Transferability and Hardness of Supervised Classification Tasks

Anh T. Tran*
VinAI Research
anstar1111@gmail.com

Cuong V. Nguyen
Amazon Web Services
nguycuo@amazon.com

Tal Hassner*
Facebook AI
talhassner@gmail.com

Abstract

We propose a novel approach for estimating the difficulty and transferability of supervised classification tasks. Unlike previous work, our approach is solution agnostic and does not require or assume trained models. Instead, we estimate these values using an information theoretic approach: treating training labels as random variables and exploring their statistics. When transferring from a source to a target task, we consider the conditional entropy between two such variables (i.e., label assignments of the two tasks). We show analytically and empirically that this value is related to the loss of the transferred model. We further show how to use this value to estimate task hardness. We test our claims extensively on three large scale data sets—CelebA (40 tasks), Animals with Attributes 2 (85 tasks), and Caltech-UCSD Birds 200 (312 tasks)—together representing 437 classification tasks. We provide results showing that our hardness and transferability estimates are strongly correlated with empirical hardness and transferability. As a case study, we transfer a learned face recognition model to CelebA attribute classification tasks, showing state of the art accuracy for tasks estimated to be highly transferable.

1. Introduction

How easy is it to transfer a representation learned for one task to another? How can we tell which of several tasks is hardest to solve? Answers to these questions are vital in planning model transfer and reuse, and can help reveal fundamental properties of tasks and their relationships in the process of developing universal perception engines [3]. The importance of these questions is therefore driving research efforts, with several answers proposed in recent years.

Some of the answers to these questions established task relationship indices, as in the Taskonomy [69] and Task2Vec [1, 2] projects. Others analyzed task relationships in the context of multi-task learning [30, 36, 59, 66, 71]. Importantly, however, these and other efforts are computa-

tional in nature, and so build on specific machine learning solutions as *proxy task representations*.

By relying on such proxy task representations, these approaches are naturally limited in their application: Rather than insights on the tasks themselves, they may reflect relationships between the specific solutions chosen to represent them, as noted by previous work [69]. Some, moreover, establish task relationships by maintaining model zoos, with existing trained models already available. They may therefore also be computationally expensive [1, 69]. Finally, in some scenarios, establishing task relationships requires multi-task learning of the models, to measure the influence different tasks have on each other [30, 36, 59, 66, 71].

We propose a radically different, *solution agnostic* approach: We seek underlying relationships, irrespective of the particular models trained to solve these tasks or whether these models even exist. We begin by noting that supervised learning problems are defined not by the models trained to solve them, but rather by the *data sets of labeled examples and a choice of loss functions*. We therefore go to the source and explore tasks directly, by examining their data sets rather than the models they were used to train.

To this end, we consider supervised classification tasks defined over the same input domain. As a loss, we assume the cross entropy function, thereby including most commonly used loss functions. We offer the following surprising result: By assuming an optimal loss on two tasks, the *conditional entropy* (CE) between the label sequences of their training sets provides a bound on the *transferability* of the two tasks—that is, the log-likelihood on a target task for a trained representation transferred from a source task. We then use this result to obtain a-priori estimates of task transferability and hardness.

Importantly, we obtain effective transferability and hardness estimates by evaluating *only training labels*; we do not consider the solutions trained for each task or the input domain. This result is surprising considering that it greatly simplifies estimating task hardness and task relationships, yet, as far as we know, was overlooked by previous work.

We verify our claims with rigorous tests on a total of 437 tasks from the CelebA [34], Animals with Attributes 2

---

*Work at Amazon Web Services, prior to joining current affiliation.

(AwA2) [65], and Caltech-UCSD Birds 200 (CUB) [64] sets. We show that our approach reliably predicts task transferability and hardness. As a case study, we evaluate transferability from face recognition to facial attribute classification. On attributes estimated to be highly transferable from recognition, our results outperform the state of the art despite using a simple approach, involving training a linear support vector machine per attribute.

2. Related work

Our work is related to many fields in machine learning and computer vision, including transfer learning [67], meta learning [54], domain shifting [52], and multi-task learning [28]. Below we provide only a cursory overview of several methods directly related to us. For more principled surveys on transfer learning, we refer to others [4, 47, 63, 68].

Transfer learning. This paper is related to transfer learning [47, 63, 67] and our work can be used to select good source tasks and data sets when transferring learned models. Previous theoretical analysis of transfer learning is extensive [2, 5, 6, 7, 8, 38]. These papers allowed generalization bounds to be proven but they are abstract and hard to compute in practice. Our transferability measure, on the other hand, is easily computed from the training sets and can potentially be useful also for continual learning [43, 44, 50].

Task spaces. Tasks in machine learning are often represented as labeled data sets and a loss function. For some applications, qualitative exploration of the training data can reveal relationships between two tasks and, in particular, the biases between them [57].

Efforts to obtain more complex task relationships involved trained models. Data sets were compared using fixed dimensional lists of statistics, produced using an autoencoder trained for this purpose [18]. The successful Taskonomy project [69], like us, assumes multiple task labels for the same input images (same input domain). They train one model per-task and then evaluate transfers between tasks thereby creating a task hypergraph—their taxonomy.

Finally, Task2Vec constructs vector representations for tasks, obtained by mapping partially trained probe networks down to low dimensional task embeddings [1, 2]. Unlike these methods, we consider only the labels provided in the training data for each task, without using trained models.

Multi-task learning. Training a single model to solve multiple tasks can be mutually beneficial to the individual tasks [23, 49, 62]. When two tasks are only weakly related, however, attempting to train a model for them both can produce a model which under-performs compared to models trained for each task separately. Early multi-branch networks and their variants encoded human knowledge on the relationships of tasks in their design, joining related tasks

or separating unrelated tasks [27, 48, 51].

Others adjusted for related vs. unrelated tasks during training of a deep multi-task network. Deep cross residual learning does this by introducing cross-residuals for regularization [27], cross-stitch combines activations from multiple task-specific networks [42], and UberNet proposed a task-specific branching scheme [29].

Some sought to discover what and how should be shared across tasks during training by automatic discovery of network designs that would group similar tasks together [36] or by solving tensor factorization problems [66]. Alternatively, parts of the input rather than the network were masked according to the task at hand [59]. Finally, modulation modules were proposed to seek destructive interferences between unrelated tasks [71].

3. Transferability via conditional entropy

We seek information on the transferability and hardness of supervised classification tasks. Previous work obtained this information by examining machine learning models developed for these tasks [1, 2, 71]. Such models are produced by training on labeled data sets that represent the tasks. These models can therefore be considered views on their training data. In this work we instead use information theory to produce estimates *from the source*: the data itself.

Like others [69], we assume our tasks share the same input instances and are different only in the labels they assign to each input. Such settings describe many practical scenarios. A set of face images, for instance, can have multiple labels for each image, representing tasks such as recognition [39, 40] and classification of various attributes [34].

We estimate transferability using the CE between the label sequences of the target and source tasks. Task hardness is similarly estimated: by computing transferability from a trivial task. We next formalize our assumptions and claims.

3.1. Task transferability

We assume a single *input sequence* of training samples, $X = (x_1, x_2, \ldots, x_n) \in \mathcal{X}^n$, along with two *label sequences* $Y = (y_1, y_2, \ldots, y_n) \in \mathcal{Y}^n$ and $Z = (z_1, z_2, \ldots, z_n) \in \mathcal{Z}^n$, where y_i and z_i are labels assigned to x_i under two separate tasks: source task $T^Z = (X, Z)$ and target task $T^Y = (X, Y)$. Here, $\mathcal{X}$ is the domain of the values of X, while $\mathcal{Y} = \text{range}(Y)$ and $\mathcal{Z} = \text{range}(Z)$ are the sets of different values in Y and Z respectively. Thus, if Z contains binary labels, then $\mathcal{Z} = \{0, 1\}$.

We consider a classification model $M = (w, h)$ on the source task, T^Z. The first part, $w : \mathcal{X} \to \mathbb{R}^D$, is some transformation function, possibly learned, that outputs a D-dimensional representation $r = w(x) \in \mathbb{R}^D$ for an input $x \in \mathcal{X}$. The second part, $h : \mathbb{R}^D \to \mathcal{P}(\mathcal{Z})$, is a classifier that takes a representation r and produces a probability

distribution $h(r) \in \mathcal{P}(\mathcal{Z})$, where $\mathcal{P}(\mathcal{Z})$ is the space of all probability distributions over $\mathcal{Z}$.

This description emphasizes the two canonical stages of a machine learning system [16]: representation followed by classification. As an example, a deep neural network with Softmax output is represented by a learned w which maps the input into some feature space, producing a *deep embedding*, r, followed by classification layers (one or more), h, which maps the embedding into the prediction probability.

Now, assume we train a model (w_Z, h_Z) to solve T^Z by minimizing the cross entropy loss on Z:

$$w_Z, h_Z = \underset{w,h \in (W,H)}{\operatorname{argmin}} \mathcal{L}_Z(w, h), \qquad (1)$$

where W and H are our chosen spaces of possible values for w and h, and $\mathcal{L}_Z(w, h)$ is the cross entropy loss (equivalently, the negative log-likelihood) of the parameters (w, h):

$$\mathcal{L}_Z(w, h) = -l_Z(w, h) = -\frac{1}{n} \sum_{i=1}^{n} \log P(z_i | x_i; w, h), \qquad (2)$$

where $l_Z(w, h)$ is the log-likelihood of (w, h).

To transfer this model to target task T^Y, we fix the function w_Z and retrain only the classifier on the labels of T^Y. Denote the new classifier k_Y, selected from our chosen space K of target classifiers. Note that k_Y does not necessarily share the same architecture as h_Z. We train k_Y by minimizing the cross entropy loss on Y with the fixed w_Z:

$$k_Y = \underset{k \in K}{\operatorname{argmin}} \mathcal{L}_Y(w_Z, k), \qquad (3)$$

where $\mathcal{L}_Y(w_Z, k)$ is defined similarly to Eq. (2) but for the label set Y. Under this setup, we define the transferability of task T^Z to task T^Y as follows.

Definition 1 *The transferability of task T^Z to task T^Y is measured by the expected accuracy of the model (w_Z, k_Y) on a random test example (x, y) of task T^Y:*

$$\operatorname{Trf}(T^Z \to T^Y) = \mathbb{E}\left[\operatorname{acc}(y, x; w_Z, k_Y)\right], \qquad (4)$$

which indicates how well a representation w_Z trained on task T^Z performs on task T^Y.

In practice, if the trained model does not overfit, the log-likelihood on the *training set*, $l_Y(w_Z, k_Y)$, provides a good indicator of Eq (4), that is, how well the representation w_Z and the classifier k_Y performs on task T^Y. This non-overfitting assumption holds even for large networks that are properly trained and tested on datasets sampled from the *same distribution* [70]. Thus, in the subsequent sections, we instead consider the following log-likelihood as an alternative measure of transferability:

$$\widetilde{\operatorname{Trf}}(T^Z \to T^Y) = l_Y(w_Z, k_Y). \qquad (5)$$

3.2. The conditional entropy of label sequences

From the label sequences Y and Z, we can compute the empirical joint distribution $\hat{P}(y, z)$ for all $(y, z) \in \mathcal{Y} \times \mathcal{Z}$ by counting, as follows:

$$\hat{P}(y, z) = \frac{1}{n} |\{i : y_i = y \text{ and } z_i = z\}|. \qquad (6)$$

We now adopt the definition of CE between two random variables [13] to define the CE between our label sequences Y and Z.

Definition 2 *The CE of a label sequence Y given a label sequence Z, $H(Y|Z)$, is the CE of a random variable (or random label) $\bar{y}$ given a random variable (or random label) $\bar{z}$, where $(\bar{y}, \bar{z})$ are drawn from the empirical joint distribution $\hat{P}(y, z)$ of Eq. (6):*

$$H(Y|Z) = -\sum_{y \in \mathcal{Y}} \sum_{z \in \mathcal{Z}} \hat{P}(y, z) \log \frac{\hat{P}(y, z)}{\hat{P}(z)}, \qquad (7)$$

where $\hat{P}(z)$ is the empirical marginal distribution on $\mathcal{Z}$:

$$\hat{P}(z) = \sum_{y \in \mathcal{Y}} \hat{P}(y, z) = \frac{1}{n} |\{i : z_i = z\}|. \qquad (8)$$

CE represents a measure of the amount of information provided by the value of one random variable on the value of another. By treating the labels assigned to both tasks as random variables and measuring the CE between them, we are measuring the information required to estimate a label in one task given a (known) label in another task.

We now prove a relationship between the CE of Eq. (7) and the tranferability of Eq. (5). In particular, we show that the log-likelihood on the target task T^Y is lower bounded by log-likelihood on the source task T^Z minus $H(Y|Z)$, if the optimal input representation w_Z trained on T^Z is transferred to T^Y.

To prove our theorem, we assume the space K of target classifiers contains a classifier $\bar{k}$ whose log-likelihood lower bounds that of k_Y. We construct $\bar{k}$ as follows. For each input x, we compute the Softmax output $p_Z = h_Z(w_Z(x))$, which is a probability distribution on $\mathcal{Z}$. We then convert p_Z into a Softmax on $\mathcal{Y}$ by taking the expectation of the empirical conditional probability $\hat{P}(y|z) = \hat{P}(y, z)/\hat{P}(z)$ with respect to p_Z. That is, for all $y \in \mathcal{Y}$, we define:

$$p_Y(y) = \mathbb{E}_{z \sim p_Z}[\hat{P}(y|z)] = \sum_{z \in \mathcal{Z}} \hat{P}(y|z) \, p_Z(z), \qquad (9)$$

where $p_Z(z)$ is the probability of the label z returned by p_Z. For any input $w_Z(x)$, we let the output of $\bar{k}$ be p_Y. That is, $\bar{k}(w_Z(x)) = p_Y$. We can now prove the following theorem.
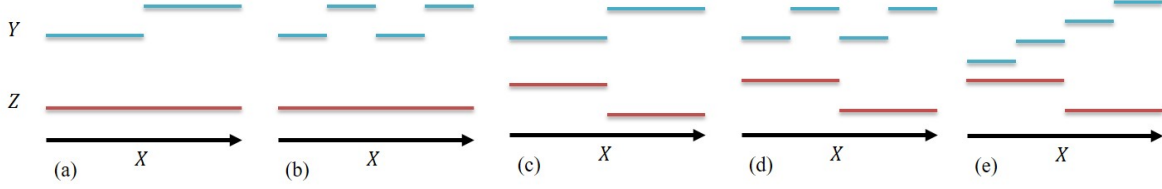
Figure 1. **Visualizing toy examples.** The transferability between two tasks, represented as sequences (X, Y) and (X, Z). The horizontal axis represent instances and the values for Z (in red) and Y (cyan). In which of these examples would it be easiest to transfer a model trained for task T^Z to task T^Y? See discussion and details in Sec. 3.3.

Theorem 1 *Under the training procedure described in Sec. 3.1, we have:*

$$\widetilde{\mathrm{Trf}}(T^Z \to T^Y) \geq l_Z(w_Z, h_Z) - H(Y|Z). \quad (10)$$

Proof sketch.[1] From the definition of k_Y and the assumption that $\bar{k} \in K$, we have $\widetilde{\mathrm{Trf}}(T^Z \to T^Y) = l_Y(w_Z, k_Y) \geq l_Y(w_Z, \bar{k})$. From the construction of $\bar{k}$, we have:

$$l_Y(w_Z, \bar{k}) = \frac{1}{n} \sum_{i=1}^{n} \log \left(\sum_{z \in \mathcal{Z}} \hat{P}(y_i|z) P(z|x_i; w_Z, h_Z) \right)$$

$$\geq \frac{1}{n} \sum_{i=1}^{n} \log \left(\hat{P}(y_i|z_i) P(z_i|x_i; w_Z, h_Z) \right) \quad (11)$$

$$= \frac{1}{n} \sum_{i=1}^{n} \log \hat{P}(y_i|z_i) + \frac{1}{n} \sum_{i=1}^{n} \log P(z_i|x_i; w_Z, h_Z). \quad (12)$$

We can easily show that the first term in Eq. (12) equals $-H(Y|Z)$, while the second term is $l_Z(w_Z, h_Z)$.

Discussion 1: Generality of our settings. Our settings for spaces W, H, K are general and include a variety of practical use cases. For example, neural networks W will include all possible (vector) values of the network weights until the penultimate layer, while H and K would include all possible (vector) values of the last layer's weights. Alternatively, we can use support vector machines (SVM) for K. In this case, K would include all possible values of the SVM parameters [55]. Our result even holds when the features are fixed, as when using tailored representations such as SIFT [35]. In these cases, space W would contain only one transformation function from raw input to the features.

Discussion 2: Assumptions. We can easily satisfy the assumption $\bar{k} \in K$ by first choosing a space K' (e.g., the SVMs) which will play the role of $K \setminus \{\bar{k}\}$. We solve the optimization problem of Eq. (3) on K' instead of K to obtain the optimal classifier k'. To get the optimal classifier k_Y on $K = K' \cup \{\bar{k}\}$, we simply compare the losses of k' and $\bar{k}$ and select the best one as k_Y.

The optimization problems of Eq. (1) and (3) are global optimization problems. In practice, for complex deep networks trained with stochastic gradient descent, we often

---

[1]Full derivations provided in the supplemental.

only obtain the local optima of the loss. In this case, we can easily change and prove Theorem 1 which would include the differences in the losses between the local optimum and the global optimum in the right-hand-side of Eq. (10). In many practical applications, the difference between local optimum and global optimum is not significant [12, 45].

Discussion 3: Extending our result to test log-likelihood. In Theorem 1, we consider the empirical log-likelihood, which is generally unbounded. If we make the (strong) assumption of bounded differences between empirical log-likelihoods, we can apply McDiarmids inequality [41] to get an upper-bound on the left hand side of Eq. (10) by the expected log-likelihood with some probability.

Discussion 4: Implications. Theorem 1 shows that the transferability from task T^Z to task T^Y depends on both the CE $H(Y|Z)$ and the log-likelihood $l_Z(w_Z, h_Z)$. Note that the log-likelihood $l_Z(w_Z, h_Z)$ is optimal for task T^Z and so it represents the hardness (or easiness) of task T^Z. Thus, from the theorem, if $l_Z(w_Z, h_Z)$ is small (i.e., the source task is hard), transferability would reduce. Besides, if the CE $H(Y|Z)$ is small, transferability would increase.

Finally, we note that when the source task T^Z is fixed, the log-likelihood $l_Z(w_Z, h_Z)$ is a constant. In this case, the transferability only depends on the CE $H(Y|Z)$. Thus, we can estimate the transferability from one source task to multiple target tasks by considering only the CE.

3.3. Intuition and toy examples

To gain intuition on CE and transferability, consider the toy examples illustrated in Fig. 1. The (joint) input set is represented by the X axis. Each input $x_i \in X$ is assigned with two labels, $y_i \in Y$ and $z_i \in Z$, for the two tasks. In Fig. 1(a,b), task T^Z is the trivial task with a constant label value (red line) and in Fig. 1(c–e) T^Z is a binary classification task, whereas T^Y is binary in Fig. 1(a–d) and multi-label in Fig. 1(e). In which of these examples would transferring a representation trained for T^Z to T^Y be hardest?

Of the five examples, (c) is the easiest transfer as it provides a 1-1 mapping from Z to Y. Appropriately, in this case, $H(Y|Z) = 0$. Next up are (d) and (e) with $H(Y|Z) = \log 2$: In both cases each class in T^Z is mapped to two classes in T^Y. Note that T^Y being non-binary is naturally handled by the CE. Finally, transfers (a) and (b) have

$H(Y|Z) = 4 \log 2$; the highest CE. Because T^Z is trivial, the transfer must account for the greatest difference in the information between the tasks and so the transfer is hardest.

4. Task hardness

A potential application of transferability is task hardness. In Sec. 3.2, we mentioned that the hardness of a task can be measured from the optimal log-likelihood on that task. Formally, we can measure the hardness of a task T^Z by:

$$\text{Hard}(T^Z) = \min_{w,h \in (W,H)} \mathcal{L}_Z(w,h) = -l_Z(w_Z, h_Z). \quad (13)$$

This definition of hardness depends on our choice of (W, H), which may determine various factors such as representation size or network architecture. The intuition behind the definition is that if the task T^Z is hard for all models in (W, H), we should expect higher loss even after training.

Using Theorem 1, we can bound $l_Z(w_Z, h_Z)$ in Eq. (13) by transferring from a trivial task T^C to T^Z. We define a *trivial task* as the task for which all input values are assigned the same, constant label. Let C be the (constant) label sequences of the trivial task T^C. From Theorem 1 and Eq. (13), we can easily show that:

$$\text{Hard}(T^Z) = -l_Z(w_Z, h_Z) \leq H(Z|C). \quad (14)$$

Thus, we can approximate the hardness of task T^Z by looking at the CE $H(Z|C)$. We note that the CE $H(Z|C)$ is also used to estimate the transferability $\widetilde{\text{Trf}}(T^C \rightarrow T^Z)$. So, $\text{Hard}(T^Z)$ is closely related to $\widetilde{\text{Trf}}(T^C \rightarrow T^Z)$. Particularly, if task T^Z is hard, we expect it is more difficult to transfer from a trivial task to T^Z.

This relationship between hardness and transferability from a trivial task is similar to the one proposed by Task2Vec [1]. They too indexed task hardness as the distance from a trivial task. To compute task hardness, however, they required training deep models, whereas we obtain this measure by simply computing $H(Z|C)$ using Eq. (7).

Of course, estimating the hardness by $H(Z|C)$ ignores the input and is hence only an approximation. In particular, one could possibly design scenarios where this measure would not accurately reflect the hardness of a given task. Our results in Sec. 5.3 show, however, that these label statistics provide a strong cue for task hardness.

5. Experiments

We rigorously evaluate our claims using three large scale, widely used data sets representing 437 classification tasks. Although Sec. 3.2 provides a bound on the training loss, test accuracy is generally more important. We thus report results on test images not included in the training data.

Benchmarks. The *Celeb Faces Attributes* (CelebA) set [34] was extensively used to evaluate transfer learning [17, 32,

33, 56]. CelebA contains over 202k face images of 10,177 subjects. Each image is labeled with subject identity as well as 40 binary attributes. We used the standard train / test splits (182,626 / 19,961 images, respectively). To our knowledge, of the three sets, it is the only one that provides baseline results for attribute classification.

Animals with Attributes 2 (AwA2) [65] includes over 37k images labeled as belonging to one of 50 animals classes. Images are labeled based on their class association with 85 different attributes. Models were trained on 33,568 training images and tested on 3,754 separate test images.

Finally, *Caltech-UCSD Birds 200* (CUB) [64] offers 11,788 images of 200 bird species, labeled with 312 attributes as well as *Turker Confidence* attributes. Labels were averaged across multiple Turkers using confidences. Finally, we kept only reliable labels, using a threshold of 0.5 on the average confidence value. We used 5,994 images for training and 5,794 images for testing.

We note that the *Task Bank* set with its 26 tasks was also used for evaluating task relationships [69]. We did not use it here as it mostly contains regression tasks rather than the classification problems we are concerned with.

5.1. Evaluating task transferability

We compared our transferability estimates from the CE to the actual transferability of Eq. (4). To this end, for each attribute T^Z in a data set, we measure the actual transferability, $\text{Trf}(T^Z \rightarrow T^Y)$, to all other attributes T^Y in that set using the test split. We then compare these transferability scores to the corresponding CE estimates of Eq. (7) using an existing correlation analysis [43].

We note again that when the source task is fixed, as in this case, the transferability estimates can be obtained by considering only the CE. Furthermore, since $\text{Trf}(T^Z \rightarrow T^Y)$ and the CE $H(Y|Z)$ are negative correlated, we compare the correlation between the test error rate, $1 - \text{Trf}(T^Z \rightarrow T^Y)$, and the CE $H(Y|Z)$ instead.

Transferring representations. We keep the learned representation, w_Z, and produce a new classifier k_Y by training on the target task (Sec. 3.1). We used ResNet18 [24], trained with standard cross entropy loss, on each source task T^Z (source attribute). These networks were selected as they were deep enough to obtain good accuracy on our benchmarks, but not too deep to overfit [70]. The penultimate layer of these networks produce embeddings $r \in \mathbb{R}^{2048}$ which the networks classified using h_Z—their last, fully connected (FC) layers—to binary attribute values.

We transferred from source to target task by freezing the networks, only replacing their FC layers with linear SVM (lSVM). These lSVM were trained to predict the binary labels of *target* tasks given the embeddings produced for the source tasks by w_Z as their input. The test errors of the lSVM, which are measures of $1 - \text{Trf}(T^Z \rightarrow T^Y)$, were
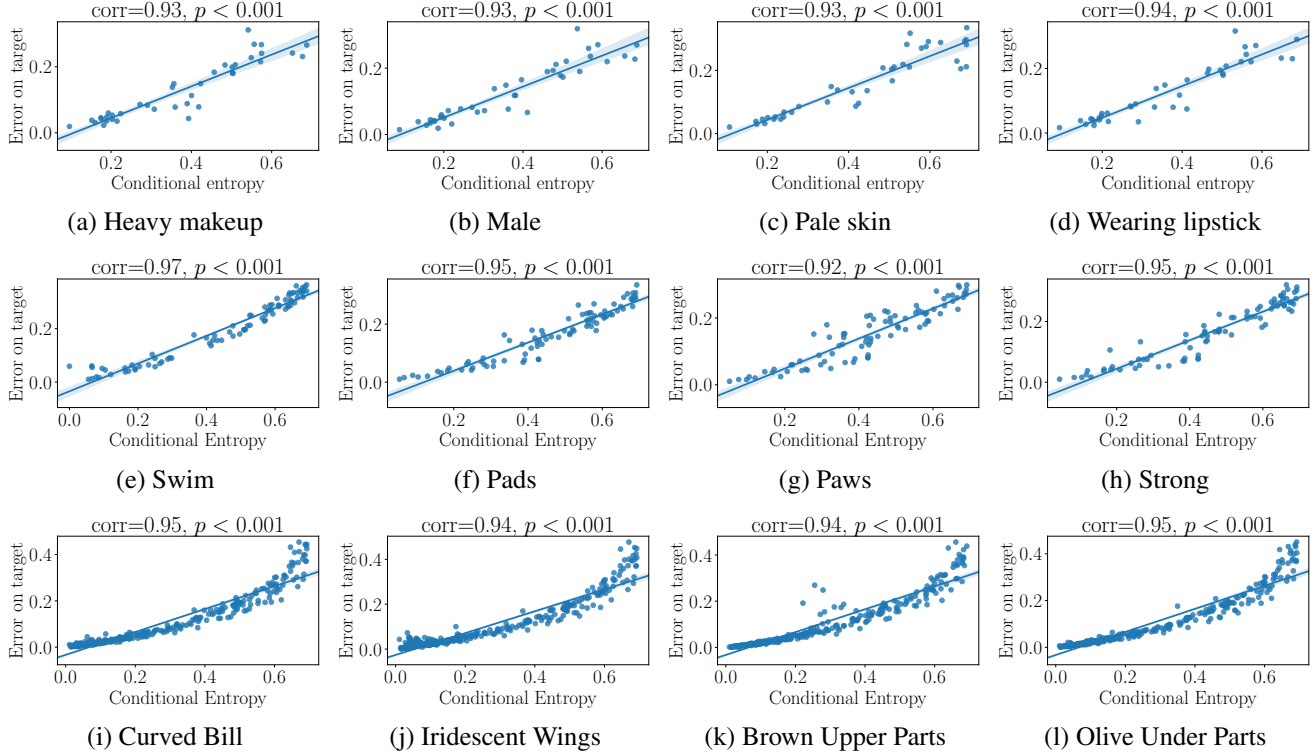
Figure 2. **Attribute prediction; CE vs. test errors on target tasks.** Examples from CelebA (a-d), AwA2 (e-h), and CUB (i-l). Plot titles name the source tasks T^Z; points represent different target tasks T^Y. Corr is the Pearson correlation coefficient between the two variables and p is the statistical significance of the correlation. In all cases, the correlation is statistically significant. See Sec. 5.1 for details.

	Attribute:	Male	Bald	Gray Hair	Mustache	Double Chin	...	Attractive	Wavy Hair	High Cheeks	Smiling	Mouth Open	Average (all)
1	LNets+ANet 2015 [34]	0.980	0.980	0.970	0.950	0.920		0.810	0.800	0.870	0.920	0.920	0.873
2	Walk and Learn 2016 [61]	0.960	0.920	0.950	0.900	0.930		0.840	0.850	0.950	0.980	0.970	0.887
3	MOON 2016 [53]	0.981	0.988	0.981	0.968	0.963		0.817	0.825	0.870	0.926	0.935	0.909
4	LMLE 2016 [25]	0.990	0.900	0.910	0.730	0.740		0.880	0.830	0.920	0.990	0.960	0.838
5	CR-I 2017 [15]	0.960	0.970	0.950	0.940	0.890		0.830	0.790	0.890	0.930	0.950	0.866
6	MCNN-AUX 2017 [22]	0.982	0.989	0.982	0.969	0.963	...	0.831	0.839	0.876	0.927	0.937	0.913
7	DMTL 2018 [21]	0.980	0.990	0.960	0.970	0.990		0.850	0.870	0.880	0.940	0.940	0.926
8	Face-SSD 2019 [26]	0.973	0.986	0.976	0.960	0.960		0.813	0.851	0.868	0.918	0.919	0.903
9	CE↑ (decreasing transferability)	0.017	0.026	0.052	0.062	0.083		0.361	0.381	0.476	0.521	0.551	-
10	Dedicated Res18	0.985	0.990	0.980	0.968	0.959		0.823	0.842	0.878	0.933	0.943	0.911
11	Transfer	0.992	0.991	0.981	0.968	0.963		0.820	0.800	0.859	0.909	0.901	0.902

Table 1. **Transferability from face recognition to facial attributes.** Results for CelebA attributes, sorted in ascending order of row 9 (decreasing transferability). Results are shown for the five attributes most and least transferable from recognition. Subject specific attributes, e.g., *male* and *bald*, are more transferable than expression related attributes such as *smiling* and *mouth open*. Unsurprisingly, transfer results (row 11) are best on the former than the latter. Rows 1-8 provide published state of the art results. Despite training only an lSVM for attribute, row 11 results are comparable with more elaborate attribute classification systems. For details, see Sec. 5.2.

then compared with the CE, $H(Y|Z)$.

We use lSVM as it allows us to focus on the information passed from T^Z to T^Y. A more complex classifier could potentially mask this information by being powerful enough to offset any loss of information due to the transfer. In practical use cases, when transferring a deep network from one task to another, it may be preferable to fine tune the last layers of the network or its entirety, provided that the training data on the target task is large enough.

Transferability results. Fig. 2 reports selected quantitative

transferability results on the three sets.[2] Each point in these graphs represents the CE, $H(Y|Z)$, vs. the target test error, $1 - \text{Trf}(T^Z \rightarrow T^Y)$. The graphs also provide the linear regression model fit with 95% confidence interval, the Pearson correlation coefficients between the two values, and the statistical significance of the correlation, p.

In all cases, the CE and target test error are highly positively correlated with statistical significance. These results testify that the CE of Eq. (7) is indeed a good predictor for the actual transferability of Eq. (4). This is remarkable es-
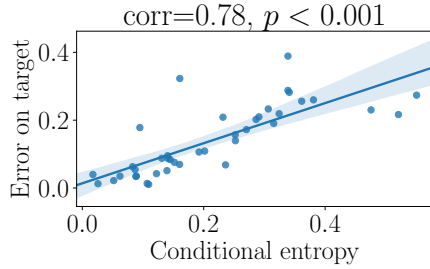
---

[2] For full results see supplemental.

Figure 3. **Identity to attribute; CE vs. test errors on target tasks.** Predicting 40 CelebA attributes using a face recognition network. Corr is the Pearson correlation coefficient between the two variables, and p is the statistical significance of the correlation.
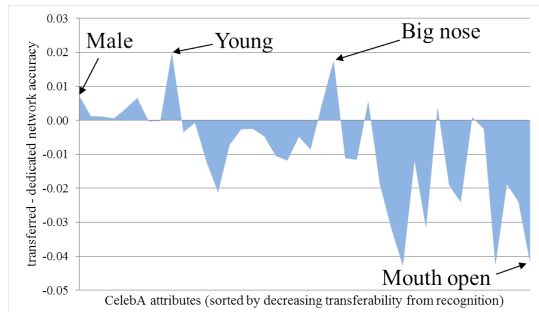

Figure 4. **Identity to attribute; transferred − dedicated accuracy.** Differences between CelebA accuracy of transferred recognition model and models trained for each attribute. Results are sorted by decreasing transferability (same as Table 1).

pecially since the relationship between tasks is evaluated without considering the input domain or the machine learning models trained to solve these tasks.

5.2. Case study: Identity to facial attributes

A key challenge when training effective attribute classifiers is the difficulty of obtaining labeled attribute training data. Whereas face images are often uploaded to the Internet along with subject names [9, 20], it is far less common to find images labeled with attributes such as *high cheek bones*, *bald*, or even *male* [31]. It is consequently harder to assemble training sets for attribute classification at the same scale and diversity as those used to train other tasks.

To reduce the burden of collecting attribute data, we therefore explore transferring a representation learned for face recognition. In this setting, we can also compute estimated transferability scores (via the CE) between the subject labels provided by CelebA and the labels of each attribute. We note that unlike the previous examples, the source labels are not binary and include over 10k values.

Face recognition network. We compare our estimated transferability vs. actual transferability using a deep face recognition network. To this end, we use a ResNet101 architecture trained for face recognition on the union of the
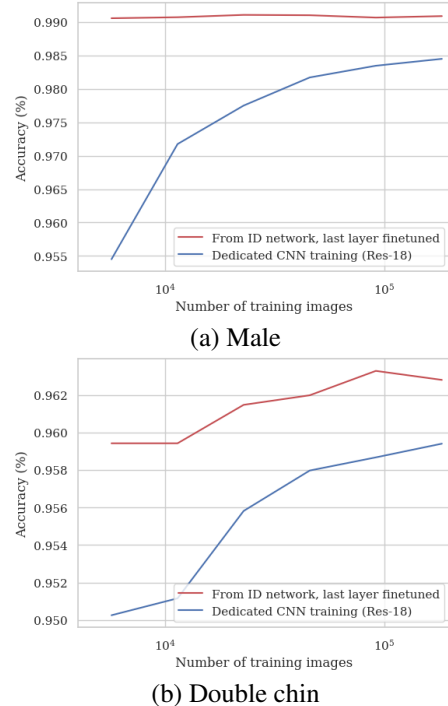

(a) Male

(b) Double chin

Figure 5. **Classification accuracy for varying training set sizes.** Top: *male*; bottom: *double chin*. Dedicated classification networks trained from scratch (blue) vs. face recognition network transferred to the attributes with an lSVM (red). Because recognition transfers well to these attributes, we obtain accurate classification with a fraction of the training data and effort.

MS-Celeb-1M [20] and VGGFace2 [9] training sets (following removal of subjects included in CelebA), with a cosine margin loss ($m = 0.4$) [60]. This network achieves accuracy comparable to the state of the art reported by others, with different systems, on standard benchmarks [14].

Transferability results: recognition to attributes. Table 1 reports results for the five attributes most transferable from recognition (smallest CE; Eq. (7)) and the five least transferable (largest CE). Columns are sorted by increasing CE values (decreasing transferability), listed in row 9. Row 11 reports accuracy of the transferred network with the lSVM trained on the target task. Estimated vs. actual transferability is further visualized in Fig. 3. Evidently, correlation between the two is statistically significant, testifying that Eq. (7) is a good predictor of actual transferability, here demonstrated on a source task with multiple labels.

For reference, Table 1 provides in Row 10 the accuracy of the dedicated ResNet18 networks trained for each attribute. Finally, rows 1 through 8 provide results for published state of the art on the same tasks.

Analysis of results. Subject specific attributes such as *male* and *bald* are evidently more transferable from recognition (left columns of Table 1) than attributes that are related to
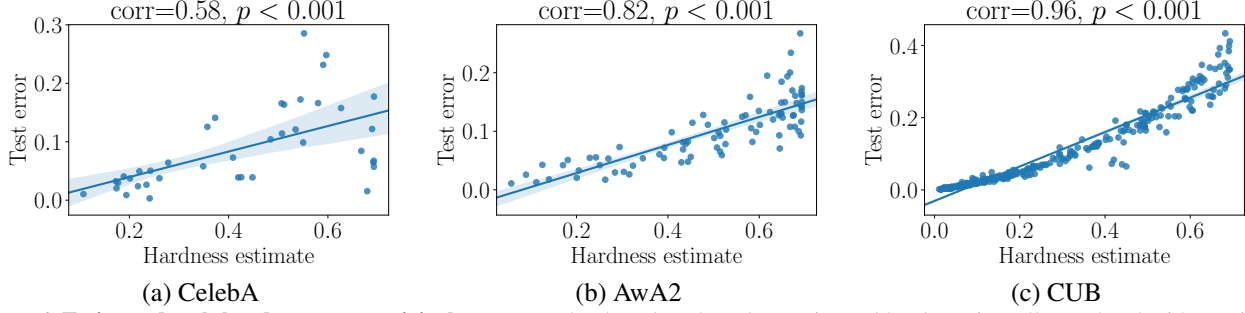
Figure 6. Estimated task hardness vs. empirical errors on the three benchmarks. Estimated hardness is well correlated with empirical hardness with significance $p < 0.001$.

expressions (e.g., *smiling* and *mouth open*, right columns). Although this relationship has been noted by others, previous work used domain knowledge to determine which attributes are more transferable from identity [34], as others have done in other domains [19, 37]. By comparison, our work shows how these relationships emerge from our estimation of transferability.

Also, notice that for the transferable attributes, our results are comparable to dedicated networks trained for each attribute, although they gradually drop off for the less transferable attributes in the last columns. This effect is visualized in Fig. 4 which shows the growing differences in attribute classification accuracy for a transferred face recognition model and models trained for each attribute. Results are sorted by decreasing transferability (same as in Table 1).

Results in Fig. 4 show a few notable exceptions where transfer performs substantially better than dedicated models (e.g., the two positive *peaks* representing attributes *young* and *big nose*). These and other occasional discrepancies in our results can be explained in the difference between the true transferability of Eq. (4), which we measure on the test sets, and Eq. (5), defined on the training sets and shown in Sec. 3.2 to be bounded by the CE.

Finally, we note that our goal is not to develop a state of the art facial attribute classification scheme. Nevertheless, results obtained by training an lSVM on embeddings transferred from a face recognition network are only 2.4% lower than the best scores reported by DMTL 2018 [21] (last column of Table 1). The effort involved in developing a state of the art face recognition network can be substantial. By transferring this network to attributes these efforts are amortized in training multiple facial attribute classifiers.

To emphasize this last point, consider Fig. 5 which reports classification accuracy on *male* and *double chin* for growing training set sizes. These attributes were selected as they are highly transferable from recognition (see Table 1). The figure compares the accuracy obtained by training a dedicated network (in blue) to a network transferred from recognition (red). Evidently, on these attributes, transferred accuracy is much higher with far less training data.

5.3. Evaluating task hardness

We evaluate our hardness estimates for all attribute classification tasks in the three data sets, using the CE $H(Z|C)$ in Eq. (14). Fig. 6 compares the hardness estimates for each task vs. the errors of our dedicated networks, trained from scratch to classify each attribute. Results are provided for CelebA, AwA2, and CUB.

The correlation between estimated hardness and classification errors is statistically significant with $p < 0.001$, suggesting that the CE $H(Z|C)$ in Eq. (14) indeed captures the hardness of these tasks. That is, in the three data sets, test error rates strongly correlate with our estimated hardness: the harder a task is estimated to be, the higher the errors produced by the model trained for the task. Of course, this result does not imply that the input domain has no impact on task hardness; only that the distribution of training labels already provides a strong predictor for task hardness.

6. Conclusions

We present a practical method for estimating the hardness and transferability of supervised classification tasks. We show that, in both cases, we produce reliable estimates by exploring training label statistics, particularly the conditional entropy between the sequences of labels assigned to the training data of each task. This approach is simpler than existing work, which obtains similar estimates by assuming the existence of trained models or by careful inspection of the training process. In our approach, computing conditional entropy is cheaper than training deep models, required by others for the same purpose.

We assume that different tasks share the same input domain (the same input images). It would be useful to extend our work to settings where the two tasks are defined over different domains (e.g., face vs. animal images). Our work further assumes discrete labels. Conditional entropy was originally defined over distributions. It is therefore reasonable that CE could be extended to non-discrete labeled tasks, such as, for faces, 3D reconstruction [58], pose estimation [10, 11] or segmentation [46].

References

[1] Alessandro Achille, Michael Lam, Rahul Tewari, Avinash Ravichandran, Subhransu Maji, Charless Fowlkes, Stefano Soatto, and Pietro Perona. Task2Vec: Task embedding for meta-learning. *arXiv preprint arXiv:1902.03545*, 2019.

[2] Alessandro Achille, Giovanni Paolini, Glen Mbeng, and Stefano Soatto. The information complexity of learning tasks, their structure and their distance. *arXiv preprint arXiv:1904.03292*, 2019.

[3] Sanjeev Arora, Aditya Bhaskara, Rong Ge, and Tengyu Ma. Provable bounds for learning some deep representations. In *Int. Conf. Mach. Learning*, pages 584–592, 2014.

[4] Hossein Azizpour, Ali Sharif Razavian, Josephine Sullivan, Atsuto Maki, and Stefan Carlsson. Factors of transferability for a generic convnet representation. *Trans. Pattern Anal. Mach. Intell.*, 38(9):1790–1802, 2015.

[5] Kamyar Azizzadenesheli, Anqi Liu, Fanny Yang, and Animashree Anandkumar. Regularized learning for domain adaptation under label shifts. In *Int. Conf. on Learning Representations*, 2019.

[6] Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. A theory of learning from different domains. *Mach. Learn.*, 79(1-2):151–175, 2010.

[7] Shai Ben-David and Reba Schuller. Exploiting task relatedness for multiple task learning. In *Learning Theory and Kernel Machines*, pages 567–580. Springer, 2003.

[8] John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman. Learning bounds for domain adaptation. In *Neural Inform. Process. Syst.*, pages 129–136, 2008.

[9] Q. Cao, L. Shen, W. Xie, O. M. Parkhi, and A. Zisserman. VGGFace2: A dataset for recognising faces across pose and age. In *Automatic Face and Gesture Recognition*, 2018.

[10] Feng-Ju Chang, Anh Tran, Tal Hassner, Iacopo Masi, Ram Nevatia, and Gérard Medioni. Faceposenet: Making a case for landmark-free face alignment. In *Proc. Int. Conf. Comput. Vision Workshops*, 2017.

[11] Feng-Ju Chang, Anh Tuan Tran, Tal Hassner, Iacopo Masi, Ram Nevatia, and Gérard Medioni. Deep, landmark-free fame: Face alignment, modeling, and expression estimation. *Int. J. Comput. Vision*, 127(6-7):930–956, 2019.

[12] Anna Choromanska, Mikael Henaff, Michael Mathieu, Gérard Ben Arous, and Yann LeCun. The loss surfaces of multilayer networks. In *Artificial Intelligence and Statistics*, pages 192–204, 2015.

[13] Thomas M Cover and Joy A Thomas. *Elements of information theory*. John Wiley & Sons, 2012.

[14] Prithviraj Dhar, Carlos Castillo, and Rama Chellappa. On measuring the iconicity of a face. In *Winter Conf. on App. of Comput. Vision*, pages 2137–2145. IEEE, 2019.

[15] Qi Dong, Shaogang Gong, and Xiatian Zhu. Class rectification hard mining for imbalanced deep learning. In *Proc. Conf. Comput. Vision Pattern Recognition*, pages 1851–1860, 2017.

[16] Richard O Duda, Peter E Hart, and David G Stork. *Pattern classification*. John Wiley & Sons, 2012.

[17] Emilien Dupont. Learning disentangled joint continuous and discrete representations. In *Neural Inform. Process. Syst.*, pages 708–718, 2018.

[18] Harrison Edwards and Amos Storkey. Towards a neural statistician. *arXiv preprint arXiv:1606.02185*, 2016.

[19] Deepti Ghadiyaram, Du Tran, and Dhruv Mahajan. Large-scale weakly-supervised pre-training for video action recognition. In *Proc. Conf. Comput. Vision Pattern Recognition*, pages 12046–12055, 2019.

[20] Yandong Guo, Lei Zhang, Yuxiao Hu, Xiaodong He, and Jianfeng Gao. MS-Celeb-1M: A dataset and benchmark for large scale face recognition. In *European Conf. Comput. Vision*. Springer, 2016.

[21] Hu Han, Anil K Jain, Fang Wang, Shiguang Shan, and Xilin Chen. Heterogeneous face attribute estimation: A deep multi-task learning approach. *Trans. Pattern Anal. Mach. Intell.*, 40(11):2597–2609, 2018.

[22] Emily M Hand and Rama Chellappa. Attributes for improved attributes: A multi-task network utilizing implicit and explicit relationships for facial attribute classification. In *AAAI Conf. on Artificial Intelligence*, 2017.

[23] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proc. Int. Conf. Comput. Vision*, pages 2961–2969, 2017.

[24] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proc. Conf. Comput. Vision Pattern Recognition*, June 2016.

[25] Chen Huang, Yining Li, Chen Change Loy, and Xiaoou Tang. Learning deep representation for imbalanced classification. In *Proc. Conf. Comput. Vision Pattern Recognition*, pages 5375–5384, 2016.

[26] Youngkyoon Jang, Hatice Gunes, and Ioannis Patras. Registration-free face-ssd: Single shot analysis of smiles, facial attributes, and affect in the wild. *Comput. Vision Image Understanding*, 2019.

[27] Brendan Jou and Shih-Fu Chang. Deep cross residual learning for multitask visual recognition. In *Int. Conf. Multimedia*, pages 998–1007. ACM, 2016.

[28] Alex Kendall, Yarin Gal, and Roberto Cipolla. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7482–7491, 2018.

[29] Iasonas Kokkinos. Ubernet: Training a universal convolutional neural network for low-, mid-, and high-level vision using diverse datasets and limited memory. In *Proc. Conf. Comput. Vision Pattern Recognition*, pages 6129–6138, 2017.

[30] Giwoong Lee, Eunho Yang, and Sung Hwang. Asymmetric multi-task learning based on task relatedness and loss. In *Int. Conf. Mach. Learning*, pages 230–238, 2016.

[31] Gil Levi and Tal Hassner. Age and gender classification using convolutional neural networks. In *Proc. Conf. Comput. Vision Pattern Recognition Workshops*, June 2015.

[32] Yu Liu, Fangyin Wei, Jing Shao, Lu Sheng, Junjie Yan, and Xiaogang Wang. Exploring disentangled feature representation beyond face identification. In *Proc. Conf. Comput. Vision Pattern Recognition*, pages 2080–2089, 2018.

[33] Yen-Cheng Liu, Yu-Ying Yeh, Tzu-Chien Fu, Sheng-De Wang, Wei-Chen Chiu, and Yu-Chiang Frank Wang. Detach and adapt: Learning cross-domain disentangled deep representation. In *Proc. Conf. Comput. Vision Pattern Recognition*, pages 8867–8876, 2018.

[34] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proc. Int. Conf. Comput. Vision*, 2015.

[35] David G Lowe. Object recognition from local scale-invariant features. In *Proc. Int. Conf. Comput. Vision*, page 1150, 1999.

[36] Yongxi Lu, Abhishek Kumar, Shuangfei Zhai, Yu Cheng, Tara Javidi, and Rogerio Feris. Fully-adaptive feature sharing in multi-task networks with applications in person attribute classification. In *Proc. Conf. Comput. Vision Pattern Recognition*, pages 5334–5343, 2017.

[37] Dhruv Mahajan, Ross Girshick, Vignesh Ramanathan, Kaiming He, Manohar Paluri, Yixuan Li, Ashwin Bharambe, and Laurens van der Maaten. Exploring the limits of weakly supervised pretraining. In *European Conf. Comput. Vision*, pages 181–196, 2018.

[38] Yishay Mansour, Mehryar Mohri, and Afshin Rostamizadeh. Domain adaptation: Learning bounds and algorithms. In *Conference on Learning Theory*, 2009.

[39] I. Masi, F. J. Chang, J. Choi, S. Harel, J. Kim, K. Kim, J. Leksut, S. Rawls, Y. Wu, T. Hassner, W. AbdAlmageed, G. Medioni, L. P. Morency, P. Natarajan, and R. Nevatia. Learning pose-aware models for pose-invariant face recognition in the wild. *Trans. Pattern Anal. Mach. Intell.*, 2018.

[40] Iacopo Masi, Anh Tuan Tran, Tal Hassner, Gozde Sahin, and Gérard Medioni. Face-specific data augmentation for unconstrained face recognition. *Int. J. Comput. Vision*, 127(6-7):642–667, 2019.

[41] Colin McDiarmid. On the method of bounded differences. *Surveys in combinatorics*, 141(1):148–188, 1989.

[42] Ishan Misra, Abhinav Shrivastava, Abhinav Gupta, and Martial Hebert. Cross-stitch networks for multi-task learning. In *Proc. Conf. Comput. Vision Pattern Recognition*, pages 3994–4003, 2016.

[43] Cuong V Nguyen, Alessandro Achille, Michael Lam, Tal Hassner, Vijay Mahadevan, and Stefano Soatto. Toward understanding catastrophic forgetting in continual learning. *arXiv:1908.01091*, 2019.

[44] Cuong V Nguyen, Yingzhen Li, Thang D Bui, and Richard E Turner. Variational continual learning. In *Int. Conf. on Learning Representations*, 2018.

[45] Quynh Nguyen and Matthias Hein. The loss surface of deep and wide neural networks. In *Int. Conf. Mach. Learning*, pages 2603–2612, 2017.

[46] Yuval Nirkin, Iacopo Masi, Anh Tran Tuan, Tal Hassner, and Gerard Medioni. On face segmentation, face swapping, and face perception. In *Int. Conf. on Automatic Face and Gesture Recognition*, pages 98–105. IEEE, 2018.

[47] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *Trans. Knowledge and Data Eng.*, 22(10):1345–1359, 2010.

[48] Rajeev Ranjan, Vishal M Patel, and Rama Chellappa. Hyperface: A deep multi-task learning framework for face detection, landmark localization, pose estimation, and gender recognition. *Trans. Pattern Anal. Mach. Intell.*, 41(1):121–135, 2019.

[49] Rajeev Ranjan, Swami Sankaranarayanan, Carlos D Castillo, and Rama Chellappa. An all-in-one convolutional neural network for face analysis. In *Int. Conf. on Automatic Face and Gesture Recognition*, pages 17–24. IEEE, 2017.

[50] Mark B Ring. CHILD: A first step towards continual learning. *Mach. Learn.*, 28(1):77–104, 1997.

[51] Rasmus Rothe, Radu Timofte, and Luc Van Gool. Dex: Deep expectation of apparent age from a single image. In *Proc. Int. Conf. Comput. Vision Workshops*, pages 10–15, 2015.

[52] Artem Rozantsev, Mathieu Salzmann, and Pascal Fua. Beyond sharing weights for deep domain adaptation. *Trans. Pattern Anal. Mach. Intell.*, 41(4):801–814, 2019.

[53] Ethan M Rudd, Manuel Günther, and Terrance E Boult. Moon: A mixed objective optimization network for the recognition of facial attributes. In *European Conf. Comput. Vision*, pages 19–35. Springer, 2016.

[54] Andrei A Rusu, Dushyant Rao, Jakub Sygnowski, Oriol Vinyals, Razvan Pascanu, Simon Osindero, and Raia Hadsell. Meta-learning with latent embedding optimization. In *Int. Conf. on Learning Representations*, 2019.

[55] Bernhard Scholkopf and Alexander J Smola. *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press, 2001.

[56] Sukrit Shankar, Duncan Robertson, Yani Ioannou, Antonio Criminisi, and Roberto Cipolla. Refining architectures of deep convolutional neural networks. In *Proc. Conf. Comput. Vision Pattern Recognition*, pages 2212–2220, 2016.

[57] A Torralba and AA Efros. Unbiased look at dataset bias. In *Proc. Conf. Comput. Vision Pattern Recognition*, pages 1521–1528. IEEE Computer Society, 2011.

[58] Anh Tuan Tran, Tal Hassner, Iacopo Masi, Eran Paz, Yuval Nirkin, and Gérard Medioni. Extreme 3D face reconstruction: Looking past occlusions. In *Proc. Conf. Comput. Vision Pattern Recognition*, 2018.

[59] Andreas Veit, Serge Belongie, and Theofanis Karaletsos. Conditional similarity networks. In *Proc. Conf. Comput. Vision Pattern Recognition*, pages 830–838, 2017.

[60] Hao Wang, Yitong Wang, Zheng Zhou, Xing Ji, Dihong Gong, Jingchao Zhou, Zhifeng Li, and Wei Liu. Cosface: Large margin cosine loss for deep face recognition. In *Proc. Conf. Comput. Vision Pattern Recognition*, pages 5265–5274, 2018.

[61] Jing Wang, Yu Cheng, and Rogerio Schmidt Feris. Walk and learn: Facial attribute representation learning from egocentric video and contextual data. In *Proc. Conf. Comput. Vision Pattern Recognition*, pages 2295–2304, 2016.

[62] Peng Wang, Xiaohui Shen, Zhe Lin, Scott Cohen, Brian Price, and Alan L Yuille. Towards unified depth and semantic prediction from a single image. In *Proc. Conf. Comput. Vision Pattern Recognition*, pages 2800–2809, 2015.

[63] Karl Weiss, Taghi M Khoshgoftaar, and DingDing Wang. A survey of transfer learning. *Journal of Big Data*, 3(1):9, 2016.

[64] P. Welinder, S. Branson, T. Mita, C. Wah, F. Schroff, S. Belongie, and P. Perona. Caltech-UCSD Birds 200. Technical Report CNS-TR-2010-001, California Institute of Technology, 2010.

[65] Yongqin Xian, Christoph H Lampert, Bernt Schiele, and Zeynep Akata. Zero-shot learning-a comprehensive evaluation of the good, the bad and the ugly. *Trans. Pattern Anal. Mach. Intell.*, 2018.

[66] Yongxin Yang and Timothy Hospedales. Deep multi-task representation learning: A tensor factorisation approach. In *Int. Conf. on Learning Representations*, 2017.

[67] Wei Ying, Yu Zhang, Junzhou Huang, and Qiang Yang. Transfer learning via learning to transfer. In *Int. Conf. Mach. Learning*, pages 5072–5081, 2018.

[68] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? In *Neural Inform. Process. Syst.*, pages 3320–3328, 2014.

[69] Amir R Zamir, Alexander Sax, William Shen, Leonidas J Guibas, Jitendra Malik, and Silvio Savarese. Taskonomy: Disentangling task transfer learning. In *Proc. Conf. Comput. Vision Pattern Recognition*, pages 3712–3722, 2018.

[70] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. In *Int. Conf. on Learning Representations*, 2017.

[71] Xiangyun Zhao, Haoxiang Li, Xiaohui Shen, Xiaodan Liang, and Ying Wu. A modulation module for multi-task learning with applications in image retrieval. In *European Conf. Comput. Vision*, pages 401–416, 2018.