

One-Shot Learning of Scene Locations via Feature Trajectory Transfer

Roland Kwitt, Sebastian Hegenbart
University of Salzburg, Austria

roland.kwitt@gmail.com, sebastian.hegenbart@gmail.com

Marc Niethammer
UNC Chapel Hill, NC, United States

mn@cs.unc.edu

Abstract

The appearance of (outdoor) scenes changes considerably with the strength of certain transient attributes, such as “rainy”, “dark” or “sunny”. Obviously, this also affects the representation of an image in feature space, e.g., as activations at a certain CNN layer, and consequently impacts scene recognition performance. In this work, we investigate the variability in these transient attributes as a rich source of information for studying how image representations change as a function of attribute strength. In particular, we leverage a recently introduced dataset with fine-grain annotations to estimate feature trajectories for a collection of transient attributes and then show how these trajectories can be transferred to new image representations. This enables us to synthesize new data along the transferred trajectories with respect to the dimensions of the space spanned by the transient attributes. Applicability of this concept is demonstrated on the problem of one-shot recognition of scene locations. We show that data synthesized via feature trajectory transfer considerably boosts recognition performance, (1) with respect to baselines and (2) in combination with state-of-the-art approaches in one-shot learning.

1. Introduction

Learning new visual concepts from only a *single* image is a remarkable ability of human perception. Yet, the predominant setting of recognition experiments in computer vision is to measure success of a learning process with respect to hundreds or even thousands of training instances. While datasets of such size were previously only available for object-centric recognition [2], the emergence of the *Places* database [31] has made a large corpus of scene-centric data available for research. Coupled with advances in CNN-based representations and variants thereof, the performance of scene recognition systems has improved remarkably in the recent past [8, 31, 3], even on already well established (and relatively small) benchmarks such as “15 Scenes”. However, large scale databases are typically constructed

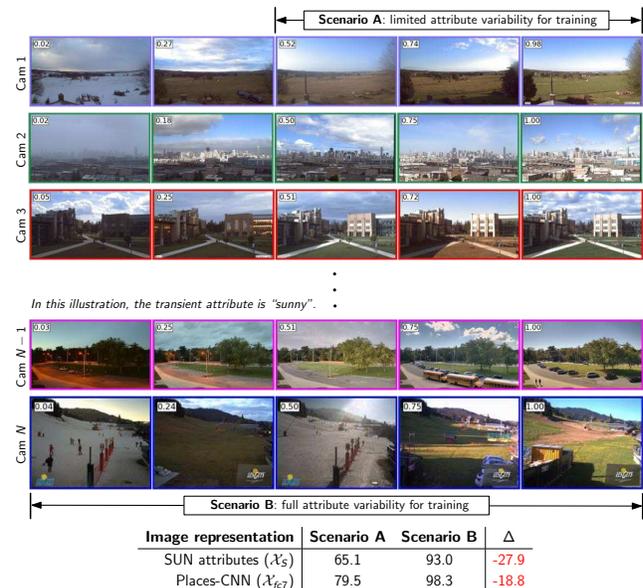


Figure 1: Introductory experiment on the *Transient Attributes Database* of [11]. The task is to distinguish scenes from different webcams. In *Scenario A*, images with annotated attribute strengths (in the range of $[0, 1]$) less than 0.4 are excluded during training and used for testing. Results are averaged over 40 attributes. *Scenario B* represents a standard five-fold cross-validation setup, using random splits of the data. The size of each training split is set to approximately the size as in *Scenario A*. In summary, significant drops in recognition accuracy occur when only limited variability is present in the training data with respect to (transient) attributes.

by fetching images from the web and subsequently crowdsourcing the annotation task. Consequently, a considerable user bias [25] is expected, e.g., with respect to captured conditions. It is, e.g., highly unlikely that scenes of “city skylines” or “beaches” are captured during rainy or cloudy conditions. Hence, we expect that the observed variability of transient states in readily available scene databases will be limited. The question then is if this existing visual data is rich enough to sufficiently inform the process of learning to recognize scenes from single instances, i.e., the aforementioned task on which humans perform so exceptionally well.

While the majority of approaches to scene recognition either rely on variants of Bag-of-Words [6, 13], Fisher vectors [22], or outputs of certain CNN layers [8, 31, 3], several works have also advocated more abstract, semantic (attribute) representations [21, 23, 10]. Largely due to the absence of fine-grained semantic annotations, the axes of the semantic space typically correspond to the scene category labels. To alleviate this problem and to enable attribute representations of scene images, Patterson *et al.* [18] have recently introduced the SUN attribute database. This is motivated by the success of attributes in object-centric recognition [4, 12]. However, construction of the attribute vocabulary is guided by the premise of enhancing discriminability of scene categories. Somewhat orthogonal to this objective, Laffont *et al.* [11] recently introduced the concept of *transient* attributes, such as “sunny” or “foggy”. This is particularly interesting in the context of studying scene variability, since the collection of attributes is designed with high variability in mind (as opposed to discriminability). Equipped with a collection of trained attribute strength regressors, this enables us to more thoroughly assess variability issues.

Is limited scene variability a problem? To motivate our work, we start with three simple experiments. For technical details, we refer the reader to Section 4.

First, we consider the (supposedly) simple recognition problem of distinguishing images from the 101 webcams used in the *Transient Attribute Database (TADB)* [11]. We refer to this task as recognition of *scene locations*, since each webcam records images from one particular location. Each image is hand-annotated by the strength of 40 different transient attributes. We use activations of the final fully-connected layer (‘fc7’) in *Places-CNN* [31] as our feature space $\mathcal{X}_{fc7} \subset \mathbb{R}^{4096}$ and a linear support vector classifier. In *Scenario A*, images with annotated attribute strength below a certain threshold are excluded during training and used for testing. In *Scenario B*, we perform five-fold cross-validation using random splits of *all* available data. Fig. 1 shows that almost perfect accuracy (98.3%) is achieved in *Scenario B*. However, the recognition rate drops by almost 20 percentage points in *Scenario A*. This clearly highlights that limited variability with respect to the transient state of a scene can severely impact recognition accuracy, even when using one of the state-of-the-art CNN representations.

In our *second* experiment, the task is to predict the scene category of images in *TADB*, this time by directly using the 205 category labels as outputted by the pre-trained *Places-CNN*. In this setup, images from different scene locations (*i.e.*, webcams) could be from the same scene category (*e.g.*, “mountain”). However, under the assumption of invariance against the transient states of a scene, we would expect to obtain *consistent* predictions over *all* images from the same webcam. This is not always the case, though, as can be seen from Fig. 2 showing scene locations from *TADB* with highly

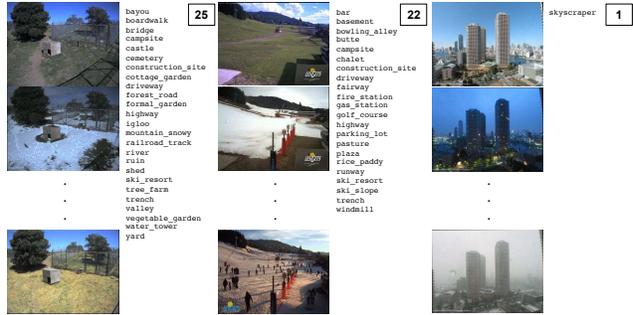


Figure 2: Illustration of the two locations (left) from *TADB* with *most inconsistent* predictions by *Places-CNN*. The number in the top-right hand corner of each column denotes the number of different predictions. The rightmost column shows the *only* scene with a consistent prediction over all its images. The mean number of different category predictions over *all TADB* webcams is 10.3 ± 4.6 .

inconsistent predictions, as well as the *only* scene location where the *Places-CNN* prediction is consistent over all images from this webcam. On average, we get 10.3 ± 4.6 different predictions per webcam. This is in line with our observation from the previous experiment. It further strengthens our conjecture that features learned by *Places-CNN* are *not* invariant with respect to transient states of a scene.

Third, we assess how much variability in transient attributes is covered by existing image corpora for scene recognition. In particular, we train a collection of regressors to predict the strength of all 40 attributes in *TADB*, using the same CNN-based image representation of the previous two experiments. We then use the trained regressors to map each image of the *SUN397* database [27] into the 40-dimensional attribute space. Note that the transient attributes only apply to outdoor scenes; consequently only images from the outdoor categories in *SUN397* were used. For each category and attribute, we record the 5th and the 95th percentile of the predicted attribute strength (denoted by p_5 and p_{95}) and compute the range ($r = p_{95} - p_5$). Fig. 3 shows a series of boxplots, where each column represents the distribution of range values r , collected from all images and attributes in a particular scene category. The median over *all* range values is 0.33, indicating that the observed variability is limited with respect to the transient attributes. We also conducted the same experiment on 50000 randomly sampled images from 10 outdoor categories of the *Places* dataset*. While the median increases to 0.41, this is still far below a complete coverage of the transient attribute space. For comparison, when running the same experiment on *TADB* itself, the median is close to 1 by construction.

In summary, the first two experiments highlight the negative effect of limited variability in the training data, the

*abbey, alley, amphitheater, apartment building (outdoor), aqueduct, arch, badlands, butte, basilica and bayou

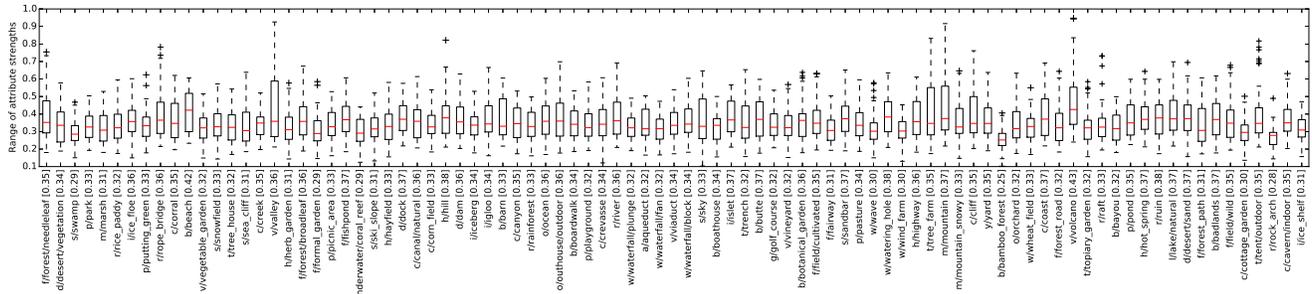


Figure 3: Illustration of limited scene variability with respect to transient attributes in the outdoor scenes of *SUN397*. Each column represents the distribution of range values r — defined as $r = (p_{95} - p_5)$, where p_i denotes the i th percentile of the data — collected from all images and attributes in a scene category (listed on the x -axis).

third experiment is a direct illustration of the lack of variability. Even in case of *Places*, our experiments suggest that the image data does not sufficiently cover variability in transient attributes so that the CNN can learn features that exhibit the required degree of invariance. Note that the most extreme scenario of limited variability is learning from single instances, *i.e.*, one-shot recognition. Only a *single* transient state is observed per scene category. To the best of our knowledge, this has gained little attention so far in the scene recognition community. The motivating question for this work boils down to asking *whether we can artificially increase variability – via data synthesis – by learning how image representations change depending on the strength of transient attributes*.

Organization. In Section 2 we review related work. Section 3 introduces the proposed concept of *feature trajectory transfer*. Section 4 presents our experimental evaluation of the main parts of our pipeline and demonstrates one-shot recognition of scene locations. Section 5 concludes the paper with a discussion of the main findings and open issues.

2. Related work

Most previous works in the literature consider the problem of scene recognition in a setting where sufficient training data per scene category is available, *e.g.*, [28, 10, 3, 31]. To the best of our knowledge, one-shot recognition has not been attempted so far. However, there have been many efforts to one-shot learning in the context of object-centric recognition [17, 24, 30, 5]. In our review, we primarily focus on these approaches.

A consensus among most avenues to one-shot learning is the idea of “knowledge transfer”, *i.e.*, to let knowledge about previously seen (*external*) data influence the process of learning from only a single instance of each new class.

In early work, Miller *et al.* [17] follow the idea of synthesizing data for classes (in their case digits) with only a single training instance. This is done through an iterative process called *congealing* which aligns the external images

of a given category by optimizing over a class of geometric transforms (*e.g.*, affine transforms). The learned transforms are then applied to each single instances of the new categories to augment the image data.

In [5], Fei-Fei *et al.* coin the term one-shot learning in the context of visual recognition problems. They propose a Bayesian approach where priors on the parameters of models for known object categories (in the external data) are learned and later adapted to object categories with few (or only a single) training instances.

In [7], Fink advocates to leverage pseudo-metric learning to find a linear projection of external training data that maximizes class separation. The learned projection is then used to transform the one-shot instances of new categories and a classifier is learned in the transform space. Tang *et al.* [24] follow a similar idea, but advocate the concept of learning with *micro-sets*. The key to achieve better results is to learn over multiple training sets with only a single instance per category, *i.e.*, the *micro-sets*. This has the effect of already simulating a one-shot recognition setting during metric learning on external data.

Bart and Ullmann [1] propose to use feature adaptation for one-shot recognition. In particular, features (*e.g.*, informative image fragments) in new, single instances of a category are selected based on their similarity to features in the external training data and their performance in discriminating the external categories.

Yu and Aloimonos [30] tackle one-shot learning of object categories by leveraging attribute representations and a topic model that captures the distribution of features related to attributes. During learning, training data is synthesized using the attribute description of each new object category.

Pfister *et al.* [20] propose one-shot learning of gestures using information from an external weakly-labeled gesture dataset. Exemplar SVMs [16] are used to train detectors for the single gesture instances and the detectors are then used to mine more training data from the external database.

Salakhutdinov *et al.* [26] consider one-shot learning by

adding a hierarchical prior over high-level features of Deep Boltzmann Machines. The model is trained on external data plus the one-shot instances of the novel categories. While the approach generalizes well on the categories with few (or only a single) training instance(s), information transfer happens during training which requires to retrain in case new categories are added.

Recently, Yan *et al.* [29] have explored multi-task learning for one-shot recognition of events in video data. External data from different categories is added to the single instances of the new categories; then, multi-task learning is used to distinguish the categories with the intuition that tasks with lots of training data inform the process of learning to distinguish the new categories.

Conceptually, our approach is related to Miller *et al.* [17] and Yu and Aloimonos [30] in the sense that we also synthesize data. However, Miller *et al.* synthesize in the input space of *images*, while we synthesize image representations in *feature space*. The generative model of [30] also allows feature synthesis, however, does not specifically leverage the structure of the space as a function of a continuous state. With respect to other work, we argue that our approach is *not* a direct competitor, but rather complementary, *e.g.*, to the pseudo-metric learning approaches of Fink [7] or Tang *et al.* [24]. In Section 4, we present experiments that specifically highlight this point.

3. Methodology

On an abstract level, the key idea of our approach is to use information obtained from an *external* training corpus to synthesize additional samples starting from a limited amount of previously unseen data. The objective is to increase the variability of transient states available for training. Specifically, we want to use knowledge about how the appearance of a scene location – as captured by its image representation – varies depending on the state of some transient scene attributes. This concept, illustrated in Fig. 4, is based on two assumptions: *First*, we can predict the transient state of a scene image based on its representation. *Second*, we can model the functional dependency between such a transient state and the elements of the feature representation as a *trajectory* in feature space. We will provide empirical evidence backing both assumptions in Section 4.

3.1. Feature trajectory transfer

We adhere to the following conventions. We denote by I_1, \dots, I_N the *external* training corpus of N images. Each image I_i is assigned to one of C scene locations (categories) with label $y_i \in [C]^\dagger$ and represented by a D -dimensional feature vector $\mathbf{x}_i \in \mathcal{X} \subset \mathbb{R}^D$. Additionally, each image is annotated with a vector of transient attribute strengths $\mathbf{a}_i \in$

$^\dagger [n]$ is the set $\{1, \dots, n\}$ with $n \in \mathbb{N}$

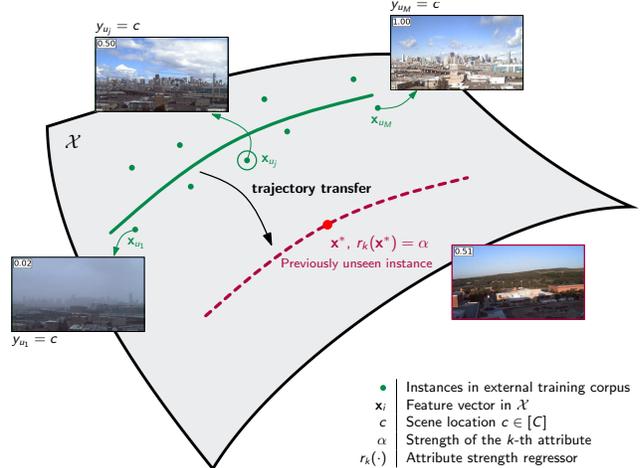


Figure 4: Illustration of *feature trajectory transfer* and synthesis. For a previously unseen image, given by $\mathbf{x}^* \in \mathcal{X}$, we *first* compute its representation in the transient attribute space by evaluating the learned attribute regressors r_k . *Second*, we iterate over all scene locations (here only **one scene location** is shown) from the external image corpus and transfer the learned feature trajectories for each attribute to \mathbf{x}^* . *Third*, this allows us to predict features along the **trajectory** for this attribute (over its full range). The final synthesized image representation is a weighted combination of predictions from all scene locations, see Eq. (4).

\mathbb{R}_+^A , where \mathcal{A}_T denotes the set of attributes and $A = |\mathcal{A}_T|$. Also, let I^* be a previously unseen image, represented by $\mathbf{x}^* \in \mathcal{X}$, and let $\mathbf{x}[d]$ denotes the d -th component of \mathbf{x} .

Since no attribute representation is available for unseen data, the idea is to use a collection of learned regressors $r_k : \mathcal{X} \rightarrow \mathbb{R}_+$, $k \in [A]$ to estimate the attribute strength vector $[r_1(\mathbf{x}^*), \dots, r_A(\mathbf{x}^*)]$ from \mathbf{x}^* . These attribute strength regressors can be learned, *e.g.*, using support vector regression, or Gaussian process regression (cf. Section 4).

For a given scene location, we wish to estimate the path $\gamma_k : \mathbb{R}_+ \rightarrow \mathcal{X}$ for every attribute in \mathcal{A}_T . In our case, we rely on a simple linear model to represent this path. Formally, let's fix the scene location c and let $S_c = \{i : y_i = c\}$ be the index set of the $M = |S_c|$ images from this location. The model, for the k -th attribute, can then be written as

$$\mathbf{x}_i = \mathbf{w}_k \cdot \mathbf{a}_i[k] + \mathbf{b}_k + \epsilon_k \quad (1)$$

where $\mathbf{w}_k, \mathbf{b}_k \in \mathbb{R}^D$ are the slope and intercept parameters and ϵ_k denotes (component-wise) Gaussian noise. We can easily estimate this model, for a specific choice of c , using linear regression with data vectors \mathbf{z} and \mathbf{v} , *i.e.*,

$$\mathbf{z} = \begin{bmatrix} \mathbf{x}_{u_1}[d] \\ \vdots \\ \mathbf{x}_{u_M}[d] \end{bmatrix}, \mathbf{v} = \begin{bmatrix} \mathbf{a}_{u_1}[k] \\ \vdots \\ \mathbf{a}_{u_M}[k] \end{bmatrix}, u_i \in S_c, d \in [D]. \quad (2)$$

Note that for each dimension d and attribute k , we obtain one tuple $(\mathbf{w}_k[d], \mathbf{b}_k[d])$ of slope and intercept that

parametrizes the linear model. In summary, we estimate $(\mathbf{w}_1, \mathbf{b}_1), \dots, (\mathbf{w}_A, \mathbf{b}_A)$ for every scene location c , describing the feature trajectories in the external data corpus.

Synthesizing new data, starting from a previously unseen instance \mathbf{x}^* , can now be done in the following way. Lets consider the feature trajectory of the k -th attribute and scene location c (parameterized by \mathbf{w}_k and \mathbf{b}_k). We define the synthesis function $s_k : \mathbb{R}_+ \times \mathcal{X} \rightarrow \mathcal{X}$ as

$$s_k(t, \mathbf{x}^*) = \mathbf{w}_k \cdot (t - r_k(\mathbf{x}^*)) + \mathbf{x}^*. \quad (3)$$

Up to this point, we have fixed c for readability. However, in practice the question remains how to select the *most informative* scene location from the external data to perform trajectory transfer. Conceptually, we follow the principle (cf. [1]) of identifying informativeness as the similarity of I^* to each scene location in the external training corpus. In particular, our objective is to select locations which are similar in appearance, *e.g.*, as measured in the feature space \mathcal{X} . To avoid relying on hard decisions for one location, we advocate to take the trajectories of *all* locations into account. This can be achieved by weighing the contribution of each model with the similarity of I^* to a scene location. In our approach, weights are determined by the posterior probability of \mathbf{x}^* under all locations, *i.e.*, $\forall c : p_c = P[c|\mathbf{x}^*]$. This can be obtained, *e.g.*, by using the output of a probability-calibrated support vector classifier that is trained (on the external training corpus) to distinguish between locations. The synthesis function of Eq. (3) is then (re)formulated as

$$s_k(t, \mathbf{x}^*) = \left[\sum_{c \in [C]} \underbrace{P[c|\mathbf{x}^*]}_{p_c} \mathbf{w}_k^c \cdot (t - r_k(\mathbf{x}^*)) \right] + \mathbf{x}^*. \quad (4)$$

Alternatively, the contribution of each location could be ranked by $P[c|\mathbf{x}^*]$. *This process of synthesizing additional feature representations as a function of the desired attribute strength $t \in \mathbb{R}_+$ is repeated for every attribute in \mathcal{A}_T .*

Relation to data augmentation. While the process of synthesizing image representations in \mathcal{X} , depending on attribute strength, is particularly useful in our target application of one-shot recognition, it can potentially be used as an augmentation technique in other scenarios as well.

4. Experiments

Our experiments are structured as follows: *First*, we assess the performance of predicting attribute strength, *i.e.*, the quality of r_k (see Section 3.1). *Second*, we evaluate the quality of our linear model for synthesizing different image representations. *Third*, we address the problem of one-shot recognition of scene locations as our target application.

Datasets. We use the *Transient Attributes Database (TADB)* [11] that we already briefly introduced in Section 1

in all recognition experiments. In detail, the dataset contains 8571 images from 101 webcams which serve as our *scene locations*. The recognition task is to *assign an image to the correct scene location*. While this might sound artificial and overly simplistic at first, Figs. 1 and 2 reveal that the task is actually fairly difficult, especially when the training data is constrained to only contain a certain amount of variability in the transient states. Each image in *TADB* is hand-annotated with 40 transient attributes; this set is denoted as \mathcal{A}_T and corresponds to the attribute set in Section 3. Further, we also use the *SUN Attributes Database (SADB)* [18] which contains 14340 images, each hand-annotated with respect to a collection \mathcal{A}_S of 102 discriminative attributes. *Note that SADB is only used for training attribute detectors (i.e., presence/absence).* These detectors are then employed to map images, represented in \mathcal{X}_{fc7} (specified below), into the alternative feature space $\mathcal{X}_S \subset \mathbb{R}^{|\mathcal{A}_S|}$.

Image representations. All of our image representations build upon *Places-CNN* [31], one of the state-of-the-art CNN architectures for scene recognition. The CNN is based on the AlexNet architecture of [9]. In particular, we use the activations of the 4096 neurons in the last fully-connected layer ('fc7') and denote this as our feature space \mathcal{X}_{fc7} . We either represent images *directly* in that space, use it as an intermediate image representation to detect discriminative attributes from \mathcal{A}_S , or use it to regress the strength of transient attributes from \mathcal{A}_T . In all recognition experiments, we further perform dimensionality reduction of features in \mathcal{X}_{fc7} via PCA to 200 dimensions. This retains $\approx 90\%$ of the variance in the data. In case of features in \mathcal{X}_S , we also perform PCA, not to reduce dimensionality, but to decorrelate the feature dimensions. PCA is followed by component-wise normalization to $[-1, 1]$, similar to [24].

Implementation. To obtain regressors for the strength of the 40 transient attributes in \mathcal{A}_T , we train a collection of linear support vector regressors (SVR). To obtain detectors for the presence/absence of the 102 attributes in \mathcal{A}_S , we train a collection of linear support vector classifiers (SVC), configured for probability outputs. We do *not* binarize the predictions (as in [18]), but use the probability for an attribute being present to represent images in \mathcal{X}_S . Probability outputs of a SVC are also used to obtain the weights p_c required to compute Eq. (4). In that case, the SVC is trained (one-vs-all) to distinguish scene locations (*i.e.*, webcams) from the external *TADB* training data. The latter setup is also used for recognition of scene locations in the one-shot learning experiments. In all cases, we use the SVM implementation available as part of the `scikit-learn` [19] toolbox; the SVR/SVC cost parameter C is configured using five-fold cross-validation on a left-out portion of the training data. The *source code* to reproduce the results of the paper is available online at <https://github.com/rkwitt/TrajectoryTransfer>.

Transient attributes \mathcal{A}_T [11]		SUN attributes \mathcal{A}_S [18]
MSE	R^2	MAP
0.05 ± 0.01	0.28 ± 0.17	0.90 ± 0.04

Table 1: Performance of *regressing* transient attribute strength, evaluated on *TADB*, and attribute *detection*, evaluated on *SADB*, using images represented in \mathcal{X}_{fc7} .

4.1. Attribute regression / detection performance

Attribute regression. We start by assessing the first assumption of Section 3.1, *i.e.*, predicting a transient state of a scene from its image representation, in our case $\mathbf{x} \in \mathcal{X}_{fc7}$. We choose the following performance metrics: we report the mean-squared-error (MSE) and the R^2 score per attribute. The R^2 score is an informative measure, capturing how much variation in the dependent variable (*i.e.*, the attribute strength) is explained by the model. A R^2 score of 0 indicates no linear dependency, 1 indicates perfect linear dependency. All results are averaged over 10 random splits of *TADB* (split by scene location). In particular, the data for 61 webcams is selected for training, the data for the remaining 40 webcams is selected for testing. This is equivalent to the *Holdout* experiment reported in [11]. Table 1 lists the performance measures. Overall, the MSE of 0.05 is comparable to the results of [11]. From the R^2 statistic of 0.28, we see that there *is* a linear relationship, yet this relationship is not particularly strong. Nevertheless, the computational advantages of using a *linear* SVR (*e.g.*, training/testing in linear time) nicely balances the tradeoff between model choice and computational resources.

Attribute detection. Since we will experiment with the attribute-based image representation of [18] (*i.e.*, images represented in \mathcal{X}_S) in our one-shot recognition tests, we first need to assess the performance of detecting the presence/absence of the 102 SUN attributes \mathcal{A}_S . For this assessment, we use the images from *SADB* and the original splits provided by [18]. Table 1 reports the *mean average precision* (MAP). Note that the MAP of 0.90 is slightly higher than the 0.88 originally reported in [18], presumably due to our use of *Places-CNN* features. Most notably, the AP per attribute rarely drops below 0.8.

Trajectory estimation. Next, we address the second assumption of Section 3.1, *i.e.*, whether it is reasonable to assume linear dependency between the strength of a transient attribute and the entries of an image representation. In particular, we measure the quality of regressing the entries of representations in \mathcal{X}_{fc7} and \mathcal{X}_S , as a function of the attribute strength. Our evaluation measures are MSE and the R^2 score. Since, according to Section 3, we fit one trajectory per attribute and scene location, the MSE and R^2 are averaged over all models. From Table 2, we see that a linear model fits best for data in \mathcal{X}_{fc7} with a MSE of 0.03 and a

Image representation	#Dim.	R^2	MSE
Places-CNN (\mathcal{X}_{fc7})	200	0.14	0.03
SUN attributes (\mathcal{X}_S)	102	0.11	0.03

Table 2: Evaluation of a linear model for regressing image representations, evaluated on *TADB*, in \mathcal{X}_{fc7} and \mathcal{X}_S as a function of the transient attribute strengths.

R^2 score of 0.14. For representations in the SUN attribute space \mathcal{X}_S , performance is lower in both measures. We hypothesize that this result is due to the additional layer of indirection, *i.e.*, detecting attribute presence/absence from features in \mathcal{X}_{fc7} . Overall, the R^2 scores are relatively low, however, they do indicate some linear relationship. We argue that these results still warrant the use of a linear model, but obviously leave room for potential improvements.

4.2. One-shot recognition of scene locations

The objective of our one-shot recognition experiments is to distinguish scene locations from only a *single* image per location available for training. Due to the characteristics of one-shot learning, *i.e.*, a very limited amount of available data and therefore no variability in the transient states, trajectory transfer is a natural approach to this problem.

We run all recognition experiments on TADB. We use only this database, since it is specifically designed to have a high degree of variability in the transient attributes. This allows us to assess the quality of trajectory transfer on suitable evaluation data. Note that we deliberately do not use *SUN397*, or *Places* to demonstrate our approach, since scene variability (*cf.* Section 1, Fig. 3) is quite limited, certainly not with respect to scene configurations, but with respect to the range of transient attribute strengths.

Evaluation setup. We follow a standard cross-validation protocol. In detail, we repeatedly (for 10 times) split the number of scene locations into 61 training and 40 testing locations at random. The 61 locations serve as our *external* training corpus that is used for (1) training the attribute strength regressors for all attributes in \mathcal{A}_T , (2) estimating feature trajectories and (3) computing the weights for trajectory transfer. For one-shot recognition, we select one image at random from each of the 40 test locations. Random chance for this classification problem is at 2.5%. We stick to the following notation in Table 3: whenever the one-shot instances of the 40 scenes are used as training data, we denote this by (A). Experiments with data synthesized using trajectory transfer are indicated by (+S). Recognition rates are reported for the remaining (*i.e.*, after removal of the one-shot instances) images of the test split, averaged over the cross-validation runs. We synthesize data for *all* attributes in \mathcal{A}_T , using R different attribute strengths, linearly spaced in $[0, 1]$. This is done, since the hand-annotated attribute strengths in [11] are also normalized to that range.

Comparison(s). To establish *baselines* for one-shot recognition, we first train a linear SVC on the single instances of the 40 scene locations in the testing portion of each split and refer to this first baseline as SVC (A) . Our second baseline is obtained by adding data synthesized from *random* trajectories, indicated as SVC (A+Random) .

Regarding one-shot approaches from the literature, we compare against the approaches of Fink [7] and Tang *et al.* [24] which are both based on learning a suitable pseudo-metric from the external training data. In both cases, we learn a *linear transformation*. In particular, we implement [7] using LSML [14] for metric learning (instead of POLA). Tang *et al.* [24] adapt neighborhood component analysis (NCA) to the setting of one-shot learning and advocate the use of so called *micro-sets* during optimization of the transform. We closely follow the original setup of [24] by choosing 10 instances of every class for testing, one for training and SGD for optimization. Whenever we report results with metric learning, we add (+ML) and the reference to the specific approach, *e.g.*, SVC (A+ML) .

Additionally, we *combine* feature trajectory transfer with the aforementioned pseudo-metric learning approaches of [7] and [24]. As mentioned earlier, those approaches are not necessarily competitors, but rather provide complementary information. In fact, learning a suitable metric can be naturally combined with data synthesized by our approach: the learned transform is not only applied to the one-shot instances, but also on the synthetic instances. We indicate such combinations as (A+S+ML) when reporting results.

4.2.1 Results

Table 3 lists the results for our one-shot recognition experiments. The number of synthesized samples for trajectory transfer is set to $R = 5$. This leads to a total of $5 \cdot 40 = 200$ additional features per one-shot instance. Interestingly, we found that increasing the number of synthesized samples per attribute does not improve the recognition accuracy. In fact, $R \geq 2$ already produces stable results. One reason for this behavior could be that our approach *independently* synthesizes data for each attribute. This is certainly suboptimal and introduces redundancies. For example, sweeping “daylight” from $0 \rightarrow 1$ might have similar effects than sweeping “night” from $1 \rightarrow 0$. A second reason might be that linearly spacing $R = 2$ values over the range $[0, 1]$, by design, covers the extreme ends of the transient attribute strengths.

A *first* observation is that using all available data (*i.e.*, all images of the 40 locations in each test split) in a five-fold cross-validation setup, denoted by SVC (Ref, CV) in Table 3, leads to almost perfect recognition accuracy. The observed behavior is in line with our initial experiment in Fig. 1. The small difference in the numerical results can be explained by the fact that in Fig. 1 data from all 101 scene locations was used, whereas in Table 3 cross-validation was

	Image representation	
	\mathcal{X}_{fc7}	\mathcal{X}_S
SVC (A)	78.0 ± 4.5	61.2 ± 4.9
→ SVC (A+S)	80.7 ± 4.5	61.7 ± 5.0
SVC (A+Random)	77.7 ± 4.5	60.5 ± 5.1
SVC (A+ML [7])	88.2 ± 3.9	61.6 ± 5.2
→ SVC (A+S+ML [7])	89.2 ± 3.9	61.2 ± 5.2
SVC (A+ML [24])	80.6 ± 4.6	61.8 ± 4.5
→ SVC (A+S+ML [24])	84.1 ± 4.1	63.5 ± 4.4
SVC (Ref, CV)	99.5 ± 0.1	97.4 ± 0.4

Table 3: Accuracy ($\pm 1\sigma$, averaged over ten runs) of **one-shot recognition**; (1) using one-shot instances only (A), (2) using one-shot instances + **synthesized data** (A+S) and (3) using (A, or A+S) in combination with metric learning (+ML). Random chance is at $1/40 = 2.5\%$. SVC (Ref, CV) reports the accuracy that can be obtained using *all* available data. Results obtained with synthesized data are marked by ‘→’; **highlighted** cells indicate improvements from adding synthesized data. For comparison, SVC (A+Random) lists the results obtained via *random* trajectories.

performed on data from only 40 locations.

Second, we see that using data synthesized via trajectory transfer always improves recognition accuracy over *not* including synthetic data for image representations in \mathcal{X}_{fc7} . This effect is less pronounced for representations in \mathcal{X}_S .

Third, and most notably, the results indicate the complementary nature of synthetic data (obtained via trajectory transfer) in combination with pseudo-metric learning [7, 24]. While, relying on the learned transformations alone, *i.e.*, (A+ML), already leads to fairly good results, including synthetic data is beneficial with accuracy gains ranging from 1 to 3 percentage points. Overall, the top performance is obtained by SVC (A+S+ML [7]) with 89.2%. This is a considerable improvement over the baseline SVC (A) and reduces the gap between using all available data and only using the one-shot instances to ≈ 10 percentage points. Extracting the minimum (81.5%) and maximum (93.4%) accuracies obtained with SVC (A+S+ML [7]) over all splits also shows that performance depends on the specific instances that are used as one-shot examples. This is not surprising, since the transient states of these images essentially *anchor* the process of trajectory transfer.

Since, we cannot *directly* visualize the synthesized feature representations, Fig. 6 shows a selection of image retrieval results instead. Starting from our one-shot examples, we synthesized representations using trajectory transfer (for “sunny”) and used these to search for the closest (*i.e.*, nearest-neighbor) image in the external data. While, Fig. 6 only shows a small selection, the retrieved images appear to reflect the desired change in attribute strength. Combined with the results from Table 3, this indicates that trajectory transfer produces meaningful representations.

As an additional comparison, we note that a simplified[‡] automatic variant of Laffont *et al.*'s [11] appearance transfer (to create, for each desired attribute strength and $R = 5$, additional *images*) leads to an accuracy of 71.7 ± 4.7 . Notably, this result is below the accuracy of training with one-shot instances alone (SVC+A). While, we obtain reasonable synthetic images in many cases, transferring appearance *without* segment matching [11, §6.2] (not implemented in our setup) tends to introduce spurious image details in the final synthesis results, see Fig. 5; this eventually leads to features in \mathcal{X}_{fc7} that apparently confuse the SVM classifier. Nevertheless, exploring the *full* pipeline of [11] is a promising avenue for future work, regardless of the relatively high computational demand (compared to trajectory transfer).

Finally, we remark that improvements in recognition accuracy for image representations in \mathcal{X}_S are minor and the the gap between SVC (A) and SVC (Ref, CV) of ≈ 36 percentage points is substantial. In fact, we can only demonstrate marginal improvements of about 3 percentage points. This behavior is somewhat expected, since the additional layer of abstraction when predicting attributes adds another source of uncertainty when estimating feature trajectories. Table 2 also support this explanation. By using \mathcal{X}_{fc7} directly, we (1) avoid this indirection and (2) use image representations that already are highly discriminative.

5. Discussion

Throughout all experiments of this work, we have seen that changes in the appearance of a scene with respect to transient attributes affect the underlying image representation, *e.g.*, in \mathcal{X}_{fc7} . In the ideal case, features would be totally invariant with respect to the particular state of a transient attribute. Consequently, one-shot recognition would “reduce” to the problem of achieving good generalization with respect to the *configuration* of a particular scene category (*i.e.*, different variants of beaches, alleys, arches, etc.). However, as we have seen in Figs. 1 and 2, changes in transient states manifest as changes in the image representations and negatively impact scene recognition performance.

While the introduction of large scene recognition datasets, such as *Places*, has led to better generalization with respect to variability in scene configuration, the data collection process (*e.g.*, via image search engines) does not seem to achieve complete coverage with respect to transient attribute variability (cf. Section 3). However, the recently released dataset of [11] allows us to study and eventually model the functional dependency between transient attributes and feature representations. This enables the proposed *feature trajectory transfer* and allows us to increase variability via synthesized data. As demonstrated in our experiments on one-shot recognition of scene locations, the

[‡]We closely follow [11], but *do not* implement *segment matching* as in [11, §6.2] and *no* human is involved in the selection of “target” images.



Figure 5: Some qualitatively good (*top*) and bad (*bottom*) results of a simplified variant of Laffont *et al.*'s appearance transfer [11].

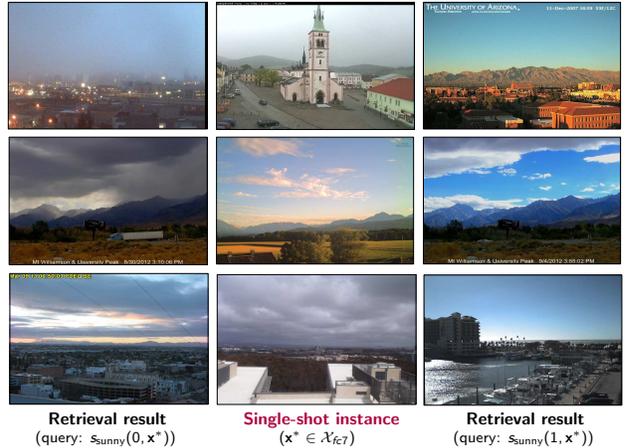


Figure 6: Exemplary retrieval results (*left/right*) for three single-shot images (*middle*), when the query is executed based on the feature representation, synthesized for a desired attribute strength (here: “sunny” of 0 and 1) via trajectory transfer.

resulting increased coverage of the feature space has a positive effect on recognition accuracy.

The proposed approach also raises several interesting issues. First, we remark that regressing feature dimensions independently only makes sense after decorrelation via PCA. However, this is done one attribute at a time and ignores potential relationships between the attributes (such as “sunny” & “daylight”). An alternative strategy could be, *e.g.*, to regress feature dimensions from combinations of related transient attributes. A second question is, how trajectory transfer performs as an augmentation technique beyond one-shot recognition scenarios. Finally, it will be interesting to (qualitatively) study the results of strategies, *e.g.*, Mahendran and Vedaldi [15], that allow for an *inversion* of feature representations. While this might be challenging for activations from the ‘fc7’ layer of *Places-CNN*, inversion results from features (produced via trajectory transfer) at earlier layers could potentially lead to greater insight into what is being captured by the proposed approach.

Acknowledgements. This work has been supported, in part, by the Austrian Science Fund (FWF KLI project 429) and the NSF grant ECCS-1148870.

References

- [1] E. Bart and S. Ullman. Cross-generalization: learning novel classes from a single example by feature replacement. In *CVPR*, 2005. 3, 5
- [2] J. Deng, W. Dong, R. Socher L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009. 1
- [3] M. Dixit, S. Chen, D. Gao, N. Rasiwasia, and N. Vasconcelos. Scene classification with semantic fisher vectors. In *CVPR*, 2015. 1, 2, 3
- [4] A. Farhadi, I. Endres, D. Hoiem, and D. Forsyth. Describing objects by their attributes. In *CVPR*, 2009. 2
- [5] L. Fei-Fei, R. Fergus, and P. Perona. One-shot learning for object categorization. *TPAMI*, 28(4):594–611, 2006. 3
- [6] L. Fei-Fei and P. Perona. A Bayesian hierarchical model for learning natural scene categories. In *CVPR*, 2005. 2
- [7] M. Fink. Object classification from a single example utilizing relevance metrics. In *NIPS*, 2005. 3, 4, 7
- [8] Y. Gong, L. Wang, R. Guo, and S. Lazebnik. Multi-scale orderless pooling of deep convolutional activation features. In *ECCV*, 2014. 1, 2
- [9] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012. 5
- [10] R. Kwitt, N. Vasconcelos, and N. Rasiwasia. Scene recognition on the semantic manifold. In *ECCV*, 2012. 2, 3
- [11] P.-Y. Laffont, R. Zhile, X. Tao, C. Qian, and J. Hays. Transient attributes for high-level understanding and editing of outdoor scenes. *ACM TOG*, 33(4), 2014. 1, 2, 5, 6, 8
- [12] C.H. Lampert, H. Nickisch, and S. Harmeling. Learning to detect unseen object classes by between-class attribute transfer. In *CVPR*, 2009. 2
- [13] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing scene categories. In *CVPR*, 2006. 2
- [14] E.Y. Liu, Z. Guo, X. Zhang, and V. Jojic W. Wang. Metric learning from relative comparisons by minimizing squared residual. In *ICDM*, 2012. 7
- [15] A. Mahendran and A. Vedaldi. Understanding deep image representations by inverting them. In *CVPR*, 2015. 8
- [16] T. Malisiewicz, A. Gupta, and A.A. Efros. Ensemble of exemplar-SVMs for object detection and beyond. In *ICCV*, 2011. 3
- [17] E.G. Miller, N.E. Matsakis, and P. Viola. Learning from one-example through shared density transforms. In *CVPR*, 2000. 3, 4
- [18] G. Patterson, C. Xu, H. Su, and J. Hayes. The SUN attribute database: Beyond categories for deeper scene understanding, 2014. 2, 5, 6
- [19] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *JMLR*, 12:2825–2830, 2011. 5
- [20] T. Pfister, J. Charles, and A. Zisserman. Domain-adaptive discriminative one-shot learning of gestures. In *ECCV*, 2014. 3
- [21] N. Rasiwasia and N. Vasconcelos. Scene classification with low-dimensional semantic spaces and weak supervision. In *CVPR*, 2008. 2
- [22] J. Sánchez, F. Perronnin, T. Mensink, and J. Verbeek. Image classification with the Fisher vector: Theory and practice. *IJCV*, 105(3):222–245, 2013. 2
- [23] Y. Su and F. Jurie. Improving image classification using semantic attributes. *IJCV*, 100(1):59–77, 2012. 2
- [24] K.D. Tang, M.F. Tappen, R. Sukthankar, and C.H. Lampert. Optimizing one-shot recognition with micro-set learning. In *CVPR*, 2010. 3, 4, 5, 7
- [25] A. Torralba and A.A. Efros. Unbiased look at dataset bias. In *CVPR*, 2011. 1
- [26] R. Salakhutdinov A. Torralba and J.B. Tenenbaum. Learning to learn with compound HD models. In *NIPS*, 2011. 3
- [27] J. Xiao, K.A. Ehinger, J. Hays, A. Torralba, and A. Oliva. SUN database: Exploring a large collection of scene categories. *IJCV*, pages 1–20, 2014. 2
- [28] J. Xiao, J. Hays, K. Ehinger, A. Oliva, and A. Torralba. SUN database: Large-scale scene recognition from abbey to zoo. In *CVPR*, 2010. 3
- [29] W. Yan, J. Yap, and G. Mori. Multi-task transfer methods to improve one-shot learning for multimedia events. In *BMVC*, 2015. 4
- [30] X. Yu and Y. Aloimonos. Attribute-based transfer learning for object categorization with zero/one training example. In *ECCV*, 2010. 3, 4
- [31] B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, and A. Oliva. Learning deep features for scene recognition using Places database. In *NIPS*, 2014. 1, 2, 3, 5