	23	
City	20	20	13	14	12	14	10	11	-	10	10	10	20	20	21	12	19	
IDD	29	26	14	15	13	13	13	12	14	-	11	10	26	23	22	13	26	
BDD	29	26	14	15	13	14	14	14	16	11	-	11	26	23	23	14	27	
Mapillary	45	39	15	16	14	14	21	18	26	12	12	-	40	32	34	18	41	
SUN	18	18	15	15	13	14	21	22	20	19	19	20	-	12	19	20	18	
ScanNet	20	20	16	16	15	15	22	23	21	21	20	21	13	-	20	21	18	
SUIM	19	17	19	19	19	18	23	23	23	22	20	21	21	20	-	21	19	
vKITTI2	26	23	14	14	14	14	12	13	13	12	12	12	24	21	21	-	22	
vGallery	20	19	16	15	15	15	20	20	19	19	19	18	15	15	20	19	-	

TABLE 9: Domain distance between datasets (Eq. (1), asymmetric). Rows are targets, columns are sources. Lowest distance is within a manually defined image domain, as expected. Consumer to driving and indoor domains yield close distance values, but not vice versa. This shows that consumer images *include* both the driving and indoor domains.

Assignment method	$\tau$
(1) Earth Mover’s Distance	0.20
(2) Target image $\rightarrow$ closest source image	0.42
(3) Source image $\rightarrow$ closest target image	0.06
(4) Source $\rightarrow$ Target & Target $\rightarrow$ Source	0.27

TABLE 10: Kendall- $\tau$  correlation between transfer gains and dataset distance for semantic segmentation in the small source small target setting (Section 5.4). As distance we take the average distance between images from the source and target datasets, while varying the assignment method (*i.e.* min function in Eq. (1)). The assignment with the highest correlation measures *inclusion* of the target by the source.

for COCO (Fig. 3b) and SUN RGB-D (Fig. 3c). We observe that each model creates compact cluster representations, showing there is a larger distance between an identical image in the feature spaces of two different models than between two different images in the feature space of one model.

**A7: Transfer within-task-type and within-domain yields very positive effects.** Combining the observations A3 and A6, our recommendation to obtain positive transfer is to use a source model trained on the same domain and for the same task type as the target task. In this setting, a total of 69% of all source-target pairs exhibits positive transfer (Tab. 7). Moreover, the best available source in this setting leads to positive transfer for 73% of the target tasks (and 64% very positive).

	same dataset			same domain		
	P	N	#	P	N	#
seg $\rightarrow$ obj det	100%	0%	2	63%	25%	8
seg $\rightarrow$ kp det	0%	100%	1	0%	100%	2
seg $\rightarrow$ depth	33%	0%	3	33%	33%	3
obj det $\rightarrow$ seg	0%	0%	1	30%	50%	10
obj det $\rightarrow$ kp det	100%	0%	1	100%	0%	2
kp det $\rightarrow$ seg	0%	100%	1	0%	100%	3
kp det $\rightarrow$ obj det	100%	0%	1	0%	33%	3
depth $\rightarrow$ seg	100%	0%	1	100%	0%	1

TABLE 11: Percentage of positive and negative transfer experiments in the cross-task-type, small target setting. Semantic segmentation is denoted as ‘seg’, object detection as ‘obj det’, keypoint detection as ‘kp det’, depth estimation as ‘depth’. We consider the setting where the images of the source are *the same* (dataset) as the target, and when the source and target share the *same domain* but are different datasets. Note that the two experiments where segmentation to object detection fails for the same domain, are for Underwater Trash as a target. Arguably, this is a rather different underwater domain than that of SUIM which acts as the source (once alone, once as a multi-source model).

**A8: Transfer naturally flows from larger to smaller datasets.** To study the effect of dataset size, we look at transfer learning within-domain and within-task-type. In this setting, the transfer effect is best visible in Tab. 4a and Tab. 5a, when looking at the blocks around the diagonal. Within each block of datasets of the same domain, they are ordered by training set size. We observe a clear *upper-triangle* pattern, where the larger datasets positively transfer towards the smaller datasets. The reverse is much less present.

Quantitatively, we measure a positive transfer in 60% of within-domain and within-task-type experiments where the source training set is larger than the target dataset. Conversely, when the source training set is smaller than the target training set, only 5% of our experiments result in positive transfer.

**A9: Transfer learning effects are larger for small target training sets.** We expect this to hold because when the target task already offers a large training set, the target model can learn the required visual knowledge from the target task directly. We can clearly see such effects by comparing the within-domain segmentation results of the small and full target training set setting, *i.e.* Tab. 4a vs Tab. 5a. The absolute transfer gains are higher in the small target training set setting than in the full one, while overall patterns remain roughly the same.

We now compare Tab. 4a and Tab. 5a quantitatively. We first examine positive transfer gains: out of all experiments with positive transfer gains in the full target setting, 80% also have a positive transfer in the small target setting. Vice versa, out of all experiments with positive transfer gains in the small target setting, 43% also have significant gains in the full target setting (for most of these experiments the gains become insignificant). When looking at negative transfer effects: if transfer effects are negative on a small target training set, these effects remain negative or become

neutral in the full target set in 100% of these experiments.

Practically, this suggests a quick test for whether a source dataset may be beneficial for a target. First train on a small subset of the target training set. Then, if transfer effects are negative, discard this source for this target. If transfer effects are positive, explore this source further, since there is a good chance its benefits are kept when using the full target training set.

**A10: The source domain including the target is more important than the number of source samples.** Here we aim to disentangle the size of the source training set from the breadth of the image domain it spans (e.g. COCO covers a broad image domain as visualized in Fig. 3b). The influence of source size is eliminated in our experiment in Tab. 6, where we use a small training set for all sources (and a small target training set too). It is instructive to compare this Tab. 4a, where full source sets are used.

Even after fixing source size, most patterns stay roughly the same. For example, COCO remains a good source for other consumer and indoor datasets. Indeed, COCO is not only the largest but also the most diverse consumer dataset. Similarly, Mapillary remains a good source for most other driving datasets, while other driving sources are generally worse. Mapillary was designed to capture driving conditions around all continents, whereas the India Driving Dataset covers only Indian roads, Berkeley Deep Drive is US only, while CityScapes is (mostly) Germany. Camvid consists of frames of four videos, and therefore has a narrow image domain. This suggests that much of the effects in Tab. 4a which we attributed to dataset size in A8, can in fact be attributed to the property of the source domain *including* the target domain. Of course, source size is generally correlated to the breadth of its domain (e.g. COCO and Mapillary were both designed to span broader domains), which is why care needs to be taken to disentangle their effects.

## 7 ADDITIONAL SCENARIOS

In this section we perform several additional experiments to verify whether our work generalizes to other scenarios. We only report the main observations while details are provided in Appendix B.

**Fixed image resolution.** We redo the segmentation experiments in the small target training setting (i.e. Tab. 4a) but now fixing the image resolution to be  $713 \times 713$  pixels across all datasets. Results in Tab. A.1 (see Appendix) show highly similar transfer patterns and all our previous conclusions hold.

**ResNet50.** We change the backbone to ResNet50 [44] and redo the experiments in the small target training setting for segmentation and partially for detection (i.e. redo Tab. 4a partially Tab. 4c). Results in Tab. A.2a (see Appendix) suggest that ResNet50 benefits more from transfer learning, possibly due to improving its ability for localized predictions. But again, we observe similar transfer patterns and all our previous conclusions hold.

**Self-supervised ILSVRC'12 training.** We now redo the experiments with ResNet50, but instead we start from a checkpoint which was obtained by using self-supervised learning on ILSVRC'12 using the publicly available SimCLR V2 implementation [15]. We then train our sources fully

supervised as before. Results in Tab. A.2b (see Appendix) show primarily changes in the baseline: directly training from self-supervised ILSVRC'12 weights is better than directly training from ILSVRC'12 fully supervised classification weights, as also found in [29], [133]. At the same time patterns are again very similar and our main conclusions remain unaltered.

**Self-supervised Transfer Chain.** We do a single experiment where we use SimCLR [15] to create a self-supervised transfer chain: ILSVRC'12 self-supervised  $\rightarrow$  COCO self-supervised  $\rightarrow$  target. We find that this transfer chain is mildly beneficial for COCO *image classification*. However, for segmentation our results in Tab. A.4 (see Appendix) show that that this transfer chain is worse than directly training from ILSVRC'12 self-supervised weights for *all* target datasets. On average it is 0.07 IoU worse, while even for COCO as a target results are 0.02 IoU worse. This result suggests that self-supervised pre-training is biased towards image classification. Furthermore, image classification results on self-supervised models are not very predictive of performance on other tasks, as also shown in the dedicated study of [29].

## 8 CONCLUSION

In this paper we performed over 2000 transfer learning experiments across a wide variety of image domains and task types. Our systematic analysis of these experiments lead to the following conclusions: (1) for most tasks there exists a source which significantly outperforms ILSVRC'12 pre-training; (2) the image domain is the most important factor for achieving positive transfer; (3) the source task should *include* the image domain of the target task to achieve best result; (4) at the same time, we observe only small negative effects when the image domain of the source task is much broader than that of the target; (5) transfer across task types can be beneficial, but its success is heavily dependent on both the source and target task types.

Our findings provide support for the success of large-scale pre-training for transfer learning, with a single very large source spanning a mixture of domains [51], [59], [72], [102], [127]. If a good source set should include the target but can be arbitrarily broad, training on a massive dataset which covers most of the visual domains should yield a good source model for most target tasks. However, this form of pre-training inevitably requires transfer across task types, whose success depends on both the source and target task types. This suggests that future works exploring pre-training should focus also on structured prediction tasks.

## REFERENCES

- [1] Common objects in context (COCO) keypoint challenge - evaluation. <https://cocodataset.org/#keypoints-eval>.
- [2] The PASCAL visual object classes challenge. <http://host.robots.ox.ac.uk/pascal/VOC/>.
- [3] Robust vision challenge. <http://www.robustvision.net/>.
- [4] Imagenet large scale visual recognition challenge (ILSVRC). <http://www.image-net.org/challenges/LSVRC/2012/index>, 2012.
- [5] H. Alhaja, S. Mustikovela, L. Mescheder, A. Geiger, and C. Rother. Augmented reality meets computer vision: Efficient data generation for urban driving scenes. *IJCV*, 2018.



- [6] M. Andriluka, L. Pishchulin, P. Gehler, and B. Schiele. 2d human pose estimation: New benchmark and state of the art analysis. In *CVPR*, 2014.
- [7] H. Azizpour, A. S. Razavian, J. Sullivan, A. Maki, and S. Carlsson. Factors of transferability for a generic convnet representation. *IEEE Trans. on PAMI*, 2015.
- [8] B. Biggs, O. Boyne, J. Charles, A. Fitzgibbon, and R. Cipolla. Who left the dogs out?: 3D animal reconstruction with expectation maximization in the loop. In *ECCV*, 2020.
- [9] K. Bousmalis, N. Silberman, D. Dohan, D. Erhan, and D. Krishnan. Unsupervised pixel-level domain adaptation with generative adversarial networks. In *CVPR*, 2017.
- [10] G. J. Brostow, J. Fauqueur, and R. Cipolla. Semantic object classes in video: A high-definition ground truth database. *Patt. Rec. Letters*, 30(2):88–97, 2009.
- [11] Y. Cabon, N. Murray, and M. Humenberger. Virtual kitti 2, 2020.
- [12] H. Caesar, J. Uijlings, and V. Ferrari. COCO-Stuff: Thing and stuff classes in context. In *CVPR*, 2018.
- [13] W.-G. Chang, T. You, S. Seo, S. Kwak, and B. Han. Domain-specific batch normalization for unsupervised domain adaptation. In *CVPR*, 2019.
- [14] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. Yuille. DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs. *IEEE Trans. on PAMI*, 2018.
- [15] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton. A simple framework for contrastive learning of visual representations. In *ICML*, 2020.
- [16] T. Chen, S. Kornblith, K. Swersky, M. Norouzi, and G. Hinton. Big self-supervised models are strong semi-supervised learners. In *NeurIPS*, 2020.
- [17] W.-Y. Chen, Y.-C. Liu, Z. Kira, Y.-C. F. Wang, and J.-B. Huang. A closer look at few-shot classification. In *ICLR*, 2019.
- [18] X. Chen and K. He. Exploring simple siamese representation learning. In *CVPR*, 2021.
- [19] Y. Chen, T. Mensink, and E. Gavves. 3d neighborhood convolution: Learning depth-aware features for rgb-d and rgb semantic segmentation. In *3DV*, 2019.
- [20] B. Chu, V. Madhavan, O. Beijbom, J. Hoffman, and T. Darrell. Best practices for fine-tuning visual classifiers to new domains. In *ECCV*, 2016.
- [21] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele. The cityscapes dataset for semantic urban scene understanding. In *CVPR*, 2016.
- [22] E. D. Cubuk, B. Zoph, D. Mane, V. Vasudevan, and Q. V. Le. Autoaugment: Learning augmentation strategies from data. In *CVPR*, 2019.
- [23] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Nießner. ScanNet: Richly-annotated 3d reconstructions of indoor scenes. In *CVPR*, 2017.
- [24] G. Davidson and M. C. Mozer. Sequential mastery of multiple visual tasks: Networks naturally learn to learn and forget to forget. In *CVPR*, 2020.
- [25] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-fei. ImageNet: A large-scale hierarchical image database. In *CVPR*, 2009.
- [26] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.
- [27] K. Duan, S. Bai, L. Xie, H. Qi, Q. Huang, and Q. Tian. Centernet: Keypoint triplets for object detection. In *ICCV*, 2019.
- [28] D. Eigen, C. Puhrsch, and R. Fergus. Depth map prediction from a single image using a multi-scale deep network. In *NeurIPS*, 2014.
- [29] L. Ericsson, H. Gouk, and T. M. Hospedales. How well do self-supervised models transfer? In *CVPR*, 2021.
- [30] M. Everingham, S. Eslami, L. van Gool, C. Williams, J. Winn, and A. Zisserman. The PASCAL visual object classes challenge: A retrospective. *IJCV*, 2015.
- [31] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results. <http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html>, 2012.
- [32] C. Finn, P. Abbeel, and S. Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *ICML*, 2017.
- [33] H. Fu, M. Gong, C. Wang, K. Batmanghelich, and D. Tao. Deep ordinal regression network for monocular depth estimation. In *CVPR*, 2018.
- [34] M. Fulton, J. Hong, and J. Sattar. Trash-icra19: A bounding box labeled dataset of underwater trash. In *Proc. Intl. Conf. on Robotics and Automation*, 2020.
- [35] A. Gaidon, Q. Wang, Y. Cabon, and E. Vig. Virtual worlds as proxy for multi-object tracking analysis. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 4340–4349, 2016.
- [36] Y. Ganin and V. Lempitsky. Unsupervised domain adaptation by backpropagation. In *ICML*, 2014.
- [37] W. Ge and Y. Yu. Borrowing treasures from the wealthy: Deep transfer learning through selective joint fine-tuning. In *CVPR*, 2017.
- [38] S. Gidaris and N. Komodakis. Dynamic few-shot visual learning without forgetting. In *CVPR*, 2018.
- [39] R. Girshick. Fast R-CNN. In *ICCV*, 2015.
- [40] A. Gordon, H. Li, R. Jonschkowski, and A. Angelova. Depth from videos in the wild: Unsupervised monocular depth learning from unknown cameras. In *ICCV*, 2019.
- [41] J.-B. Grill, F. Strub, F. Althé, C. Tallec, P. H. Richemond, E. Buchatskaya, C. Doersch, B. A. Pires, Z. D. Guo, M. G. Azar, B. Piot, K. Kavukcuoglu, R. Munos, and M. Valko. Bootstrap your own latent: A new approach to self-supervised learning. In *NeurIPS*, 2020.
- [42] M. Gygli, J. Uijlings, and V. Ferrari. Training neural networks to produce compatible features. In *AAAI*, 2020.
- [43] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask R-CNN. In *ICCV*, 2017.
- [44] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016.
- [45] J. Hoffman, D. Wang, F. Yu, and T. Darrell. Fcns in the wild: Pixel-level adversarial and constraint-based adaptation. *arXiv preprint arXiv:1612.02649*, 2016.
- [46] S. Huang and D. Tao. All you need is a good representation: A multi-level and classifier-centric representation for few-shot learning. *arXiv preprint arXiv:1911.12476*, 2019.
- [47] M. Huh, P. Agrawal, and A. Efros. What makes imagenet good for transfer learning? In *NIPS LSCVS workshop*, 2016.
- [48] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, 2015.
- [49] M. J. Islam, C. Edge, Y. Xiao, P. Luo, M. Mehtaz, C. Morse, S. S. Enan, and J. Sattar. Semantic Segmentation of Underwater Imagery: Dataset and Benchmark. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020.
- [50] A. Janoch, S. Karayev, Y. Jia, J. T. Barron, M. Fritz, K. Saenko, and T. Darrell. A category-level 3-d object dataset: Putting the kinect to work. In *ICCV Workshop*, 2011.
- [51] A. Joulin, L. van der Maaten, A. Jabri, and N. Vasilache. Learning visual features from large weakly supervised data. In *ECCV*, 2016.
- [52] M. Kendall. A new measure of rank correlation. *Biometrika*, 1938.
- [53] A. Khosla, N. Jayadevaprakash, B. Yao, and L. Fei-Fei. Novel dataset for fine-grained image categorization. In *Workshop on Fine-Grained Visual Categorization, CVPR Workshop*, 2011.
- [54] D. P. Kingma and J. L. Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.
- [55] A. Kirillov, K. He, R. Girshick, C. Rother, and P. Dollár. Panoptic segmentation. In *CVPR*, 2019.
- [56] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proc. Nat. Acad. Sci. USA*, 2017.
- [57] G. Koch, R. Zemel, and R. Salakhutdinov. Siamese neural networks for one-shot image recognition. In *ICML Workshop*, 2015.
- [58] I. Kokkinos. Ubernet: Training a ‘universal’ cnn for low-, mid-, and high- level vision using diverse datasets and limited memory. In *CVPR*, 2017.
- [59] A. Kolesnikov, L. Beyer, X. Zhai, J. Puigcerver, J. Yung, S. Gelly, and N. Houlsby. Big Transfer (BiT): general visual representation learning. In *ECCV*, 2020.
- [60] S. Kornblith, J. Shlens, and Q. V. Le. Do better imagenet models transfer better? In *CVPR*, 2019.
- [61] A. Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

- [62] J. N. Kundu, N. Venkat, A. Revanur, R. V. Babu, et al. Towards inheritable models for open-set domain adaptation. In *CVPR*, 2020.
- [63] A. Kuznetsova, H. Rom, N. Alldrin, J. Uijlings, I. Krasin, J. Pont-Tuset, S. Kamali, S. Popov, M. Mallocci, A. Kolesnikov, et al. The open images dataset v4. *IJCV*, 2020.
- [64] L. Ladicky, J. Shi, and M. Pollefeys. Pulling things out of perspective. In *CVPR*, 2014.
- [65] J. Lambert, Z. Liu, O. Sener, J. Hays, and V. Koltun. MSeg: A composite dataset for multi-domain semantic segmentation. In *CVPR*, 2020.
- [66] H. Law and J. Deng. Cornernet: Detecting objects as paired keypoints. In *ECCV*, 2018.
- [67] S. Lee, J. Hyun, H. Seong, and E. Kim. Unsupervised domain adaptation for semantic segmentation by content transfer. In *AAAI*, 2020.
- [68] J. Li, R. Klein, and A. Yao. A two-streamed network for estimating fine-scaled depth maps from single RGB images. In *ICCV*, 2017.
- [69] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár. Focal loss for dense object detection. In *ICCV*, 2017.
- [70] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. Zitnick. Microsoft COCO: Common objects in context. In *ECCV*, 2014.
- [71] M. Long, Y. Cao, J. Wang, and M. Jordan. Learning transferable features with deep adaptation networks. In *ICML*, 2015.
- [72] D. Mahajan, R. Girshick, V. Ramanathan, K. He, M. Paluri, Y. Li, A. Bharambe, and L. van der Maaten. Exploring the limits of weakly supervised pretraining. In *ECCV*, 2018.
- [73] F. Maria Carlucci, L. Porzi, B. Caputo, E. Ricci, and S. Rota Bulo. AutoDIAL: Automatic Domain Alignment Layers. In *ICCV*, 2017.
- [74] T. Minka, J. Verbeek, F. Perronnin, and G. Csurka. Distance-based image classification: Generalizing to new classes at near-zero cost. *IEEE Trans. on PAMI*, 2013.
- [75] R. Mottaghi, X. Chen, X. Liu, N.-G. Cho, S.-W. Lee, S. Fidler, R. Urtasun, and A. Yuille. The role of context for object detection and semantic segmentation in the wild. In *CVPR*, 2014.
- [76] B. Mustafa, A. Loh, J. Freyberg, P. MacWilliams, A. Karthikesalingam, N. Houlsby, and V. Natarajan. Supervised transfer learning at scale for medical imaging. *arXiv preprint arXiv:2101.05913*, 2021.
- [77] G. Neuhold, T. Ollmann, S. Rota Bulò, and P. Kotschieder. The mapillary vistas dataset for semantic understanding of street scenes. In *ICCV*, 2017.
- [78] A. Newell, K. Yang, and J. Deng. Stacked hourglass networks for human pose estimation. In *ECCV*. Springer, 2016.
- [79] J. Ngiam, D. Peng, V. Vasudevan, S. Kornblith, Q. V. Le, and R. Pang. Domain adaptive transfer learning with specialist models. *arXiv preprint arXiv:1811.07056*, 2018.
- [80] A. Nichol, J. Achiam, and J. Schulman. On first-order meta-learning algorithms. *arXiv preprint arXiv:1803.02999*, 2018.
- [81] X. Peng, B. Usman, N. Kaushik, D. Wang, J. Hoffman, and K. Saenko. Visda: A synthetic-to-real benchmark for visual domain adaptation. In *CVPR Workshop*, 2018.
- [82] J. Puigcerver, C. Riquelme, B. Mustafa, C. Renggli, A. S. Pinto, S. Gelly, D. Keysers, and N. Houlsby. Scalable transfer learning with expert models. In *ICLR*, 2020.
- [83] H. Qi, M. Brown, and D. G. Lowe. Low-shot learning with imprinted weights. In *CVPR*, 2018.
- [84] A. Raghu, M. Raghu, S. Bengio, and O. Vinyals. Rapid learning or feature reuse? towards understanding the effectiveness of MAML. *ICLR*, 2020.
- [85] S. Ravi and H. Larochelle. Optimization as a model for few-shot learning. In *ICLR*, 2016.
- [86] S.-A. Rebuffi, H. Bilen, and A. Vedaldi. Learning multiple visual domains with residual adapters. In *NeurIPS*, 2017.
- [87] S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *NeurIPS*, 2015.
- [88] F. Rottensteiner, G. Sohn, M. Gerke, and J. D. Wegner. Theme section — urban object detection and 3d building reconstruction. *ISPRS Journal of Photogrammetry and Remote Sensing*, 93:143 – 144, 2014.
- [89] S. Roy, A. Siarohin, E. Sangineto, S. R. Bulo, N. Sebe, and E. Ricci. Unsupervised domain adaptation using feature-whitening and consensus loss. In *CVPR*, 2019.
- [90] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015.
- [91] P. Russo, F. M. Carlucci, T. Tommasi, and B. Caputo. From source to target and back: symmetric bi-directional adaptive gan. In *CVPR*, 2018.
- [92] K. Saenko, B. Kulis, M. Fritz, and T. Darrell. Adapting visual category models to new domains. In *ECCV*, 2010.
- [93] S. Sankaranarayanan, Y. Balaji, C. D. Castillo, and R. Chellappa. Generate to adapt: Aligning domains using generative adversarial networks. In *CVPR*, 2018.
- [94] A. Shaban, S. Bansal, Z. Liu, I. Essa, and B. Boots. One-shot learning for semantic segmentation. In *arXiv*, 2017.
- [95] E. Shelhamer, J. Long, and T. Darrell. Fully convolutional networks for semantic segmentation. *IEEE Trans. on PAMI*, 2016.
- [96] K. Shmelkov, C. Schmid, and K. Alahari. Incremental learning of object detectors without catastrophic forgetting. In *ICCV*, 2017.
- [97] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus. Indoor segmentation and support inference from rgbd images. In *ECCV*, 2012.
- [98] J. Snell, K. Swersky, and R. Zemel. Prototypical networks for few-shot learning. In *NeurIPS*, 2017.
- [99] S. Song, S. Lichtenberg, and J. Xiao. SUN RGB-D: A RGB-D scene understanding benchmark suite. In *CVPR*, 2015.
- [100] J.-C. Su, S. Maji, and B. Hariharan. When does self-supervision improve few-shot learning? In *ECCV*, 2020.
- [101] B. Sun and K. Saenko. Deep coral: Correlation alignment for deep domain adaptation. In *ECCV Workshop*, 2016.
- [102] C. Sun, A. Shrivastava, S. Singh, and A. Gupta. Revisiting unreasonable effectiveness of data in deep learning era. In *ICCV*, 2017.
- [103] Y. Sun, E. Tzeng, T. Darrell, and A. A. Efros. Unsupervised domain adaptation through self-supervision. *arXiv preprint arXiv:1909.11825*, 2019.
- [104] F. Sung, Y. Yang, L. Zhang, T. Xiang, P. H. Torr, and T. M. Hospedales. Learning to compare: Relation network for few-shot learning. In *CVPR*, 2018.
- [105] Y. Tian, Y. Wang, D. Krishnan, J. B. Tenenbaum, and P. Isola. Rethinking few-shot image classification: a good embedding is all you need? In *ECCV*, 2020.
- [106] Y.-H. Tsai, W.-C. Hung, S. Schuster, K. Sohn, M.-H. Yang, and M. Chandraker. Learning to adapt structured output space for semantic segmentation. In *CVPR*, 2018.
- [107] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell. Adversarial discriminative domain adaptation. In *CVPR*, 2017.
- [108] J. Uijlings, S. Popov, and V. Ferrari. Revisiting knowledge transfer for training object class detectors. In *CVPR*, 2018.
- [109] G. Varma, A. Subramanian, A. Namboodiri, M. Chandraker, and C. Jawahar. Idd: A dataset for exploring problems of autonomous navigation in unconstrained environments. In *Proc. WACV*, 2019.
- [110] H. Venkateswara, J. Eusebio, S. Chakraborty, and S. Panchanathan. Deep hashing network for unsupervised domain adaptation. In *CVPR*, 2017.
- [111] O. Vinyals, C. Blundell, T. Lillicrap, K. Kavukcuoglu, and D. Wierstra. Matching networks for one shot learning. *NeurIPS*, 2016.
- [112] J. Wang, K. Sun, T. Cheng, B. Jiang, C. Deng, Y. Zhao, D. Liu, Y. Mu, M. Tan, X. Wang, W. Liu, and B. Xiao. Deep high-resolution representation learning for visual recognition. *IEEE Trans. on PAMI*, 2020.
- [113] K. Wang, J. H. Liew, Y. Zou, D. Zhou, and J. Feng. Panet: Few-shot image semantic segmentation with prototype alignment. In *ICCV*, October 2019.
- [114] X. Wang, T. Huang, J. Gonzalez, T. Darrell, and F. Yu. Frustratingly simple few-shot object detection. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119, pages 9919–9928. PMLR, 2020.
- [115] S. Waqas Zamir, A. Arora, A. Gupta, S. Khan, G. Sun, F. Shahbaz Khan, F. Zhu, L. Shao, G.-S. Xia, and X. Bai. isaid: A large-scale dataset for instance segmentation in aerial images. In *CVPR Workshops*, 2019.
- [116] P. Weinzaepfel, G. Csurka, Y. Cabon, and M. Humenberger. Visual localization by learning objects-of-interest dense match regression. In *CVPR*, 2019.

- [117] L. Weng. Meta-learning: Learning to learn fast. *lilianweng.github.io/lil-log*, 2018.
- [118] J. Wu, S. Liu, D. Huang, and Y. Wang. Multi-scale positive sample refinement for few-shot object detection. In A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm, editors, *ECCV*. Springer International Publishing, 2020.
- [119] G.-S. Xia, X. Bai, J. Ding, Z. Zhu, S. Belongie, J. Luo, M. Datcu, M. Pelillo, and L. Zhang. Dota: A large-scale dataset for object detection in aerial images. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.
- [120] J. Xiao, A. Owens, and A. Torralba. SUN3D: A database of big spaces reconstructed using SfM and object labels. In *ICCV*, 2013.
- [121] J. Xie, M. Kiefel, M.-T. Sun, and A. Geiger. Semantic instance annotation of urban scenes by 3d to 2d label transfer. In *CVPR*, 2016.
- [122] X. Yan, D. Acuna, and S. Fidler. Neural data server: A large-scale search engine for transfer learning data. *CVPR*, 2020.
- [123] F. Yu, H. Chen, X. Wang, W. Xian, Y. Chen, F. Liu, V. Madhavan, and T. Darrell. Bdd100k: A diverse driving dataset for heterogeneous multitask learning. In *CVPR*, June 2020.
- [124] F. Yu, M. Zhang, H. Dong, S. Hu, B. Dong, and L. Zhang. Dast: Unsupervised domain adaptation in semantic segmentation based on discriminator attention and self-training. In *AAAI*, 2021.
- [125] S. Zagoruyko and N. Komodakis. Wide residual networks. In *BMVC*, 2016.
- [126] A. R. Zamir, A. Sax, W. Shen, L. J. Guibas, J. Malik, and S. Savarese. Taskonomy: Disentangling task transfer learning. In *CVPR*, 2018.
- [127] X. Zhai, J. Puigcerver, A. Kolesnikov, P. Ruysen, C. Riquelme, M. Lucic, J. Djolonga, A. S. Pinto, M. Neumann, A. Dosovitskiy, L. Beyer, O. Bachem, M. Tschannen, M. Michalski, O. Bousquet, S. Gelly, and N. Houlsby. A large-scale study of representation learning with the visual task adaptation benchmark. *arXiv preprint arXiv:1910.04867*, 2019.
- [128] H. Zhang, H. Ouyang, S. Liu, X. Qi, X. Shen, R. Yang, and J. Jia. Human pose estimation with spatial contextual information. In *arXiv*, 2019.
- [129] Y. Zhao, H. Ali, and R. Vidal. Stretching domain adaptation: How far is too far? *arXiv preprint arXiv:1712.02286*, 2017.
- [130] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, and A. Torralba. Scene parsing through ADE20K dataset. In *CVPR*, 2017.
- [131] X. Zhou, D. Wang, and P. Krähenbühl. Objects as points. In *arXiv*, 2019.
- [132] J. Zhu and Y. Fang. Learning object-specific distance from a monocular image. In *ICCV*, 2019.
- [133] B. Zoph, G. Ghiasi, T.-Y. Lin, Y. Cui, H. Liu, E. D. Cubuk, and Q. V. Le. Rethinking pre-training and self-training. In *NeurIPS*, 2020.



**Thomas Mensink** is a Research Scientist at Google since 2019. He obtained his PhD from the University of Grenoble in 2012, working at the LEAR team of INRIA and at Xerox Research Centre Europe. He worked as Assistant Professor (2017-2019) and post-doc (2012-2017) at University of Amsterdam. His research interest include transfer learning for dense prediction tasks and learning visual representations for zero-shot reasoning. He has received multiple best paper awards (ACM Multimedia 2014,

ICMR 2016), a VENI grant (NWO 2015), and the Koenderink Prize for fundamental contributions in computer vision (ECCV 2020).



**Jasper Uijlings** is a Research Scientist at Google since 2016, working in a team led by Vittorio Ferrari. His research topics are on lifelong learning and human-machine collaboration for image annotation. Before that, he worked as a post-doc at the University of Edinburgh with Prof. Vittorio Ferrari and at the University of Trento with Prof. Nicu Sebe. He obtained his PhD in 2011 at the University of Amsterdam under supervision of Prof. Dr. Ir. Remko Scha and Prof. Dr. Ir. Arnold Smeulders.



and transfer learning.

**Michael Gygli** received his PhD from ETH Zurich in 2017. He is a technical co-founder of Cerrion, which uses Computer Vision to increase the stability of manufacturing processes. He is also an experienced computer vision researcher. During this PhD, his time as head of AI at gifs.com and as a Research Scientist at Google he has published over 20 research papers at venues such as CVPR, ICML, ICCV, IJCV, AAI and ACM Multimedia. His main interests are real-time video analysis, image segmentation



**Alina Kuznetsova** received her Ph.D. from Leibniz University of Hannover in 2016. During her studies, she interned at Disney Research, working with Dr. Sung Ju Hwang and Dr. Leonid Sigal, and at Microsoft Research. She has co-authored over 15 technical publications in various fields of computer vision during her PhD and time in Google Research. Her research interests are large-scale annotation and transfer learning.



**Vittorio Ferrari** is a Senior Staff Research Scientist at Google, where he leads a research group on visual learning. He received his PhD from ETH Zurich in 2004, then was a post-doc at INRIA Grenoble (2006-2007) and at the University of Oxford (2007-2008). Between 2008 and 2012 he was an Assistant Professor at ETH Zurich, funded by a Swiss National Science Foundation Professorship grant. In 2012-2018 he was faculty at the University of Edinburgh, where he became a Full Professor in 2016 (now a Honorary Professor). In 2012 he received the prestigious ERC Starting Grant, and the best paper award from the European Conference in Computer Vision. He is the author of over 130 technical publications. He regularly serves as an Area Chair for the major computer vision conferences, he was a Program Chair for ECCV 2018 and a General Chair for ECCV 2020. He is an Associate Editor of the International Journal of Computer Vision, and formerly of IEEE Pattern Analysis and Machine Intelligence. His current research interests are in learning visual models with minimal human supervision, human-machine collaboration, and 3D Deep Learning.



# Appendices

In these appendices we mainly provide more details on the network architectures in [Appendix A](#), generalization experiments in [Appendix B](#), and more extensive experimental results in [Appendix C](#). For all experiments not only the relative transfer gain is shown, but also the absolute value of the task-specific performance metric is shown. In [Appendix D](#) we discuss the additional computation costs of transfer chains. Finally, in [Appendix E](#) we discuss potential data overlap in our collection of datasets.

## APPENDIX A

### BACKBONE AND TASK TYPE SPECIFIC NETWORK ARCHITECTURES

In this section we describe in detail the network architectures used throughout our study.

#### A.1 Backbone Architecture

Transfer learning through pre-training is only possible when the models for all task types share the same backbone architecture. To have meaningful results, we need to choose a backbone which works well across all task types we explore. Therefore we choose the recent high-resolution backbone HRNetV2 [112] (illustrated in [Figure A.1](#)). It has two main advantages. First, HRNetV2 was shown to outperform ResNet [44], ResNeXt [121], Wide ResNet [125], and stacked hourglass networks [78] on three of the tasks we explore: semantic segmentation, keypoint detection, and object detection. Second, its design allows to use relative shallow task type specific heads compared to the number of parameters in the backbone. In [Table 2](#) (in main paper) we provide an overview of the number of trainable parameters. The backbone consists of 69M parameters, making up 87%-99.99% of all parameters depending on the task type and the number of classes.

**Architecture.** The HRNetV2 backbone extends the ResNet architecture to preserve high-resolution spatial features. A regular ResNet consists of blocks organised in four ‘stages’, each reducing the image resolution by factor 2. HRNetV2 follows this design, but after each stage it *also* keeps a parallel high-resolution branch ([Figure A.1](#)). All branches from one stage are fed into the next, so this next stage incorporates information from the representations at different

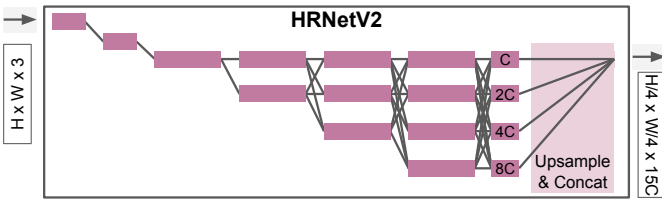
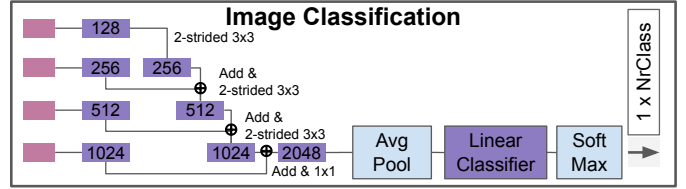


Fig. A.1: Schematic illustration of the HRNet backbone. Each horizontal layer depicts a resolution and each block represents a series of residual units. The blocks are fully connected using all resolutions with either downscaling or upsampling. The final representation is obtained by concatenating all upsampled layers to  $H/4 \times W/4$ .



(a)

Description	Top-1 Acc
ResNet-50 [112]	76.9
HRNetV2-W44 [112]	77.0
<b>ours HRNetV2-W48</b>	<b>79.5</b>

(b)

Fig. A.2: Illustration of the classification head architecture ([A.2a](#)) and table with top-1 accuracies for different networks and backbones on the ILSVRC’12 validation set ([A.2b](#)).

resolutions. The backbone produces 4 feature maps, each with different resolution and number of channels. These are combined into a single feature map by upscaling and concatenating them along the channel dimension. This results in a final output feature map of shape  $W/4 \times H/4 \times 15C$ , where  $W$  and  $H$  are the original image width and height, and  $C = 48$  is the number of channels in the highest resolution output layer. Hence the final feature map has  $15 \times 48 = 720$  channels.

**Training.** In order to validate our backbone architecture we use image classification on ILSVRC’12. We extend the backbone with the classification head proposed in [112] ([Figure A.2a](#)). This head increases the number of channels, while reducing the resolution to  $H/32 \times W/32$ , and then uses an average pooling layer with a linear classifier. The network is trained for 200 epochs, with SGD using momentum and a large batch-size of 4096, learning-rate decay after 60, 120, and 180 epochs.

**Evaluation.** Performance is evaluated using Top-1 accuracy on the ILSVRC’12 validation set.

**Results.** In [Figure A.2b](#) we show the results of our network compared to the ResNet-50 and HRNetV2-W44 architectures from [112]. Our model performs competitively, reaching 79.5% Top-1 accuracy, validating our choice of architecture.

#### A.2 Semantic Segmentation

**Architecture.** For semantic segmentation, we adopt the network head proposed in [112] ([Figure A.3a](#)). It consists of three stages: (1) a  $1 \times 1$  convolution changing the dimensionality of the backbone output to the number of classes  $K$ . This implements a linear classifier at each pixel; (2) a softmax non-linearity; (3) a bi-linear up sampling layer to produce the final predicted segmentation map  $H \times W \times K$ .

**Training.** The network is trained using the cross-entropy loss at each pixel, ignoring the background class:

$$L_{\log} = \frac{1}{N} \sum_{k=1}^N \delta(y_k \neq \text{bg}) \log p(y_k | x_k), \quad (\text{A.1})$$

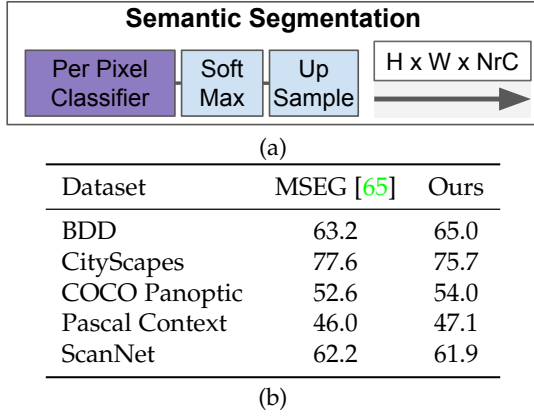


Fig. A.3: Semantic segmentation model and performance. **A.3a** shows an illustration of the head architecture, where purple blocks indicate trainable parameters (in total  $(720 + 1) * NrC$ ) and blue blocks have no trainable parameters. **A.3b** compares our performance against that of the MSEG setup [65] on several datasets of the MSEG collection. Our setup is validated by the fact we perform on par with [65].

where the  $\delta$  denotes the Dirac-delta function to return 1 iff the pixel depicts a not background class, and  $p(y_k|x_k)$  is computed using the softmax function.

Starting from ILSVRC’12 pre-trained weights, we train on the source dataset using SGD with momentum. We use multiple Google Cloud TPU-v3-8 accelerators using synchronized batch norm and a batch size of 32. For almost all sources datasets we use a stepwise learning rate decay, and optimize per source the starting learning rate, the number of steps after which the learning rate is lowered (by a fixed factor of 10), and the total number of training steps. However, we found performance to be unstable for the small SUIM and KITTI datasets. For these we found that switching to a “poly” learning rate policy ( $\text{base\_lr} \times (1 - \frac{\text{curr\_step}}{\text{max\_steps}})^p$ ,  $p = 0.9$ ) stabilizes training, as also done in [14], [65].

**Evaluation.** Performance is evaluated by the Intersection-over-Union (IoU), averaged over classes [30]. During evaluation we process complete images at a single resolution only. We scale each image to match the resolution used during training.

**Results.** To validate our training setup, we compare the performance of our networks on a subset of the MSEG dataset collection [65]. While these datasets are also part of our collection, for this experiment we use the annotation provided in MSEG, which has fewer classes than in our transfer setting setup, *c.f.* Section 5 (in main paper). As Figure A.3b shows, our models perform on par with those of [65], which also use a HRNetV2 backbone and linear semantic segmentation layer, but have different normalisation and data augmentation steps and a different implementation.

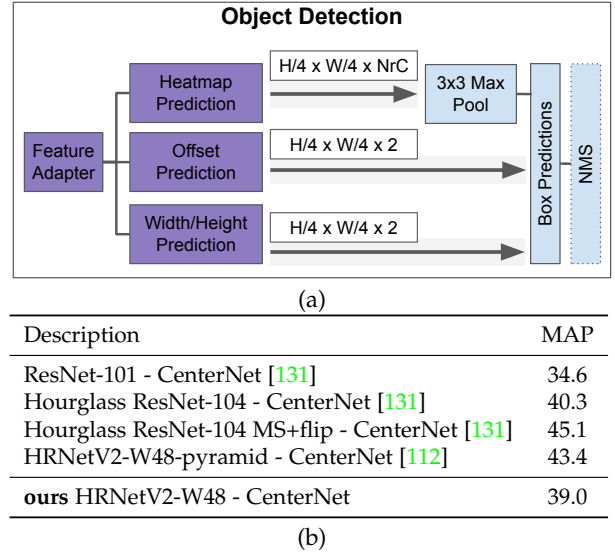


Fig. A.4: Object detection model and performance. **A.4a** illustrates our CenterNet-based [131] object detection head. **A.4b** compares the performance to recent works. Our model performs close to the CenterNet Hourglass ResNet-14 [131] under the same conditions: no data augmentation during evaluation (neither Multi-Scale (MS) nor horizontal flipping (flip)). This demonstrates that we base our transfer learning experiments on a strong modern object detection framework.

### A.3 Object Detection model architecture details

**Architecture.** We follow the CenterNet<sup>3</sup> approach of Zhou et al. [131] (Figure A.4a). For each pixel in the feature map and for each class, a binary classifier evaluates whether this pixel is the center point of an object bounding box. Additionally, at these points we predict a translation correcting offset (in x- and y- coordinates) and the size of the bounding box (width and height). Using center points to predict bounding boxes results in a simpler model than a two-stage architecture like Faster-RCNN [87], while being about as accurate [27], [131].

Following [112], [131] we add a feature adapter (*neck*) between the backbone and the head. This neck consists of 1x1 convolution followed by a 3x3 convolution, batch-normalization [48] and ReLu. The detection CenterNet head outputs three prediction maps, each is the result of a 3x3 convolution, ReLu, and another 3x3 convolution. The three pixel-wise prediction maps are:

- M1 a  $K$ -channel heatmap  $\hat{Y}_{xyc}$ , one per class. Each pixel is assigned the likelihood to be the center on an object of a certain class computed using Gaussian kernel.
- M2 a two-channel offset map  $\hat{O}_p$  representing x- and y- offsets of the point to the real center of the bounding box  $p$ ; and
- M3 a two-channel width/height map  $\hat{S}_p$ , which predicts the width and height of an object centered at the pixel

3. With CenterNet we denote a family of object detection models which predicts boxes via their center points with fully convolutional architectures [27], [131]. The paper of [27] is called CenterNet while the GitHub of [131] is also called CenterNet <https://github.com/xingyizhou/CenterNet>. We mostly follow the approach of [131].

$p$ .

Following [131] the final box predictions are obtained by combining the top  $T = 100$  highest scoring pixels from the class-specific heatmaps, with their respective x/y offsets to define the box-centers. The box dimensions are extracted from the width/height prediction map. Then, optionally, non-maximum suppression (NMS) is applied.

### Training.

During training, three losses are computed, one for each head [131]:

- L1 a logistic regression focal loss [66], [69] for the class heatmap, acting on all pixels:

$$L_{\text{cls}} = -\frac{1}{N} \sum_{xyc} \begin{cases} (1 - \hat{Y}_{xyc})^\alpha \log(\hat{Y}_{xyc}), \\ (1 - Y_{xyc})^\beta (\hat{Y}_{xyc})^\alpha \log(1 - \hat{Y}_{xyc}), \end{cases} \quad Y_{xyc} = \begin{cases} 1, \\ Y_{xyc} \neq 1 \end{cases} \quad (\text{A.2})$$

Here  $Y_{xyc}$  is a sum of Gaussians around from all ground-truth boxes object centers, projected to low-resolution feature map. For  $\alpha = 2$  and  $\beta = 4$  we use default parameters provided in [131]

- L2 a L1-loss for offset prediction, acting only on  $N$  pixels that are close to center points of objects in the ground-truth (those where  $Y_{xyc} = 1$ ); here  $o_k$  are the ground-truth offsets:

$$L_{\text{off}} = \frac{1}{N} \sum_{k=1}^N |\hat{O}_{p_k} - o_k| \quad (\text{A.3})$$

- L3 a L1-loss for width and height prediction, again acting only on  $N$  pixels close to ground-truth centers;  $s_k$  are ground-truth box sizes:

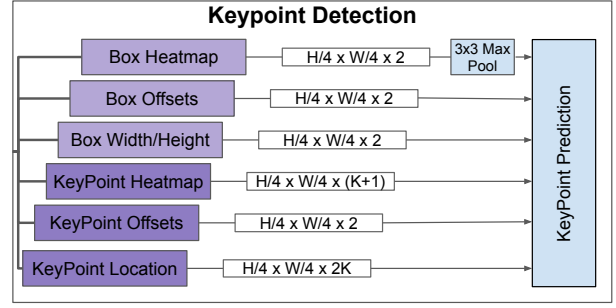
$$L_{\text{size}} = \frac{1}{N} \sum_{k=1}^N |\hat{S}_{p_k} - s_k| \quad (\text{A.4})$$

The final training objective is  $L = L_{\text{cls}} + \lambda_{\text{off}} L_{\text{off}} + \lambda_{\text{size}} L_{\text{size}}$ . We train using the same procedure as for semantic segmentation, but using a batch size of 128 and the Adam optimizer [54] until convergence.

**Evaluation.** We use the COCO definition of mean Average Precision (mAP) [70]. They evaluate mAP at various IoU thresholds and average them. For COCO we do not use Non Maximum Suppression, since this did not improve results on this dataset as also observed by [131] (on other datasets NMS was useful, see details in Appendix C).

**Results.** We validate our setup by training on the COCO training set and evaluating performance on its validation set. We compare to results reported in [131] and for a HRNetV2-W48 backbone with a feature pyramid using the CenterNet variant [112].

As can be seen in Figure A.4b, the best results are obtained by using use data augmentation during evaluation (multi-scale and horizontal flip) or by using a feature pyramid. Without these enhancements, our implementation performs close to the Hourglass ResNet-104 results of [131] (39.0 mAP and 40.3 mAP respectively). Hence we conclude that we can base our transfer learning experiments on a strong, modern object detection framework.



(a)

Description	AP50
Hourglass-104 - CenterNet - single stage [131]	84.2
HRNetV1-W44 - two stage cascade [112]	90.8
our HRNetV2-W48 - CenterNet - single stage	81.1

(b)

Fig. A.5: Keypoint Detection model and performance. A.5a illustrates our model head architecture. A.5b provides a performance comparison. The two stage cascade model first detects persons, crops out their image pixels, and uses a separate network to predict keypoints. We follow the simpler single stage approach used in e.g. [131].

## A.4 Keypoint Detection

**Architecture.** We again follow the CenterNet approach of [131] (Figure A.5a). The keypoint head architecture resembles the object detection architecture, yet with some differences.

We create 6 pixelwise prediction heads, each composed of a 3x3 convolution, ReLU, and another 3x3 convolution. We do not use a neck, but directly feed the feature maps produced by the backbone into these heads. The maps output by the first three heads are analogous to the object detection ones, but for a single class only, to detect the box around the person (or dog depending on the dataset). The remaining three maps are:

- M4 a  $K$ -channel keypoint heatmap, where each channel is the output of a binary classifier predicting one of the  $K$  keypoints;
- M5 a keypoint offset map: a two-channel x/y offset map for a translation correction of the keypoint;
- M6 a keypoint allocation map: a  $2 \times K$ -channel map marking the displacement from the center of an object at this pixel to each of the  $K$  keypoints belonging to that object.

We obtain keypoints for a image by processing these maps: First, a bounding box around the person/dog is obtained by combining the box centers (M1), with the offsets (M2) and the box dimensions (M3). Then, M4 and M5 are combined to create a set of candidate keypoint locations with accompanying confidence scores, within the person/dog bounding box. Finally, M6 is used to associate each candidate keypoint to the person/dog object it belongs to.

**Training.** For M1-M3 we use the same losses L1-L3 as for object detection. For M4-M6 we add the following [131]:

- L4 a focal loss is used for multi-class joint classification, analogous to object detection loss L1 as described by



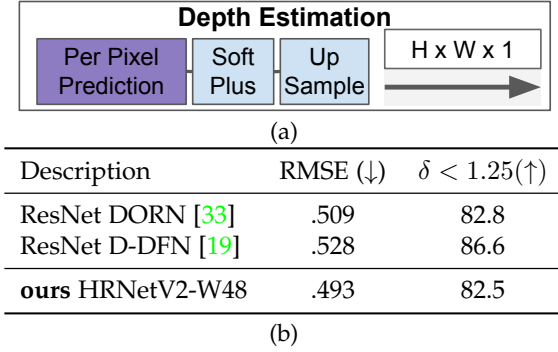


Fig. A.6: Monocular Depth Estimation. A.6a illustrates the task head architecture. A.6b shows a comparison of performance on NYUDepth-V2 [97] dataset using a RMSE and  $\delta < 1.25$  accuracy as performance measures. While our architecture features a light-weighted head, it still outperforms depth-specific architectures in terms of RMSE and is on par with  $\delta < 1.25$ .

Eq. (A.2);

L5 a L1-loss is used for offset prediction, analogous to object detection loss L2 as described by Eq. (A.3);

L6 a L1-loss is used to directly regress the initial keypoint locations w.r.t. the object centers predicted by M1+M2.

We largely follow the training setup for object detection, using a batch size of 128 and the Adam optimizer [54]. We set the starting learning rate and number of steps per dataset.

**Evaluation.** We report the mean Average Precision at 0.5 Object Keypoint Similarity (OKS), as defined by the COCO challenge [1].

**Results.** To validate our setup we train for keypoint prediction on the COCO dataset. In Figure A.5b we report the performance of our network architecture and compare to [131]. They use as backbone a stacked hourglass [78] consisting of two consecutive hourglass models, which they call Hourglass-104. Their model yields 84.2 AP50.<sup>4</sup> For completeness, we include the results of a more complex two stage cascade keypoint detection approach [112], which obtains very good performance. In this paper we use the HRNetV2 backbone with the simpler single-stage CenterNet heads. Our model yields 81.1 mAP, which is near [131] and thus strong enough for our transfer learning exploration.

## A.5 Depth Estimation

**Architecture.** For monocular depth estimation, we modify the architecture for semantic segmentation. We propose to add a single regression layer on top of the output of the backbone, followed by a soft-plus layer and a bi-linear upsampling layer to match the original resolution (Figure A.6a). The softplus activation converts the logit value to depth:  $\log(\exp(x) + 1)$  and is also used in [40], [132]. This is a differentiable clipping function which ensures that depth predictions are positive.

4. Result obtained via correspondence with the authors: The public available model in the zoo obtaining 64.0 AP50 on `coco-test`, obtains 84.2 mAP on `coco-val`, when evaluated without flipping, see: <https://github.com/xingyizhou/CenterNet>

**Training.** To train the network, we combine two losses following [19], [68]:

- a L1-loss between the predicted depth map and the ground truth map at each pixel;

$$L_{L1} = \frac{1}{N} \sum_{k=1}^N |y_k - \hat{y}_k| \quad (\text{A.5})$$

- a smoothness loss which encourages differences in neighbouring depth pixels to be the same in the predicted depth map and the ground-truth. More precisely, we take the derivative in x-direction for both the predicted and ground-truth maps, and compare them with an L1-loss. We repeat this for the y-direction. Resulting in:

$$L_{\text{smooth}} = \frac{1}{N} \sum_{k=1}^N |(\nabla_x y)_k - (\nabla_x \hat{y})_k| + |(\nabla_y y)_k - (\nabla_y \hat{y})_k|, \quad (\text{A.6})$$

where  $y$  denotes the ground-truth depth map,  $\hat{y}$  the predicted depth map and  $N$  the number of pixels.

For training we use a batch size of 64 and the Adam optimizer [54].

**Evaluation.** We validate on the NYUDepthV2 [97] dataset, using the root mean squared error (RMSE) metric and the  $\delta < 1.25$  accuracy, where  $\delta = \max(\frac{\hat{z}}{z}, \frac{z}{\hat{z}})$  is a measure of relative accuracy defined in [64].

**Results.** We compare our proposed network to two recent ResNet based models: [33] which uses depth-specific losses based on ordinal regression; and [19] which uses the same losses as we do, but with a depth-specific network architecture on top of the ResNet backbone. The results are presented in Figure A.6b. We observe that our model is slightly better than both others, when measured using the root mean squared error (RMSE), while slightly behind on the  $\delta < 1.25$  measure. Hence, we conclude that our light-weight depth prediction head is well suited for monocular depth estimation.

## APPENDIX B ADDITIONAL SCENARIOS

In this section we perform several additional experiments to verify whether our work generalizes to different scenarios. In particular, while we optimized image resolution per dataset, *c.f.* Section 4.1 (in main paper), we explore what happens when we fix the image resolution across all datasets. In addition, we partially redo experiments using ResNet50 [44] as backbone. Finally, we explore self-supervision in our transfer learning chain.

**Fixed image resolution.** We redo the segmentation experiments in the small target training setting (i.e. Table 4a (in main paper)) but now fixing the image resolution to be  $713 \times 713$  pixels across all datasets. Results are shown in Table A.1. The average accuracy across all experiments goes down by -0.01 IoU. However, when comparing the results tables (optimized vs fixed resolution) we qualitatively observe that all transfer patterns remain very similar.

Quantitatively, for 88% of the experiments with significant positive or negative gains in Table 4a (in main paper),

To: Segmentation	Aerial			Consumer			Driving						Indoor			Underwater			Synthetic		ILSVRC'12
	ISPRS 40K	iSAID 20K	PContext 5K	PVOC 10K	ADE20K 20K	COCO 120K	KITTI 20K	CamVid 30K	City 30K	IDD 7K	BDD 7K	Mapillary 18K	SceneNet 5K	SUN 1.5K	SUN 1.5K	WITZ 2K	vGallery 40K	Search 1.2M			
ISPRS	-	0.48	0.43	0.42	0.44	0.45	0.45	0.44	0.45	0.43	0.45	0.44	0.41	0.41	0.44	0.45	0.42	0.29	0.46		
iSAID	0.36	-	0.37	0.35	0.39	0.39	0.38	0.34	0.37	0.39	0.37	0.36	0.37	0.37	0.39	0.35	0.34	0.39	0.41		
PContext	0.21	0.24	-	0.36	0.26	0.41	0.23	0.23	0.23	0.23	0.23	0.23	0.22	0.20	0.23	0.23	0.23	0.06	0.25		
PVOC	0.34	0.40	0.35	-	0.43	0.70	0.40	0.39	0.38	0.40	0.40	0.39	0.37	0.33	0.39	0.40	0.40	0.07	0.45		
ADE20K	0.21	0.22	0.23	0.21	-	0.21	0.22	0.22	0.22	0.22	0.21	0.22	0.22	0.21	0.21	0.22	0.22	0.07	0.22		
COCO	0.29	0.30	0.31	0.30	0.30	-	0.30	0.30	0.29	0.29	0.29	0.29	0.28	0.26	0.29	0.30	0.30	0.07	0.31		
KITTI	0.44	0.44	0.45	0.46	0.50	0.46	-	0.46	0.52	0.48	0.47	0.51	0.44	0.44	0.46	0.46	0.45	0.35	0.45		
CamVid	0.47	0.48	0.46	0.50	0.52	0.52	0.49	-	0.54	0.49	0.51	0.57	0.48	0.47	0.47	0.50	0.50	0.35	0.49		
City	0.40	0.42	0.42	0.41	0.43	0.46	0.43	0.43	-	0.47	0.47	0.52	0.39	0.37	0.41	0.42	0.41	0.25	0.42		
IDD	0.36	0.39	0.39	0.39	0.40	0.42	0.39	0.38	0.41	-	0.41	0.44	0.36	0.36	0.38	0.39	0.38	0.20	0.39		
BDD	0.40	0.43	0.46	0.44	0.49	0.50	0.45	0.45	0.50	0.51	-	0.57	0.39	0.39	0.42	0.43	0.42	0.25	0.44		
Mapillary	0.24	0.25	0.26	0.25	0.26	0.28	0.25	0.25	0.29	0.28	0.27	-	0.23	0.23	0.25	0.25	0.25	0.10	0.26		
SUN	0.18	0.19	0.24	0.22	0.33	0.32	0.20	0.18	0.18	0.18	0.18	0.18	-	0.32	0.20	0.20	0.18	0.06	0.20		
ScanNet	0.12	0.14	0.16	0.15	0.23	0.23	0.14	0.13	0.13	0.13	0.13	0.13	0.24	-	0.14	0.13	0.13	0.05	0.15		
SUNM	0.49	0.53	0.52	0.55	0.50	0.56	0.51	0.51	0.50	0.52	0.53	0.50	0.50	0.47	-	0.52	0.50	0.28	0.54		
vKITTI2	0.54	0.55	0.54	0.54	0.55	0.56	0.56	0.56	0.57	0.58	0.56	0.56	0.54	0.53	0.55	-	0.55	0.43	0.55		
vGallery	0.75	0.76	0.76	0.76	0.76	0.76	0.76	0.76	0.76	0.76	0.76	0.76	0.76	0.76	0.75	0.76	-	0.63	0.76		

To: Segmentation	Aerial			Consumer			Driving						Indoor			Underwater			Synthetic		ILSVRC'12
	ISPRS 40K	iSAID 20K	PContext 5K	PVOC 10K	ADE20K 20K	COCO 120K	KITTI 20K	CamVid 30K	City 30K	IDD 7K	BDD 7K	Mapillary 18K	SceneNet 5K	SUN 1.5K	SUN 1.5K	WITZ 2K	vGallery 40K	Search 1.2M			
ISPRS	-	5	-6	-10	-5	-2	-3	-4	-1	-6	-1	-4	-10	-12	-5	-2	-8	-38			
iSAID	-13	-	-9	-15	-14	-4	-8	-17	-11	-5	-9	-11	-10	-9	-4	-14	-17	-53			
PContext	-16	-5	-	-44	13	63	-6	-9	-9	-9	-8	-9	-12	-19	-10	-7	-7	-77			
PVOC	-25	-11	-44	-	-5	54	-11	-15	-16	-12	-12	-14	-18	-28	-13	-12	-13	-84			
ADE20K	-5	-2	0	-7	-	20	-2	-2	-1	-1	-5	-3	-4	-8	-5	-1	-3	-70			
COCO	-6	-1	0	-1	-3	-	-2	-3	-4	-5	-6	-7	-9	-15	-5	-3	-2	-77			
KITTI	-2	-1	0	3	11	4	-	4	16	9	6	13	0	-2	3	2	1	-21			
CamVid	-3	-1	-6	1	7	6	0	-	10	1	4	16	-3	-4	-3	1	3	-28			
City	-6	-2	0	-4	1	7	1	1	-	10	10	22	-8	-12	-4	-2	-2	-42			
IDD	-6	-1	2	1	2	8	0	-2	6	-	5	14	-7	-8	-3	1	-1	-49			
BDD	-8	-3	5	1	11	13	2	3	13	16	-	31	-11	-11	-5	-2	-3	-44			
Mapillary	-7	-3	1	-3	2	11	-3	-3	11	11	7	-	-11	-17	-4	-4	-4	-63			
SUN	-12	-6	18	7	63	59	-4	-10	-10	-12	-9	-10	-	-59	-2	-4	-11	-69			
ScanNet	-21	-8	11	1	55	57	-6	-9	-11	-9	-11	-12	66	-	-6	-12	-14	-64			
SUNM	-8	0	-3	2	-7	4	-4	-5	-6	-3	-1	-6	-7	-11	-	-3	-6	-47			
vKITTI2	-1	0	-2	-1	1	1	1	2	3	5	2	3	-2	-5	0	-	0	-21			
vGallery	-1	0	1	0	1	0	0	0	1	1	0	1	1	0	-1	0	-	-16			

TABLE A.1: HRNetV2 backbone using a fixed crop ( $713 \times 713$ ) instead of dataset specific crops. Absolute performance and the corresponding relative transfer gains for semantic segmentation as both a source and target task type. Top table: mean Intersection-over-Union. Bottom table: the corresponding relative transfer gains.

similar significant gains are observed at a fixed image resolution in Table A.1. More importantly, for all experiments with significant positive transfer gains, we find that the best source remains the best source. The only exception is for SUN RGB-D as target: the best three sources are the same but have a different order. All these sources yield high transfer gains. We conclude that our analysis (Section 6 (in main paper)) holds when fixing the image resolution.

**ResNet50.** We change the backbone to ResNet50 [44] and redo the experiments in the small target training setting for segmentation (*i.e.* redo Table 4a (in main paper)) and partially for detection (*i.e.* redo Table 4c (in main paper)). Results are shown in Table A.2a.

For segmentation, on average results are significantly worse (-0.11 IoU) compared to the HRNetV2 backbone used

in our main experiments. But again, most transfer patterns stay the same. Quantitatively, for 81% of the experiments with significant positive or negative gains with HRNetV2 backbone (Table 4a (in main paper)), similar gains are observed for ResNet50. In fact, gains are stronger for 69% of all experiments, both in positive and negative direction. There are some difference though: generally for ResNet50 there are more source-target combinations with significant positive transfer gains. This is for iSAID as a target dataset, vKITTI2 as a source for *driving*, and for *consumer* as a source for *consumer*, *driving*, and *indoor*. Measured quantitatively, in 13% of the experiments where we before saw negative transfer gains, we now see positive transfer gains. This suggests that ResNet50 benefits more from transfer learning. We hypothesize this is because its activations are at a lower resolution, and training on a segmentation source dataset makes the network better suited for localized predictions.

For detection (Table A.3) there are a few more changes when using ResNet50. Within detection, COCO remains a good source, VOC07 becomes less good (but not negative), while BDD now yields positive transfer instead of negative for VOC07 and Underwater Trash as a target. When using segmentation as a source, all but one sources which had positive transfer gains using HRNetV2 still yield positive transfer. More interestingly, the number of source-target combinations which yield positive transfer gains are doubled (from 6 to 12). Again, this suggests training on segmentation makes the ResNet50 model better suited for localized predictions.

To conclude, ResNet50 models perform less than HRNetV2 in terms of the absolute performance, however ResNet50 generally benefits more from transfer learning, possibly due to improving its ability for localized predictions. At the same time, the overall trends which we observed in Section 6 (in main paper) still hold.

**Self-supervised ILSVRC'12 training.** We now redo the experiments with ResNet50, but instead we start from a checkpoint which was obtained by using self-supervised learning on ILSVRC'12 using the publicly available SimCLR V2 implementation [15]. Starting from these weights, we train the sources fully supervised as before. Results are shown in Table A.2b.

We compare results with our other ResNet50 experiments (*i.e.* Table A.2a (in main paper)). We see that the best or top 3 sources remain the same for all experiments with positive transfer gains, while most previously observed patterns hold. However, there are noticeably fewer positive gains overall, some even turning into negative gains. This is especially visible for *consumer* as source and *driving* as target. But a closer look reveals that the most significant changes happens on the ILSVRC'12 baseline: results starting from self-supervised weights are on average 0.01 IoU higher than starting from fully supervised classification weights, while results for the full transfer chains are comparable. This manifests itself as lower measured transfer gains. The only exception is iSAID, where the self-supervised ILSVRC'12 baseline is 0.03 IoU lower, while results for the full transfer chain are comparable. Hence here we measure higher transfer gains.

To conclude, the main observation is that the self-supervised ILSVRC'12 pre-training generally yields slightly

	Aerial										Consumer										Driving										Indoor										Underwater										Synthetic										ILSVRC'12																																		
	ISPRS 4.0K	ISAD 2.8K	PContext 3K	PVOC 10K	ADE20K 20K	COCO 120K	KITTI 120	CamVid 300	City 3.0K	IDD 7K	BDD 7K	Mapillary 18K	SUN 5K	ScanNet 1.5K	SUM 1.5K	vkITTI2 42K	vGallery 44K	Scratch	ISPRS 4.0K	ISAD 2.8K	PContext 3K	PVOC 10K	ADE20K 20K	COCO 120K	KITTI 120	CamVid 300	City 3.0K	IDD 7K	BDD 7K	Mapillary 18K	SUN 5K	ScanNet 1.5K	SUM 1.5K	vkITTI2 42K	vGallery 44K	Scratch	ISPRS 4.0K	ISAD 2.8K	PContext 3K	PVOC 10K	ADE20K 20K	COCO 120K	KITTI 120	CamVid 300	City 3.0K	IDD 7K	BDD 7K	Mapillary 18K	SUN 5K	ScanNet 1.5K	SUM 1.5K	vkITTI2 42K	vGallery 44K	Scratch	ISPRS 4.0K	ISAD 2.8K	PContext 3K	PVOC 10K	ADE20K 20K	COCO 120K		KITTI 120	CamVid 300	City 3.0K	IDD 7K	BDD 7K	Mapillary 18K	SUN 5K	ScanNet 1.5K	SUM 1.5K	vkITTI2 42K	vGallery 44K	Scratch	ISPRS 4.0K	ISAD 2.8K	PContext 3K	PVOC 10K	ADE20K 20K	COCO 120K	KITTI 120	CamVid 300	City 3.0K	IDD 7K	BDD 7K	Mapillary 18K	SUN 5K	ScanNet 1.5K	SUM 1.5K	vkITTI2 42K	vGallery 44K	Scratch				
ISPRS	-	0.41	0.40	0.40	0.38	0.41	0.38	0.39	0.38	0.40	0.40	0.41	0.38	0.36	0.41	0.38	0.38	0.31	0.40	ISPRS	-	2	0	0	-5	1	-5	-4	-5	-1	0	1	-5	-11	1	-6	-6	-22	ISPRS	-	4	-10	-9	-10	-5	-9	-9	-11	-10	-11	-5	-11	-21	-9	-12	-16	-30	ISPRS	-	0.42	0.39	0.40	0.39	0.41	0.40	0.40	0.39	0.39	0.41	0.39	0.34	0.39	0.38	0.37	0.31	0.44	ISPRS	-	7	-15	12	6	20	26	16	17	25	18	20	14	13	26	27	6	-36

(a) ResNet50 Classification on ILSVRC'12

	Aerial										Consumer										Driving										Indoor										Underwater										Synthetic										ILSVRC'12														
	ISPRS 4.0K	ISAD 2.8K	PContext 3K	PVOC 10K	ADE20K 20K	COCO 120K	KITTI 120	CamVid 300	City 3.0K	IDD 7K	BDD 7K	Mapillary 18K	SUN 5K	ScanNet 1.5K	SUM 1.5K	vkITTI2 42K	vGallery 44K	Scratch	ISPRS 4.0K	ISAD 2.8K	PContext 3K	PVOC 10K	ADE20K 20K	COCO 120K	KITTI 120	CamVid 300	City 3.0K	IDD 7K	BDD 7K	Mapillary 18K	SUN 5K	ScanNet 1.5K	SUM 1.5K	vkITTI2 42K	vGallery 44K	Scratch	ISPRS 4.0K	ISAD 2.8K	PContext 3K	PVOC 10K	ADE20K 20K	COCO 120K	KITTI 120	CamVid 300	City 3.0K	IDD 7K	BDD 7K	Mapillary 18K	SUN 5K	ScanNet 1.5K	SUM 1.5K	vkITTI2 42K	vGallery 44K	Scratch	ISPRS 4.0K	ISAD 2.8K	PContext 3K	PVOC 10K	ADE20K 20K	COCO 120K		KITTI 120	CamVid 300	City 3.0K	IDD 7K	BDD 7K	Mapillary 18K	SUN 5K	ScanNet 1.5K	SUM 1.5K	vkITTI2 42K	vGallery 44K	Scratch		
ISPRS	-	0.42	0.39	0.40	0.39	0.41	0.40	0.40	0.39	0.39	0.41	0.39	0.34	0.39	0.38	0.37	0.31	0.44	ISPRS	-	4	-10	-9	-10	-5	-9	-9	-11	-10	-11	-5	-11	-21	-9	-12	-16	-30	ISPRS	-	7	-15	12	6	20	26	16	17	25	18	20	14	13	26	27	6	-36	ISPRS	-	0.42	0.39	0.40	0.39	0.41	0.40	0.40	0.39	0.39	0.41	0.39	0.34	0.39	0.38	0.37	0.31	0.44

(b) ResNet50 Self Supervised on ILSVRC'12

TABLE A.2: Comparison of using different network architectures and pre-training task: (a) ResNet50 pre-trained on ILSVRC'12 classification; (b) ResNet50 trained self-supervised on ILSVRC'12 training set [16]. Absolute performance and the corresponding relative transfer gains for semantic segmentation as both a source and target task type. Left tables: mean Intersection-over-Union. Right tables: the corresponding relative transfer gains.

	Detection				Segmentation										ILSVRC'12									
	VOC07 5K	BDD 70K 6K	COCO 120K	PContext 5K	SUN 5K	BDD 7K	Mapillary 18K	vkITTI2 42K	COCO 120K	SUM 1.5K	PContext 5K	SUN 5K	BDD 7K	Mapillary 18K		vkITTI2 42K								
VOC07	-	0.11	0.14	0.41	0.10	0.20	0.10	0.10	0.15	0.07	0.23	0.13	VOC07	-	20	4	206	-23	47	-29	-23	15	-47	76
UWT	0.25	-	0.26	0.35	0.23	0.23	0.26	0.25	0.25	0.22	0.28	0.24	UWT	1	-	8	43	-5	-7	8	5	4	-9	14
BDD	0.03	0.03	-	0.06	0.02	0.03	0.02	0.03	0.04	0.03	0.03	0.03	BDD	1	-5	-	93	-19	-16	-30	10	45	12	-6
COCO	0.04	0.04	0.03	-	0.03	0.04	0.03	0.04	0.04	0.04	0.07	0.04	COCO	2	-7	-23	-	-23	-5	-30	-15	5	-12	61

(a) Average Precision

	Detection				Segmentation										ILSVRC'12									
	VOC07 5K	BDD 70K 6K	COCO 120K	PContext 5K	SUN 5K	BDD 7K	Mapillary 18K	vkITTI2 42K	COCO 120K	SUM 1.5K	PContext 5K	SUN 5K	BDD 7K	Mapillary 18K		vkITTI2 42K								
VOC07	-	0.11	0.14	0.41	0.10	0.20	0.10	0.10	0.15	0.07	0.23	0.13	VOC07	-	20	4	206	-23	47	-29	-23	15	-47	76
UWT	0.25	-	0.26	0.35	0.23	0.23	0.26	0.25	0.25	0.22	0.28	0.24	UWT	1	-	8	43	-5	-7	8	5	4	-9	14
BDD	0.03	0.03	-	0.06	0.02	0.03	0.02	0.03	0.04	0.03	0.03	0.03	BDD	1	-5	-	93	-19	-16	-30	10	45	12	-6
COCO	0.04	0.04	0.03	-	0.03	0.04	0.03	0.04	0.04	0.04	0.07	0.04	COCO	2	-7	-23	-	-23	-5	-30	-15	5	-12	61

(b) Relative transfer gain

TABLE A.3: Absolute metrics and the corresponding relative transfer gains for object detection as a target task type when using ResNet50 as backbone.



	To: Segmentation				To: Segmentation		
	(segmentation)	(self-supervised)	(self-supervised)		(segmentation)	(self-supervised)	(self-supervised)
ISPRS	0.41	0.34	0.44	ISPRS	-6	-23	
ISAID	0.30	0.21	0.23	ISAID	27	-12	
PCContext	0.34	0.16	0.18	PCContext	86	-13	
PVOC	0.61	0.30	0.31	PVOC	98	-3	
ADE20K	0.20	0.11	0.14	ADE20K	38	-25	
COCO	-	0.17	0.19	COCO	-	-10	
KITTI	0.39	0.32	0.41	KITTI	-4	-21	
CamVid	0.41	0.32	0.43	CamVid	-5	-25	
City	0.39	0.30	0.37	City	3	-21	
IDD	0.35	0.27	0.33	IDD	6	-18	
BDD	0.45	0.30	0.38	BDD	18	-20	
Mapillary	0.24	0.16	0.21	Mapillary	14	-25	
SUN	0.25	0.11	0.14	SUN	87	-21	
ScanNet	0.16	0.07	0.09	ScanNet	76	-24	
SUIM	0.51	0.40	0.50	SUIM	3	-19	
vKITTI2	0.49	0.45	0.52	vKITTI2	-6	-13	
vGallery	0.68	0.63	0.71	vGallery	-4	-12	

TABLE A.4: Self-supervised Transfer Chain. Mean IoU (left), Relative (right).

better segmentation results than fully supervised classification pre-training, which is in line with [29], [133]. Still, overall patterns remain the same and the overall conclusions in Section 6 (in main paper) remain unaltered.

**Self-supervised Transfer Chain.** We do a single experiment where we use SimCLR [15] to create a self-supervised transfer chain: ILSVRC’12 self-supervised  $\rightarrow$  COCO self-supervised  $\rightarrow$  target. We first verify whether the self-supervision works for COCO image classification as a target following a standard self-supervised evaluation protocol [15], [18], [41]: we take the trained backbone, freeze its weights, attach a linear classifier, and train on COCO image classification. We do this for both ILSVRC’12 weights and ILSVRC’12  $\rightarrow$  COCO weights. We find that the additional self-supervised training on COCO leads to a small improvement in classification accuracy of 0.4%.

We now test the transfer chain on segmentation for the small target training setting. Results are shown in Table A.4. Interestingly, we find that this chain is worse than directly training from ILSVRC’12 self-supervised weights for *all* target datasets. On average it is 0.07 IoU worse, while even for COCO as a target results are 0.02 IoU worse.

This result suggests that self-supervised pre-training using SimCLR [15] is biased towards image classification. Furthermore, image classification results on self-supervised models are not very predictive of performance on other tasks, as also shown in the dedicated study of [29].

## APPENDIX C COMPLETE RESULT TABLES

This appendix reports more extensive results for all of our experiments, *i.e.* including both absolute performance of the task type specific metric as well as relative transfer gain calculated *w.r.t.* the ILSVRC’12 column of the absolute performance, *c.f.* Eq. (2) (in main paper). The experiments and metrics are summarized as follows:

- For semantic segmentation we use Intersection-over-Union (IoU), averaged over classes [30]. See Table A.5 and Table A.6.
- For semantic segmentation we also include experiments using ResNet50 as backbone, both when pre-trained on the ILSVRC’12 image classification task, and when self-supervisedly trained on the ILSVRC’12 training set. The results are in Table A.2. For self-supervised training we follow the SimCLR [15], [16] approach. For both ResNet models we use the publicly available checkpoints from the SimCLR [repo](#).
- For semantic segmentation we also include experiments using the HRNetV2 backbone, but with a fixed image cropping instead of the dataset specific cropping used in the main experiments (*c.f.* Section 4.1 (in main paper)). The results are in Table A.1.
- For object detection, we use mean Average Precision (mAP) at a threshold of 0.5 (box-based) IoU [30] for Pascal VOC and Underwater Trash. For COCO and BDD we use the COCO mAP variant [70], which evaluates mAP at multiple IoU thresholds and averages them. As earlier observed in [131], the common post-processing stage of Non-Maximum Suppression (NMS) did not improve results on COCO or BDD, but we found it beneficial on Pascal VOC and SUIM. For the latter two dataset we use NMS at an IoU of 0.3. Results are in Table A.7.
- For object detection we have also used ResNet50 as a backbone, using pre-training on ILSVRC’12 image classification. The results are in Table A.3.
- For keypoint detection we report the mean Average Precision at 0.5 Object Keypoint Similarity (OKS), as defined by the COCO challenge [1]. Results are in Table A.8.
- For depth estimation we use the Root Mean Square Error (RMSE), linear version, as defined in [28] (lower is better), and also the standard metric  $\delta < 1.25$ , where  $\delta = \max(\frac{\hat{z}}{z}, \frac{z}{\hat{z}})$  is a measure of relative accuracy as defined in [64] (higher is better). Results are in Table A.9.

To: Segmentation	Aerial		Consumer			Driving			Indoor			Underwater			Synthetic			vGallery	Multi	Scratch	IS/NRC12
	ISPRS 4DK	ISAD	PContext	ADE20K	COCO	City	IDD	BDD	Mapillary	SUN	ScanNet	Sum	WTT2	WTT3	WTT4	WTT5	WTT6				
ISPRS	-	0.49	0.45	0.43	0.45	0.46	0.45	0.43	0.46	0.43	0.45	0.45	0.42	0.42	0.44	0.44	0.45	0.43	0.48	0.31	0.46
ISAD	0.35	-	0.39	0.37	0.37	0.39	0.37	0.34	0.36	0.39	0.37	0.39	0.37	0.39	0.36	0.38	0.39	0.34	0.35	0.20	0.42
PContext	0.24	0.26	-	0.36	0.28	0.39	0.26	0.25	0.25	0.26	0.25	0.26	0.23	0.20	0.25	0.25	0.26	0.26	0.37	0.06	0.28
PVOC	0.43	0.50	0.63	-	0.47	0.71	0.49	0.47	0.47	0.47	0.47	0.47	0.40	0.37	0.47	0.47	0.48	0.68	0.08	0.54	
ADE20K	0.22	0.23	0.22	0.21	-	0.26	0.23	0.22	0.23	0.23	0.22	0.23	0.22	0.22	0.22	0.22	0.22	0.26	0.08	0.23	
COCO	0.29	0.30	0.30	0.29	0.29	-	0.30	0.30	0.29	0.29	0.29	0.29	0.28	0.26	0.29	0.30	0.30	-	0.07	0.31	
KITTI	0.45	0.47	0.47	0.47	0.49	0.49	-	0.47	0.51	0.50	0.49	0.51	0.43	0.45	0.46	0.46	0.47	0.50	0.36	0.46	
CamVid	0.49	0.50	0.51	0.50	0.54	0.50	0.54	-	0.56	0.54	0.55	0.58	0.51	0.51	0.49	0.49	0.51	0.55	0.40	0.49	
City	0.40	0.42	0.43	0.42	0.44	0.45	0.43	0.43	-	0.47	0.46	0.51	0.41	0.39	0.42	0.42	0.42	0.49	0.26	0.43	
IDD	0.36	0.38	0.39	0.39	0.39	0.41	0.38	0.37	0.40	-	0.40	0.43	0.37	0.35	0.38	0.38	0.37	0.42	0.23	0.38	
BDD	0.41	0.45	0.46	0.46	0.49	0.51	0.46	0.46	0.50	0.51	-	0.59	0.42	0.40	0.44	0.45	0.43	0.55	0.27	0.45	
Mapillary	0.25	0.26	0.27	0.26	0.27	0.29	0.26	0.25	0.29	0.29	0.28	-	0.24	0.23	0.26	0.26	0.25	-	0.11	0.26	
SUN	0.18	0.19	0.22	0.20	0.31	0.32	0.20	0.18	0.18	0.18	0.18	-	0.33	0.20	0.18	0.19	0.19	0.35	0.07	0.20	
ScanNet	0.12	0.13	0.15	0.14	0.19	0.22	0.14	0.14	0.13	0.13	0.13	0.14	-	0.24	0.14	0.13	0.13	-	0.05	0.14	
SUIM	0.47	0.53	0.53	0.54	0.53	0.55	0.52	0.50	0.51	0.53	0.52	0.50	0.49	0.48	-	0.53	0.51	-	0.29	0.53	
vkITTI2	0.54	0.56	0.55	0.55	0.56	0.56	0.56	0.56	0.56	0.56	0.56	0.57	0.55	0.55	0.55	-	0.56	-	0.47	0.56	
vGallery	0.76	0.78	0.78	0.77	0.78	0.78	0.78	0.78	0.78	0.78	0.78	0.77	0.78	0.77	0.77	0.78	-	-	0.70	0.78	

(a) Small target training set.

To: Segmentation	Aerial		Consumer			Driving			Indoor			Underwater			Synthetic			vGallery	Multi	Scratch	IS/NRC12
	ISPRS 4DK	ISAD	PContext	ADE20K	COCO	City	IDD	BDD	Mapillary	SUN	ScanNet	Sum	WTT2	WTT3	WTT4	WTT5	WTT6				
ISPRS	-	0.50	0.46	0.46	0.47	0.47	0.45	0.45	0.44	0.45	0.47	0.45	0.44	0.44	0.45	0.46	0.45	0.47	0.31	0.47	
ISAD	0.82	-	0.82	0.81	0.80	0.80	0.82	0.81	0.82	0.82	0.81	0.82	0.80	0.78	0.81	0.81	0.82	0.83	0.46	0.82	
PContext	0.47	0.47	-	0.49	0.47	0.51	0.47	0.47	0.47	0.46	0.46	0.47	0.46	0.45	0.46	0.47	0.47	0.49	0.25	0.47	
PVOC	0.78	0.77	0.77	-	0.77	0.79	0.78	0.78	0.77	0.77	0.77	0.78	0.77	0.76	0.78	0.78	0.78	0.78	0.49	0.78	
ADE20K	0.40	0.40	0.40	0.39	-	0.41	0.41	0.40	0.41	0.41	0.40	0.41	0.40	0.39	0.40	0.40	0.40	0.42	0.25	0.41	
COCO	0.51	0.51	0.51	0.51	0.51	-	0.51	0.51	0.51	0.51	0.51	0.51	0.51	0.51	0.51	0.51	0.51	0.53	0.34	0.51	
KITTI	0.46	0.47	0.48	0.47	0.48	0.48	-	0.46	0.50	0.48	0.48	0.50	0.43	0.44	0.47	0.45	0.46	0.50	0.34	0.45	
CamVid	0.53	0.51	0.54	0.50	0.56	0.53	0.53	-	0.57	0.57	0.56	0.62	0.51	0.53	0.54	0.55	0.56	0.59	0.42	0.55	
City	0.54	0.54	0.56	0.53	0.53	0.53	0.54	0.54	-	0.55	0.54	0.60	0.53	0.51	0.56	0.55	0.54	0.57	0.39	0.57	
IDD	0.52	0.52	0.51	0.51	0.52	0.52	0.52	0.52	-	0.52	0.53	0.52	0.52	0.50	0.52	0.52	0.52	0.53	0.41	0.52	
BDD	0.64	0.63	0.62	0.63	0.64	0.64	0.64	0.65	0.65	0.65	-	0.64	0.62	0.60	0.64	0.64	0.64	0.66	0.43	0.64	
Mapillary	0.48	0.48	0.47	0.47	0.46	0.48	0.48	0.48	0.49	0.49	0.47	-	0.46	0.46	0.47	0.48	0.47	0.49	0.35	0.49	
SUN	0.42	0.42	0.41	0.41	0.44	0.45	0.42	0.42	0.43	0.43	0.42	0.42	-	0.43	0.42	0.42	0.42	0.45	0.29	0.43	
ScanNet	0.42	0.42	0.42	0.42	0.43	0.44	0.43	0.43	0.43	0.43	0.42	0.43	0.43	-	0.42	0.43	0.42	0.44	0.28	0.42	
SUIM	0.73	0.73	0.71	0.71	0.74	0.73	0.74	0.73	0.73	0.73	0.74	0.72	0.74	0.73	-	0.73	0.73	0.74	0.48	0.76	
vkITTI2	0.58	0.58	0.58	0.58	0.58	0.58	0.58	0.58	0.61	0.63	0.63	0.62	0.58	0.63	-	0.58	0.61	0.56	0.58	0.58	
vGallery	0.78	0.79	0.79	0.78	0.79	0.79	0.79	0.79	0.79	0.79	0.79	0.79	0.79	0.79	0.78	0.79	-	0.80	0.72	0.79	

(b) Full target training set.

To: Segmentation	Aerial		Consumer			Driving			Indoor			Underwater			Synthetic			vGallery	Multi	Scratch	IS/NRC12
	ISPRS 4DK	ISAD	PContext	ADE20K	COCO	City	IDD	BDD	Mapillary	SUN	ScanNet	Sum	WTT2	WTT3	WTT4	WTT5	WTT6				
ISPRS	-	0.48	0.47	0.47	0.44	0.46	0.45	0.43	0.44	0.45	0.45	0.46	0.43	0.43	0.44	0.44	0.46	0.44	0.31	0.46	
ISAD	0.37	-	0.41	0.41	0.38	0.38	0.37	0.34	0.36	0.38	0.37	0.38	0.39	0.37	0.39	0.34	0.35	0.20	0.42		
PContext	0.26	0.27	-	0.31	0.27	0.32	0.26	0.25	0.26	0.26	0.25	0.27	0.24	0.23	0.25	0.26	0.27	0.06	0.28		
PVOC	0.46	0.50	0.64	-	0.48	0.59	0.49	0.47	0.47	0.50	0.49	0.50	0.45	0.43	0.48	0.48	0.49	0.08	0.54		
ADE20K	0.22	0.23	0.22	0.22	-	0.23	0.23	0.22	0.23	0.23	0.22	0.23	0.22	0.21	0.22	0.23	0.22	0.08	0.23		
COCO	0.30	0.31	0.30	0.30	0.29	-	0.30	0.30	0.30	0.30	0.30	0.29	0.28	0.29	0.30	0.31	-	0.07	0.31		
KITTI	0.45	0.45	0.46	0.45	0.45	0.47	-	0.47	0.50	0.47	0.47	0.47	0.44	0.41	0.46	0.46	0.46	0.36	0.46		
CamVid	0.49	0.51	0.51	0.52	0.51	0.51	0.54	-	0.55	0.54	0.55	0.53	0.49	0.51	0.50	0.50	0.50	0.40	0.49		
City	0.42	0.42	0.42	0.41	0.43	0.43	0.43	0.43	-	0.44	0.44	0.46	0.39	0.38	0.41	0.41	0.41	0.26	0.43		
IDD	0.37	0.38	0.39	0.38	0.37	0.38	0.38	0.37	0.40	-	0.39	0.41	0.35	0.35	0.37	0.39	0.37	0.22	0.38		
BDD	0.43	0.44	0.46	0.47	0.46	0.45	0.46	0.46	0.49	0.49	-	0.52	0.41	0.41	0.44	0.44	0.44	0.27	0.46		
Mapillary	0.25	0.26	0.27	0.26	0.26	0.26	0.26	0.25	0.28	0.28	0.28	-	0.24	0.23	0.26	0.26	0.26	0.11	0.26		
SUN	0.19	0.20	0.21	0.20	0.24	0.23	0.20	0.18	0.19	0.19	0.18	0.19	-	0.26	0.19	0.18	0.19	0.07	0.20		
ScanNet	0.13	0.14	0.15	0.14	0.16	0.16	0.14	0.14	0.13	0.14	0.13	0.14	0.21	-	0.14	0.14	0.14	0.05	0.14		
SUIM	0.52	0.53	0.50	0.54	0.51	0.54	0.52	0.50	0.51	0.51	0.52	0.52	0.48	-	0.51	0.52	0.29	0.53			
vkITTI2	0.54	0.55	0.54	0.53	0.54	0.54	0.56	0.56	0.56	0.54	0.55	0.55	0.54	0.53	0.54	-	0.55	0.47	0.56		
vGallery	0.76	0.77	0.77	0.77	0.77	0.77	0.78	0.78	0.77	0.77	0.77	0.77	0.77	0.76	0.76	0.77	-	0.70	0.78		

(c) Small source and small target training set.

TABLE A.5: Absolute performance and the corresponding relative transfer gains for semantic segmentation as both a source and target task type. Left tables: mean Intersection-over-Union. Right tables: relative transfer gains.

	Depth			Detection			Keypoints			ILSVRC12 1.2M		
	SUN 5K	VKIT12 42K	Vgallery 44K	COCO 120K	BDD 70K	UWT 6K	Multi	St. Dogs 12K	MPII 25K		COCO 120K	
ISAIID	0.37	0.28	0.34	0.30	0.25	0.31	0.34	0.35	0.37	0.27	0.20	0.42
PContext	0.24	0.17	0.24	0.27	0.15	0.17	0.26	0.24	0.26	0.18	0.06	0.28
COCO	0.28	0.27	0.29	0.30	0.17	0.23	0.29	0.28	0.29	0.20	0.07	0.31
KITTI	0.45	0.43	0.44	0.44	0.47	0.43	0.45	0.44	0.44	0.42	0.36	0.46
CamVid	0.47	0.49	0.51	0.49	0.54	0.49	0.51	0.43	0.47	0.44	0.40	0.49
IDD	0.37	0.35	0.37	0.36	0.37	0.33	0.38	0.36	0.37	0.34	0.22	0.38
SUN	0.22	0.12	0.18	0.18	0.13	0.14	0.20	0.18	0.19	0.15	0.07	0.20
ScanNet	0.16	0.09	0.12	0.12	0.09	0.10	0.13	0.13	0.14	0.10	0.05	0.14
SUIM	0.49	0.44	0.51	0.50	0.43	0.48	0.51	0.50	0.52	0.45	0.29	0.53

	Depth			Detection			Keypoints			ILSVRC12 1.2M	
	SUN 5K	VKIT12 42K	Vgallery 44K	COCO 120K	BDD 70K	UWT 6K	Multi	St. Dogs 12K	MPII 25K		COCO 120K
ISAIID	-13	-35	-21	-29	-40	-28	-21	-17	-13	-36	-52
PContext	-16	-40	-15	-4	-47	-39	-7	-14	-6	-37	-80
COCO	-8	-14	-6	-2	-44	-24	-5	-8	-5	-36	-77
KITTI	-3	-6	-4	-4	3	-6	-1	-3	-3	-8	-21
CamVid	-4	0	4	-1	10	-1	4	-13	-4	-11	-18
IDD	-4	-9	-5	-6	-3	-13	-2	-6	-3	-11	-44
SUN	8	-39	-12	-9	-34	-31	-3	-10	-4	-28	-66
ScanNet	13	-38	-13	-18	-36	-27	-11	-9	-3	-27	-62
SUIM	-8	-16	-4	-6	-19	-9	-3	-5	-1	-14	-46

TABLE A.6: Transfer from other task types to semantic segmentation, in the small target training setting. Left: absolute metric (Intersection-over-Union) and the corresponding relative transfer gains (right).

	Detection			Depth			Keypoints			Segmentation			ILSVRC12 1.2M						
	VOC07 5K	UWT 6K	BDD 70K	COCO 120K	SUN 5K	VKIT12 42K	Multi	St. Dogs 12K	MPII 25K	COCO 120K	SUIM 1.5K	PContext 5K							
VOC07	-	0.20	0.21	0.71	0.12	0.06	0.13	0.20	0.20	0.23	0.18	0.24	0.14	0.17	0.18	0.17	0.29	0.25	0.23
UWT	0.35	-	0.30	0.51	0.28	0.24	0.28	0.32	0.31	0.37	0.31	0.30	0.32	0.30	0.29	0.28	0.31	0.31	0.34
BDD	0.07	0.06	-	0.12	0.05	0.05	0.06	0.06	0.06	0.07	0.06	0.06	0.05	0.07	0.07	0.06	0.06	0.07	0.06
COCO	0.10	0.08	0.08	-	0.06	0.02	0.07	0.08	0.08	0.09	0.08	0.08	0.07	0.08	0.08	0.08	0.09	0.09	0.08

	Detection			Depth			Keypoints			Segmentation			ILSVRC12 1.2M					
	VOC07 5K	UWT 6K	BDD 70K	COCO 120K	SUN 5K	VKIT12 42K	Multi	St. Dogs 12K	MPII 25K	COCO 120K	SUIM 1.5K	PContext 5K						
VOC07	-	-14	-8	207	-49	-74	-45	-15	-11	0	-21	4	-38	-28	-22	-26	25	8
UWT	6	-	-12	52	-18	-29	-17	-5	-8	9	-9	-11	-6	-12	-14	-16	-8	-9
BDD	15	-1	-	94	-16	-23	-2	-4	-3	19	-3	5	-12	14	23	-4	-1	15
COCO	30	5	-5	-	-26	-72	-14	-6	1	8	-6	-1	-11	-6	-4	-1	16	12

(a) Small target training set. Top: mean Average Precision. Bottom: relative transfer gain.

	Detection			Depth			Keypoints			Segmentation			ILSVRC12 1.2M					
	VOC07 5K	UWT 6K	BDD 70K	COCO 120K	SUN 5K	VKIT12 42K	Multi	St. Dogs 12K	MPII 25K	COCO 120K	SUIM 1.5K	PContext 5K						
VOC07	-	0.65	0.60	0.85	0.80	0.64	0.43	0.66	0.70	0.67	0.69	0.67	0.70	0.70	0.70	0.60	0.71	0.72
UWT	0.51	-	0.44	0.52	0.40	0.39	0.34	0.40	0.48	0.49	0.49	0.43	0.42	0.45	0.41	0.47	0.47	0.45
BDD	0.26	0.26	-	0.27	0.27	0.26	0.24	0.26	0.26	0.26	0.26	0.26	0.26	0.26	0.26	0.19	0.26	0.26
COCO	0.40	0.39	0.39	-	0.38	0.38	0.36	0.38	0.38	0.39	0.38	0.38	0.38	0.39	0.37	0.38	0.38	0.39

	Detection			Depth			Keypoints			Segmentation			ILSVRC12 1.2M				
	VOC07 5K	UWT 6K	BDD 70K	COCO 120K	SUN 5K	VKIT12 42K	Multi	St. Dogs 12K	MPII 25K	COCO 120K	SUIM 1.5K	PContext 5K					
VOC07	-	-10	-16	18	11	-11	-41	-8	-3	-7	-4	-7	-3	-2	-17	-1	-1
UWT	15	-	-1	16	-9	-12	-24	-10	7	10	11	-3	-5	0	-9	6	
BDD	1	0	-	2	2	-1	-8	0	0	0	0	-1	0	0	-25	1	
COCO	1	0	1	-	-2	-3	-9	-3	-2	0	-2	-2	-2	-1	-5	-3	

(b) Full target training set. Top: mean Average Precision. Bottom: relative transfer gain.

TABLE A.7: Absolute metrics and the corresponding relative transfer gains for object detection as a target task type.



	Keypoints			Depth		Detection		Segmentation		ILSVRC12 1.2M
	St. Dogs 12K	COCO 120K	MPII 25K	SUN 5K	Multi	COCO 120K	Multi	COCO 120K	Multi	
To: Keypoints	-	0.48	0.32	0.17	0.02	0.55	0.52	0.15	0.18	0.23
MPII	0.26	-	0.75	0.20	0.05	0.38	0.39	0.20	0.15	0.28
COCO	0.16	0.43	-	0.10	0.00	0.32	0.31	0.11	0.08	0.23

	Keypoints			Depth		Detection		Segmentation		ILSVRC12 1.2M
	St. Dogs 12K	COCO 120K	MPII 25K	SUN 5K	Multi	COCO 120K	Multi	COCO 120K	Multi	
To: Keypoints	-	0.92	0.92	0.90	0.43	0.91	0.90	0.85	0.91	0.92
MPII	0.70	-	0.82	0.69	0.22	0.70	0.71	0.64	0.65	0.66
COCO	0.81	0.81	-	0.81	0.53	0.79	0.80	0.80	0.81	0.81

	Keypoints			Depth		Detection		Segmentation	
	St. Dogs 12K	COCO 120K	MPII 25K	SUN 5K	Multi	COCO 120K	Multi	COCO 120K	Multi
To: Keypoints	-	110	42	-25	-90	141	128	-33	-20
MPII	-8	-	166	-30	-81	36	37	-31	-46
COCO	-29	90	-	-54	-98	39	39	-51	-63

	Keypoints			Depth		Detection		Segmentation	
	St. Dogs 12K	COCO 120K	MPII 25K	SUN 5K	Multi	COCO 120K	Multi	COCO 120K	Multi
To: Keypoints	-	0	-1	-2	-53	-2	-3	-8	-1
MPII	7	-	26	6	-66	7	9	-2	-1
COCO	0	0	-	0	-34	-2	-2	-1	0

(a) Small target training set. Top: mean Average Precision at 0.5 Object Keypoint Similarity. Bottom: relative transfer gain.

(b) Full target training set. Top: mean Average Precision at 0.5 Object Keypoint Similarity. Bottom: relative transfer gain.

TABLE A.8: Absolute metrics and the corresponding relative transfer gains for keypoint detection as target task type.

	Depth		Detection			Keypoints			Segmentation			ILSVRC12 1.2M						
	SUN 5K	vKITTI2 42K	COCO 120K	BDD 70K	UWT 6K	Multi	St. Dogs 12K	MPII 25K	COCO 120K	BDD 7K	SUN 5K		vKITTI2 42K	vGallery 44K	Multi	Scratch		
To: Depth	-	0.90	0.81	0.82	0.89	0.88	0.83	0.82	0.83	0.86	0.76	0.84	0.80	0.89	0.86	0.77	1.11	0.83
vKITTI2	4.66	-	4.72	4.69	4.67	4.66	4.67	4.75	4.73	4.65	4.68	4.72	4.65	4.67	4.72	4.60	5.40	4.66
vGallery	0.87	0.93	-	0.91	0.90	0.90	0.92	0.88	0.90	0.93	0.84	0.88	0.91	0.89	0.87	0.84	1.27	0.87

	Depth		Detection			Keypoints			Segmentation								
	SUN 5K	vKITTI2 42K	COCO 120K	BDD 70K	UWT 6K	Multi	St. Dogs 12K	MPII 25K	COCO 120K	BDD 7K	SUN 5K	vKITTI2 42K	vGallery 44K	Multi	Scratch		
To: Depth	-	-9	2	1	-7	-6	0	1	0	-3	9	-1	4	-7	-3	7	-33
vKITTI2	0	-	-1	-1	0	0	0	-2	-1	0	0	-1	0	0	-1	1	-16
vGallery	0	-6	-	-5	-3	-3	-5	-1	-3	-6	3	-1	-5	-2	1	3	-46

(a) Small target training set. Top: RMSE. Bottom: relative transfer gain.

	Depth			ILSVRC12 1.2M	
	SUN 5K	vKITTI2 42K	vGallery 44K		
To: Depth	-	0.629	0.610	0.613	0.592
vKITTI2	4.051	-	4.132	4.068	4.182
vGallery	0.850	0.859	-	0.883	0.861

	Depth			
	SUN 5K	vKITTI2 42K	vGallery 44K	
To: Depth	-	-6	-3	-3
vKITTI2	3	-	1	3
vGallery	1	0	-	-3

(b) Full target training set. Top: RMSE. Bottom: relative transfer gain.

	Depth		Detection			Keypoints			Segmentation			ILSVRC12 1.2M						
	SUN 5K	vKITTI2 42K	COCO 120K	BDD 70K	UWT 6K	Multi	St. Dogs 12K	MPII 25K	COCO 120K	BDD 7K	SUN 5K		vKITTI2 42K	vGallery 44K	Multi	Scratch		
To: Depth	-	0.50	0.57	0.56	0.52	0.53	0.55	0.56	0.56	0.55	0.61	0.55	0.60	0.52	0.54	0.61	0.41	0.56
vKITTI2	0.88	-	0.89	0.89	0.89	0.89	0.90	0.90	0.88	0.89	0.89	0.89	0.89	0.89	0.89	0.90	0.87	0.90
vGallery	0.48	0.46	-	0.48	0.49	0.49	0.48	0.49	0.50	0.48	0.50	0.49	0.47	0.48	0.49	0.50	0.36	0.49

	Depth		Detection			Keypoints			Segmentation								
	SUN 5K	vKITTI2 42K	COCO 120K	BDD 70K	UWT 6K	Multi	St. Dogs 12K	MPII 25K	COCO 120K	BDD 7K	SUN 5K	vKITTI2 42K	vGallery 44K	Multi	Scratch		
To: Depth	-	-9	2	2	-6	-4	-1	1	1	-2	10	-1	7	-6	-3	9	-26
vKITTI2	-2	-	0	-1	-1	0	0	0	-1	-1	-1	-1	-1	-1	0	-3	
vGallery	-3	-7	-	-3	-1	-1	-3	0	1	-3	1	0	-4	-2	0	1	-27

(c) Small target training set. Top:  $\delta < 1.25$ . Bottom: relative transfer gain.

	Depth			ILSVRC12 1.2M	
	SUN 5K	vKITTI2 42K	vGallery 44K		
To: Depth	-	0.74	0.76	0.76	0.77
vKITTI2	0.93	-	0.93	0.93	0.93
vGallery	0.50	0.49	-	0.47	0.49

	Depth			
	SUN 5K	vKITTI2 42K	vGallery 44K	
To: Depth	-	-4	-1	-1
vKITTI2	0	-	0	0
vGallery	1	0	-	-4

(d) Full target training set. Top:  $\delta < 1.25$ . Bottom: relative transfer gain.

TABLE A.9: Absolute metrics and the corresponding relative transfer gains for depth estimation as a target task type.

## APPENDIX D COMPUTATIONAL COSTS OF TRANSFER CHAINS

For performed the experiments in this paper on Google Cloud TPU-v3 chips. If we calculate costs in terms of TPU-hours, defined as the number of TPUs multiplied with the computation time:

- ILSVRC'12 pretraining (ImageNet) takes 364 hours.
- Training a COCO segmentation source model (largest *consumer* source) takes an additional 92 hours.
- Training a Mapillary segmentation source model (*i.e.* the largest *driving* source) takes an additional 352 hours (due to higher resolution images).
- Training a multi-source model takes 228 hours.

This means that pre-training costs for our transfer chains are increased by 25%-97%, compared to the standard practice of pre-training on ILSVRC'12.

However, the source models are trained only once. The same source models can then be repeatedly used for transfer learning to many target tasks. This means that the relative additional costs of pre-training becomes increasingly small as more researchers reuse the same source model for new target tasks.

## APPENDIX E DATASET OVERLAP

When using 21 datasets there is always a risk that images in the test set of one dataset are used as training in another dataset, this is deemed undesirable for fair evaluation. Since we have not quantified such an overlap, we can not assure there is no overlap at all. Still, we do believe that the risk of such potential overlap significantly changes the results of our experiments to be minimal, for the following reasons:

- 1) We evaluate structured prediction tasks, while the largest dataset (ILSVRC'12) had only classification annotation, thus even if an image has already been seen it had different type of annotation.
- 2) Within the same annotation type datasets either differ in visual domain (*e.g.* underwater and aerial are not expected to have any overlap), or they are acquired in different geographical regions (*e.g.* BDD and IDD).
- 3) Since the collection consists of popular public benchmarks, major overlaps of these datasets would have been reported.