Transferability Metrics for Selecting Source Model Ensembles

Andrea Agostinelli Jasper Uijlings Thomas Mensink Vittorio Ferrari
Google Research
{agostinelli, jrru, mensink, vittoferrari}@google.com

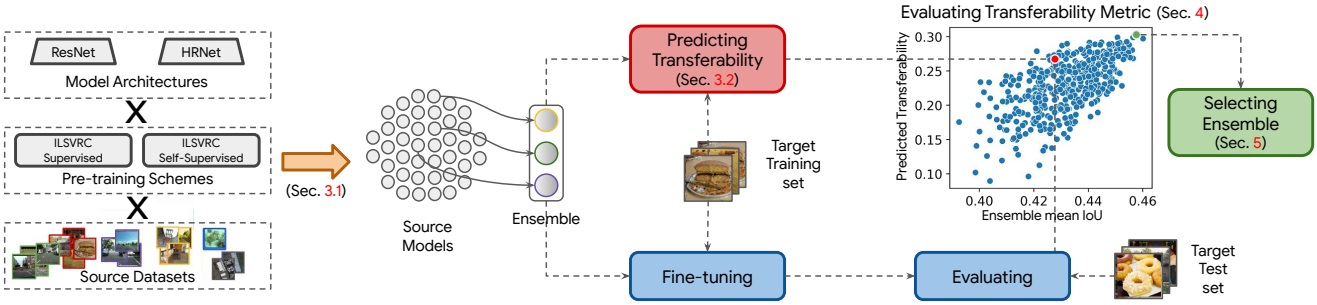

Figure 1. Overview of our method for predicting transferability of source model ensembles. We start from a large pool of pre-trained source models, and form candidate ensembles by combining multiple source models. For each candidate ensemble we predict performance on the target dataset with an efficient transferability metric. We show experimentally that this metric correlates well with actual performance after fine-tuning the ensemble on the target dataset. Hence, it can be used to select the best ensemble *without the expensive fine-tuning stage*.

Abstract

We address the problem of ensemble selection in transfer learning: Given a large pool of source models we want to select an ensemble of models which, after fine-tuning on the target training set, yields the best performance on the target test set. Since fine-tuning all possible ensembles is computationally prohibitive, we aim at predicting performance on the target dataset using a computationally efficient transferability metric. We propose several new transferability metrics designed for this task and evaluate them in a challenging and realistic transfer learning setup for semantic segmentation: we create a large and diverse pool of source models by considering 17 source datasets covering a wide variety of image domain, two different architectures, and two pre-training schemes. Given this pool, we then automatically select a subset to form an ensemble performing well on a given target dataset. We compare the ensemble selected by our method to two baselines which select a single source model, either (1) from the same pool as our method; or (2) from a pool containing large source models, each with similar capacity as an ensemble. Averaged over 17 target datasets, we outperform these baselines by 6.0% and 2.5% relative mean IoU, respectively.

1. Introduction

In transfer learning we want to re-use knowledge previously learned on a source task to help learning a target task. The most common form of transfer learning in computer vision is to pre-train a single source model on the generic ILSVRC dataset [3, 14, 26, 31, 33, 38, 59, 80] and then fine-tune it on the target dataset. However, often a more domain-specific approach can lead to better results [49, 54, 75]. Hence, it is beneficial to have a large pool of diverse source models such that it contains models suited for many different target tasks. The problem then becomes: how can we automatically and efficiently select good source models for a given target task?

Recently, transferability metrics were introduced to address this problem [4, 44, 55, 65, 66, 76]. The general goal is to select a single source model which, after fine-tuning on the target training set, yields the best performance on the target test set. Transferability metrics enable to select this model efficiently without carrying out expensive fine-tuning on the target training set.

While previous transferability metrics consider selecting a single source model, in this paper we aim at selecting a subset containing multiple source models to form an *ensemble*. Ensembles are a general technique used to improve model accuracy, out-of-distribution robustness, and to estimate uncertainty [7, 17, 23, 24, 39, 40, 56, 70]. Fur-

thermore, by aggregating multiple source models, we can combine knowledge coming from multiple source datasets and image domains, which may be beneficial for a particular target task. Hence, in this paper we extend previous work on transferability by proposing several transferability metrics designed for *ensemble selection*.

To evaluate ensemble selection we introduce a challenging experimental setup. We consider semantic segmentation as a task, with a truly diverse pool of source models, as we train them on 17 complete datasets spanning a wide variety of images domains, while also varying their model architectures and pre-training schemes (Fig. 1). In contrast, previous works typically focus on image classification [4, 44, 55, 65, 66, 76], consider a narrower range of at most 4 source datasets [44, 55, 65, 76], and often generate multiple datasets artificially by sampling different subsets of classes out of a single actual dataset [4, 55, 65, 66].

To summarize, we make the following contributions: (1) We design transferability metrics for *ensemble* selection. (2) We consider a challenging application scenario on semantic segmentation featuring a large and truly diverse pool of source models. (3) We compare the ensemble selected by our method to two baselines which select a single source model, either from the same pool as our method; or from a pool containing large source models, each with similar capacity as an ensemble. Averaged over 17 target datasets, we outperform these baselines by 6.0% and 2.5% relative mean IoU, respectively (Sec. 5.2).

2. Related Work

Transfer Learning. The most common form of transfer learning in computer vision is to pre-train a model on ILSVRC'12 [18, 37], and fine-tune it on the target dataset. Several works extend this to using a larger source dataset such as ImageNet21k (9M images), JFT-300M (300M images) [18, 37, 52], or Open Images (1.7M images) [75]. Other works consider self-supervised pre-training, enabling the use of unlabeled source datasets (*e.g.* [12, 13, 30, 34, 43]).

Several studies explore in-depth under which circumstances transfer learning works. Mensink et al. [49] study transfer learning across datasets with vastly different image domains and multiple visual tasks. Mustafa et al. [51] studies transfer learning in medical imaging. Finally, Taskonomy [78] establishes relationships between visual tasks (*e.g.* semantic segmentation, depth prediction, etc.). Following the success of Taskonomy, several works investigate whether visual task relatedness can be predicted [6, 19, 20, 61, 62] rather than calculated by brute force [78].

Only a few works conduct transfer learning from multiple source datasets at the same time. Liu et al. [46] train a student model using knowledge from multiple teachers (*i.e.* source models), which is expensive in both memory and computation. Zoo-Tuning [60] learns to aggregate the parameters of multiple source models into a target model. This requires storing all source models in memory during training, limiting scalability. In contrast, we select an ensemble from a large pool of source models in a computationally and memory efficient manner.

Transferability Metrics. Recently, several papers introduced transferability metrics. H-score [4] measures the discriminativeness of source model features on the target task in terms of inter-class and intra-class variance. LEEP [55] measures how well a classifier built on top of source model predictions performs on the target task. $\mathcal{N}$LEEP [44] trains a Gaussian Mixture Model (GMM) on top of source model features. Then it measures how well a classifier built on top of these GMM predictions performs on the target task. LogME [76] estimates accuracy on the target task based on a formulation which integrates over all possible linear classifiers built on top of the source model features. OTCE [65] applies a source model to extract image features from both the source and target dataset. Then it uses optimal transport between these features to calculate domain difference and task difference. Finally, NCE [66] considers a more restrictive setting where the source and target datasets consist of identical images. Their method uses conditional entropy between ground truth source and target labels, which avoids training models and is thus computationally efficient.

To put our work in context: (1) Instead of selecting a single source model [4, 44, 55, 65, 76], we do *ensemble* selection. (2) Instead of image classification [4, 44, 55, 65, 66, 76], we address semantic segmentation. (3) We consider a larger variety of source datasets than previous works (17 vs at most 4 [65]). (4) We consider complete datasets, whereas previous works often sample different subsets of classes out of a single actual dataset [4, 55, 65, 66].

Ensemble of Models. Ensembling machine learning models is a classical method for increasing accuracy [7, 17, 24, 29, 39, 42], where having diverse models is typically important. More recently, ensembles of deep neural network have been studied in the context of uncertainty estimation and out-of-distribution robustness [2, 23, 40, 56].

3. Methods

We consider the problem of source model ensemble selection for semantic segmentation. Given N source models and a target dataset, the goal is to select an *ensemble* of source models which, after fine-tuning on the target training set, yields the best performance on the target test set. Since fine-tuning all possible ensembles is too computationally expensive, we predict performance on the target dataset using a computationally efficient transferability metric.

Sec. 3.1 discusses what makes for a good pool of source models and describes how we construct this pool. Sec. 3.2 describes our setup to work with transferability metrics in semantic segmentation tasks. Sec. 3.3 describes LEEP [55],

	Ours Sec. 5	Ours Sec. 4	LEEP [55]	LogME [76]	OTCE [65]	$\mathcal{N}$LEEP [44]
# source datasets	17	10-15	1	1	4	3
# pre-training schemes	2	1-2	1	1	1	4
# model architectures	2	1-2	9	10	1	13
# source models	68	15	9	10	4	41
# candidates	41K	455	9	10	4	41

Table 1. Comparing our experimental setup (see Sec. 4 and 5) to previous works on cross-dataset source selection. We compare the diversity of source models in terms of the number of source datasets, pre-training schemes, and model architectures. The last row denotes the number of candidate source models (or ensembles in our case) that are in the pool for a given target dataset. For [44, 55, 65, 76] we consider their largest source selection experiment.

a transferability metric for single-source selection. We use this as a starting point in Sec. 3.4 to define our four transferability metrics for *ensemble* selection.

3.1. Preparing source models

We want to create a pool with a large variety of source models for three reasons: (1) this increases the chance that for any given target dataset there exists at least one good source model. (2) we need bad source models to verify that our transferability metrics correctly select good source models while discarding bad ones. (3) an ensemble can only outperform its individual members if they are diverse (and therefore complementary) [5, 17, 24, 29, 39]. Hence, we construct our source model pool by incorporating diversity in three ways: we use 17 different source datasets, two model architectures and two pre-training strategies. Tab. 1 summarizes how this setup compares to related work.

Source datasets. The image domain is one of the most important factors to influence whether transfer learning will succeed [49, 54, 57], and therefore we want to cover a wide array of image domains. Furthermore, the most natural way to perform transfer learning is to consider each dataset as a whole (rather than subsampling a dataset to simulate dataset variations [4, 55, 65, 66]). Therefore, we adopt the realistic cross-dataset transfer learning setup for semantic segmentation by [49]: 17 source datasets from 6 image domains (consumer photos, driving, aerial, indoor, underwater, synthetic; Tab. 2). While this setup was defined in [49], that work did not explore any transferability metric.

Model architectures. We consider two semantic segmentation architectures, each with a backbone and a linear classification layer. As the first backbone we choose HRNetV2 [69], a high-resolution alternative to ResNet. It maintains parallel feature representations at different resolutions, which helps dense prediction tasks [41, 49, 69]. As ensembles contain multiple models, we choose a lightweight version: HRNetV2-W28 (23M parameters).

As the second backbone, we adopt a high-resolution variant of ResNet50 [32]. First, we remove the downsampling operations in the last two ResNet blocks while increasing the dilation rate [73]. Second, we add an upsampling layer using 5 parallel atrous convolutions with different dilation rates, which enlarges the field of view of the filters without compromising on spatial resolution [11]. Finally, we remove the last four layers of the last ResNet block to make this backbone have the same number of parameters as HRNetV2-W28 (we call this model ResNet23M). As all ensembles we build contain the same number of source models, this ensures that they also have the same number of parameters, enabling a fair comparison between them.

Dataset	Domain	# classes	# train images
Pascal Context [50]	Consumer	60	5K
Pascal VOC [22]	Consumer	22	10K
ADE20K [79]	Consumer	150	20K
COCO Panoptic [10, 36, 45]	Consumer	134	118K
KITTI [1]	Driving	30	150
CamVid [8]	Driving	23	367
CityScapes [15]	Driving	33	3K
India Driving Dataset (IDD) [67]	Driving	35	7K
Berkeley Deep Drive (BDD) [77]	Driving	20	7K
Mapillary Vista Dataset [53]	Driving	66	18K
ISPRS [58]	Aerial	6	4K
iSAID [71, 74]	Aerial	16	27K
SUN RGB-D [63]	Indoor	37	5K
ScanNet [16]	Indoor	41	19K
SUIM [35]	Underwater	8	1525
vKITTI2 [9, 25]	Synthetic driving	9	43K
vGallery [72]	Synthetic indoor	8	44K

Table 2. Semantic segmentation datasets used in our paper.

Pre-training schemes. Fully supervised pre-training on ILSVRC'12 generally benefits semantic segmentation [49, 59]. Furthermore, self-supervised pre-training is making rapid progress and can even outperform fully supervised pre-training [34, 43]. To maximize model diversity, we create two variants of each source model by using two types of ILSVRC'12 pre-trained weights: fully supervised and using the self-supervised SimCLR method [12].

Training source models. We have 17 source datasets, two architectures, and two pre-training schemes. We train a source model for each combination, i.e. 68 in total (details in suppl. mat.). These models are trained only once and reused in all experiments.

3.2. Setting up for semantic segmentation

In most previous works [4, 44, 55, 65, 66, 76], a transferability metric is primarily applied to image classification, where an image is associated to one label. In semantic segmentation instead we have predictions at the pixel level, and therefore we consider for each pixel x_i and its ground-truth label y_i as an individual example (x_i, y_i).

The number of datapoints in semantic segmentation is approximately 6 orders of magnitude higher than in image classification. To reduce the computational cost, we sample 1000 pixels per image to calculate each transferability metric. Furthermore, semantic segmentation datasets often have large class imbalance, which can negatively affect results. Therefore, we sample pixels inversely proportionally to the frequency of their class labels in the target dataset.

3.3. LEEP as single-source transferability metric

We want a transferability metric suitable for selecting an ensemble of models for semantic segmentation. We start from LEEP [55], which is based on probability distributions, for several reasons. First, both per-pixel predictions and ensemble selection increase computational and memory complexity compared to the usual image classification and single-source selection. LEEP is computationally cheap, requiring just a single forward pass of the target training set through the source model, without additional training. Second, LEEP is memory-efficient as it stores predictions instead of features as opposed to alternative metrics [4,65,76]. Finally, starting from LEEP we can derive clear mathematical formulations for the multi-source setting.

LEEP calculates a transferability score between a single source model s and a target training set $\mathcal{D}_t$, containing a set of training samples $(x_i, y_i) \in \mathcal{D}_t$. Applying s to a target sample produces the probability $p_s(z|x_i)$, for each source class z in the source label space $\mathcal{Z}$. The core idea of LEEP is to associate predictions in the source label space $\mathcal{Z}$ to predictions in the target label space $\mathcal{Y}$. To do so, we apply s to all samples in $\mathcal{D}_t$, and then compute the empirical joint distribution $\hat{P}(y, z)$ measuring co-occurrences between all pairs of labels $(y, z) \in \mathcal{Y} \times \mathcal{Z}$.

Next, we can calculate the empirical conditional distribution $\hat{P}(y|z)$ as $\frac{\hat{P}(y,z)}{\hat{P}(z)}$. Given the source model s, we can now construct a classifier, called **E**xpected **E**mpirical **P**redictor (EEP):

$$p_s(y_i|x_i) = \sum_{z \in \mathcal{Z}} \hat{P}(y_i|z) p_s(z|x_i) \tag{1}$$

Here $p_s(y_i|x_i)$ is the probability that model s assigns to the ground-truth label y_i at pixel x_i. LEEP is defined as the log-average of the predictor over $\mathcal{D}_t$:

$$\text{LEEP} = \frac{1}{n} \sum_{i=1}^{n} \log p_s(y_i|x_i) \tag{2}$$

We can see that LEEP measures how well the constructed classifier EEP performs on $\mathcal{D}_t$, where better transferability is associated with higher LEEP scores.

3.4. Multi-source selection transferability metrics

We design four transferability metrics suitable for multi-source selection. We base all our approaches on the EEP predictor (1), which provides a mathematical foundation to establish relationships between source models. In all cases our ensembles contain a fixed number S of source models.

Multi-Source LEEP (MS-LEEP). A natural way of extending LEEP to the multi-source setting is to compute the joint probability distribution over the S source model predictions, where each model makes predictions in its own label space. But this requires calculating the joint probability distribution for every possible subset of S models, which is infeasible for a large pool of N models. Instead, we assume the source models to be independent, yielding a simplified joint conditional distribution:

$$\hat{P}(y|z_1, z_2, ..., z_S) \approx \prod_{s=1}^{S} \hat{P}(y|z_s) \tag{3}$$

We extend (2) by applying (3) to define a new metric:

$$\begin{aligned} \text{MS}-\text{LEEP} &= \frac{1}{n} \sum_{i=1}^{n} \log \left(\prod_{s=1}^{S} \left(\sum_{z_s \in \mathcal{Z}_s} \hat{P}(y_i|z_s) p_s(z_s|x_i) \right) \right) \\ &= \frac{1}{n} \sum_{i=1}^{n} \left(\sum_{s=1}^{S} \log p_s(y_i|x_i) \right) = \sum_{s=1}^{S} \text{LEEP}_s \end{aligned} \tag{4}$$

Hence MS-LEEP can be seen as taking the best source models according to the single-model metric LEEP. This suggests that other existing transferability metrics [4,44,66,76] can be similarly adapted to ensemble selection.

Ensemble LEEP (E-LEEP). We now approach the problem from a different prospective, stressing that we want to predict the transfer performance of an *ensemble* of models. For this we consider the ensemble prediction as the average of the S individual models predictions. Considering (1) as a single-source predictor $p_s(y_i|x_i)$, we can construct the prediction of the ensemble as

$$p_{\text{ens}}(y_i|x_i) = \frac{1}{S} \sum_{s=1}^{S} p_s(y_i|x_i) \tag{5}$$

By reformulating (2) accordingly, we get a new metric:

$$\text{E}-\text{LEEP} = \frac{1}{n} \sum_{i=1}^{n} \log p_{\text{ens}}(y_i|x_i) \tag{6}$$

The difference with MS-LEEP (4) is the order of the log and the sum: E-LEEP uses the log of the mean predictions, while MS-LEEP uses the mean of the log predictions.

IoU-EEP. While MS-LEEP and E-LEEP estimate pixel classification accuracy, semantic segmentation performance is usually measured as Intersection-over-Union (IoU). In practice, for IoU we count True Positives (TP), False Positives (FP), False Negatives (FN), and calculate $\frac{TP}{TP+FP+FN}$. IoU is calculated separately per class, and then averaged into a single metric (mean IoU). In order to design a transferability metric that more directly approximates mean IoU, we first use the ensemble predictor (5) to compute the predicted semantic segmentation over the target training set

$$y_i^* = \arg\max_{y \in \mathcal{Y}} p_{\text{ens}}(y|x_i) \tag{7}$$

where y now iterates over the entire target label space (as opposed to being the one ground-truth label y_i). Finally, we use these predictions y_i^* to calculate the mean IoU, arriving at the IoU-EEP transferability metric. This metric is also less dependent on the probabilistic output of the source classifiers, which are often poorly calibrated [27] and therefore could negatively affect the metric.

SoftIoU-EEP. Applying the $\arg\max$ in Eq. (7) alleviates calibration errors. But it also loses the fine-grained information in the probability distribution $p_{\text{ens}}(\cdot|x_i)$, which could be helpful in ranking similar models. Hence, we propose to introduce a "Soft-IoU" relaxation. For every pixel x_i, instead of counting True Positives, we aggregate their confidences $p_{\text{ens}}(y_i|x_i)$ (where y_i is the ground-truth label of that pixel). Therefore the higher the confidences of correct predictions, the higher the predicted transferability. We do the analogue for errors (FP and FN) by aggregating $1-p_{\text{ens}}(y_i|x_i)$. Since errors are in the denominator, the higher their confidences the lower the predicted transferability.

4. Evaluating transferability metrics

4.1. Experimental setup

The standard procedure to evaluate transferability metrics is to consider several candidate source models for a given target dataset, and measure the correlation between (A) the transferability score and (B) the actual performance on the target test set after the source model has been fine-tuned on the target training set [44, 55, 65, 76].

In this paper we consider multiple candidate source model *ensembles* for a given target dataset. To measure the desired correlation, we calculate transferability as in Sec. 3.4 and we also need to calculate the actual performance of each ensemble on the target dataset.

Considering ensembles leads to additional computational challenges. The number of possible ensembles of S source models out of a pool of N is very large (the binomial $\binom{N}{S}$). Hence computational efficiency is an important requirement in our experimental design.

Measuring actual performance of ensembles efficiently. Given a candidate ensemble, we fine-tune each member model individually and apply it individually on the target test set. Then we take the average of their predictions as the ensemble prediction. Finally, we measure actual ensemble performance as mean IoU. By considering the predictions of each ensemble member individually, we can reuse them across ensembles. Hence, we only need to run inference for each model on a target test sample *once*, regardless of the number of ensembles the model participates in.

While it would also be possible to fine-tune each ensemble as a whole, this is too computationally expensive for this experiment. We do it in a different setting in Sec. 5.

Ensemble size. We fix the number of models in an ensemble to $S = 3$. We found $S = 3$ to be a good compromise to benefit from a diverse ensemble, while limiting overall computation (more details in suppl. mat.).

Subsampling candidate source models. For a given target dataset, the total number of candidate source models is 64 (68 minus the 4 trained on that target). We reduce the number of candidates to gain two types of speed-ups. Firstly, this requires fine-tuning fewer source models on the target training set. But more importantly, the number of candidate ensembles grows factorially with N: for $N = 64$ and $S = 3$ there are 41k possible ensembles. While this is not a problem for calculating our transferability metrics, evaluating the performance of 41k ensembles on the target test set is computationally prohibitive. Hence we limit $N = 15$ as described below, yielding 455 candidate ensembles.

For each target dataset we sample 15 source models as follows. As the final goal is to select a high-performing ensemble, we pick 5 good source models for that target. We first compute all our transferability metrics for all ensembles of 3 source models from the complete pool ($N = 64$). We then select the 5 most frequent models in the top-ranked ensembles across all metrics. Since we want to evaluate the ability to distinguish between good and bad sources, we include an additional 10 source models at random.

Target datasets. We consider five target datasets: Camvid [8], ISPRS [58], vKITTI2 [9], KITTI [1] and Pascal VOC [21]. Since transfer learning is particularly interesting in scenarios with limited training images [37, 49, 52], we follow [49] and limit each target training set to 150 images.

Correlation measure: weighted Kendall Tau. A transferability metric is useful when it can order the candidate source models (or ensembles) according to their actual performance. The exact predicted performance value is less important. Therefore we use a rank correlation measure as in [44, 49]. Moreover, as in practice we mainly care about selecting a high performing ensemble, we follow [76] and use a weighted version of Kendall τ considering top-ranked items more important than low-ranked ones [68].

Baseline transferability metric. As there is no previously proposed transferability metric for source model ensembles, we introduce a simple baseline. It is based on three factors: (1) source model performance P_s evaluated on a source test set, (2) source dataset size N_s in terms of number of images, and (3) source dataset richness measured by the number of source classes C_c. For a candidate ensemble containing S source models, we calculate this baseline as: $\text{BASE} = \sum_{s=1}^{S}(P_s \times N_s \times C_s)$. This baseline is target-agnostic, favours broad source datasets (COCO, ADE20k, Mapillary), and the best model architecture (HRNetV2-W28, pre-trained fully supervised; as it typically leads to higher P_s).

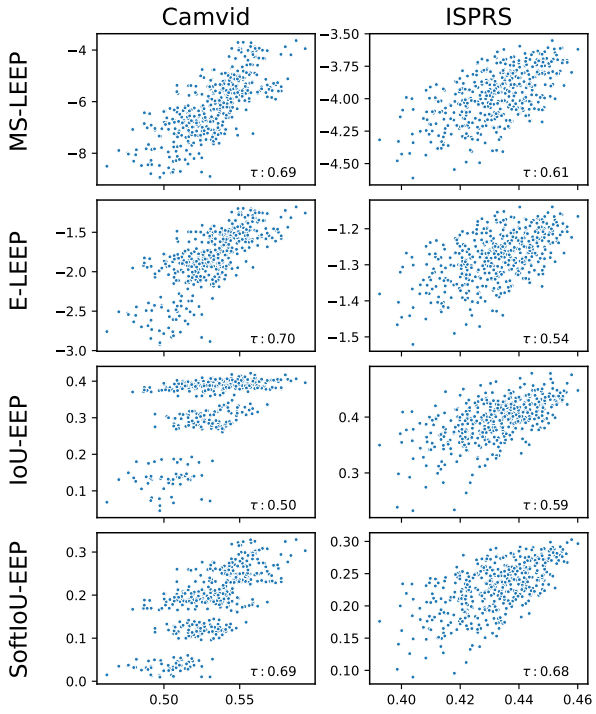

Figure 2. Relation between the predicted transferability (y-axis) and the actual mean IoU performance (x-axis) on the target test set (for 2 of the 5 target datasets). Each plot shows 455 candidate ensembles as a separate dot, and also reports the corresponding weighted Kendall's τ correlation score. These scores are generally high, demonstrating the success of our transferability metrics.

4.2. Results

Fig. 2 shows qualitatively the relation between predicted transferability and actual mean IoU for all our transferability metrics on two target datasets. In all experiments we see good positive correlations, demonstrating that our metrics work well. We also see that ensemble performance varies greatly depending on the sources used, justifying the importance of having a good ensemble selection mechanism.

Quantitatively, Fig. 3 reports the weighted Kendall's τ for our transferability metrics and the baseline (for all 5 target datasets). All our transferability metrics generally achieve high scores, significantly outperforming the baseline on each target dataset. Among our metrics, the direct LEEP variants, MS-LEEP and E-LEEP, perform equally well. Next, IoU-EEP performs the worst, suggesting that it is important to consider the probability distribution $P_{ens}(\cdot|x_i)$. Finally, SoftIoU-EEP performs best, confirming the benefits of directly approximating the performance measure of the target test set (mean IoU). On average, SoftIoU-EEP achieves τ of 69.3%, outperforming the baseline by 20.3%, MS-LEEP by 3.1%, E-LEEP by 3.6%, IoU-EEP by 8.4%.

As related works on single-source transferability metrics sometimes report the standard Kendall τ [44] or the (linear) Pearson coefficient [44, 55, 65], we also report them in Tab. 3. Despite each correlation measure is based on different mathematical assumptions, the general trend among target datasets and transferability metrics remains the same.

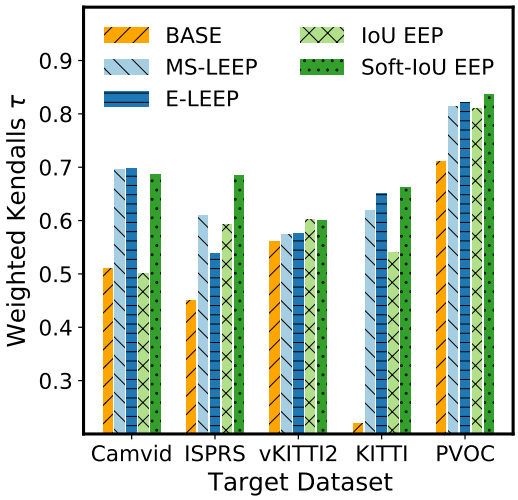

Figure 3. Comparison of our transferability metrics, over 5 target datasets. We show the weighted Kendall's τ between a metric and actual mean IoU on the target test set, where each ranks all 455 candidate ensembles. SoftIoU-EEP performs best overall.

5. Evaluating ensemble selection

5.1. Experimental Setup

We now turn to ensemble selection: we use our best transferability metric, SoftIoU-EEP, to select the ensemble with the highest predicted transferability. We fine-tune this ensemble on the target training set and evaluate on the target test set. Selecting only a single ensemble per target dataset on which to perform expensive fine-tuning and evaluation enables us to expand our experimental setup beyond what we did in Sec. 4. Instead of using $N = 15$, we now use all $N = 64$ source models as candidates. Moreover, instead of only fine-tuning individual ensemble members, we add an additional step of ensemble-specific fine-tuning. We compare to two baselines which select a single source model.

Improved actual performance of ensembles. Given an ensemble, we start by fine-tuning each member individually on the target training set as in Sec. 4.1. Afterwards, we perform additional ensemble-specific fine-tuning to improve it by re-weighting the class predictions of its members. Specifically, we freeze the backbone of each member and attach to it a light head that assigns per-class weights and biases, before averaging their predictions into a single ensemble prediction. We then fine-tune the light head of this ensemble on the target training set. While the total number of additional parameters introduced by this head depends on the number of target classes, in practice it is between 144 and 2700, which is negligible compared to the total number of parameters in the models composing the ensemble.

Ensemble size. We set $S = 3$ as in Sec. 4.

Measure	Weighted Kendall's τ					Kendall's τ					Pearson				
Method	BASE	MS	E	IoU	sIoU	BASE	MS	E	IoU	sIoU	BASE	MS	E	IoU	sIoU
Camvid	0.51	0.69	**0.70**	0.50	0.68	0.30	**0.56**	0.55	0.37	0.51	0.46	**0.74**	0.73	0.56	0.70
ISPRS	0.45	0.61	0.54	0.59	**0.68**	0.22	0.41	0.37	0.41	**0.44**	0.29	0.60	0.55	0.59	**0.63**
vKITTI2	0.56	0.57	0.58	**0.60**	**0.60**	0.38	0.49	0.51	0.51	**0.53**	0.54	0.66	0.67	0.62	**0.70**
KITTI	0.22	0.62	0.65	0.54	**0.66**	0.17	0.50	**0.54**	0.42	0.53	0.25	0.69	0.73	0.62	**0.74**
PVOC	0.71	0.81	0.82	0.81	**0.83**	0.42	0.59	**0.61**	0.58	**0.61**	0.66	**0.84**	0.83	0.73	0.81
Average	0.49	0.66	0.66	0.61	**0.69**	0.30	0.51	**0.52**	0.46	**0.52**	0.44	0.71	0.70	0.62	**0.72**

Table 3. Correlation measures for 5 different target datasets, obtained by comparing the transferability predicted by a metric to the actual Mean IoU on the target test set. The methods shortcuts refer to: (BASE=Baseline), (MS=MS-LEEP), (E=E-LEEP), (IoU=IoU-EEP), (sIoU=SoftIoU-EEP). SoftIoU-EEP performs best overall. See Sec. 4.1 for more details about the setup.

Target datasets. We now consider each of the 17 datasets in Tab. 2 as a target dataset in turn (instead of 5 in Sec. 4). As in Sec. 4, we study transfer learning in the low data regime defined by [49] (i.e. 150 target training images for each dataset, except 1000 for COCO and ADE20k as they contain many classes).

Baseline B1: select a single source model. We test whether aggregating multiple source models is beneficial to transfer learning, compared to the standard of selecting a single source model [44,55,65,76]. To do so, for a given target dataset we use the same pool of 64 source models as for ensemble selection. Then we select the single model with the highest predicted transferabiliy according to LEEP [55]. Finally, we fine-tune it on the target training set, and evaluate on the target test set.

Baseline B1L: select a single *large* source model. We also propose a stronger baseline which selects a single large model with the same number of parameters as one of our ensembles. An ensemble of $S = 3$ has 69M parameters, as both HRNetV2-W28 and ResNet23M have 23M parameters. We use the best such model according to preliminary experiments: HRNetV2-W48 pre-trained fully supervised on ILSVRC'12. It offers excellent performance for semantic segmentation [69] and was also used in [41,49].

For this baseline we need to construct a new pool of large source models. To do so, we train 17 large models, one for each source dataset (Tab. 2). Then, given a target dataset, we use LEEP [55] to select a single model (out of 16, excluding that target dataset).

5.2. Results

Comparison to baselines B1 and B1L. We first consider the improvement made by the ensemble selected by our transferability metric SoftIoU-EEP over the baseline B1. On average across all 17 dataset, selecting an ensemble instead of a single model out of the same pool (B1) improves results by a relative +6.0% mean IoU. This demonstrates that the source models in our pool are complementary and composing several into an ensemble brings clear benefits.

Next, we show in Fig. 4 that selecting an ensemble out of our pool even outperforms baseline B1L, which selects a single model out of a pool of *larger models*, each with capacity similar to one of our ensembles (+2.5% relative mean IoU). When looking at individual target datasets, the selected ensemble improves over B1L by more than 2% in 10 of them. In contrast, B1L outperforms the ensemble only once (on Pascal Context). We conclude that selecting an ensemble brings solid accuracy benefits over selecting a single model, even when controlling for total capacity.

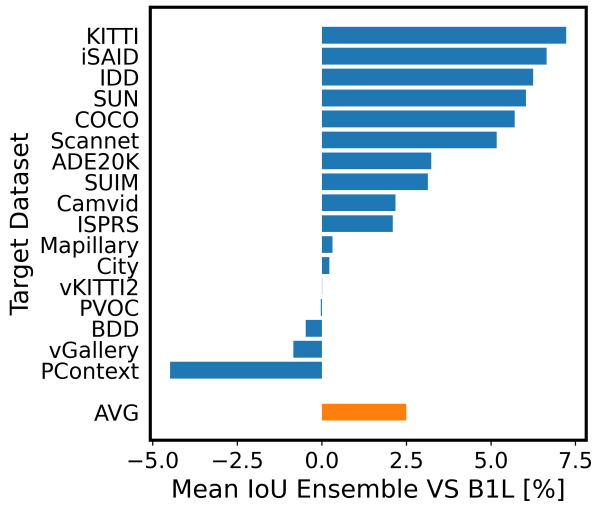

Figure 4. Relative mean IoU gain on each target test set made by the ensemble selected by our SoftIoU-EEP metric over baseline B1L. For clarity, we set the 0 vertical line to corresponds to the performance of B1L. On average over all target datasets, the improvement is 2.5%. We do not explicitly display baseline B1, as it performs worse than B1L (see main text for discussion).

What factors of diversity matter more? Generally, ensembles benefit from the predictive diversity of its member models [5,17,28,47]. To understand which factors of diversity are important in our transfer learning setting (Sec. 3.1), Fig. 5 shows which source models are selected by SoftIoU-EEP as part of the winning ensemble for each target dataset.

We observe the following patterns: (1) Some target datasets are well covered by multiple source datasets. For these cases, our metric generates diversity by selecting source models trained on these different datasets. For example, for ScanNet (indoor) as a target, our metric selects source models trained on Sun (indoor), COCO, and

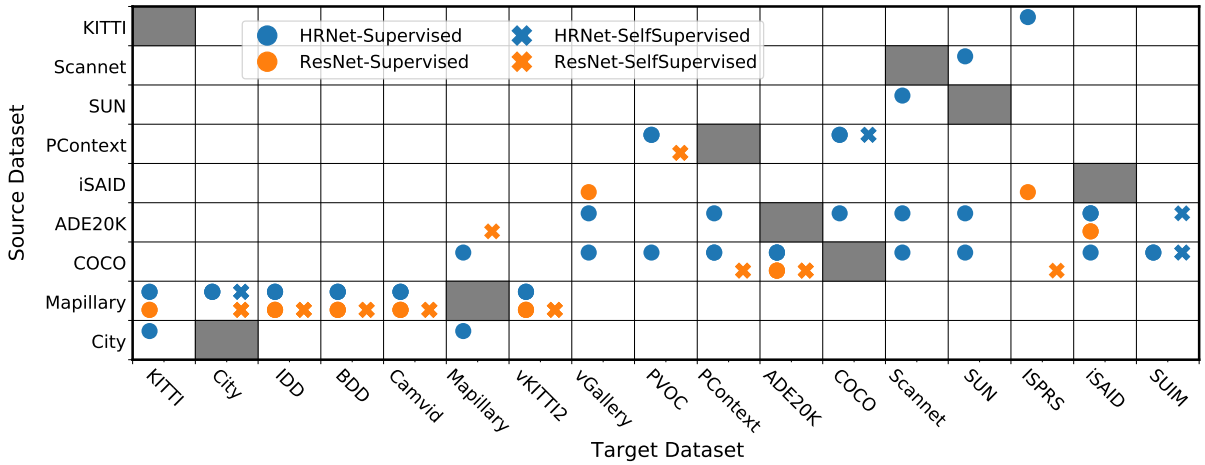

Figure 5. For each target dataset (x-axis), we show the 3 source models in the ensemble selected by SoftIoU-EEP. When multiple source datasets cover the image domain of the target, our metric generates diversity by selecting models trained on these different source datasets (see target ScanNet and Sun). When there exists a single strong source dataset for a target, our metric selects only this source dataset, and instead generates diversity by varying architectures and pre-training schemes (see targets from the driving domain). Note: we do not consider source models trained on the same dataset as the target (gray cells).

ADE20K, as COCO and ADE20K contain many indoor images. In these cases the backbone of choice is HRNetV2-W28 using fully supervised pre-training, which is generally the best perfoming backbone (76% of all source models selected). (2) For other target datasets there exists a strong source dataset which alone already covers most variations in the target domain. For example, Mapillary is the largest dataset in the driving domain and covers all continents. Our method nicely picks Mapillary almost exclusively as the sole source dataset for all target driving datasets (Cityscapes, IDD, BDD, Camvid, vKITTI2, and KITTI as the only case with a second source - Cityscapes). In these cases, our metric generates diversity by varying model architectures and pre-training schemes within the selected ensemble. (3) The most frequently selected source datasets have a greater number of training samples, greater number of labels, and larger diversity of images (Mapillary, COCO, ADE20K). These observations are in line with earlier work which show the benefits of in-domain source images [49,54,75] and large, broad source sets [37,48,49,64].

Ensemble vs. its members. In Fig. 6 we compare the mean IoU of the selected ensemble to that of its member models. On average the ensemble improves over its best member by 4.6%, and over its worst member by 18.6%. These improvements demonstrate that the sources selected to be part of the winning ensemble are indeed diverse (otherwise the ensemble could not outperform its best member). Hence we conclude that our transferability metric is good in selecting a diverse set of source models.

Finally, to evaluate how important it is to *learn* to combine model predictions, we re-evaluate the results but now using a simple unweighted average of the model predictions instead of the learned head. This still outperforms the best single ensemble member by +2.9%. Hence, our improvements are truly due to ensembling models (be it with a fixed combiner head, or with a learned one).

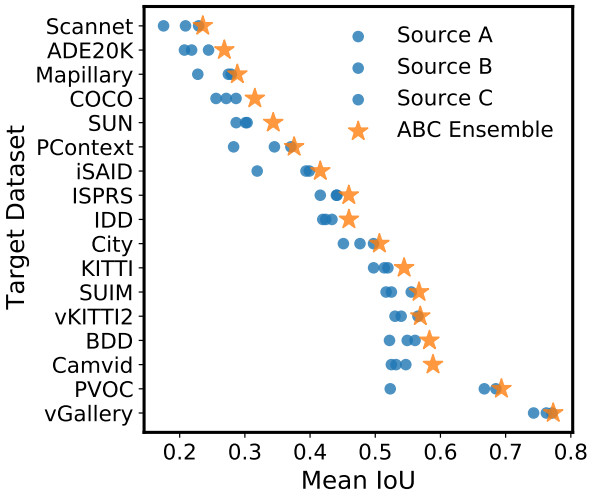

Figure 6. Comparison of the mean IoU of the winning ensemble vs its component source models.

6. Conclusion

We design for the first time transferability metrics for ensemble selection. We evaluate them in a challenging and realistic transfer learning setup for semantic segmentation, featuring 17 source datasets covering a wide variety of image domain, two model architectures, and two pre-training schemes. We show experimentally that our transferability metrics rank correlate well with actual transfer learning performance. Moreover, our best metric selects an ensemble performing better than two baselines which select a single source model (even after equalizing capacity).

References

[1] Hassan Alhaija, Siva Mustikovela, Lars Mescheder, Andreas Geiger, and Carsten Rother. Augmented reality meets computer vision : Efficient data generation for urban driving scenes. *International Journal of Computer Vision*, 2018. 3, 5

[2] James Urquhart Allingham, Florian Wenzel, Zelda E Mariet, Basil Mustafa, Joan Puigcerver, Neil Houlsby, Ghassen Jerfel, Vincent Fortuin, Balaji Lakshminarayanan, Jasper Snoek, Dustin Tran, Carlos Riquelme, and Rodolphe Jenatton. Sparse MoEs meet efficient ensembles. In *arXiv*, 2021. 2

[3] Hossein Azizpour, Ali Sharif Razavian, Josephine Sullivan, Atsuto Maki, and Stefan Carlsson. Factors of transferability for a generic convnet representation. *IEEE Trans. on PAMI*, 2015. 1

[4] Yajie Bao, Yang Li, Shao-Lun Huang, Lin Zhang, Lizhong Zheng, Amir Zamir, and Leonidas Guibas. An information-theoretic approach to transferability in task transfer learning. In *ICIP*, 2019. 1, 2, 3, 4

[5] Yijun Bian and Huanhuan Chen. When does diversity help generalization in classification ensembles? *IEEE Transactions on Cybernetics*, 2021. 3, 7

[6] Daniel Bolya, Rohit Mittapalli, and Judy Hoffman. Scalable diverse model selection for accessible transfer learning. *Advances in Neural Information Processing Systems*, 34, 2021. 2

[7] Leo Breiman. Bagging predictors. *Machine learning*, 1996. 1, 2

[8] 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. 3, 5

[9] Yohann Cabon, Naila Murray, and Martin Humenberger. Virtual kitti 2. *arXiv*, 2020. 3, 5

[10] Holger Caesar, Jasper Uijlings, and Vittorio Ferrari. COCO-Stuff: Thing and stuff classes in context. In *CVPR*, 2018. 3

[11] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE Trans. on PAMI*, 2017. 3

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

[13] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *CVPR*, 2021. 2

[14] Brian Chu, Vashisht Madhavan, Oscar Beijbom, Judy Hoffman, and Trevor Darrell. Best practices for fine-tuning visual classifiers to new domains. In *ECCV*, 2016. 1

[15] 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. 3

[16] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. ScanNet: Richly-annotated 3d reconstructions of indoor scenes. In *CVPR*, 2017. 3

[17] Thomas G Dietterich. Ensemble methods in machine learning. In *International workshop on multiple classifier systems*, pages 1–15. Springer, 2000. 1, 2, 3, 7

[18] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021. 2

[19] Kshitij Dwivedi, Jiahui Huang, Radoslaw Martin Cichy, and Gemma Roig. Duality diagram similarity: a generic framework for initialization selection in task transfer learning. In *European Conference on Computer Vision*, pages 497–513. Springer, 2020. 2

[20] Kshitij Dwivedi and Gemma Roig. Representation similarity analysis for efficient task taxonomy & transfer learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12387–12396, 2019. 2

[21] Mark Everingham and SM Ali Eslami. Van˜ gool, l. *Williams, CKI, Winn, J. and Zisserman, A. The PASCAL Visual Object Classes Challenge*, 2012. 5

[22] 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. 3

[23] Stanislav Fort, Huiyi Hu, and Balaji Lakshminarayanan. Deep ensembles: A loss landscape perspective. *arXiv*, 2019. 1, 2

[24] Yoav Freund and Robert Schapire. Experiments with a new boosting algorithm. In *ICML*, 1996. 1, 2, 3

[25] Adrien Gaidon, Qiao Wang, Yohann Cabon, and Eleonora Vig. Virtual worlds as proxy for multi-object tracking analysis. In *CVPR*, 2016. 3

[26] R. Girshick. Fast R-CNN. In *ICCV*, 2015. 1

[27] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *ICML*, 2017. 5

[28] Huaping Guo, Hongbing Liu, Ran Li, Changan Wu, Yibo Guo, and Mingliang Xu. Margin & diversity based ordering ensemble pruning. *Neurocomputing*, 2018. 7

[29] Lars Kai Hansen and Peter Salamon. Neural network ensembles. *IEEE transactions on pattern analysis and machine intelligence*, 12(10):993–1001, 1990. 2, 3

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

[31] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask R-CNN. In *ICCV*, 2017. 1

[32] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 3

[33] M. Huh, P. Agrawal, and A.A. Efros. What makes imagenet good for transfer learning? *NeurIPS LSCVS workshop*, 2016. 1

[34] Ashraful Islam, Chun-Fu Chen, Rameswar Panda, Leonid Karlinsky, Richard Radke, and Rogerio Feris. A broad study on the transferability of visual representations with contrastive learning. *ICCV*, 2021. 2, 3

[35] Md Jahidul Islam, Chelsey Edge, Yuyang Xiao, Peigen Luo, Muntaqim Mehtaz, Christopher Morse, Sadman Sakib Enan, and Junaed Sattar. Semantic Segmentation of Underwater Imagery: Dataset and Benchmark. In *IROS*, 2020. 3

[36] A. Kirillov, K. He, R. Girshick, C. Rother, and P. Dollár. Panoptic segmentation. In *CVPR*, 2019. 3

[37] Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Joan Puigcerver, Jessica Yung, Sylvain Gelly, and Neil Houlsby. Big transfer (bit): General visual representation learning. In *ECCV*, 2020. 2, 5, 8

[38] Simon Kornblith, Jonathon Shlens, and Quoc V Le. Do better imagenet models transfer better? In *CVPR*, 2019. 1

[39] Anders Krogh and Jesper Vedelsby. Neural network ensembles, cross validation, and active learning. In *NeurIPS*, 1995. 1, 2, 3

[40] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *NeurIPS*, 2017. 1, 2

[41] John Lambert, Zhuang Liu, Ozan Sener, James Hays, and Vladlen Koltun. MSeg: A composite dataset for multi-domain semantic segmentation. In *CVPR*, 2020. 3, 7

[42] Stefan Lee, Senthil Purushwalkam, Michael Cogswell, David Crandall, and Dhruv Batra. Why m heads are better than one: Training a diverse ensemble of deep networks. In *arXiv*, 2015. 2

[43] Suichan Li, Dongdong Chen, Yinpeng Chen, Lu Yuan, Lei Zhang, Qi Chu, Bin Liu, and Nenghai Yu. Improve unsupervised pretraining for few-label transfer. In *CVPR*, pages 10201–10210, 2021. 2, 3

[44] Yandong Li, Xuhui Jia, Ruoxin Sang, Yukun Zhu, Bradley Green, Liqiang Wang, and Boqing Gong. Ranking neural checkpoints. In *CVPR*, 2021. 1, 2, 3, 4, 5, 6, 7

[45] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. Microsoft COCO: Common objects in context. In *ECCV*, 2014. 3

[46] Iou Jen Liu, Jian Peng, and Alexander G Schwing. Knowledge flow: Improve upon your teachers. In *ICLR*, 2019. 2

[47] Zhenyu Lu, Xindong Wu, Xingquan Zhu, and Josh Bongard. Ensemble pruning via individual contribution ordering. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 871–880, 2010. 7

[48] 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. 8

[49] Thomas Mensink, Jasper Uijlings, Alina Kuznetsova, Michael Gygli, and Vittorio Ferrari. Factors of influence for transfer learning across diverse appearance domains and task types. *arXiv*, 2021. 1, 2, 3, 5, 7, 8

[50] 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. 3

[51] Basil Mustafa, Aaron Loh, Jan Freyberg, Patricia MacWilliams, Alan Karthikesalingam, Neil Houlsby, and Vivek Natarajan. Supervised transfer learning at scale for medical imaging. *arXiv*, 2021. 2

[52] Basil Mustafa, Carlos Riquelme, Joan Puigcerver, André Susano Pinto, Daniel Keysers, and Neil Houlsby. Deep ensembles for low-data transfer learning. *arXiv*, 2020. 2, 5

[53] Gerhard Neuhold, Tobias Ollmann, Samuel Rota Bulò, and Peter Kontschieder. The mapillary vistas dataset for semantic understanding of street scenes. In *ICCV*, 2017. 3

[54] Jiquan Ngiam, Daiyi Peng, Vijay Vasudevan, Simon Kornblith, Quoc V. Le, and Ruoming Pang. Domain adaptive transfer learning with specialist models. In *arXiv*, 2018. 1, 3, 8

[55] Cuong Nguyen, Tal Hassner, Matthias Seeger, and Cedric Archambeau. LEEP: A new measure to evaluate transferability of learned representations. In *ICML*, 2020. 1, 2, 3, 4, 5, 6, 7

[56] Yaniv Ovadia, Emily Fertig, J. Ren, Zachary Nado, D. Sculley, Sebastian Nowozin, Joshua V. Dillon, Balaji Lakshminarayanan, and Jasper Snoek. Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. In *NeurIPS*, 2019. 1, 2

[57] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 2009. 3

[58] Franz Rottensteiner, Gunho Sohn, Markus Gerke, Jan Dirk Wegner, Uwe Breitkopf, and Jaewook Jung. Results of the isprs benchmark on urban object detection and 3d building reconstruction. *ISPRS journal of photogrammetry and remote sensing*, 2014. 3, 5

[59] E. Shelhamer, J. Long, and T. Darrell. Fully convolutional networks for semantic segmentation. *IEEE Trans. on PAMI*, 2016. 1, 3

[60] Yang Shu, Zhi Kou, Zhangjie Cao, Jianmin Wang, and Mingsheng Long. Zoo-tuning: Adaptive transfer from a zoo of models. In *ICML*, 2021. 2

[61] Jie Song, Yixin Chen, Xinchao Wang, Chengchao Shen, and Mingli Song. Deep model transferability from attribution maps. *Advances in Neural Information Processing Systems*, 32, 2019. 2

[62] Jie Song, Yixin Chen, Jingwen Ye, Xinchao Wang, Chengchao Shen, Feng Mao, and Mingli Song. Depara: Deep attribution graph for deep knowledge transferability. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3922–3930, 2020. 2

[63] S. Song, S. Lichtenberg, and J. Xiao. SUN RGB-D: A RGB-D scene understanding benchmark suite. In *CVPR*, 2015. 3

[64] C. Sun, A. Shrivastava, S. Singh, and A. Gupta. Revisiting unreasonable effectiveness of data in deep learning era. In *ICCV*, 2017. 8

[65] Yang Tan, Yang Li, and Shao-Lun Huang. Otce: A transferability metric for cross-domain cross-task representations. In *CVPR*, 2021. 1, 2, 3, 4, 5, 6, 7

[66] Anh T Tran, Cuong V Nguyen, and Tal Hassner. Transferability and hardness of supervised classification tasks. In *CVPR*, 2019. 1, 2, 3, 4

[67] Girish Varma, Anbumani Subramanian, Anoop Namboodiri, Manmohan Chandraker, and C.V. Jawahar. Idd: A dataset for exploring problems of autonomous navigation in unconstrained environments. In *Proc. WACV*, 2019. 3

[68] Sebastiano Vigna. A weighted correlation index for rankings with ties. In *Proceedings of the 24th international conference on World Wide Web*, 2015. 5

[69] 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. 3, 7

[70] Xiaofang Wang, Dan Kondratyuk, Eric Christiansen, Kris M Kitani, Yair Movshovitz-Attias, and Elad Eban. On the surprising efficiency of committee-based models. *arXiv*, 2020. 1

[71] Syed Waqas Zamir, Aditya Arora, Akshita Gupta, Salman Khan, Guolei Sun, Fahad Shahbaz Khan, Fan Zhu, Ling Shao, Gui-Song Xia, and Xiang Bai. isaid: A large-scale dataset for instance segmentation in aerial images. In *CVPR Workshops*, 2019. 3

[72] Philippe Weinzaepfel, Gabriela Csurka, Yohann Cabon, and Martin Humenberger. Visual localization by learning objects-of-interest dense match regression. In *CVPR*, 2019. 3

[73] Zifeng Wu, Chunhua Shen, and Anton Van Den Hengel. Wider or deeper: Revisiting the resnet model for visual recognition. *Pattern Recognition*, 2019. 3

[74] Gui-Song Xia, Xiang Bai, Jian Ding, Zhen Zhu, Serge Belongie, Jiebo Luo, Mihai Datcu, Marcello Pelillo, and Liangpei Zhang. Dota: A large-scale dataset for object detection in aerial images. In *CVPR*, June 2018. 3

[75] Xi Yan, David Acuna, and Sanja Fidler. Neural data server: A large-scale search engine for transfer learning data. In *CVPR*, 2020. 1, 2, 8

[76] Kaichao You, Yong Liu, Jianmin Wang, and Mingsheng Long. LogME: Practical assessment of pre-trained models for transfer learning. In *ICML*, 2021. 1, 2, 3, 4, 5, 7

[77] Fisher Yu, Haofeng Chen, Xin Wang, Wenqi Xian, Yingying Chen, Fangchen Liu, Vashisht Madhavan, and Trevor Darrell. Bdd100k: A diverse driving dataset for heterogeneous multitask learning. In *CVPR*, 2020. 3

[78] Amir R Zamir, Alexander Sax, William Shen, Leonidas J Guibas, Jitendra Malik, and Silvio Savarese. Taskonomy: Disentangling task transfer learning. In *CVPR*, 2018. 2

[79] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, and A. Torralba. Scene parsing through ADE20K dataset. In *CVPR*, 2017. 3

[80] Xingyi Zhou, Dequan Wang, and Philipp Krähenbühl. Objects as points. In *arXiv*, 2019. 1