Semantically Consistent Regularization for Zero-Shot Recognition

Pedro Morgado Nuno Vasconcelos
Department of Electrical and Computer Engineering
University of California, San Diego
{pmaravil,nuno}@ucsd.edu

Abstract

The role of semantics in zero-shot learning is considered. The effectiveness of previous approaches is analyzed according to the form of supervision provided. While some learn semantics independently, others only supervise the semantic subspace explained by training classes. Thus, the former is able to constrain the whole space but lacks the ability to model semantic correlations. The latter addresses this issue but leaves part of the semantic space unsupervised. This complementarity is exploited in a new convolutional neural network (CNN) framework, which proposes the use of semantics as constraints for recognition.Although a CNN trained for classification has no transfer ability, this can be encouraged by learning an hidden semantic layer together with a semantic code for classification. Two forms of semantic constraints are then introduced. The first is a loss-based regularizer that introduces a generalization constraint on each semantic predictor. The second is a codeword regularizer that favors semantic-to-class mappings consistent with prior semantic knowledge while allowing these to be learned from data. Significant improvements over the state-of-the-art are achieved on several datasets.

1. Introduction

Significant advances in object recognition have been recently achieved with the introduction of deep convolutional neural networks (CNNs). The main limitation of this approach is the effort required to 1) collect and annotate millions of images necessary to train these models, and 2) the complexity of training a CNN from scratch. In fact, most recent computer vision papers use or adapt a small set of popular models, such as AlexNet [28], GoogLeNet [54], and VGG [51], learned from the Imagenet dataset [13]. Hence, there is an interest in techniques for transfer learning, where a model learned on a dataset is used to recognize object classes that are not represented in it. Ideally, transfer learning methods would replicate the human ability to recognize objects from a few example images or even from a description in terms of concepts in some semantic vocabulary.

This has motivated the introduction of semantic representations for object recognition [34, 44, 45, 55, 56], which rely on a predefined vocabulary of visual concepts to define a semantic space $\mathcal{S}$ and a set of classifiers to map each image into that space. The scores of these classifiers can then be used as semantic features for object classification. Furthermore, because simple rules of thumb can be designed, a priori, to describe new object classes in terms of these semantics, the image mapping into $\mathcal{S}$ can be exploited to recognize previously unseen objects. This is known as *zero-shot learning* (ZSL) [2, 4, 14, 31, 48, 49].

The fundamental difficulty of ZSL is that training cannot be guided by the end goal of the classifier. While the recognizer is learned from a set of *training* classes, it must provide accurate predictions for image classification into a non-overlapping set of *unseen* or *zero-shot* (ZS) classes. Historically, early efforts were devoted to the identification of good semantics for ZSL. This motivated the collection of datasets containing images annotated with respect to semantics such as *visual attributes* [14,31]. Subsequent works addressed the design of the semantic space $\mathcal{S}$, using one of two strategies previously proposed in the semantic representation literature. The first, *recognition using independent semantics* (RIS), consists of learning an independent classifier per semantic [34, 55, 56]. Due to its simplicity, RIS became widely popular in the attribute recognition literature [14,31,42,48,53,58]. Notwithstanding efforts in discriminant attribute discovery [9, 14, 30, 42, 46] or modeling of uncertainty [25,31,58], learning semantics independently proved too weak to guarantee reliable ZS predictions.

This motivated a shift to the second strategy, which ties the design of $\mathcal{S}$ to the goal of recognition, by learning a single multi-class classifier that optimally discriminates between all training classes [44, 45]. The difficulty of extending this approach to ZSL is that the semantics of interest

This work was funded by graduate fellowship SFRH/BD/109135/2015 from the Portuguese Ministry of Sciences and Education and NRI Grants IIS-1208522 and IIS-1637941 from the National Science Foundation.

are not the classes themselves. [2] proposed an effective solution to this problem by noting that there is a fixed linear transformation, or embedding, between the semantics of interest and the class labels, which can be specified by hand, even for ZS classes. This was accomplished using a label embedding function ϕ, to map each class y into a vector $\phi(y)$ in the space of attributes. Recently, various works have proposed variations on this approach [1,4,35,43,47,49]. We refer to this class of methods as *recognition using semantic embeddings* (RULE). By learning all semantics simultaneously, RULE is able to leverage dependencies between concepts, thus addressing the main limitation of RIS.

In this work, we investigate the advantages and disadvantages of the two approaches for implementations based on deep learning and CNNs. We show that, in this context, the two methods reduce to a set of *constraints* on the CNN architecture: RIS learns a bank of independent CNNs, and RULE uses a single CNN with fixed weights in the final layer. It follows that the performance of the two approaches is constrained by the form in which supervision is provided on the space $\mathcal{A}$ of image attributes. While RIS provides supervision along each dimension independently, RULE does so along the subspace spanned by the label embedding directions $\phi(y)$. Because the number of attributes is usually larger than classes, this exposes the strengths and weaknesses of the two approaches. On one hand, RIS supervises all attributes but cannot model their dependencies. On the other, RULE models dependencies but leaves a large number of dimensions of $\mathcal{A}$ unconstrained.

To exploit this complementarity, we propose a new framework denoted *Semantically COnsistent REgularization* (SCoRe) that leverages the advantages of *both* RIS and RULE. This is achieved by recognizing that the two methods exploit *semantics* as *constraints for recognition*. While RIS enforces first-order constraints (single semantics), RULE focuses second-order (linear combinations). However, both are suboptimal for ZSL. RIS ignores the recognition of training classes, sacrificing the modeling of semantic dependencies, and RULE ignores a large subspace of $\mathcal{A}$ and fixes network weights. SCoRe addresses these problems by exploiting the view of a CNN as an optimal classifier with respect to a multidimensional classification code, implemented at the top CNN layer. It interprets this code as a mapping between semantics (layer before last) and classes (last layer). It then enforces *both* first and second-order regularization constraints through a combination of 1) an RIS like *loss-based regularizer* that constraints semantic predictions, and 2) a *codeword regularizer* that favors classification codes consistent with RULE embeddings.

2. Previous Work

Semantics Semantics are visual descriptions that convey meaning about an image $\mathbf{x} \in \mathcal{X}$, and may include any measurable visual property: discrete or continuous, numerical or categorical. Given a semantic vocabulary $\mathcal{V} = \{v_1, \ldots, v_Q\}$, a semantic feature space $\mathcal{S}$ is defined as the Cartesian product of the vector spaces $\mathcal{S}_k$ associated with each semantic v_k, $\mathcal{S} = \mathcal{S}_1 \times \cdots \times \mathcal{S}_Q$. A classifier is denoted semantic if it operates on $\mathcal{S}$. As an example, for animal recognition, a semantic vocabulary containing visual attributes, e.g. $\mathcal{V} \in \{\textit{furry}, \textit{has legs}, \textit{is brown}, etc.\}$, is usually defined along with their corresponding vector spaces. In this case, since all semantics are binary, $\mathcal{S}_k = \mathbb{R}$ where large positive values indicate the attribute presence, and large negative values, its absence.

Early approaches to semantic recognition [45] used the set of image classes to be recognized as the semantic vocabulary. The rationale is to create a feature space with a high-level abstraction, where operations such as image search [44] or classification [34, 45] can be performed more robustly. More recently, there has been substantial interest in semantic feature spaces for transfer learning, which use an auxiliary semantic vocabulary, defined by mid-level visual concepts. Three main categories of concepts have been explored, including visual attributes, hierarchies and word vector representations. Attributes were introduced in [14, 31] and quickly adopted in many other works [2,8,21,23–25,27,48,53,58,60]. Semantic concepts extracted from hierarchies/taxonomies were later explored in [2,4,48,60], and vector representations for words/entities in [4,8,16,18,20,41,43,47,60].

Zero-shot learning Most current solutions to ZSL fall under two main categories: RIS and RULE. Early approaches adopted the RIS strategy. One of the most popular among these is the direct attribute prediction (DAP) method [31], which learns attributes *independently* using SVMs and infers ZS predictions by a maximum a posteriori rule that assumes attribute independence. Several enhancements have been proposed to account for attribute correlations *a posteriori,* e.g. by using CRFs to model attribute/class correlations [10], directed Bayesian networks to merge attribute predictions into class scores [58], or random forests learned so as to mitigate the effect of unreliable attributes [25]. More recently, [37] proposed a multiplicative framework that enables class-specific attribute classifiers, and [5] learns independent attributes which were previously discovered from Word2Vec representations.

RULE is an alternative strategy that exploits the one-to-one relationship between semantics and object classes. The central idea is to define an embedding $\phi(\cdot)$ that maps each class y into a Q-dimensional vector of attribute states $\phi(y)$ that identifies it. A bilinear compatibility function

$$h(\mathbf{x}, y; \mathbf{T}) = \phi(y)^T \mathbf{T}^T \theta(\mathbf{x}) \quad (1)$$

of parameters $\mathbf{T} \in \mathbb{R}^{d \times Q}$ is then defined between the feature vector $\theta(\mathbf{x}) \in \mathbb{R}^d$ of image $\mathbf{x}$ and the encoding of its

class y. In the first implementation of RULE for ZSL [2], $\mathbf{T}$ is learned by a variant of the structured SVM. Several variants have been proposed, such as the addition of different regularization terms [43, 49], the use of least-squares losses for faster training [49], or improved semantic representations of objects learned from multiple text sources [1, 47].

3. Semantics and deep learning

We now discuss the CNN implementation of RIS and RULE. For simplicity, we assume attribute semantics. Sections 5 and 6 extend the treatment to other concepts. For quick consultation, Table 1 summarizes important notation used in the rest of the paper.

3.1. Deep-RIS

Under the independence assumption that underlies RIS, the CNN implementation reduces to learning Q independent attribute predictors. Inspired by the success of multi-task learning, it is advantageous to share CNN parameters across attributes, and rely on a common feature extractor $\theta(\mathbf{x};\Theta)$ of parameters Θ, which can be implemented with one of the popular CNNs in the literature. Thus, each attribute predictor a_k of Deep-RIS takes the form

$$a_k(\mathbf{x};\mathbf{t}_k,\Theta) = \sigma\left(\mathbf{t}_k^T\theta(\mathbf{x};\Theta)\right) \tag{2}$$

where $\sigma(\cdot)$ is the sigmoid function and $\mathbf{t}_k$ a parameter vector. Given a training set $\mathcal{D} = \{(\mathbf{x}^{(i)},\mathbf{s}^{(i)})_{i=1}^N\}$, where $\mathbf{s}^{(i)} = (s_1^{(i)},\ldots,s_Q^{(i)})$ are attribute labels, $\mathbf{t}_k$ and Θ are learned by minimizing the risk

$$\mathcal{R}[a_1,\ldots,a_Q,\mathcal{D}] = \sum_i\sum_k L_b(a_k(\mathbf{x}^{(i)};\mathbf{t}_k,\Theta),s_k^{(i)}) \tag{3}$$

where L_b is a binary loss function, typically the cross-entropy loss $L_b(v,y) = -y\log(v) - (1-y)\log(1-v)$.

3.2. Deep-RULE

The implementation of RULE follows immediately from the bilinear form of (1). Note that $\phi(y)$ is a *fixed* mapping from the space of attributes to the space of class labels. For example, if there are Q binary attributes and C class labels, $\phi(y)$ is a Q dimensional vector that encodes the presence/absence of the Q attributes in class y

$$\phi_k(y) = \begin{cases} 1 & \text{if class } y \text{ contains attribute } k, \\ -1 & \text{if class } y \text{ lacks attribute } k. \end{cases} \tag{4}$$

We denote $\phi(y)$ the *semantic code of class* y. To implement (1) in a CNN, it suffices to use one of the popular models to compute $\theta(\mathbf{x};\Theta)$, add a fully-connected layer of Q units and parameters $\mathbf{T}$, so that $a(\mathbf{x}) = \mathbf{T}^T\theta(\mathbf{x};\Theta)$ is a vector of attribute scores, and define the CNN class outputs

$$h(\mathbf{x};\mathbf{T},\Theta) = \Phi^T a(\mathbf{x}) = \Phi^T\mathbf{T}^T\theta(\mathbf{x};\Theta), \tag{5}$$

Table 1. Notation.

Symbol	Meaning
Φ/Φ_{ZS}	Semantic codeword matrix for training/ZS classes
$\phi(y)$	Semantic codeword of class y (column of Φ)
$\phi_k(y)$	Semantic-state codewords ("building blocks" of $\phi(y)$)
$\mathbf{W}$	Classification codeword matrix (related to Φ through (11))
$\mathbf{w}_y$	Classification codewords (columns of $\mathbf{W}$)
$\mathcal{A}'$	Effective attribute space
$\mathcal{A}'_T$ / $\mathcal{A}'_{ZS}$	Subspace of $\mathcal{A}'$ spanned by the columns of Φ / Φ_{ZS}

where $\Phi = [\phi(1),\ldots,\phi(C)] \in \mathbb{R}^{Q\times C}$. Given a training set $\mathcal{D} = \{(\mathbf{x}^{(i)},y^{(i)})_{i=1}^N\}$, where $y^{(i)}$ is the class label of image $\mathbf{x}^{(i)}$, $\mathbf{T}$ and Θ are learned by minimizing

$$\mathcal{R}[h,\mathcal{D}] = \textstyle\sum_i L\left(h(\mathbf{x}^{(i)};\mathbf{T},\Theta),y^{(i)}\right) \tag{6}$$

where L is some classification loss, typically the cross-entropy $L(\mathbf{v},y) = -\log(\rho_y(\mathbf{v}))$ of softmax outputs $\rho(\mathbf{v})$.

3.3. Relationships

Both Deep-RIS and Deep-RULE have advantages and disadvantages, which can be observed by comparing the risks of (3) and (6). Since the attributes $a_k(\mathbf{x})$ are the quantities of interest for ZSL, it is useful to understand how the two methods provide supervision to the space $\mathcal{A}$ of attributes. From (3), Deep-RIS provides supervision to the individual attributes $a_k(\mathbf{x})$. Since $a_k(\cdot) = \mathbf{1}_k^T a(\cdot)$, where $\mathbf{1}_k$ is the k^{th} vector in the canonical basis (1 in the k^{th} position and 0 elsewhere), the supervision is along the *canonical directions of* $\mathcal{A}$. On the other hand, (5)-(6) only depend on the projections $\phi(y)^T a(\mathbf{x})$ of $a(\cdot)$ along the vector encodings $\phi(\cdot)$ of all training classes. Hence, RULE only provides supervision to the *the column space* $\mathcal{C}(\Phi)$ *of* Φ.

In practice, we are often on the regime of Figure 1, where the number of attributes Q is larger than the number of training classes C. It follows that $\mathcal{C}(\Phi)$ can be fairly low dimensional (dimension C) and the left null space $\mathcal{N}(\Phi^T)$ fairly high dimensional (dimension $Q-C$). Hence, while RIS constraints all attributes, RULE leaves $Q-C$ attribute dimensions unconstrained. In this case, ZS classes with semantic codes ϕ_{ZS} misaligned with $\mathcal{C}(\Phi)$ cannot be expected to be accurately predicted. In the limit, RULE is completely inadequate to discriminate ZS classes when ϕ_{ZS} is perpendicular to $\mathcal{C}(\Phi)$, such as $\phi_{ZS}(1)$ in Figure 1. This suggests the superiority of RIS over RULE. However, because RIS supervises attributes independently, it has no ability to learn attribute dependencies, e.g. that the attributes "has wings" and "lives in the water" have a strong negative correlation. These dependencies can be thought of as constraints that reduce the effective dimensionality of the attribute space. They imply that the attribute vectors $a(\mathbf{x})$ of natural images do not span $\mathcal{A}$, but only an *effective attribute subspace* $\mathcal{A}'$ of dimension $Q' < Q$. By learning only on $\mathcal{C}(\Phi) \subset \mathcal{A}'$, Deep-RULE provides supervision explicitly in this space. This suggests that Deep-RULE should outperform Deep-RIS.

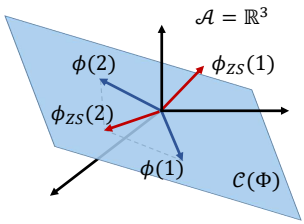

Figure 1. Attribute space ($Q = 3$). Semantic codes for two training classes shown in blue, and two ZS classes shown in red.

Overall, the relative performance of the two approaches depends on the overlap between the subspaces of $\mathcal{A}'$ covered by the training and ZS classes, denoted $\mathcal{A}'_T$ and $\mathcal{A}'_{ZS}$ respectively. If $\mathcal{A}'_T$ contains all the directions ϕ_{ZS} that define $\mathcal{A}'_{ZS}$, Deep-RULE will outperform Deep-RIS. If the ZS classes are defined by directions ϕ_{ZS} not contained in $\mathcal{A}'_T$, Deep-RIS will likely outperform Deep-RULE.

4. Semantically consistent regularization

In this section, we introduce the *Semantically COnsistent REgularizer* (SCoRe) architecture.

4.1. Attributes as regularization constraints

In the previous section, we saw that the relative performance of Deep-RIS and Deep-RULE depends on the alignment between the subspaces of $\mathcal{A}'$ that define the training and ZS classes, $\mathcal{A}'_T$ and $\mathcal{A}'_{ZS}$. In an ideal scenario, $\mathcal{A}'_T = \mathcal{A}'$ and so $\phi_{ZS}(c) \in \mathcal{A}'_T$ for any ZS class c. However, this is unlikely to happen for datasets of tractable size, and the subsets $\mathcal{A}'_T$ and $\mathcal{A}'_{ZS}$ are most likely not aligned.

Under this scenario, Deep-RIS and Deep-RULE compliment each other. While Deep-RIS enforces first-order constraints on the statistics of single attributes, Deep-RULE enforces second-order constraints, by constraining the statistics of linear attribute combinations. If the two strategies are combined, Deep-RULE can explain attribute dependencies that appear on both training and ZS classes, leaving to Deep-RIS the task of constraining the attribute distribution on the remainder of the space. It is, thus, natural to combine the two strategies. We accomplish this by mapping them into regularization constraints.

4.2. Recognition and regularization

An object recognizer maps an image $\mathbf{x}$ into a class

$$y^* = \underset{c\in\{1,\dots,C\}}{\arg\max}\ h_c(\mathbf{x}), \tag{7}$$

where $h(\mathbf{x}) = (h_1(\mathbf{x}), \dots, h_C(\mathbf{x}))$ is a vector of confidence scores for the assignment of $\mathbf{x}$ to each class, and y^* the class prediction. The score function $h(\cdot)$ is usually learned by minimizing an empirical risk $\mathcal{R}_E[h]$, under a complexity constraint $\Omega[h]$ to improve generalization, i.e.

$$h^* = \arg\min_h \mathcal{R}_E[h] + \lambda\Omega[h] \tag{8}$$

where $\lambda \geq 0$ is a Lagrange multiplier, and $\Omega[\cdot]$ a regularizer that favors score functions of low complexity. Common usages of $\Omega[\cdot]$ include shrinkage [22], sparse representations [11] or weight decay [29]. Since all these approaches simply favor solutions of low complexity, they are a form of *task-insensitive* regularization. For ZSL, this type of regularization has indeed been used to control the variance of 1) semantic scores or 2) backward projections of object embeddings into the feature space [49], as well as to suppress noisy semantics [43].

In this work, rather than a *generic* penalty on the complexity of $h(.)$, we propose a *task-sensitive* form of regularization, which favors score functions $h(\cdot)$ with the *added functionality* of attribute prediction. This regularization is implemented with two complimentary mechanisms, introduced in the next two sections.

4.3. Codeword regularization

The first mechanism exploits the fact that the score functions of (7) are always of the form

$$h_c(\mathbf{x}) = \langle \mathbf{w}_c, f(\mathbf{x}) \rangle\,, \tag{9}$$

where $\langle\cdot,\cdot\rangle$ denotes an inner product, $f(\cdot)$ a *predictor*, and $\{\mathbf{w}_1, \dots, \mathbf{w}_C\}$ a set of C class *codewords*. We denote $\mathbf{w}_c$ as the *classification code* of class c. For example, in binary classification, algorithms such as boosting [15] or SVM [12] simply choose $1/-1$ as the codewords of the positive/negative class.Similarly, for C-ary classification, neural networks [33] or multi-class SVMs [59] rely on one-hot encodings that lead to the typical decision rule $y^* = \arg\max_{j\in\{1,\dots,C\}} f_j(\mathbf{x})$. There is, however, no reason to be limited by these classical sets.

By comparing the score functions of (5) and (9), Deep-RULE can be interpreted as learning the optimal predictor $a(\mathbf{x})$ for a classification code given by (4), i.e. $\mathbf{w}_c = \phi(c)$. Hence, Deep-RULE can be seen as a form of very strict CNN regularization, where the final fully-connected layer is set to these semantic codes. In general, fixing network weights is undesirable, as better results are usually obtained by learning them from data. We avoid this by using the semantic codes $\phi(c)$ as loose regularization constraints, under the framework of (8). Similarly to Deep-RULE, we learn the predictor f using cross-entropy as the empirical risk $\mathcal{R}_E$, and score functions of the form

$$h(\mathbf{x}; \mathbf{W}, \mathbf{T}, \Theta) = \mathbf{W}^T f(\mathbf{x}) = \mathbf{W}^T\mathbf{T}^T\theta(\mathbf{x}; \Theta) \tag{10}$$

where the columns of $\mathbf{W}$ contain the weight vectors $\mathbf{w}_c$ of the last CNN layer. This is complemented by a *codeword regularizer*

$$\Omega[\mathbf{W}] = \tfrac{1}{2}\textstyle\sum_{c=1}^{C} ||\mathbf{w}_c - \phi(c)||^2 \tag{11}$$

that favors classification codes $\mathbf{w}_c$ similar to the semantic codes $\phi(c)$. Note that, up to terms that do not depend

on $\mathbf{w}_c$, this can be written as $\Omega[\mathbf{W}] \sim \sum_{c=1}^{C} \frac{1}{2}||\mathbf{w}_c||^2 - \sum_{c=1}^{C} \mathbf{w}_c^T \phi(c)$. In the Lagrangian of (8), the first summation becomes the "weight decay" regularizer already implemented by most CNN learning packages. Thus, effectively,

$$\Omega[\mathbf{W}] = -\sum_{c=1}^{C} \mathbf{w}_c^T \phi(c). \tag{12}$$

In sum, the use of codeword regularization forces the CNN to model attribute dependencies by aligning the learned classification codes $\mathbf{w}_c$ with semantic codes $\phi(c)$.

4.4. Loss-based regularization

The second mechanism, denoted *loss-based regularization*, aims to constraint attributes beyond $\mathcal{A}'_T$, and provides explicit regularization to attribute predictions. It is implemented by introducing an *auxiliary risk* $\mathcal{R}_A[f]$ in the optimization, i.e. replacing $\mathcal{R}_E[h]$ in (8) by $\mathcal{R}_E[h] + \lambda \mathcal{R}_A[f]$ where $\mathcal{R}_A[f]$ is the sum of attribute prediction risks of (3). This drives the score function to produce accurate attribute predictions, in addition to classification.

4.5. SCoRe

Given a training set of images $\mathbf{x}^{(i)}$, attribute labels $(s_1^{(i)}, \ldots, s_Q^{(i)})$, and class labels $y^{(i)}$, the regularizers of the previous sections are combined into the SCoRe objective

$$\begin{aligned} \underset{\Theta,\mathbf{T},\mathbf{W}}{\text{minimize}} \quad & \textstyle\sum_i L\left(h(\mathbf{x}^{(i)}; \mathbf{W}, \mathbf{T}, \Theta), y^{(i)}\right) \\ & + \lambda \textstyle\sum_i \sum_k L_b\left(f_k(\mathbf{x}^{(i)}; \mathbf{t}_k, \Theta), s_k^{(i)}\right) \\ & + \beta \Omega[\mathbf{W}], \end{aligned} \tag{13}$$

where $h(\cdot)$ is given by (10), $f_k(\mathbf{x}; \mathbf{t}_k, \Theta) = \mathbf{t}_k^T \theta(\mathbf{x}; \Theta)$ is the k^{th} semantic predictor, $\Omega[\mathbf{W}]$ the codeword regularizer of (11), and λ and β Lagrange multipliers that control the tightness of the regularization constraints.

Depending on the value of these multipliers, SCoRe can learn a standard CNN, Deep-RIS, or Deep-RULE. When $\lambda = \beta = 0$, all the regularization constraints are disregarded and the classifier is a standard recognizer for the training classes. Increasing λ and β improves its transfer ability. On one hand, regardless of β, increasing λ makes SCoRe more like Deep-RIS. In the limit of $\lambda \to \infty$, the first summation plays no role in the optimization, Ω is trivially minimized by $\mathbf{w}_c = \phi(c)$, and (13) is reduced to the Deep-RIS optimization problem of (3). On the other hand, maintaining $\lambda = 0$ while increasing β makes SCoRe similar Deep-RULE. For large values of β, the learning algorithm emphasizes the similarity between classification and semantic codes, trading off classification performance for semantic alignment. Finally, when both λ and β are non-zero, SCoRe learns the classifier that best satisfies the corresponding trade-off between the three goals: recognition, attribute predictions, and alignment with the semantic code.

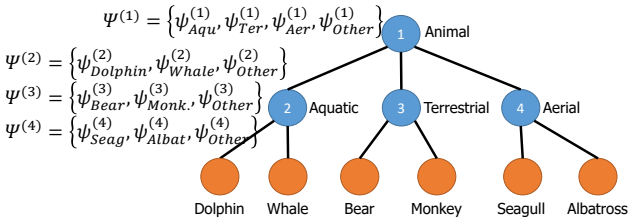

Figure 2. Semantic encoding for a taxonomy of six animal classes.

5. Semantics

In this section, we discuss the encoding of different semantics under the SCoRe framework.

5.1. Attributes

So far, we assumed that semantics are binary attributes. Each attribute is mapped into an entry of the semantic code according to (4), which is used to represent each class, i.e.

$$\phi(y) = concat(\phi_1(y), \ldots, \phi_Q(y)). \tag{14}$$

To support different degrees of certainty on class/attribute associations, continuous attributes are also easily implemented by making $\phi_k(y) \in [-1, 1]$.

5.2. Beyond binary semantics

SCoRe can be easily extended to semantics with more than two states. Consider a semantic k with S_k states. In this case, each state is itself represented by a codeword, i.e.

$$\phi_k(y) \in \Psi^{(k)} = \{\psi_1^{(k)}, \ldots, \psi_{S_k}^{(k)}\}, \tag{15}$$

where $\psi_i^{(k)}$ are *semantic state codewords*. Then, the semantic code $\phi(y)$ of class y is built by concatenating $\phi_k(y)$ for all k, as in (14). Similarly to the binary case, a predictor $f(\mathbf{x})$ learned under this codeword set will attempt to approximate $\phi_k(y)$ for images $\mathbf{x}$ of class y. The state of the k^{th} semantic can thus be recovered from f with $s_k^* = \arg\max_{i=1,\ldots,S_k} \langle \psi_i^{(k)}, f_k(\mathbf{x}) \rangle$ where $\psi_i^{(k)}$ is the codeword of state i of the k^{th} semantic, and $f_k(\cdot)$ the corresponding subspace of $f(\cdot)$. Many semantic state codewords can be defined. We now provide some examples.

Taxonomies In this work, we consider taxonomic encodings that emphasize node specific decisions, by interpreting each node as a semantic concept. As illustrated in Figure 2, a semantic state codeword set $\Psi^{(k)}$ is defined per node k. Its state codewords identify all possible children nodes plus a reject option. For example, the codeword set $\Psi^{(2)}$ of node 2 contains codewords $\psi^{(2)}_{dolphin}$ and $\psi^{(2)}_{whale}$, plus the reject codeword $\psi^{(2)}_{other}$. Under this taxonomic encoding, the semantic code $\phi(y)$ identifies the relevance of each node to

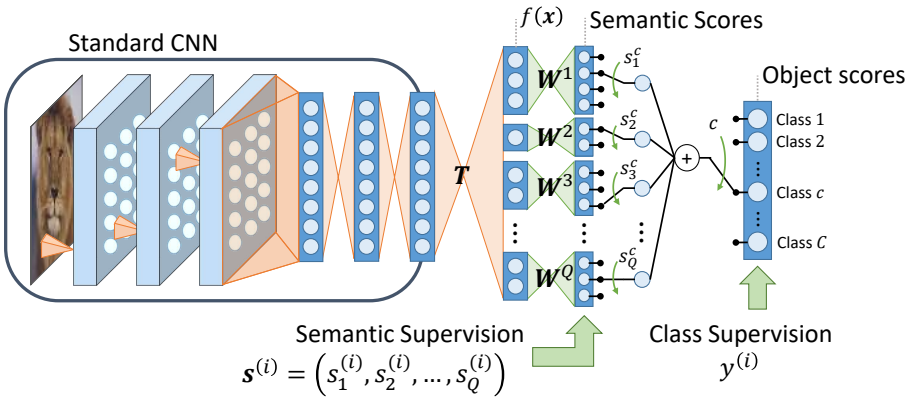

Figure 3. Deep-SCoRe. Feature extraction based on common CNN architectures. Classification is performed by first computing semantic scores through codewords $\mathbf{W}^k$, and then combining them into class scores using known class/semantics relations s_k^c.

the class y. An internal node that is an ancestor of y contributes with the codeword corresponding to the branch selection (needed to reach the class) at that node. A node that is not an ancestor contributes with the reject codeword. For example, in Figure 2, the class "bear" receives the code $\phi(bear) = concat\left(\psi_{Ter}^{(1)}, \psi_{Other}^{(2)}, \psi_{Bear}^{(3)}, \psi_{Other}^{(4)}\right)$.

It remains to define the codeword sets $\mathcal{V}^{(k)}$. These could be used to reflect further semantic information. In the tree of Figure 2, $\mathcal{V}^{(1)}$ could encode a set of attributes that distinguish aquatic, terrestrial, and aerial animals, such as "has fins," "has legs" or "has wings". In this work, since no semantic information is available beyond the taxonomy itself, we rely on the maximally separated codeword sets of [50]. Under this procedure, a Q-ways decision is mapped into the set of codewords defined as the vertices of a Q-sided regular polygon in $Q-1$ dimensions centered at the origin.

Word2Vec Word2Vec is a procedure to generate word embeddings. A word w is mapped into a high-dimensional vector $\xi(w) \in \chi$ by a neural network trained from large text corpora to reconstruct linguistic contexts of words. For semantic annotation, this mapping is used as the semantic code, i.e. each class y is encoded by the vector $\phi(y) = \xi(y)$.

In this work, we use the skip-gram architecture proposed by Mikolov *et al*. [39]. Its embeddings are determined by two parameters: size of the encoding layer and the window size that defines a context for each word. Rather than relying on a single model, we learn Q Word2Vec embeddings $\xi_k(y), k \in \{1, \ldots, Q\}$, using Q different combinations of the two parameters. This creates Q codeword sets $\mathcal{V}^{(k)}$. The semantic code then represents class c by a string of the resulting vectors $\phi_k(y) = \xi_k(y)$, using (14).

6. Deep-SCoRe

Deep-SCoRe implements (10) using a CNN to compute $\theta(\mathbf{x}; \Theta)$. Parameters Θ, $\mathbf{W}$ and $\mathbf{T}$ are learned from (13), using a semantic code that combines various semantic state codeword sets $\mathcal{V}^{(k)}$. These can be relative to attributes, taxonomy nodes, Word2Vec mappings, or any other semantic encoding. From (9), class scores decompose into

$$h_c(\mathbf{x}) = \sum_k h_c^{(k)}(\mathbf{x}) = \sum_k \langle \mathbf{w}_{s_k^c}^{(k)}, f_k(\mathbf{x}) \rangle \qquad (16)$$

where s_k^c is the state of the k^{th} semantic under class c, $\mathbf{w}_{s_k^c}^{(k)}$ the corresponding codeword, and $f_k(\cdot)$ the corresponding subspace of $f(\cdot)$. Semantic predictions are obtained by computing the dot-products

$$u_i^{(k)}(\mathbf{x}) = \langle \mathbf{w}_i^{(k)}, f_k(\mathbf{x}) \rangle \qquad (17)$$

for all states i of semantic k and choosing the state

$$s_k^* = \arg\max_i u_i^{(k)}(\mathbf{x}). \qquad (18)$$

While (16) and (17) could be computed separately, the structure of (16) allows shared computation. This can be accomplished by adding two layers to the semantic predictor $f(\mathbf{x})$, which we denote *semantic encoding* (SE) layers.

As shown in Figure 3, a CNN is used to compute the predictor $f(\mathbf{x}) = (f_1, \ldots, f_Q)(\mathbf{x})$. Similarly to Deep-RIS and Deep-RULE, this is implemented through a linear transformation $\mathbf{T}$ of a feature vector $\theta(\mathbf{x})$ computed with one of the popular CNN models. The first SE layer then consists of Q parallel fully-connected layers that compute the semantic scores $u_i^{(k)}(\mathbf{x})$ for each of the Q semantics. The weights of each branch k contain the classification codewords $\mathbf{w}_i^{(k)}$ and are learned under the codeword regularizer of (11). The second SE layer then selects, for each class c, a single output from each branch k corresponding to the state s_k^c of the k^{th} semantic of class c. These outputs are added to obtain the class recognition score $h_c(\mathbf{x})$. This is easily implemented by a fully connected layer of predetermined sparse weights of 0s and 1s that remain fixed throughout training.

Learning: Consider a training set of three-tuples: (a) the image $\mathbf{x}^{(i)}$; (b) the vector of semantic states $\mathbf{s}^{(i)}$; and (c)

the class label $y^{(i)}$. As shown in Figure 3, the state vectors $\mathbf{s}^{(i)}$ are used as supervisory signals for the first SE layer and the labels $y^{(i)}$ as supervisory signals for the second. These supervisory signals and the semantic codes $\phi(y)$ are used to compute the Lagrangian risk of (13), and all parameters are optimized by back-propagation using Caffe toolbox [26].

Deep-SCoRe models were trained by fine tuning pre-trained CNNs using stochastic gradient descent (SGD) with momentum of 0.9 and weight decay of 0.0005. The learning rate was chosen empirically for each experiment.

7. Experiments

In this section, we discuss several experiments carried out to evaluate the ZSL performance of Deep-SCoRe. Source code is available at https://github.com/pedro-morgado/score-zeroshot.

7.1. Experimental setup

Datasets: Three datasets were considered: Animals with Attributes [31] (AwA), Caltech-UCSD Birds 200-2011 [57] (CUB), and a subset of the Imaging FlowCytobot [52] (IFCB) dataset. Table 2 summarizes their statistics. On AwA and CUB, the partition into source and target classes for ZSL is as specified by [31] and [2], respectively. On IFCB, which is now first used for ZSL, classes were partitioned randomly. A separate set of validation classes (10/50/6 for the AwA/CUB/IFCB datasets, respectively) was also drawn randomly to tune SCoRe parameters.

Image representation: Images were resized to 256×256 pixels, with the exception of IFCB, where aspect ratios differ widely and resizing introduces considerable distortion. Instead, each image was first resized along the longest axis and the shortest axis then padded with the average pixel value, to preserve the aspect ratio. Typical data augmentation techniques were used for training (random cropping and mirroring), and the center crop was used for testing. Three CNN architectures were used to implement $\theta(\mathbf{x})$: AlexNet [28] (layer fc7), GoogLeNet [54] (layer pool5) and VGG19 [51] (layer fc7).

Semantics: Three sources of semantics were evaluated.

Visual attributes: Continuous attributes have been shown to be superior to their binary counterparts and were used on AwA and CUB. On IFCB, where no attributes were defined previously, a list of 35 visual attributes was assembled and annotated by an expert with binary labels, using several sources from the oceanographic community [7, 38].

Taxonomies were created by pruning the WordNet tree [40] for the training and ZS classes, and eliminating dummy nodes containing a single child. In the rare situations where WordNet was not fine-grained enough to distinguish between a set of classes, the taxonomy was expanded by simply assigning each object into its own leaf.

Table 2. Summary of dataset statistics.

Dataset	Images	Train/ZS Classes	Attributes	Hierarchy Source
AwA	30,475	40/10	85	WordNet [40]
CUB	11,788	150/50	312	WordNet [40]
IFCB	28,853	22/8	35	—

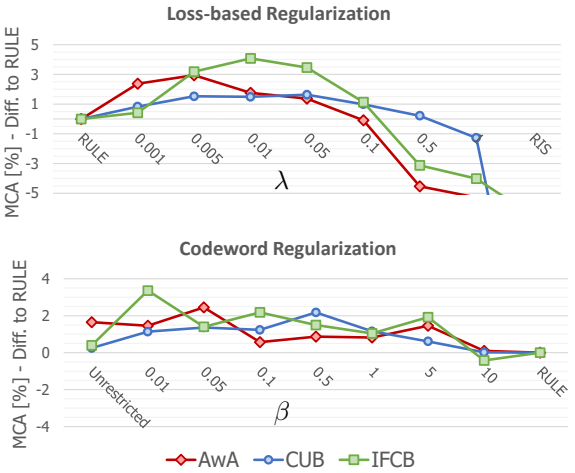

Figure 4. Influence of loss-based and codeword regularization on ZSL. Absolute improvement over RULE in ZS-MCA.

Word2Vec models were trained on a Wikipedia archive, dated June 1st, 2016. Three different window sizes (3, 5 and 10) and vector dimensions (50, 100 and 500) were used, leading to a total of 9 Word2Vec codeword sets.

7.2. Results

Gains of regularization: We started by evaluating codeword and loss-based regularization. The importance of the two regularizers was assessed separately on all datasets using visual attributes and GoogLeNet. In both cases, we measured the gains over Deep-RULE, in which classification codewords are set to $\mathbf{w}_c = \phi(c)$ and $\lambda = 0$. The gains of loss-based regularization were evaluated by increasing λ while keeping $\beta = 0$. Under this setting, the classifier converges to Deep-RIS in the limit of $\lambda \to \infty$. Conversely, the gains of codeword regularization were measured by increasing β while keeping $\lambda = 0$. In this case, the classifier converges to an unrestricted object recognizer when $\beta = 0$ and to Deep-RULE when $\beta \to \infty$. Figure 4 presents the absolute improvement in ZS mean class accuracy (ZS-MCA) over Deep-RULE, as a function of the Lagrange multipliers.

Both regularizers produced gains over Deep-RULE with absolute gains as high as 3 ZS-MCA points. This demonstrates the importance of learning the classification codewords, rather than fixing them. Note that, for codeword regularization, best results were obtained for intermediate values of β, which encourage consistency between the se-

Table 3. ZS-MCA[%] of various methods. A - AlexNet [28]; G - GoogLeNet [54]; V - VGG19 [51].

	AwA			CUB		
	A	G	V	A	G	V
DAP [32]	45.3†	59.5‡	-	16.9†	36.6‡	-
SJE [4]	61.9	66.7	-	40.3	50.1	-
ES-ZSL§ [49]	53.0	74.2	74.4	40.6	53.1	49.0
Huang *et al*. [23]	45.6	-	-	17.5	-	-
Liang *et al*. [37]	48.6	-	-	18.2	-	-
Changpinyo *et al*. [8]	-	72.9	-	-	54.7	-
Xian *et al*. [60]	-	72.5	-	-	45.6	-
Zhang *et al*. [61]	-	-	76.3	-	-	30.4
Gan *et al*. [21]	-	-	73.8	-	-	43.7
Deep-RIS	56.6	68.9	66.4	24.3	37.5	39.1
Deep-RULE	65.3	76.3	78.0	46.0	57.1	57.9
Deep-SCoRe	**66.7**	**78.3**	**82.8**	**48.5**	**58.4**	**59.5**

†As reported by Liang *et al*. [37]. ‡As reported by Al-Halah *et al*. [6]. §Self implementation.

mantic and classification codes, but leave enough flexibility to learn a classification code superior to its semantic counterpart. In all cases, the MCA of SCoRe was much superior to that of RIS, confirming the importance of modeling attribute dependencies through the first term of (13). Finally, SCoRe performance was also superior to that of the unrestricted CNN. This demonstrates the benefits of regularization. Interestingly, this was *not* the case of RIS, which always underperformed the unrestricted CNN, or RULE that only achieved on par results in CUB and IFCB[1].

In Section 3.3, we hypothesized that loss-based regularization becomes more important as the alignment between the subspaces of $\mathcal{A}'$ spanned by training and ZS classes decreases. To test this hypothesis, we measured this alignment by computing the average orthogonal distance between the semantic codeword $\phi(c)$ of each ZS class and the subspace spanned by the codewords of training classes. The average distances were 0.1244 for CUB, 0.3063 for AwA, and 0.4181 for IFCB, indicating that the transfer is easiest for CUB and hardest for IFC. This is consistent with the plots of Figure 4, which show largest gains of loss-based regularization on IFCB followed by AwA and then CUB.

Comparisons to state-of-the-art methods: A comparison to the literature is not trivial since methods differ in 1) CNN implementation, 2) train/ZS class partitioning, and 3) semantic space representation. To mitigate these differences, we focused on attribute semantics which have most available results. Methods that use alternative semantics [1, 19, 43, 47] or that use unlabeled images from ZS classes for training [17, 18, 27, 36] were disregarded for this comparison. Deep-SCoRe hyper-parameters λ and β were tuned on a subset of the training classes.

Table 3 compares our ZS-MCA to previous approaches

[1]The unrestricted CNN is initialized with semantic codes. If random initialization was used, ZSL would not be possible.

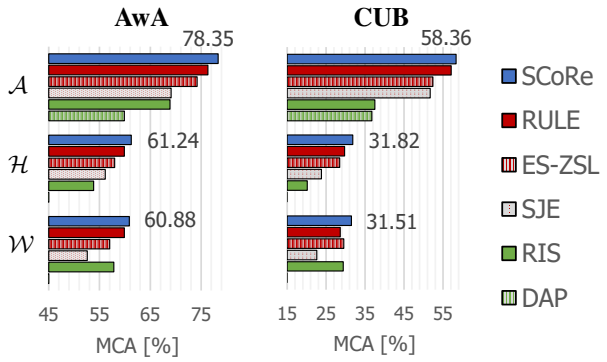

Figure 5. ZSL performance using different semantics. $\mathcal{A}$ – Attributes; $\mathcal{H}$ – Hierarchies; $\mathcal{W}$ – Word2Vec. DAP results reported in [5]. SJE and ES-ZSL self-implemented.

using three CNN architectures: AlexNet, GoogLeNet and VGG19. Although results vary drastically with CNN, it is clear that Deep-SCoRe outperforms all previous approaches on all datasets, achieving impressive gains over the state-of-the-art for every architecture: 4.8%, 4.1% and 6.5% on AwA and 7.9%, 3.7% and 10.5% on CUB with AlexNet, GoogLeNet and VGG19, respectively.

Multiple semantics: We finally studied the performance of Deep-SCoRe with attributes, taxonomies, and Word2Vec embeddings. Figure 5 compares Deep-SCoRe and its variants to popular RIS and RULE approaches in the literature: DAP [31] (RIS), SJE [4] and ES-ZSL [49] (RULE). All approaches were implemented with the semantic codes of Section 5. The best results, which were all obtained with Deep-SCoRe, are also shown. Figure 5 supports two main conclusions. First, as shown in [3, 8, 60], attributes enable by far the most effective transfer. This is not surprising since attributes tend to be discriminant properties of the various object classes. Taxonomies or Word2Vec are most informative of grouping or contextual information. Second, while all approaches rely on regularization, the nature of this regularization matters. The task-sensitive regularization of Deep-SCoRe always outperformed the task-insensitive regularization of ES-ZSL, and the combination of loss-based and codeword regularization (Deep-SCoRe) always outperformed a fixed semantic code (Deep-RULE and SJE) or loss-based regularization (Deep-RIS and DAP).

8. Conclusion

In this work, we analyzed the type of supervision provided by previous approaches. The complementarity found between class and semantic supervision lead to the introduction of a new ZSL procedure, denoted SCoRe, where a CNN is learned together with a semantic codeword set and two forms of semantic constraints: loss-based and codeword regularization. State-of-the-art zero-shot performance was achieved in various datasets.

References

[1] Z. Akata, M. Malinowski, M. Fritz, and B. Schiele. Multi-cue zero-shot learning with strong supervision. In *Computer Vision and Pattern Recognition (CVPR), IEEE Conf. on*, 2016.

[2] Z. Akata, F. Perronnin, Z. Harchaoui, and C. Schmid. Label-embedding for attribute-based classification. In *Computer Vision and Pattern Recognition (CVPR), IEEE Conf. on*, 2013.

[3] Z. Akata, F. Perronnin, Z. Harchaoui, and C. Schmid. Label-embedding for image classification. *Pattern Analysis and Machine Intelligence (TPAMI), IEEE Trans. on*, 38(7):1425–1438, 2016.

[4] Z. Akata, S. Reed, D. Walter, H. Lee, and B. Schiele. Evaluation of output embeddings for fine-grained image classification. In *Computer Vision and Pattern Recognition (CVPR), IEEE Conf. on*, 2015.

[5] Z. Al-Halah and R. Stiefelhagen. How to transfer? Zero-shot object recognition via hierarchical transfer of semantic attributes. In *Applications of Computer Vision, IEEE Winter Conf. on*, 2015.

[6] Z. Al-Halah, M. Tapaswi, and R. Stiefelhagen. Recovering the missing link: Predicting class-attribute associations for unsupervised zero-shot learning. In *Computer Vision and Pattern Recognition (CVPR), IEEE Conf. on*, 2016.

[7] D. Cassis. Phytopedia - the phytoplankton encyclpaedia project. Available at: http://www.eos.ubc.ca/research/phytoplankton/. Accessed: 2015-11-04.

[8] S. Changpinyo, W.-L. Chao, B. Gong, and F. Sha. Synthesized classifiers for zero-shot learning. In *Computer Vision and Pattern Recognition (CVPR), IEEE Conf. on*, 2016.

[9] C.-Y. Chen and K. Grauman. Inferring analogous attributes. In *Computer Vision and Pattern Recognition (CVPR), IEEE Conf. on*, 2014.

[10] H. Chen, A. Gallagher, and B. Girod. Describing clothing by semantic attributes. In *Computer Vision (ECCV), European Conf. on*, 2012.

[11] H. Cheng. *Sparse representation, modeling and learning in visual recognition*. Springer, 2015.

[12] C. Cortes and V. Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

[13] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition (CVPR), IEEE Conf. on*, 2009.

[14] A. Farhadi, I. Endres, D. Hoiem, and D. Forsyth. Describing objects by their attributes. In *Computer Vision and Pattern Recognition (CVPR), IEEE Conf. on*, 2009.

[15] Y. Freund and R. E. Schapire. A desicion-theoretic generalization of on-line learning and an application to boosting. In *Computational Learning Theory, European Conf. on*. Springer, 1995.

[16] A. Frome, G. S. Corrado, J. Shlens, S. Bengio, J. Dean, T. Mikolov, et al. Devise: A deep visual-semantic embedding model. In *Advances in Neural Information Processing Systems (NIPS)*, 2013.

[17] Y. Fu, T. M. Hospedales, T. Xiang, and S. Gong. Transductive multi-view zero-shot recognition and annotation. In *Computer Vision (ECCV), European Conf. on*, 2014.

[18] Y. Fu, T. M. Hospedales, T. Xiang, and S. Gong. Transductive multi-view zero-shot learning. *Pattern Analysis and Machine Intelligence (TPAMI), IEEE Trans. on*, 37(11):2332–2345, 2015.

[19] Y. Fu and L. Sigal. Semi-supervised vocabulary-informed learning. In *Computer Vision and Pattern Recognition (CVPR), IEEE Conf. on*, 2016.

[20] Z. Fu, T. Xiang, E. Kodirov, and S. Gong. Zero-shot object recognition by semantic manifold distance. In *Computer Vision and Pattern Recognition (CVPR), IEEE Conf. on*, 2015.

[21] C. Gan, T. Yang, and B. Gong. Learning attributes equals multi-source domain generalization. In *Computer Vision and Pattern Recognition (CVPR), IEEE Conf. on*, 2016.

[22] M. Gruber. *Improving Efficiency by Shrinkage: The James–Stein and Ridge Regression Estimators*, volume 156. CRC Press, 1998.

[23] S. Huang, M. Elhoseiny, A. Elgammal, and D. Yang. Learning hypergraph-regularized attribute predictors. *arXiv*, 2015.

[24] S. J. Hwang, F. Sha, and K. Grauman. Sharing features between objects and their attributes. In *Computer Vision and Pattern Recognition (CVPR), IEEE Conf. on*, pages 1761–1768, 2011.

[25] D. Jayaraman and K. Grauman. Zero-shot recognition with unreliable attributes. In *Advances in Neural Information Processing Systems (NIPS)*, 2014.

[26] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv:1408.5093*, 2014.

[27] E. Kodirov, T. Xiang, Z. Fu, and S. Gong. Unsupervised domain adaptation for zero-shot learning. In *Computer Vision and Pattern Recognition (CVPR), IEEE Conf. on*, 2015.

[28] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems (NIPS)*, 2012.

[29] A. Krogh and J. A. Hertz. A simple weight decay can improve generalization. In *Advances in Neural Information Processing Systems (NIPS)*, 1991.

[30] N. Kumar, A. C. Berg, P. N. Belhumeur, and S. K. Nayar. Attribute and simile classifiers for face verification. In *Computer Vision (ICCV), IEEE International Conf. on*, 2009.

[31] C. H. Lampert, H. Nickisch, and S. Harmeling. Learning to detect unseen object classes by between-class attribute transfer. In *Computer Vision and Pattern Recognition (CVPR), IEEE Conf. on*, 2009.

[32] C. H. Lampert, H. Nickisch, and S. Harmeling. Attribute-based classification for zero-shot visual object categorization. *Pattern Analysis and Machine Intelligence (TPAMI), IEEE Trans. on*, 36(3), 2013.

[33] Y. LeCun and Y. Bengio. Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks*, 3361(10), 1995.

[34] L.-J. Li, H. Su, L. Fei-Fei, and E. P. Xing. Object bank: A high-level image representation for scene classification & semantic feature sparsification. In *Advances in Neural Information Processing Systems (NIPS)*, 2010.

[35] X. Li and Y. Guo. Max-margin zero-shot learning for multi-class classification. In *Artificial Intelligence and Statistics (ICAIS), International Conf. on*, 2015.

[36] X. Li, Y. Guo, and D. Schuurmans. Semi-supervised zero-shot classification with label representation learning. In *Computer Vision (ICCV), IEEE International Conf. on*, 2015.

[37] K. Liang, H. Chang, S. Shan, and X. Chen. A unified multiplicative framework for attribute learning. In *Computer Vision (ICCV), IEEE International Conf. on*, 2015.

[38] J. Mees, G. Boxshall, M. Costello, et al. World Register of Marine Species (WoRMS). Available at: http://www.marinespecies.org. Accessed: Dec-2016.

[39] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems (NIPS)*, 2013.

[40] G. A. Miller. WordNet: A lexical database for English. *Communications of the ACM*, 38(11):39–41, 1995.

[41] M. Norouzi, T. Mikolov, S. Bengio, Y. Singer, J. Shlens, A. Frome, G. S. Corrado, and J. Dean. Zero-shot learning by convex combination of semantic embeddings. *arXiv:1312.5650*, 2013.

[42] D. Parikh and K. Grauman. Relative attributes. In *Computer Vision (ICCV), IEEE International Conf. on*, 2011.

[43] R. Qiao, L. Liu, C. Shen, and A. v. d. Hengel. Less is more: zero-shot learning from online textual documents with noise suppression. In *Computer Vision and Pattern Recognition (CVPR), IEEE Conf. on*, 2016.

[44] N. Rasiwasia, P. J. Moreno, and N. Vasconcelos. Bridging the gap: Query by semantic example. *Multimedia, IEEE Trans. on*, 9(5):923–938, 2007.

[45] N. Rasiwasia and N. Vasconcelos. Holistic context models for visual recognition. *Pattern Analysis and Machine Intelligence (TPAMI), IEEE Trans. on*, 34(5):902–917, 2012.

[46] M. Rastegari, A. Farhadi, and D. Forsyth. Attribute discovery via predictable discriminative binary codes. In *Computer Vision (ECCV), European Conf. on*, 2012.

[47] S. Reed, Z. Akata, B. Schiele, and H. Lee. Learning deep representations of fine-grained visual descriptions. In *Computer Vision and Pattern Recognition (CVPR), IEEE Conf. on*, 2016.

[48] M. Rohrbach, M. Stark, and B. Schiele. Evaluating knowledge transfer and zero-shot learning in a large-scale setting. In *Computer Vision and Pattern Recognition (CVPR), IEEE Conf. on*, 2011.

[49] B. Romera-Paredes and P. Torr. An embarrassingly simple approach to zero-shot learning. In *Machine Learning (ICCV), International Conf. on*, pages 2152–2161, 2015.

[50] M. J. Saberian and N. Vasconcelos. Multiclass boosting: Theory and algorithms. In *Advances in Neural Information Processing Systems (NIPS)*, 2011.

[51] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[52] H. M. Sosik and R. J. Olson. Automated taxonomic classification of phytoplankton sampled with imaging-in-flow cytometry. *Limnology and Oceanography: Methods*, 5(6):204–216, 2007.

[53] Y. Su, M. Allan, and F. Jurie. Improving object classification using semantic attributes. In *British Machine Vision Conference (BMVC)*, 2010.

[54] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. *arXiv preprint arXiv:1409.4842*, 2014.

[55] L. Torresani, M. Szummer, and A. Fitzgibbon. Efficient object category recognition using classemes. In *Computer Vision (ECCV), European Conf. on*, 2010.

[56] J. Vogel and B. Schiele. Semantic modeling of natural scenes for content-based image retrieval. *Computer Vision, International Journal of*, 72(2):133–157, 2007.

[57] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The Caltech-UCSD Birds-200-2011 Dataset. Technical Report CNS-TR-2011-001, California Institute of Technology, 2011.

[58] X. Wang and Q. Ji. A unified probabilistic approach modeling relationships between attributes and objects. In *Computer Vision (ICCV), IEEE International Conf. on*, 2013.

[59] J. Weston and C. Watkins. Multi-class support vector machines. Technical report, Citeseer, 1998.

[60] Y. Xian, Z. Akata, G. Sharma, Q. Nguyen, M. Hein, and B. Schiele. Latent embeddings for zero-shot classification. In *Computer Vision and Pattern Recognition (CVPR), IEEE Conf. on*, 2016.

[61] Z. Zhang and V. Saligrama. Zero-shot learning via semantic similarity embedding. In *Computer Vision and Pattern Recognition (CVPR), IEEE Conf. on*, 2015.