ETran: Energy-Based Transferability Estimation

Mohsen Gholami[1,2], Mohammad Akbari[1], Xinglu Wang[1], Behnam Kamranian [1], Yong Zhang[1]
[1]Huawei Technologies Canada Co., Ltd., [2]University of British Columbia
{mohsen.gholami,mohammad.akbari,xinglu.wang,behnam.kamranian,yong.zhang3}@huawei.com

Abstract

This paper addresses the problem of ranking pre-trained models for object detection and image classification. Selecting the best pre-trained model by fine-tuning is an expensive and time-consuming task. Previous works have proposed transferability estimation based on features extracted by the pre-trained models. We argue that quantifying whether the target dataset is in-distribution (IND) or out-of-distribution (OOD) for the pre-trained model is an important factor in the transferability estimation. To this end, we propose ETran*, an energy-based transferability assessment metric, which includes three scores: 1) energy score, 2) classification score, and 3) regression score. We use energy-based models to determine whether the target dataset is OOD or IND for the pre-trained model. In contrast to the prior works,* ETran *is applicable to a wide range of tasks including classification, regression, and object detection (classification+regression). This is the first work that proposes transferability estimation for object detection task. Our extensive experiments on four benchmarks and two tasks show that* ETran *outperforms previous works on object detection and classification benchmarks by an average of 21% and 12%, respectively, and achieves SOTA in transferability assessment. Code is available here[1].*

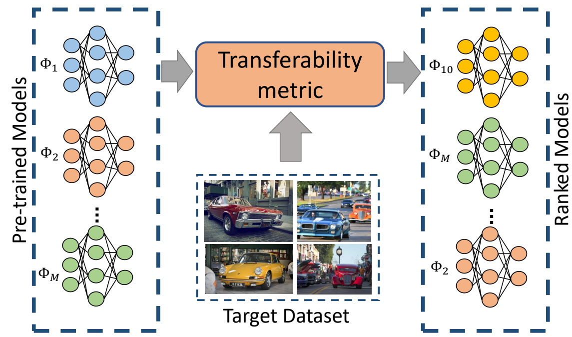

Figure 1: The overall framework of transferability assessment, given M pre-trained models and a target dataset.

1. Introduction

Pre-trained neural networks are widely available on platforms such as HuggingFace [50] and TensorFlowHub [35] for different tasks such as classification, object detection, segmentation, and natural language processing. These pre-trained models, which have acquired the fundamental knowledge in vision [25] or language [9] domains, are very important in transfer learning to downstream target tasks with limited training data. Since training these models from scratch is computationally expensive, task-specific fine-tuning from the pre-trained checkpoints is commonly considered a time- and cost-efficient alternative solution.

One of the major challenges in transfer learning is to select the best pre-trained model for a target task (or dataset), given numerous pre-trained models. The trivial solution to this problem is brute-force fine-tuning, where all the given pre-trained models need to be fine-tuned on the given dataset and then the best-performing fine-tuned model is chosen for the target task. However, this procedure is very time-consuming and computationally expensive although it is highly accurate. Recent studies have proposed fast transferability assessment and ranking solutions to properly and quickly rank the models and select the best ones (Figure 1).

Most of the previous works extract features from the target dataset using the pre-trained models and try to find the model whose features can be more effectively mapped to the labels from the target dataset (e.g., $\mathcal{N}$LEEP [29], LogME [52], PACTran [10], and SFDA [43]). In other words, these methods try to mitigate the fine-tuning process in order to find a model that is more compatible with the given data samples based on the extracted features.

One limitation of these approaches is that the extracted features are drastically different before and after fine-tuning, which is due to the difference in source and target domains. As a result, the transferability assessment only based on the extracted features cannot lead to a reliable,

[1]https://developer.huaweicloud.com/develop/aigallery/notebook/detail?id=56e6645a-8133-49f6-a7ef-d877ef608fa5

general, and task-agnostic solution for obtaining an optimal pre-trained model. This uncertainty arises when the pre-trained model sees the input that differs from its training data (called in-distribution data), which can result in unreliable features and predictions [31]. In other words, if the target dataset does not follow the data distribution with which the model has been trained, the extracted features cannot be reliable for transferability assessment. Thus, determining whether the target dataset is in-distribution (IND) or out-of-distribution (OOD) is an essential assessment factor in finding the best pre-trained model.

In this work, we propose an energy-empowered transferability estimation method (called *ETran*) that exploits energy-based models (EBM) [28] to detect whether a target dataset is IND or OOD for a given pre-trained model. To this end, the higher the energy score for a target dataset, the more IND this dataset is for the pre-trained model [31, 2, 3, 4]. As a consequence, the corresponding model has a high likelihood to provide the best accuracy after fine-tuning (for the given target dataset) compared to the models with lower energy scores. In contrast to the previous transferability metrics, the energy score is label- and optimization-free, which makes it highly efficient and easy to use.

Another major limitation of most of the previous works is that they are only applicable to classification tasks. H-Score [6] and LogME [52] are the only works in the literature that introduce a solution for regression as well. Unlike the previous works, our method can deal with all classification, regression, and object detection tasks. To the best of our knowledge, *ETran* is the first transferability assessment approach that is also applicable to object detection (i.e., a combination of classification and regression tasks).

In addition to the energy score, we also propose to use classification and regression scores to benefit from the feature-based characteristics of general transferability metrics as in the previous methods. For the classification score, we use Linear Discriminate Analysis (LDA) [19] to project features to a discriminate space by maximizing the variance between class labels and minimizing the variance within the classes. Bayes' theorem is then applied to measure the probability of class labels given the input features. For the regression score, we employ a solution based on Singular Value Decomposition (SVD) [17] that has fewer assumptions and better performance compared to LogME. Our experiments show that the regression score is crucial for an accurate transferability measurement on the object detection tasks. Figure 2 shows the overall framework of *ETran*.

The major **contributions** of this work are as follows:

- An energy-based transferability assessment metric that is label- and optimization-free; and also applicable to both classification and detection.
- Proposing two more scores based on LDA and SVD for transferability measurement on classification and object detection.
- Proposing the first transferability metric for object detection tasks, and introducing multiple corresponding benchmarks and baselines that avail future research studies.
- Achieving SOTA results in transferability assessment for image classification and object detection.

2. Related Work

In this section, we discuss the transferability estimation methods introduced in the literature for both classification (including LEEP [36], $\mathcal{N}$LEEP [29], PACTran [10], SFDA [43], and GBC [38]) and regression (including H-Score [6] and LogME [52]) tasks.

LEEP [36] estimates the transferability of a source model to a target dataset by estimating two probabilities: 1) the predicted classes of the target dataset by the source model. LEEP used the prediction head of the source model which makes the transferability estimation limited to the source models trained in a supervised fashion for classification tasks. In contrast to LEEP, $\mathcal{N}$LEEP [29], PACTran [10], SFDA [43], and GBC [38] use the features extracted from the target dataset by the source model to estimate the transferability. $\mathcal{N}$LEEP is indeed an extension of LEEP, which replaces the head detection of the source model with a Gaussian Mixture Model (GMM) fitted on the target data and then computes the LEEP score. PACTran [10] argues that LEEP and $\mathcal{N}$LEEP overlook the generalization of source models and emphasize the training error of the source dataset. Therefore, PACTran uses a linear model with a flatness regularizer to fit features to target labels using an optimization approach.

SFDA [43] and GBC [38] propose that class separability of the target dataset in the features space of source models is an important factor for transferability in classification scenarios. SFDA proposes to project features to a class separable space before applying a Bayes classifier. GBC uses Bhattacharyya coefficients to estimate the overlap between target classes in the features extracted by the source models.

All of the above-mentioned works are exclusively applicable to classification tasks. These methods cannot be directly applied to the regression tasks due to the use of cross-entropy loss function [10] or the class separating-based metrics [38, 43]. On the other hand, H-Score [6] and LogME [52] are the two methods in the literature that use least squares optimization for regression tasks. Their methods are also applicable to classification, where the problem is treated as a multi-variant regression problem.

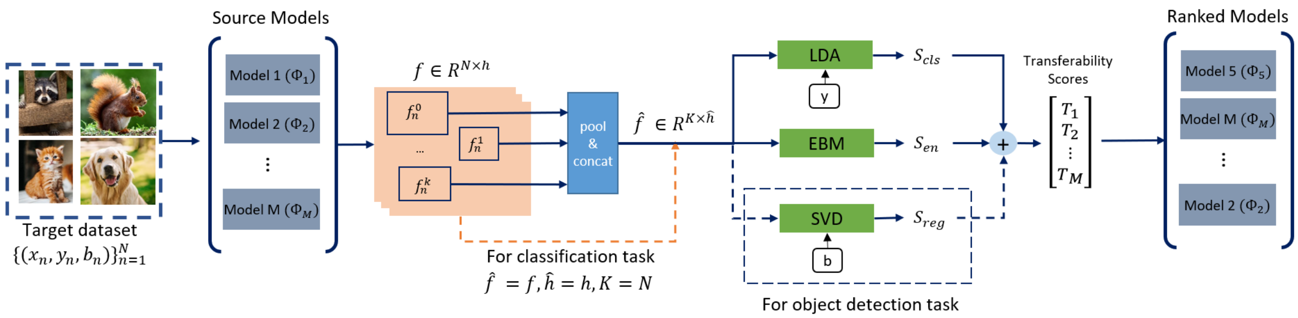

Figure 2: Overview of *ETran*'s framework. Φ_m: mth pre-trained source model, f: extracted features for the entire image, $f_{(n)}^k$: extracted features from n-th image and k-th bbox in the image (for object detection case), S_{cls}: LDA-based classification score, S_{reg}: SVD-based regression score, S_{en}: Energy-based score, T_m: the overall transferability score for the mth model.

3. Method

3.1. Problem Formulation

Classification. Given M pre-trained models $\{\Phi_m\}_{m=1}^M$ and a target dataset $\mathcal{D} = \{(x_n, y_n)\}_{n=1}^N$ (N: number of samples and y: ground-truth labels), the transferability metric for the m-th model is then computed as a scalar score T_m as follows:

$$T_m = \frac{1}{N}\sum_{n=1}^{N} p(y_n|x_n, \Phi_m), \tag{1}$$

where $p(y_n|x_n, \Phi_m)$ is the probability of label y_n, given the input data x_n and the source model Φ_m.
Object Detection. For the case of object detection, the target dataset is defined as $\mathcal{D} = \{(x_n, y_n, b_n)\}_{n=1}^N$, where b_n denotes the bounding boxes (bboxes) labels in the n-th data sample. The transferability metric is also modified to $T_m = \frac{1}{N}\sum_{n=1}^{N} p(y_n, b_n|x_n, \Phi_m)$.

3.2. Feature Construction

The transferability scores in our work are computed over the features extracted from the target dataset by the source models. In the classification scenario, the corresponding extracted features (f) are directly used as the input to the transferability metrics. However, preparing features for the object detection task is not straightforward due to the presence of bboxes in addition to the class labels. Using the entire feature vector f for this task does not precisely provide bbox-specific features aligned with the ground-truth bboxes. In order to construct bbox-specific feature vectors, denoted by f_n^k, the ground-truth bboxes from the entire target dataset are utilized. To this end, f_n^k represents the feature values in f that exist in the relative position of the k-th bbox of the n-th sample. Since bboxes are of different sizes, adaptive average pooling is applied to map f_n^k to a feature vector of size $\hat{h}$. All the pooled feature vectors are then concatenated to construct the overall feature vector $\hat{f} \in \mathbb{R}^{K\times\hat{h}}$, where K is the total number of bboxes in the target dataset. The target feature dataset is then denoted by $\mathcal{F} : \{(\hat{f}_k, y_k, b_k)\}_{k=1}^K$. In classification task $\hat{f} = f$, $\hat{h} = h$, and $K = N$.

3.3. ETran

ETran is a hybrid transferability metric including energy, classification, and regression scores. The classification and regression scores are crucial since there is no single objective function that is optimal for both, especially in the case of object detection. The energy score is also important as the other two scores are unable to determine whether the target dataset is IND or OOD for the pre-trained model. The *ETran*'s overall transferability metric is defined using the following score:

$$T = S_{\text{en}} + S_{\text{cls}} + S_{\text{reg}}, \tag{2}$$

where S_{en}, S_{cls}, and S_{reg} are the energy, classification, and regression (only for object detection) scores, respectively.

Energy Score. Energy-based models (EBM) introduce a function $E(x) : \mathbb{R}^D \to \mathbb{R}$ that maps input data x to a single, non-probabilistic scalar called *energy* [28]. Energies are uncalibrated values and can be turned into probability density $p(x)$ through *Gibbs distribution*:

$$p(y|x) = \frac{e^{-E(x,y)}}{\int_{y'} e^{-E(x,y')}}, \tag{3}$$

where the denominator $\int_{y'} e^{-E(x,y')} = e^{-E(x)}$ is called *partition function*. The negative log of the partition function is Helmholtz free energy $E(x)$ of an input data x. The $p(y|x)$ obtained by EBM can also be calculated by a machine learning model $\Phi : \mathbb{R}^D \to \mathbb{R}^C$ by applying a softmax function as follows:

$$p(y|x) = \frac{e^{\Phi^{(y)}(x)}}{\sum_c^C e^{\Phi^{(c)}(x)}}, \tag{4}$$

where $\Phi^{(y)}(x)$ is the output logits of y-th class. Due to the deep connection between EBMs and discriminative models, we can define the energy for a given data point x as $E(x, y) = -\Phi^{(y)}(x)$ by equating the Eq. 4 and 3. We can then compute the free energy $E(x)$ (defined as the negative log of the partition function) as follows:

$$E(x) = -\log \sum_{c}^{C} e^{\Phi^{(c)}(x)}. \tag{5}$$

In the transferability estimation, we aim to assign a low likelihood to the features extracted by a low-ranked source model and a high likelihood to the features from the high-ranked source models. We can use energy-based models to define the density function of data: $p(x) = \frac{e^{-E(x)}}{\int_x e^{-E(x)}}$ [31]. We take the logarithm of both sides of the above equation,

$$\log p(x) = -E(x) - \underbrace{\log Z}_{\text{constant}}, \tag{6}$$

where Z is the denominator, which is constant for all samples. Therefore, the negative free energy is correlated with the likelihood of samples. This means that samples with high energy values have less likelihood and are OOD samples for the source model Φ. On the other hand, samples with high energy values are IND for Φ. We hypothesize that Φ has a high transferability to a target dataset $\mathcal{D}$, if samples from $\mathcal{D}$ are in-distribution samples for Φ. Therefore, the transferability score of Φ to $\mathcal{D}$ is correlated to the free energy score of Φ. In the above formulas, the free energy was calculated using output logits of Φ. This has a major drawback as the logits are task-specific outputs that depend on the number of classes in the source dataset. On the contrary, features extracted by Φ are task-independent outputs that can be assumed as the output of the discriminative model Φ with $\hat{h}$ classes. Given $\hat{h}$ as the dimension of features $\hat{f}$, we calculate the energy values over features:

$$\hat{E}(x) = -\log \sum_{\eta}^{\hat{h}} e^{\hat{f}^{(\eta)}}. \tag{7}$$

Having the energy values corresponding to all data samples $\{x_k\}_{k=1}^{K}$, we define the energy-based transferability score:

$$S_{\text{en}} = -\frac{1}{K} \sum_{k=1}^{K} \hat{E}(x_k). \tag{8}$$

LDA-Based Classification Score. Features extracted by the pre-trained models are separable based on the source dataset's classes. However, after fine-tuning, the features are separable based on the target dataset's classes. The classification score in this work is obtained using Bayes theorem after projecting the features into a subspace that separates features of different target classes as much as possible.

In this work, the projection matrix, denoted by U, is computed using Linear Discriminant Analysis (LDA) [19]:

$$U = \underset{\mathrm{U}}{\text{arg max}} \frac{U^T \Sigma_\beta U}{U^T \Sigma_\omega U}, \tag{9}$$

where $\Sigma_\beta = \sum_{c=1}^{C} K_c (\mu_c - \mu)(\mu_c - \mu)^T$ and $\Sigma_\omega = \sum_{c=1}^{C} \sum_{k=1}^{K_c} (\hat{f}_k^{(c)} - \mu)(\hat{f}_k^{(c)} - \mu)^T$ are the between-scatter and within-scatter matrices. μ_c and μ are the mean of c-th class and the total mean of the target data. C is the number of classes in the target dataset and K_c is the number of samples (i.e., bboxes in object detection) in c-th class. $\hat{f}_k^{(c)}$, which is obtained by splitting $\mathcal{F}$ into classes, represents the feature vector corresponding to the k-th sample (or bbox) with c-th class. The optimization in Eq. 9 is [15] :

$$\underset{\mathrm{U}}{\text{maximize}}\ U^T \Sigma_\beta U, \quad \text{subject to: } U^T \Sigma_\omega U = 1. \tag{10}$$

The Lagrangian of this optimization is defined as:

$$\mathcal{L} = U^T \Sigma_\beta U - \lambda (U^T \Sigma_\omega U - 1), \tag{11}$$

where λ is the Lagrangian multiplier. Equating the derivative of $\mathcal{L}$ to zeros gives:

$$\frac{\partial \mathcal{L}}{\partial U} = 2\Sigma_\beta U - 2\lambda \Sigma_\omega U = 0 \quad \Rightarrow \quad \Sigma_\beta U = \lambda \Sigma_\omega U. \tag{12}$$

The above eigenvalue problem can be solved [14]:

$$U = eig((\Sigma_\omega + \epsilon I)^{-1} \Sigma_\beta), \tag{13}$$

where ϵ is a small positive number to make the Σ_w full-rank in case that Σ_w is singular. Having the projection matrix, U, we project the features by $\bar{f} = U^T \hat{f}$. We assume that each class has a normal distribution $\bar{f}^{(c)} \sim \mathcal{N}(U^T \mu_c, I)$ where I is an identity matrix. Therefore, the Bayes theorem can be applied to obtain the prediction score δ_c for each class c. The *ETran*'s classification score is then defined as the probability of the ground-truth class as follows:

$$S_{cls} = \frac{1}{K} \sum_{k=1}^{K} \frac{e^{\delta_y}}{\sum_{c=1}^{C} e^{\delta_c}}, \tag{14}$$

where y is the ground-truth label.

SVD-Based Regression Score. Singular Value Decomposition (SVD) can be utilized for approximately solving linear regression in a way that is less sensitive to errors and more effective for ill-conditioned matrices [18]. This is due to the singular values in the diagonal matrix being sorted in descending order, so the smallest values can be truncated or set to zero without significantly affecting the overall solution. Therefore, we propose to use reduced SVD [34, 17] to efficiently estimate the transferability of features obtained by the source model. We decompose the feature matrix $\hat{f} = U\text{diag}(s)V^T$ where $U \in \mathbb{R}^{K\times\hat{h}}$, $s \in \mathbb{R}^{\hat{h}}$, and $V \in \mathbb{R}^{\hat{h}\times\hat{h}}$. We then use the decomposed matrices to calculate the approximated pseudo-inverse [16] of the features as follows:

$$\hat{f}^{\dagger} = V\text{diag}(\hat{s})^{-1}U^T, \tag{15}$$

where $\hat{s}$ is the truncated singular values whose top 80% is preserved. Given the bbox position labels $b \in \mathbb{R}^{K\times 4}$, we calculate the projection of b into the subspace spanned by columns of $\hat{f}^{\dagger}$ by $\hat{f}^{\dagger}b$. The approximated labels $\hat{b}$ are calculated by $\hat{b} = \hat{f}\hat{f}^{\dagger}b$.

The *ETran*'s regression score is then computed by the mean squared error between the approximated labels and ground-truth bbox labels as follows:

$$S_r = -\frac{1}{K\times 4}\sum_{k=1}^{K}\sum_{j=1}^{4}(b_k^{(j)} - \hat{b}_k^{(j)})^2, \tag{16}$$

where $b_k^{(j)}$ denotes the j-th position value in the k-th bbox.

3.4. Baselines for Object Detection

In this section, we define 3 new baseline transferability metrics for object detection based on the SOTA methods by LogME [52], PACTran [10], and SFDA [43], which have been originally proposed for classification tasks. These classification-based metrics can be directly applied to object detection by only evaluating the compatibility between bbox features and their class labels. However, such a strategy does not consider the bbox information (as a regression problem) which is a crucial part of object detection tasks, which is required to achieve good performance in transferability assessment. In order to benefit from the bbox information in all of the 3 baselines in this work, we employ LogME's regression solution [52] to calculate our baseline regression score (denoted by S_{lmr}) as:

$$\begin{aligned} S_{\text{lmr}} = \frac{1}{4}\sum_{j=1}^{4}\big(&\frac{1}{2}\log\gamma + \frac{\hat{h}}{2}\log\alpha - \frac{K}{2}\log 2\pi \\ &- \frac{\gamma}{2}||\hat{f}q - b^{(j)}|| - \frac{\alpha}{2}q^Tq - \frac{1}{2}\log||A||\big), \end{aligned} \tag{17}$$

where $A = \alpha I + \gamma\hat{f}^T\hat{f}$ and $q = \gamma A^{-1}\hat{f}^T b^{(j)}$. Here, α and γ are positive parameters in the prior distribution of weights and observations. The weights map the features to target labels. As in Eq. 15, $b^{(j)}$ represents the jth position values, but for all K bboxes.

The overall transferability metric for our baselines is then computed as follows: $T = S_{\text{baseline}} + S_{\text{lmr}}$, where S_{baseline} is the classification score calculated via LogME, PACTran, or SFDA methods.

4. Experiments

In this section, the performance of the proposed transferability assessment method (*ETran*) compared with the previous works is numerically evaluated on image classification and object detection. An extensive set of experiments over different benchmarks along with ablation studies and computational complexity analysis are also presented. In addition, three transferability assessment benchmarks based on VOC, COCO, and HuggingFace datasets are introduced for the object detection task.

Evaluation Metric. In order to evaluate the performance of the proposed method, the ground-truth ranking scores of all the pre-trained models (Φ_m) are required. The corresponding ranking scores, denoted by G_m, are basically the validation accuracies obtained after fine-tuning each Φ_m on the target dataset. Following the previous works [10, 38, 43, 52], we use Kendall's tau, denoted by τ, as our main evaluation metric. Kendall's tau [24] is defined as the number of concordant pairs minus the number of discordant pairs divided by the overall number of pairs $\binom{M}{2}$ as follows:

$$\tau = \frac{2}{M(M-1)}\sum_{i=1}^{M}\sum_{j=i+1}^{M}\text{sgn}(G_i - G_j)\cdot\text{sgn}(T_i - T_j). \tag{18}$$

We use the weighted version of Kendall's tau by [44], τ_w, that assigns more weights to the top ranked models. In addition to τ_w, we also use the probability of correctly finding the top-k pre-trained models, denoted by $Pr(\text{topk})$, as another evaluation metric. $Pr(\text{topk})$ is the probability of the ground-truth top-ranked model being among the top k estimated models. In this work, we report $Pr(\text{top1})$, $Pr(\text{top2})$, and $Pr(\text{top3})$. Although τ_w shows whether the whole ranking of the pre-trained models matches the ground-truth ranking, $Pr(\text{top-k})$ is also important in the real-case scenarios, where we only need to find the best pre-trained model.

4.1. Image Classification

Benchmark. The experiments for the classification task are performed on the benchmark used in [43] that has 11 different source models (pre-trained on ImageNet) and 11 target datasets. The source models include ResNet-34, ResNet-50, ResNet-101, ResNet-152 [20], DenseNet-121, DenseNet-169, DenseNet-201 [21], MNet-A1 [47], MobileNetV2 [42], GoogleNet [45], and InceptionV3 [46]. The target datasets include FGVC Aircraft [33], Caltech-101 [13], Stanford Cars [26], CIFAR-10, CIFAR-100 [27], DTD [8], Oxford-102 Flowers [37], Food-101 [7], Oxford-IIIT Pets [39], SUN397 [51], and VOC2007 [11]. We fine-tune all the source models on all of the target datasets to obtain the ground-truth scores, G (details in the appendix).

Results Analysis. The *ETran*'s transferability score for the classification scenario is defined as $T = S_{\text{cls}}+S_{\text{en}}$, where

Table 1: Classification Benchmark: Performance (Kendall tau τ_w) of different methods

Method	CIFAR10	VOC	Caltech-101	AirCraft	CIFAR100	Food-101	Pets	Flowers	Cars	DTD	Sun	**Average**
LEEP [36]	0.824	0.413	0.605	-0.233	0.667	0.434	0.389	-0.242	0.317	0.417	0.697	0.390
NLEEP [29]	-0.360	-0.233	0.281	0.332	0.696	0.468	0.230	-0.162	0.367	0.378	0.511	0.228
OTCE [48]	0.562	0.639	0.104	0.099	0.285	0.474	0.056	0.265	0.439	0.082	-0.139	0.260
LogME [52]	0.852	0.564	0.352	0.334	0.725	0.385	0.411	-0.008	0.485	0.662	0.545	0.482
PACTran [10]	0.562	-0.235	0.528	-0.038	0.763	0.000	0.318	0.329	-0.121	0.522	0.301	0.266
SFDA [43]	0.849	0.518	0.555	-0.215	0.793	0.427	0.340	0.590	0.312	0.633	0.722	0.502
LEEP+S_{en}	0.897	0.413	0.626	-0.077	0.697	0.434	0.389	-0.070	0.405	0.417	0.658	0.435
LogME+S_{en}	**0.890**	0.656	**0.567**	**0.370**	0.774	0.484	0.447	-0.021	**0.586**	**0.682**	0.570	0.545
PACTran+S_{en}	0.562	-0.235	0.528	0.046	0.702	0.024	0.437	0.329	-0.163	0.599	0.378	0.291
SFDA+S_{en}	**0.890**	0.606	0.558	-0.161	0.856	0.370	0.422	0.406	0.328	0.639	**0.744**	0.514
ETran (S_{en})	0.816	0.476	0.41	0.331	0.557	0.396	0.307	0.277	0.500	0.606	0.556	0.475
ETran (S_{cls}+S_{en})	0.887	**0.667**	0.440	-0.091	**0.900**	**0.829**	**0.713**	**0.580**	0.246	0.303	0.708	**0.562**

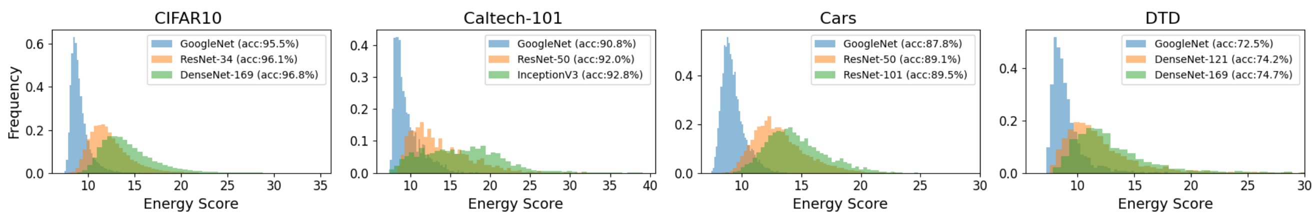

Figure 3: Energy score distributions corresponding to three source models on the CIFAR10, Caltech101, Cars, and DTD target datasets in the classification benchmark.

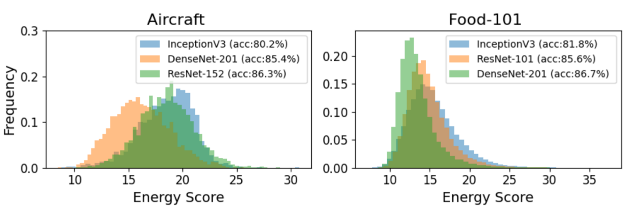

Figure 4: Some failure cases of ranking source models by energy scores.

the regression score does not exist. Table 1 demonstrates the results of *ETran* compared with the previous works on the classification benchmark [43]. *ETran* outperforms all the previous works and achieves SOTA results with an average τ_w of 0.562 which is relatively 12% better than SFDA.

In order to show the effectiveness of the proposed energy score (S_{en}), we also integrated this score with all the previous works and report the results in Table 1. It is shown that the energy score provides relative improvements for LEEP, LogME, PACTran, and SFDA by about 11%, 13%, 9%, and 2%, respectively, in terms of the average τ_w. Given the efficiency of the energy score calculation (i.e., almost 10$\times$ faster than the previous works), the corresponding improvements come with a low cost.

It is also shown that *ETran*'s performance with only the energy score can obtain an average τ_w of 0.475, which is comparable with most of the previous works and even better than LEEP and PACTran. Since the energy score is completely unsupervised (no need for labels), our proposed method can be applied in the case that labels are not provided with the target datasets. It is an important merit in real case scenarios, especially with costly labeling procedures.

Energy Analysis. Figure 3 shows the energy score distributions corresponding to three source models on the CIFAR10, Caltech101, Cars, and DTD target datasets in the classification benchmark. The ground-truth validation accuracy of the models is also provided in the legends of the figures. We can observe that the source models with a higher accuracy on the target dataset have a higher range of energy scores. For instance, on CIFAR10, Densnet169, ResNet-34, and GoogleNet are ranked as the first, second, and third models, respectively, which follow the same ranking in terms of having a higher range of energy scores. Although it is the case for the majority of the datasets, there are some rare cases, where models with higher accuracy give lower energy scores. Two examples are given in Figure 4.

4.2. Object Detection

4.2.1 Benchmarks and Setup

For the numerical analysis of our proposed transferability

Table 2: Results on VOC-FT and VOC-RH object detection benchmarks.

		VOC-FT				VOC-RH			
Method	reg	Pr(top1)	Pr(top2)	Pr(top3)	τ_w	Pr(top1)	Pr(top2)	Pr(top3)	τ_w
LogME [52]		0.071	0.107	0.250	0.180	0.107	0.250	0.393	0.340
PACTran [10]		0.143	0.214	0.321	0.131	0.143	0.286	0.357	0.242
Linear [10]		0.143	0.214	0.321	0.132	0.143	0.286	0.357	0.242
SFDA [43]		0.107	0.107	0.250	0.108	0.250	0.321	0.357	0.376
LogME+S_{lmr}	✓	0.071	0.179	0.357	0.350	0.321	0.536	0.571	0.537
PACtran+S_{lmr}	✓	0.071	0.179	0.321	0.355	0.393	0.500	0.571	0.560
Linear+S_{lmr}	✓	0.071	0.179	0.321	0.359	0.357	0.536	0.571	0.549
SFDA+S_{lmr}	✓	0.107	0.179	0.321	0.354	0.357	0.536	0.571	0.551
LogME+S_{reg}	✓	0.036	0.107	0.214	0.336	0.321	0.500	0.643	0.560
PACTran+S_{reg}	✓	0.071	0.107	0.321	0.335	0.214	0.429	0.571	0.508
Linear+S_{reg}	✓	0.071	0.143	0.393	0.352	0.250	0.429	0.607	0.555
SFDA+S_{reg}	✓	0.107	0.214	0.357	0.353	0.250	0.393	0.500	0.529
SFDA+S_{reg}+S_{en}	✓	0.214	0.321	0.536	0.462	0.143	0.393	0.500	0.528
ETran (S_{en})		**0.286**	**0.393**	0.464	0.309	0.000	0.250	0.429	0.318
ETran (S_{cls}+S_{en}+S_{reg})	✓	0.250	0.321	**0.536**	**0.464**	**0.500**	**0.536**	**0.679**	**0.590**

Table 3: Results on COCO object detection benchmark.

Method	reg	Pr(top1)	Pr(top2)	Pr(top3)	τ_w
LogME [52]		0.267	0.400	**0.600**	<u>0.269</u>
PACTran [10]		0.133	0.267	0.333	0.138
Linear [10]		0.133	0.267	0.333	0.139
SFDA [43]		0.200	0.267	0.533	0.104
LogME+S_{lmr}	✓	0.067	0.467	0.533	0.249
PACtran+S_{lmr}	✓	0.200	0.333	0.467	0.229
Linear+S_{lmr}	✓	0.200	0.333	0.467	0.227
SFDA+S_{lmr}	✓	0.200	0.333	0.467	0.183
ETran (S_{en})		<u>0.267</u>	<u>0.467</u>	**0.600**	0.213
ETran (S_{cls}+S_{en}+S_{reg})	✓	**0.400**	**0.533**	**0.600**	**0.333**

Table 4: Results on HF object detection benchmark.

Method	reg	Pr(top1)	Pr(top2)	Pr(top3)	τ_w
LogME [52]		**0.600**	0.600	0.800	0.374
PACTran [10]		0.400	0.400	0.600	0.140
Linear [10]		0.200	0.400	0.600	0.214
SFDA [43]		0.400	0.600	**0.100**	0.312
LogME+S_{lmr}	✓	**0.600**	0.600	0.800	0.400
PACtran+S_{lmr}	✓	0.200	0.400	0.800	0.322
Linear+S_{lmr}	✓	0.200	0.200	0.800	0.306
SFDA+S_{lmr}	✓	0.200	0.400	0.800	0.202
ETran (S_{en})	✓	<u>0.400</u>	**0.800**	0.800	0.412
ETran (S_{cls}+S_{en}+S_{reg})	✓	**0.600**	**0.800**	**1.000**	**0.522**

metric for object detection, we design 3 benchmarks based on VOC2012 [12], COCO [30], and HuggingFace (HF) [50] datasets.

VOC. We split the VOC2012 dataset into two clusters called the source and target clusters. VOC2012 has 20 classes from which we assign 12 classes to the source cluster and 8 classes to the target cluster. We randomly select 3 classes from the source cluster and repeat this selection 19 times to create 19 different source datasets. The YOLOv5s object detection model [40, 22] is trained on each of the source datasets resulting in 19 pre-trained source models. We also randomly select 28 pairs from the target cluster to create 28 target datasets (divided into train and validation sets). All pre-trained models are fine-tuned on the created target datasets (train set) and the *map50* value over the validation sets is used to obtain the ground-truth ranking scores of pre-trained models.

Following the method in LEEP [36], we use two approaches to fine-tune the source models: 1) fine-tuning the entire model, i.e., all layers (denoted by *FT*), 2) re-training only the detection head from the scratch with all the other layers frozen, denoted by *RH* (more details in the appendix).

COCO. We apply the same above-mentioned procedure in VOC to the COCO dataset (with 80 classes in total). We consider 65 and 15 classes for the source and target clusters, respectively. 9 source datasets each with 20 classes randomly selected from the source cluster are created. 15 target datasets each with 2 classes randomly selected from the target cluster are also created. A similar pre-training and fine-tuning process (only the *FT* case) as in VOC is performed for this benchmark.

HuggingFace (HF). In the VOC and COCO benchmarks, the architecture of the source models was fixed, but it was trained on different source datasets. In this benchmark, we fix the source dataset but use different model architectures. Six different models including YOLOv5s, YOLOv5m, YOLOv5n [22], YOLOv8s, YOLOv8m, and YOLOv8n [23] are employed all of which are pre-trained on COCO dataset. We fine-tune these source models (all the layers) on 5 object detection datasets presented in the HuggingFace platform: Blood [41], NFL [1], Valorant Video Game [32], CSGO Video Game [5], and Forklift [49]. Note that the object classes in these target datasets have not been defined in COCO dataset. Therefore, the pre-trained models have not seen the target classes beforehand.

Table 5: Ablation study of ETran scores on object detection benchmarks.

S_{cls}	S_{reg}	S_{en}	VOC-FT Pr(top3)	VOC-FT τ_w	VOC-RH Pr(top3)	VOC-RH τ_w	COCO Pr(top3)	COCO τ_w	HF Pr(top3)	HF τ_w
✓			0.29	0.14	0.43	0.37	0.53	0.13	0.80	0.38
	✓		0.39	0.36	0.50	0.54	0.40	0.12	0.80	0.51
		✓	0.46	0.31	0.43	0.32	0.60	0.21	0.80	0.41
✓	✓		0.25	0.31	0.64	0.57	0.47	0.25	1.00	0.50
	✓	✓	**0.50**	0.40	0.57	0.55	0.53	0.23	0.80	0.45
✓		✓	**0.50**	0.37	0.50	0.41	0.53	0.32	0.80	0.40
✓	✓	✓	**0.50**	**0.44**	**0.68**	**0.59**	**0.60**	**0.33**	**1.00**	**0.52**

4.2.2 Results Analysis

VOC. The experimental results of *ETran* on the VOC-FT and VOC-RH object detection benchmarks are summarized in Table 2. In both scenarios, the source models are the same, however, the ground-truth rankings are different. It is worth mentioning that the VOC-FT case provides a more challenging transferability estimation because the features extracted by the source and target models are quite different.

The baseline results with both LogME (S_{lmr}) and our SVD-based regression (S_{reg}) scores are given in Table 2, which outperforms classification-only metrics in the previous works. For example, *SFDA+S_{lmr}* with a τ_w of 0.354 achieves a relative improvement of 70% on VOC-FT compared to *SFDA* with a τ_w of 0.108. As summarized in Table 2, *ETran* outperforms all the previous works and baselines in terms of all the evaluation metrics. The results of *ETran* only with the energy score (without labels) are also presented, which provide comparable or better scores than the previous classification-only scores.

COCO. The comparison results for the COCO-based benchmark are provided in Table 3. Although adding the LogME's regression score to the previous works improves their results, *ETran* still achieves the best performance.

HuggingFace. The comparison results on HF are summarized in Table 4, which are very insightful for evaluating the performance of the transferability metrics when the source models have different architectures. Similar to the VOC benchmark, the regression-empowered baselines outperform the previous classification-only works by an average of 0.047 in τ_w (average τ_w of 0.307 vs. 0.260). Overall, the proposed *ETran* method achieves the best results in terms of all the evaluation metrics. As presented in Table 4, we obtain larger numbers in terms of $Pr(\text{top-k})$ on the HF benchmark compared with the VOC and COCO benchmarks. We argue that when the source models have different architectures, the transferability estimation is less challenging because the features extracted by the source models are more distinct. This is well-aligned with the results of the classification benchmark designed in the previous works in which the source models have different architectures [43].

Table 6: **Left**: Ablation study on ETran scores for classification benchmark. **Right**: Ablation study on the logits- vs. features-based energy scores.

S_{cls}	S_{en}	τ_w
✓		0.470
	✓	0.475
✓	✓	**0.562**

Method	τ_w CB	τ_w VOC	τ_w HF
Logits	-0.088	0.270	0.297
Features	**0.475**	**0.309**	**0.412**

Table 7: Time complexity analysis.

Object Detection (VOC-FT) Method	Run-Time (s)	Classification Method	Run-Time(s)
LogME [52]	**11**	LogME [52]	**65**
PACTran [10]	53	PACTran [10]	444
SFDA [43]	28	SFDA [43]	236
LogME+S_{lmr}	22	LogME+S_{en}	70
PACtran+S_{lmr}	63	PACTran+S_{en}	449
SFDA+S_{lmr}	39	SFDA+S_{en}	240
ETran (S_{en})	0.3	**ETran (S_{en})**	5
ETran (S_{cls}+S_{en}+S_{reg})	<u>25</u>	**ETran (S_{cls}+S_{en})**	<u>101</u>

4.3. Ablation and Complexity Study

Components of ETran. The individual performance of the proposed *ETran*'s classification, energy, and regression scores on all the object detection and classification benchmarks are summarized in Tables 5 and 6. As presented in Table 5, excluding any of the scores from *ETran* can result in a performance drop for all the benchmarks. It is also shown that the regression score has a major contribution to the object detection task. This is an important finding that emphasizes the limitation of the previous classification-only transferability metrics for object detection tasks. On the other hand, Table 6-Left shows the importance of both classification and energy scores in the classification scenarios.

Energy Score. In Table 6-Right, the performance of the proposed energy-based transferability score calculated over logits vs. features is given. As shown by the results, the feature-based energy score calculation significantly performs better than the logit-based scenario. As also discussed in Section 3.3, it is mainly because the features extracted by the source models acquire more general information about the target dataset. On the other hand, logits are the outputs of the network's head that are specific to the labels of the source datasets used for training the source models. Specifically, in the case that the number of labels in the source and target datasets is significantly different, using logits for the calculation of energy score can be misleading.

Time Complexity. The computational complexity of *ETran* compared with the previous works and the baselines over the VOC-FT and classification benchmarks are given in Table 7. The numbers in the table are the running time of the whole transferability assessment averaged over the number of target datasets (i.e., 28 for VOC-FT and 11 for the classification benchmark). Among the previous works, LogME is the fastest metric. Our *ETran* is the second model after LogME and it is faster than PACTran and SFDA. *ETran* (S_{en}) is around 10× faster than LogME, which shows the efficiency of the energy score calculation in our method. The running time of PACTran depends on the number of iterations used for optimization (i.e., 100 iterations by default). SFDA has a self-challenging mechanism that requires applying Fisher Discriminate Analysis twice on all the samples. In contrast, *ETran* only needs one round for LDA-based score calculation, which makes it 2× faster than SFDA.

5. Conclusion

In this work, we proposed an energy-based transferability metric for classification and object detection. We introduced energy score as a fast and unlabeled transferability score that is used together with labeled classification and regression scores, which outperformed previous transferability metrics on object detection and classification scenarios. In terms of running time, *ETran* is comparable with the previous works while obtaining better performance. In this work, we only showed the performance of *ETran* for vision tasks. Future work should evaluate the performance of this method for other modalities such as language models.

References

[1] Nfl-competition dataset. url https://universe.roboflow.com/home-mxzv1/nfl-competition , sep 2022. visited on 2023-01-18. 7

[2] Mohammad Akbari, Amin Banitalebi-Dehkordi, and Yong Zhang. EBJR: Energy-based joint reasoning for adaptive inference. *BMVC 2021*, 2021. 2

[3] Mohammad Akbari, Amin Banitalebi-Dehkordi, and Yong Zhang. E-lang: Energy-based joint inferencing of super and swift language models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5229–5244, 2022. 2

[4] Mohammad Akbari, Amin BANITALEBI DEHKORDI, Tianxi Xu, and Yong Zhang. System and method for unsupervised multi-model joint reasoning, 2023. US Patent App. 18/074,915. 2

[5] ASD. Wlots dataset. url https://universe.roboflow.com/asd-culfr/wlots , may 2022. visited on 2023-01-27. 7

[6] 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 *2019 IEEE international conference on image processing (ICIP)*, pages 2309–2313. IEEE, 2019. 2

[7] Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101–mining discriminative components with random forests. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part VI 13*, pages 446–461. Springer, 2014. 5

[8] Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing textures in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3606–3613, 2014. 5

[9] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. 1

[10] Nan Ding, Xi Chen, Tomer Levinboim, Soravit Changpinyo, and Radu Soricut. Pactran: Pac-bayesian metrics for estimating the transferability of pretrained models to classification tasks. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXIV*, pages 252–268. Springer, 2022. 1, 2, 5, 6, 7, 8

[11] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88:303–308, 2009. 5

[12] 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. 7

[13] Li Fei-Fei, Rob Fergus, and Pietro Perona. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. In *2004 conference on computer vision and pattern recognition workshop*, pages 178–178. IEEE, 2004. 5

[14] Benyamin Ghojogh, Fakhri Karray, and Mark Crowley. Eigenvalue and generalized eigenvalue problems: Tutorial. *arXiv preprint arXiv:1903.11240*, 2019. 4

[15] Benyamin Ghojogh, Fakhri Karray, and Mark Crowley. Fisher and kernel fisher discriminant analysis: Tutorial. *arXiv preprint arXiv:1906.09436*, 2019. 4

[16] Gene Golub and William Kahan. Calculating the singular values and pseudo-inverse of a matrix. *Journal of the Society for Industrial and Applied Mathematics, Series B: Numerical Analysis*, 2(2):205–224, 1965. 5

[17] Gene H Golub and Christian Reinsch. Singular value decomposition and least squares solutions. *Linear algebra*, 2:134–151, 1971. 2, 4

[18] Per Christian Hansen. *Rank-deficient and discrete ill-posed problems: numerical aspects of linear inversion*. SIAM, 1998. 4

[19] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. The elements of statistical learnin. *Cited on*, 33, 2009. 2, 4

[20] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 5

[21] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017. 5

[22] Glenn Jocher. YOLOv5 by Ultralytics, 5 2020. 7

[23] Glenn Jocher, Ayush Chaurasia, and Jing Qiu. YOLO by Ultralytics, 1 2023. 7

[24] Maurice G Kendall. A new measure of rank correlation. *Biometrika*, 30(1/2):81–93, 1938. 5

[25] Simon Kornblith, Jonathon Shlens, and Quoc V Le. Do better imagenet models transfer better? In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2661–2671, 2019. 1

[26] Jonathan Krause, Jia Deng, Michael Stark, and Li Fei-Fei. Collecting a large-scale dataset of fine-grained cars. 2013. 5

[27] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 5

[28] Yann LeCun, Sumit Chopra, Raia Hadsell, M Ranzato, and F Huang. A tutorial on energy-based learning. *Predicting structured data*, 1(0), 2006. 2, 3

[29] Yandong Li, Xuhui Jia, Ruoxin Sang, Yukun Zhu, Bradley Green, Liqiang Wang, and Boqing Gong. Ranking neural checkpoints. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2663–2673, 2021. 1, 2, 6

[30] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014. 7

[31] Weitang Liu, Xiaoyun Wang, John Owens, and Yixuan Li. Energy-based out-of-distribution detection. *Advances in neural information processing systems*, 33:21464–21475, 2020. 2, 4

[32] Daniels Magonis. valorant dataset. url https://universe.roboflow.com/daniels-magonis-0pjzx/valorant-9ufcp , nov 2022. visited on 2023-01-27. 7

[33] Subhransu Maji, Esa Rahtu, Juho Kannala, Matthew Blaschko, and Andrea Vedaldi. Fine-grained visual classification of aircraft. *arXiv preprint arXiv:1306.5151*, 2013. 5

[34] John Mandel. Use of the singular value decomposition in regression analysis. *The American Statistician*, 36(1):15–24, 1982. 4

[35] et al. Martín Abadi. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org. 1

[36] Cuong Nguyen, Tal Hassner, Matthias Seeger, and Cedric Archambeau. Leep: A new measure to evaluate transferability of learned representations. In *International Conference on Machine Learning*, pages 7294–7305. PMLR, 2020. 2, 6, 7

[37] Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *2008 Sixth Indian Conference on Computer Vision, Graphics & Image Processing*, pages 722–729. IEEE, 2008. 5

[38] Michal Pándy, Andrea Agostinelli, Jasper Uijlings, Vittorio Ferrari, and Thomas Mensink. Transferability estimation using bhattacharyya class separability. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9172–9182, 2022. 2, 5

[39] Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, and CV Jawahar. Cats and dogs. In *2012 IEEE conference on computer vision and pattern recognition*, pages 3498–3505. IEEE, 2012. 5

[40] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016. 7

[41] Team Roboflow. Blood cell detection dataset. url https://universe.roboflow.com/team-roboflow/blood-cell-detection-1ekwu , nov 2022. visited on 2023-01-18. 7

[42] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4510–4520, 2018. 5

[43] Wenqi Shao, Xun Zhao, Yixiao Ge, Zhaoyang Zhang, Lei Yang, Xiaogang Wang, Ying Shan, and Ping Luo. Not all models are equal: Predicting model transferability in a self-challenging fisher space. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXIV*, pages 286–302. Springer, 2022. 1, 2, 5, 6, 7, 8

[44] Grace S Shieh. A weighted kendall's tau statistic. *Statistics & probability letters*, 39(1):17–24, 1998. 5

[45] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015. 5

[46] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016. 5

[47] Mingxing Tan, Bo Chen, Ruoming Pang, Vijay Vasudevan, Mark Sandler, Andrew Howard, and Quoc V Le. Mnasnet: Platform-aware neural architecture search for mobile. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2820–2828, 2019. 5

[48] Yang Tan, Yang Li, and Shao-Lun Huang. Otce: A transferability metric for cross-domain cross-task representations. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 15779–15788, 2021. 6

[49] Mohamed Traore. Forklift dataset. url https://universe.roboflow.com/mohamed-traore-2ekkp/forklift-dsitv , mar 2022. visited on 2023-01-15. 7

[50] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*, 2019. 1, 7

[51] Jianxiong Xiao, James Hays, Krista A Ehinger, Aude Oliva, and Antonio Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *2010 IEEE computer society conference on computer vision and pattern recognition*, pages 3485–3492. IEEE, 2010. 5

[52] Kaichao You, Yong Liu, Jianmin Wang, and Mingsheng Long. Logme: Practical assessment of pre-trained models for transfer learning. In *International Conference on Machine Learning*, pages 12133–12143. PMLR, 2021. 1, 2, 5, 6, 7, 8