Enhancing Visual Embeddings through Weakly Supervised Captioning for Zero-Shot Learning

Matteo Bustreo
Istituto Italiano di Tecnologia
Via Enrico Melen 83, Genova, Italy
matteo.bustreo@iit.it

Jacopo Cavazza
Istituto Italiano di Tecnologia
Via Enrico Melen 83, Genova, Italy
jacopo.cavazza@iit.it

Vittorio Murino
Istituto Italiano di Tecnologia
Via Enrico Melen 83, Genova, Italy
vittorio.murino@iit.it

Abstract

Visual features designed for image classification have shown to be useful in zero-shot learning (ZSL) when generalizing towards classes not seen during training. In this paper, we argue that a more effective way of building visual features for ZSL is to extract them through captioning, in order not just to classify an image but, instead, to describe it. However, modern captioning models rely on a massive level of supervision, e.g. *up to 15 extended descriptions per instance provided by humans, which is simply not available for ZSL benchmarks. In the latter in fact, the available annotations inform about the presence/absence of attributes within a fixed list only. Worse, attributes are seldom annotated at the image level, but rather, at the class level only: because of this, the annotation cannot be visually grounded. In this paper, we deal with such a weakly supervised regime to train an end-to-end LSTM captioner, whose backbone CNN image encoder can provide better features for ZSL. Our enhancement of visual features, called "VisEn", is compatible with any generic ZSL method, without requiring changes in its pipeline (a part from adapting hyperparameters). Experimentally, VisEn is capable of sharply improving recognition performance on unseen classes, as we demonstrate thorough an ablation study which encompasses different ZSL approaches. Further, on the challenging fine-grained CUB dataset, VisEn improves by margin state-of-the-art methods, by using visual descriptors of one order of magnitude smaller.*

1. Introduction

Zero-shot learning (ZSL) is the problem of multi-class classification when no training data is available for some of the classes[1]. Being motivated by the well known "long tail distribution" [24], ZSL has recently attracted a vibrant interest inside the computer vision community (see [33] for a survey). In order to recognize *unseen* test classes, ZSL typically leverages auxiliary information, such as attributes, which is required to be both discriminative and shared with the *seen* training classes.

The seminal papers [16, 14] first solved zero-shot recognition (for image classification) by jointly predicting attributes. Recently, attributes prediction was replaced by *metric learning* [35, 27, 4, 6, 12, 11, 26, 20, 25, 13, 23] to implicitly infer the degree of compatibility between *semantic embeddings* - encoding attributes at the class level - and *visual embeddings* which encodes images.

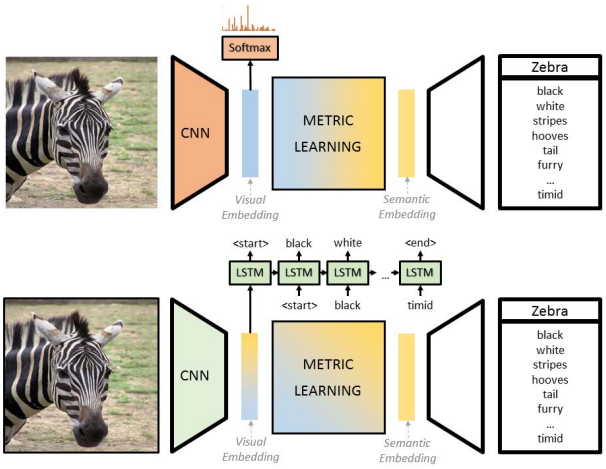

Figure 1. *Top*. In Zero-shot learning, class-related attributes are matched with visual features, the latter being usually extracted from a CNN trained for classification (represented in orange). *Bottom*. Differently, to get visual embeddings for ZSL, we exploit the CNN encoder (represented in green) of a CNN+LSTM captioner which predicts attributes at the image level. Our enhanced visual embeddings (*VisEn*) can replace default ones without modifications in the metric learning pipeline. Since containing more semantic patterns (see Figure 3), *VisEn* is capable of boosting ZSL recognition performance (see Section 4).

[1]Precisely, this is the case of *inductive* ZSL as opposed to (the easier) *transductive* ZSL case, in which un-annotated instances from the test classes are used for training.

Circumventing attributes prediction in ZSL is practically reasonable. In fact, to reliably predict attributes in the zero-shot setup, modern approaches exploit natural language processing and captioning techniques, both requiring a strong level of supervision. In fact, in *zero-shot captioning* [21, 5, 3, 31, 17, 30, 29], each training instance is annotated by humans with multiple detailed descriptions (*e.g.*, 15 each in [21]). Such wealth of annotations is unfortunately not available in ZSL benchmarks, where, instead, attributes are annotated at the instance level by registering the presence or absence of attributes within a pre-determined list (as in CUB [28]). Moreover, in some cases, attributes are annotated the class level only (as in AWA2 [15]) by providing a measure of coherence between each attribute and each class. This makes semantic embeddings **not** *visually grounded*: *e.g.*, we can expect a strong semantic coherence between the attribute "quadrapedal" and the class "zebra". But, crucially, such information can fool a ZSL model when recognizing an image of a zebra, only depicting its upper body, whose legs are not visible (as in Figure 1).

In order to tackle this problem, in this paper, we propose to enhance the semantic content of visual embeddings by extracting them from the CNN image encoder of a LSTM captioner which predicts attributes. To do so, we convert a captioner to operate in the weakly supervised regime which is common to ZSL benchmarks. That is, we do not take advantage of several natural sentences describing each instance as in [21, 5, 3, 31, 17, 30, 29]. Differently, we only rely on attributes annotated at either the image- or the class-level. In the latter case, we generate for free image-level supervision by leveraging the following observation: if an attribute is semantically incompatible with a given class, then, all instance of that class will not show that attributes as well. For instance, because zebras do not fly, we can bet on the fact that, within an (realistic) image of a zebra, wings won't be present. Hence, we train our captioner to predict which attributes are missing at the image level.

As opposed to default visual embeddings designed for classification, we posit that our captioner-based *enhancement* is capable of enriching visual features of semantic content. Consequently, ZSL is expected to be eased since our enhanced visual embeddings (termed *VisEn*) are designed to convey visually-grounded semantic cues, whereas default visual embeddings are not.

Through a broad experimental validation, we assess the capability of *VisEn* in capturing semantic patterns by evaluating attribute prediction both qualitatively and quantitatively. Further, through an ablation study, we show *VisEn* to be capable of 1) being superior to classically adopted visual embeddings (*i.e.*, GoogleNet or ResNet-101 features) and 2) boosting in performance existing ZSL methods.

In practical terms, *VisEn* is compatible with any generic technique in ZSL without requiring modifications in its pipeline (a part from hyper-parameters tuning). Also, a favorable performance is scored by *VisEn* when directly comparing to state-of-the-art methods on AWA2 and CUB databases.

In summary, the contributions of this paper are threefold.

- We claim that visual embeddings trained for classification are sub-optimal in zero-shot learning. Instead, we use a captioner, predicting attributes at the image-level, to enhance visual embeddings and allow them to capture visually-grounded semantic cues.
- With respect to zero-shot captioning [21, 5, 3, 31, 17, 30, 29], we train our captioner in a weaker supervised regime which is compatible with ZSL benchmarks. Even when attributes are not annotated at the image-level, we take advantage of attributes labelled as incompatible with a given class to deduct the visual absence of the same attribute in any image of that class, generating instance-level supervision for free.
- Our enhanced visual embeddings, called *VisEn*, can replace default ones in a plug-and-play fashion, without requiring any change in the computational pipeline (apart from hyper-parameters tuning). Further, *VisEn* is capable of improving the performance of existing methods, overall scoring a favorable performance against state-of-the-art methods on AWA2 [15] and CUB [28] datasets.

The rest of the paper is organized as follows: in Section 2, we present background material and related work, in Section 3 we detailed how we trained our captioner and present results for attributes prediction. In Section 4, we benchmark our proposed approach against the state-of-the-art. Conclusions will be drawn in Section 5.

2. Background and Related Work

In this Section, we will briefly refer to background material and related work to spot the factors of novelty of our paper with respect to existing works in the literature.

Metric Learning for Zero-Shot Learning. All inductive zero-shot methods by metric learning can be framed as follows. First, pre-computed visual features $\mathbf{v}$ are used to encode input data. Second, semantic embeddings $\mathbf{s}$ annotate the level of coherence in between a list of attributes and each seen/unseen class to recognize. Third, a metric function Φ is learnt. Usually called *compatibility function*, Φ is optimized in order to match $\mathbf{v}$ and $\mathbf{s}$ if they correspond to the same *seen* class. At the inference stage, $\Phi(\tilde{\mathbf{v}}, \mathbf{s}^{\mathrm{u}}_j)$ is computed on top of the test instance $\tilde{\mathbf{v}}$ and all semantic embeddings $\mathbf{s}^{\mathrm{u}}_j$, where j indexes unseen classes. Hence, the class $j^\star$ is predicted if $\Phi(\tilde{\mathbf{v}}, \mathbf{s}^{\mathrm{u}}_{j^\star})$ scores better than $\Phi(\tilde{\mathbf{v}}, \mathbf{s}^{\mathrm{u}}_j)$ for $j \neq j^\star$. In order to design the metric Φ for learning, different approaches have been attempted, by either considering bilinear functions [1, 9, 2, 32, 22], hidden embeddings

models [35, 27, 4, 6], dictionary learning [12, 11, 26] and eventually shallow linear networks to project visual onto semantic embeddings [20, 25, 13, 23].

Differently to ZSL by metric learning in which visual embeddings are pre-trained for classification, here we pre-trained them by using a captioner to predict attributes. As a result, our enhanced visual embeddings are expected to be richer in semantic patterns, easing the metric learning stage without any change in its computational pipeline (a part from hyper-parameters tuning).

Zero-shot captioning. Natural language processing (NLP) has been shown to successfully generate human-like captions for object and categories never seen before. It was successfully applied to both images [21, 5, 3, 31, 17] and videos [30, 29]. Far from providing a complete literature review on this topic, for our scope, it is sufficient to remind that the mainstream approaches leverage recurrent neural networks with long-short memory units (LSTM), due to their remarkable effectiveness in NLP. The LSTM is usually fed with some intermediate representation learnt from an encoder which process raw data in an end-to-end fashion (for images, CNN are usually adopted). Crucially, the common operative setup in zero-shot captioning is the annotation of each instance with many alternative extended description (*i.e.*, up to 15 sentences provided by annotators) [21, 5, 3, 31, 17, 30, 29].

Differently, in this paper, we train a captioner in the weaker supervised regime available in ZSL benchmarks: when available, we utilize the instance-level guidance about the presence/absence of attributes within a fixed list. Even when attributes are not visually grounded (since annotated at the class level only), we are still capable of training our captioner despite this extremely weakly supervised regime. We do so by using the semantic incompatibility of an attribute and a class to get for free the visual absence of the same attributes in all instances of that class. In doing so, we can find an example of **self-supervision** [8, 19, 10]. In fact, we exploit an auxiliary task (here, captioning) which 1) is solved with no need of additional supervision and 2) is preparatory for the original task of interest (here, ZSL).

3. Weakly Supervised Captioner for ZSL

In this Section, we present how we trained a captioner to predict attributes at the instance level in the weakly supervised setup of ZSL benchmarks, relaxing the strict supervised regime which is commonly adopted in zero-shot captioning methods [21, 5, 3, 31, 17, 30, 29].

In Section 3.1, we describe the datasets we used, while implementation details are available in Section 3.2. In Section 3.3, we present results in attributes' prediction.

3.1. Datasets

Caltech-UCSD Birds 200 (CUB) [28]. It is a fine-grained dataset of 200 bird species, most of which are typical in North America. In this paper, coherently with the ZSL literature, we adopted the 2011 release in which 11788 images are available. For each, up to 5 Amazon Mechanical Turkers annotated a list of 312 attributes which provides an expert level characterization of each image, specifying minute details (such as colored spots on the neck/wings) which are fundamental to disambiguate between classes, *i.e.*, bird species.

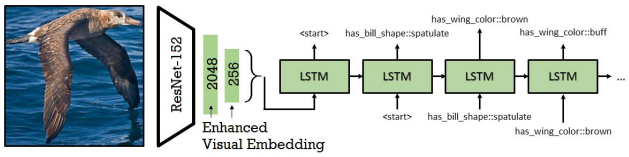

Figure 2. Architecture of our weakly supervised captioner.

Animals with Attributes 2 (AWA2) [15]. It is a coarse dataset composed of 50 different classes of animals (such as "polar bear", "zebra", "giraffe", "otter", etc. ...). To describe each class, a list of 85 attributes is provided by following Osherson's matrix (OM) [14] for class/attributes correspondence. Some values in the OM are not specified (and set to -1, like "black-colored" for the class "antelope"). The remaining entries of the OM contain a value scaled in the range [0,100], to rank how much an attribute is prototypical for that class (*e.g.*, the attribute "spots" for the class "dalmatian" has a value of 100 in the OM).

3.2. Implementation Details

As similarly done in [21, 5, 3, 31, 17], we employed an end-to-end trainable captioner which is composed by two modules: the first encodes image information in a feature vector, the second generates the caption. As first module, here we used a ResNet-152 Convolutional Neural Network, pre-trained on the Imagenet ILSVRC2012: the 2048-dimensional fully connected (FC) layer - right before logits in ResNet-152 architecture - is linearly transformed into a 256-dimensional FC layer. This latter encoding is then used as the initial state for our uni-directional LSTM captioner (256-dimensional hidden state), which is our second module. A visualization of the adopted architecture is provided in Figure 2.

Since CUB dataset provides attributes annotated at the image level, we directly used those annotations for training by considering one attribute to be present if at least one Turker annotated it. For instance, "has_bill_shape::spatulate", "has_wing_color::brown", "has_wing_color::grey", "has_wing_color::buff", "has_upperparts_color::brown" and "has_upperparts_color::buff" are some of the ground truth attributes for the image `Black_Footed_Albatross_0089_796069.jpg` which is depicted in Figure 2.

On the contrary, on AWA2, attributes are only annotated at the class level and, consequently, we do not have the guidance about which attributes are effectively present in which images. In fact, we are only given a confidence value for each attributes (like "quadrupedal") and each class (like "zebra), but this score is not capable of telling which images of the AWA2 dataset depict a zebra with visible legs.
Differently, it is advantageous to consider attributes which are semantically incoherent with a certain class (such as "black-colored" for the class "polar bear"). Then, we can assume that those incoherent attributes will be visually absent in all instances of that specific class (since none realistic photo will depict a black polar bear). This consideration is crucial to cast semantic attributes provided at the class level into (absence of) visual attributes annotate at the image level. Attributes that are annotated as incoherent with respect to a certain class[2] are labelled as missing for each instance of that class. On the contrary, for the remaining attributes, since we can not draw a better conclusion from the available annotations, we assume them to be always present in the corresponding instances. This is the weak supervision which negatively relies on the absence of attributes only and that we can generate for free from AWA2 benchmark for the sake of training our captioner.

On both CUB and AWA2, consistently to the ZSL setup, training is done only on the images belonging to the seen classes, accordingly to the proposed splits of [34].

3.3. Attribute Prediction: Results

In this Section, we validate the performance of our weakly supervised captioner for attribute prediction, in both quantitative and qualitative terms.

Quantative Results. We adopted the same binary classification framework of earlier ZSL models [14, 1]. That is, on CUB database, attributes by Turkers are used as ground truth and compared with instance-level predictions. The final reported performance is averaged across all 312 attributes and all training/testing images. Differently, on AWA2, we compare the binary predictions of our captioner (on top of a certain image I) with a binarization of the class-level attributes related to I. As in CUB, classification performance are averaged across the 85 attributes and all instances (in both training and testing).

Table 1 reports such classification accuracy values: our captioner is able to sharply improve in performance both [14] (+22.12% on CUB and +8.96% on AWA2) and [1] (+27.52% on CUB and +8.96% on AWA2). The sharper margin is registered on CUB dataset: in fact, on AWA2, we do not have a precise instance-level attributes supervision to train our captioner, whereas on CUB we do.

Qualitative Results. To visualize the image features learnt from the CNN module of our captioner, we take advantage of t-SNE [18].

[2]On average, 28.6% of attributes per class on AWA2.

	Training Set	Test Set		
	Captioner	Captioner	[14]	[1]
CUB	*87.26%*	**86.92%**	64.8%	59.4%
AWA2	*98.98%*	**81.66%**	72.7%	72.7%

Table 1. Attribute prediction in CUB and AWA2 datasets. The performance of the captioner on the training set is in italic, we highlighted in bold the best testing accuracy in attribute prediction among the captioner and the ZSL paradigms [14] and [1].

t-SNE is the state-of-the-art technique to obtain a bi-dimensional visualization of an arbitrary feature encodings. We run t-SNE on top of the 256-dimensional feature vector extracted from the CNN-module of our captioner trained on CUB database: the result is a set of bi-dimensional points $(x_i, y_i) \in \mathbb{R}^2$, each of which corresponding to the image I_i of CUB, $i = 1, \dots, 11788$. Then, we used (x_i, y_i) as anchor points where to plot I_i: in this way, we can embed all CUB images into a planar representation such that two nearby images correspond to features that are close to each other in the visual space (according to t-SNE).

To do so, we quantized (x_i, y_i) into a grid of integers point $(r_i, c_i) \in \mathbb{Z}^2$, the latter being used to align the first pixel of I_i while plotting it. Images I_i have been spatially rescaled to 50×50, preserving the original RGB color space. In order to handle overlap between images, we operated a stretching along the c_i-th coordinate of all our anchors.

The results of this visualization are provided in Figure 3, where we compare against the analogous procedure applied to the usual visual embeddings adopted for ZSL [34]: 2048-dimensional features extracted from a ResNet-101 model trained for classification over the seen classes.

ResNet-101 seems to encode similarly birds which have a similar shape, but different colors (and, therefore, different species - Figure 3 (b), orange box). In addition, when using the same descriptor, sometimes, birds appears to be clustered together accordingly to the sky on the background (Figure 3 (b), blue box).
Differently, using the 256-dimensional embedding of our captioner, we can better capture semantic patterns: in fact, we can observe a nice clustering effect of birds with green wings (Figure 3 (a), red box). Also, our enhanced visual embeddings seem to cluster birds with similar shape (long tail and neck, elongated body) *accordingly* to their colorization and the respective class as well (Figure 3 (a), green box).

In summary, even if using a visual embedding that is one order of magnitude smaller than a baseline one (ResNet-101), we are capable of capturing more semantic patterns.

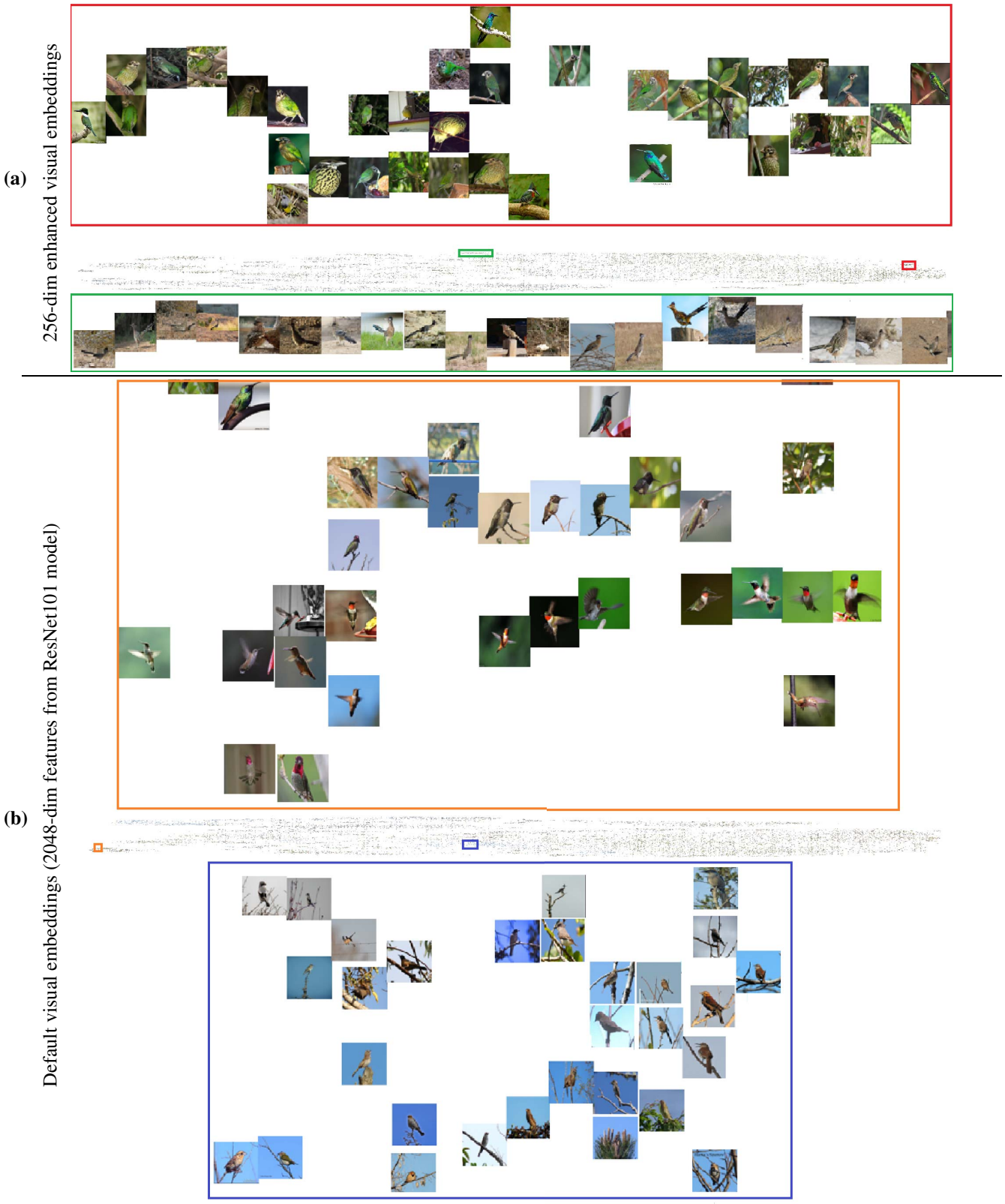

Figure 3. t-SNE visualizations of enhanced versus default semantic embedding for ZSL. **(a)**. Our enhanced visual embeddings finely learn the attribute "having-green-wings" (red box) and cluster birds with "long beack"+"long neck"+"long tail" in a color-consistent manner, so preserving species. **(b)**. Classical visual embeddings used in ZSL seems fooled by "contours", mixing up different species which very different colors (orange box). Sometimes, the sky in the background compromises a correct embedding for species (blue box).

4. Enhancing Visual Embeddings: Benchmarking the State-of-the-Art in ZSL

In this Section, we provide quantitative results to assess the effectiveness of our proposed enhancement of visual embeddings (*VisEn*). Precisely, on AWA2 and CUB benchmarks, we prove that three popular approaches in zero-shot learning [6, 13, 11] are sharply boosted in performance when replacing classical visual embeddings with ours. A part from hyper-parameters tuning, such replacement does not require changes in ther computational pipeline. Further, in Section 4.2, we setup a broad comparison between *VisEn* and the state-of-the-art performance in ZSL.

4.1. Ablation study

For our ablation, we considered the following approaches for ZSL:

• *Synthesized Classifiers* (SynC) [6] learns a latent embedding in which to combine visual and semantic embeddings in a max margin sense, by means of three different hinge losses: one-versus-one (OVO), Crammer and Singer (CS) [7] and a structured output SVM loss (struct).

• *Semantic Auto-Encoder* (SAE) [13] proposes a shallow linear encoder-decoder network to project visual embeddings into semantic ones (through a trainable projection matrix $\mathbf{W}$) and then reconstruct the visual embeddings from the semantic ones (through $\mathbf{W}^\top$). By using either the projection learnt from the encoder or the decoder, the compatibility function is the Frobenius norm between ground truth and predictions.

• *Coupled Dictionary Learning* (CDL) [11] learns a latent embedding in which semantic embeddings are projected and, by means of a synchronous dictionary learning pipeline, visual embeddings are mapped onto the latent ones and vice-versa. All such projections can be combined (or even separately used) at the inference stage which is configured as an Euclidean nearest neighbours search.

For SynC, SAE and CDL, we optimized from scratches their compatibility functions by using publicly available code[3]. We used default semantic embeddings (specifically, the ones provided in [34]) and we compare between different visual embeddings. For *VisEn*, we either alternatively used the 2048- and 256-dimensional representation learnt from our captioner and represented in Figure 2. As baseline, we adopted ResNet-101 (provided by [34]). Also for a fair comparison with Sync and SAE, we also considered the 1024-dim GoogleNet features which were originally used from those methods [6, 13].

In addition, we ablate on several factors: the OVO, CS and struct losses for SynC, all possible combinations of projections that learnt by CDL and the alternative usage of the encoder or the decoder for SAE. Moreover, for the latter method, since it is crucial aspect in ZSL [34], we tried different manners of splitting seen and unseen classes: a random extraction of 40 classes for training and 10 for testing (as commonly done in literature) in addition to standard (SS) and proposed splits (PS) from [34].

[3] SynC: https://github.com/pujols/zero-shot-learning, SAE: https://github.com/Elyorcv/SAE CDL: http://vipl.ict.ac.cn/resources/codes/code/ECCV2018_CDL_code_release.rar

Dataset: AWA2				
		SVM loss		
Visual Embedding	d	OVO	CS	struct
GoogleNet	1024	52.6%	53.4%	**59.0%**
ResNet-101	2048	53.0%	53.7%	**59.0%**
VisEn (ours)	*256*	*50.7%*	*51.0%*	*52.1%*
VisEn (ours)	*2048*	***54.6%***	***54.4%***	***59.0%***
Dataset: CUB				
		SVM loss		
Visual Embedding	d	OVO	CS	struct
GoogleNet	1024	53.4%	51.6%	54.5%
ResNet-101	2048	55.6%	49.0%	53.9%
VisEn (ours)	*256*	***59.4%***	***53.2%***	***54.6%***

Table 2. **Comparison with SynC** [6]. The performance of our visual enhancement (*VisEn*) is in italic, the best performance is in bold. In this table, we used the proposed split (PS) by [34].

Results are reported in Table 2 (for Sync), Table 3 (for SAE) and Table 4 (for CDL). In all of them, d denotes the size of the adopted visual embedding.

Discussion. In Table 2, *VisEn* exactly matches the performance of GoogleNet and ResNet-101 features on AWA2 with the OVO loss, while in all other cases is superior to both descriptors: the mean average improvement is 1.7% and 1.8% over them, respectively. In Table 3, *VisEn* improves SAE on the PS for AWA2 and in all splits for CUB. Finally, with respect to CDL , while accounting for all different projections setup (the different rows in Table 4), the 256-dimensional *VisEn* improves ResNet-101 features in 6 cases out of 7 (AWA2) and in 7 cases out of 7 on CUB.

More in details, on the AWA2 benchmark, despite the captioner was trained by only using a negative supervision about the absence of attributes, *VisEn* was frequently able to match the performance of descriptors which trained (for classification) with full supervision. When using SAE Encoder and Decoder (with 40/10 split), GoogleNet features are +1.8% and +2.7 better than 2048-dim *VisEn*, respectively. In turn, in the very same setup, 2048-dim *VisEn* improves ResNet-101 features by +1.3% and by +3.3% on the PS splits, when usind the SAE Decoder and SAE Encoder, respectively. Often, the 2048-dimensional *VisEn* are slightly superior to both GoogleNet and ResNet (SynC-OVO, SynC-CS, SAE Encoder SS, SAE Encoer PS, SAE Decoder PS, and CDL settings) guaranteeing a improvement of about one/two percentage points. For both SynC

	Visual Embedding	d	Dataset: AWA2 40/10	AWA2 SS [34]	AWA2 PS [34]	Dataset: CUB 40/10	CUB SS [34]	CUB PS [34]
Encoder	GoogleNet	1024	**84.7%**	78.5%	63.5%	61.4%	44.4%	46.2%
	ResNet-101	2048	79.6%	80.0%	64.0%	57.0%	54.4%	57.9%
	VisEn (ours)	*256*	*77.3%*	*77.3%*	*57.3%*	***62.5%***	***65.4%***	***58.6%***
	VisEn (ours)	*2048*	*82.9%*	***80.3%***	***65.7%***	-	-	-
Decoder	GoogleNet	1024	**84.0%**	**80.1%**	63.1%	60.9%	44.2%	46.2%
	ResNet-101	2048	79.0%	79.0%	63.4%	57.5%	54.8%	58.6%
	VisEn (ours)	*256*	*77.4%*	*77.4%*	*57.8%*	***63.2%***	***66.2%***	***59.2%***
	VisEn (ours)	*2048*	*80.3%*	*78.3%*	***64.1%***	-	-	-

Table 3. **Comparison with SAE** [13]. The performance of our visual enhancement (*VisEn*) is in italic, the best performance is in bold. In this case, we report the separate performance of the encoder and the decoder. Also, we ablate on several splits of seen and unseen classes: we consider the standard (SS) and proposed splits (PS) provided by [34] and the same 40/10 split (10 random classes as unseen ones, the remaining as seen ones) used in [13].

	Dataset: AWA2 ResNet-101 $d = 2048$	AWA2 *VisEn (ours)* $d = 2048$	AWA2 *VisEn (ours)* $d = 256$	Dataset: CUB ResNet-101 $d = 2048$	CUB *VisEn (ours)* $d = 256$
v	**63.8%** $\pm$ 4.3%	*59.6% $\pm$ 6.8%*	*62.0% $\pm$ 2.9%*	40.0% $\pm$ 2.3%	***57.6%*** *$\pm$ 1.5%*
a	61.4% $\pm$ 1.9%	*62.6% $\pm$ 2.1%*	***63.1%*** *$\pm$ 1.2%*	50.2% $\pm$ 1.8%	***51.5%*** *$\pm$ 0.8%*
l	53.9% $\pm$ 2.7%	*51.4% $\pm$ 3.2%*	***57.4%*** *$\pm$ 1.0%*	40.3% $\pm$ 2.2%	***50.5%*** *$\pm$ 2.7%*
v + a	**66.8%** $\pm$ 2.7%	*65.3% $\pm$ 2.9%*	*63.9% $\pm$ 1.6%*	54.6% $\pm$ 1.9%	***58.3%*** *$\pm$ 0.9%*
a+l	59.1% $\pm$ 1.7%	*55.6% $\pm$ 2.4%*	***61.4%*** *$\pm$ 0.7%*	46.3% $\pm$ 2.0%	***51.9%*** *$\pm$ 1.2%*
v+l	62.5% $\pm$ 1.7%	***66.5%*** *$\pm$ 3.0%*	*62.0% $\pm$ 0.7%*	49.5% $\pm$ 2.0%	***57.1%*** *$\pm$ 1.1%*
v+a+l	62.6% $\pm$ 1.5%	*59.5% $\pm$ 2.4%*	***62.7%*** *$\pm$ 0.8%*	50.7% $\pm$ 1.9%	***56.5%*** *$\pm$ 0.9%*

Table 4. **Comparison with CDL** [11]. The performance of our visual enhancement (*VisEn*) is in italic, the best performance is in bold. In this table, we adopted the proposed splits (PS) by [34] and compare different visual embeddings (whose *d*ensionality d is reported beneath for completeness). Also, we ablate on which combination of the three projection (visual embedding v, attributes a or latent embedding l) is used for the nearest neighbor search during inference. Since CDL leverages an iterated optimization, we provide mean and standard deviation of testing accuracy across iterations (whose number was fixed to 50 for AWA2 and 100 for CUB, as in [11]).

and SAE, 256-dimensional *VisEn* seems sub-optimal but, interestingly, the very same descriptor is able to score a remarkable performance in conjunction with CDL: despite one order of magnitude less, it is capable of improving ResNet-101 and GoogleNet features on the a, l, a+l and v+a+l settings, even by 3.5%.

Further, on AWA2, we can observe a that for both Sync and SAE, 2048-dimensional *VisEn* are always better than 256-dimensional ones, whereas, for CDL, the opposite trend is registered. This seems to suggest that CDL (which adopts dictionary learning) is capable of optimally perform even when fed with a relatively low-dimensional visual embedding. Differently, by either performing a latent embedding (Sync) or a direct mapping in between visual and semantic embeddings (SAE), a bigger dimensional visual embedding is required.

On CUB, the performance is undoubtedly coherent in its trend: 256-dim *VisEn* is *always* superior to GoogleNet and ResNet-101 features. In Table 2, while averaging across OVO, CS and struct losses, the average improvement of *VisEn* is +2.9% with respect to ResNet-101 features and +2.6% with respect to GoogleNet ones. In Table 3, across the 40/10, SS and PS splits and the usage of the Encoder or Decoder, SAE is improved by +12.0% with respect to GoogleNet features and by +5.9% with respect to ResNet-101 features. Finally, in Table 4, across all combinations of projections with CDL, ResNet-101 features are improved by +7.4% on average. All such systematic improvement on CUB are even more impressive if considering that they were achieved with a visual embedding of one order of magnitude less than baseline ones.

4.2. Comparison with the State-of-the-Art in Inductive ZSL by Metric Learning

In this Section, we directly compare the proposed enhancement of visual embeddings (*VisEn*) with state-of-the-art approaches in inductive zero-shot learning via metric learning.

Precisely, we compare *VisEn* - fed into a SAE encoder [13] - with direct and indirect attributes' prediction (DAP and IAP) [14], the cross-modal transfer (CMT) [25], the hybrid semantic and visual embedding (DEVISE) proposed in

	Dataset: AWA2		Dataset: CUB	
	SS	PS	SS	PS
DAP [14]	58.7	46.1	37.5	40.0
IAP [14]	46.9	35.9	27.1	24.0
CMT [25]	66.3	37.9	37.3	34.6
DEVISE [9]	68.6	59.7	53.2	52.0
ConSE [20]	67.8	44.5	36.7	34.3
SSE [35]	67.5	61.0	43.7	43.9
SJE [2]	69.5	61.9	55.3	53.9
ALE [1]	80.3	62.5	53.2	54.9
ESZSL [22]	75.6	58.6	55.1	53.9
LatEM [32]	68.7	55.8	49.4	49.3
SynC [6]	75.4	59.7	53.0	54.6
SAE [13]	**80.7**	54.1	33.4	33.3
expZSL [27]	79.3	63.8	53.0	49.3
LDA [12]	-	56.6	-	-
CDL [11]	-	**69.9**	-	54.5
VdSA [23]	-	-	56.7	-
PSR [4]	-	63.8	-	56.8
VisEn (ours)	*80.3*	*65.7*	***65.4***	***58.6***

Table 5. **Comparison with the state-of-the-art in inductive ZSL by metric learning**. First and second best accuracy values are highlighted in bold and underlined, respectively.

[9], the convex combination of semantic classifiers (ConSE) [20], the semantic similarity-preserving embedding (SSE) of [35], the Structured Joint Embedding (SJE)[2], label embedding for zero-shot image classification (ALE) [1], the regularized least square method for ZSL (ESZSL) [22], the latent embedding model which solves ZSL through ranking (LATEM) [32], the synthesized classifiers learnt in a max margin sense (SynC) [6]. Among the most recent ones, we considered the shallow semantic autoencoder (SAE) [13], the approach of ZSL which uses exponential family distributions (expZSL) [27], the discriminative learning of latent attributes (LDA) [12], the coupled dictionary learning approach (CDL) [11], the visually-driven semantic augmentation [23] and the metric learning approach which preserves semantic relations (PSR) [4].

Results of this extended comparison are reported in Table 5. For AWA2 and CUB, we utilized the standard (SS) and proposed splits (PS) of [34]. By doing so, we can leverage the work of the survey [33] which reported the performance of DAP, IAP, CMT, DEVISE, ConSE, SSE, SJE, ALE, ESZSL, LatEM, Sync, SAE and expZSL for those splits while using ResNet-101 features as visual embeddings. Also for LDA, CDL, VdSA and PSR, we are reporting published classification accuracy values extracted from the respective publications.

Discussion. Overall, our proposed approach scores a solid performance, so that *VisEn* locates itself as the second-best scoring method on AWA2 and the best one on CUB.

On AWA2, *VisEn* sets a second-best performance on both SS and PS. The performance is still notable due to the following aspect. Existing state-of-the-art methods extract their visual embeddings in a fully supervised regime, whereas, differently, our captioner was forced to operate in a more challenging setting in which only a negative weak supervision was provided (absence of attributes). Despite this make the comparison more demanding for us, still, our proposed enhancement of visual embedding scores a favorable performance with respect to the state-of-the-art.

Instead, on CUB, due to the availability of visually grounded attributes, *VisEn* can express its full potential and proves the effectiveness of doing metric learning by combining the coherence score of an attribute for a class (semantic embedding) and the visually-grounded predicted presence of an attribute inside an image of that class (*VisEn*). In fact, on CUB, our proposed enhancement registers an improvement over the previous best scoring method of +2.2% (on PS) and +9.7% (on SS).

5. Conclusions

In this paper, we propose to replace usual visual embeddings in ZSL (which are trained for fully supervised classification) with the intermediate representation of an end-to-end captioner which predicts attributes at the instance-level. Without the usage of the usual supervision adopted in zero-shot captioning (multiple extended descriptions per instance), we still proved the effectiveness of training a captioner in the weakly supervised regime which is typical of zero-shot recognition (list of attributes). In fact, even when attributes are annotated at the class level, we can rely on the semantic incompatibility between a given attribute and a certain class: all instances of that class do not contain the specific attribute. Leveraging this observation, we generate for free visually-grounded supervision to train our captioner, using it to extract visual embeddings which are richer in semantic content with respect to baseline ones (ResNet-101 features). Our proposed enhancement of visual embeddings *VisEn* is compatible with any generic ZSL method, without requiring changes in its pipeline (a part from hyper-parameter tuning). We proved that *VisEn* systematically improves the recognition performance of three popular approaches in ZSL [6, 13, 11], eventually outperforming classical GoogleNet and ResNet-101 features. Experimentally, *VisEn* achieves the second-best performance on AWA2 benchmark despite *VisEn* were obtained by means of negative weak supervision. Differently, on CUB, leveraging clean instance-level annotations, we sharply boosted the best scoring methods on CUB by +2.2% and +9.7%, on both standard/proposed splits of [34].

References

[1] Z. Akata, F. Perronnin, Z. Harchaoui, and C. Schmid. Label-embedding for image classification. *transactions on Pattern Analysis and Machine Intelligence (tPAMI)*, 38(7):1425–1438, 2016. 2, 4, 8

[2] 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, 2015. 2, 8

[3] P. Anderson, B. Fernando, M. Johnson, and S. Gould. Guided open vocabulary image captioning with constrained beam search. In *Empirical Methods in Natural Language Processing (EMNLP)*, 2016. 2, 3

[4] Y. Annadani and S. Biswas. Preserving semantic relations for zero-shot learning. In *Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2018. 1, 3, 8

[5] L. Anne Hendricks, S. Venugopalan, M. Rohrbach, R. Mooney, K. Saenko, and T. Darrell. Deep compositional captioning: Describing novel object categories without training data. In *Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2016. 2, 3

[6] S. Changpinyo, W.-L. Chao, B. Gong, and F. Sha. Synthesized classifiers for zero-shot learning. In *Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2016. 1, 3, 6, 8

[7] K. Crammer and Y. Singer. On the algorithmic implementation of multiclass kernel-based vector machines. *Journal of Machine Learning Research (JMLR)*, 2(Dec):265–292, 2001. 6

[8] C. Doersch, A. Gupta, and A. A. Efros. Unsupervised visual representation learning by context prediction. In *International Conference on Computer Vision (ICCV)*. IEEE, 2015. 3

[9] 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. 2, 8

[10] S. Jenni and P. Favaro. Self-supervised feature learning by learning to spot artifacts. In *Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2018. 3

[11] H. Jiang, R. Wang, S. Shan, and X. Chen. Learning class prototypes via structure alignment for zero-shot recognition. In *European Conference on Computer Vision (ECCV)*. Springer, 2018. 1, 3, 6, 7, 8

[12] H. Jiang, R. Wang, S. Shan, Y. Yang, and X. Chen. Learning discriminative latent attributes for zero-shot classification. In *International Conference on Computer Vision (ICCV)*. IEEE, 2017. 1, 3, 8

[13] E. Kodirov, T. Xiang, and S. Gong. Semantic autoencoder for zero-shot learning. In *Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2017. 1, 3, 6, 7, 8

[14] 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, 2009. 1, 3, 4, 7, 8

[15] C. H. Lampert, H. Nickisch, and S. Harmeling. Attribute-based classification for zero-shot visual object categorization. *transactions on Pattern Analysis and Machine Intelligence (tPAMI)*, 36(3):453–465, 2014. 2, 3

[16] H. Larochelle, D. Erhan, and Y. Bengio. Zero-data learning of new tasks. In *Conference on Artificial Intelligence (AAAI)*. AAAI, 2008. 1

[17] J. Lu, J. Yang, D. Batra, and D. Parikh. Neural baby talk. In *Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2018. 2, 3

[18] L. v. d. Maaten and G. Hinton. Visualizing data using t-sne. *Journal of machine learning research (JMLR)*, 9(Nov):2579–2605, 2008. 4

[19] M. Noroozi, H. Pirsiavash, and P. Favaro. Representation learning by learning to count. In *International Conference on Computer Vision (ICCV)*. IEEE, 2017. 3

[20] 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. In *International Conference on Learning Representation (ICLR)*, 2014. 1, 3, 8

[21] S. Reed, Z. Akata, H. Lee, and B. Schiele. Learning deep representations of fine-grained visual descriptions. In *Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2016. 2, 3

[22] B. Romera-Paredes and P. Torr. An embarrassingly simple approach to zero-shot learning. In *International Conference on Machine Learning (ICML)*, 2015. 2, 8

[23] A. Roy, J. Cavazza, and V. Murino. Visually-driven semantic augmentation for zero-shot learning. In *British Machine Vision Conference (BMVC)*. BMVA, 2018. 1, 3, 8

[24] R. Salakhutdinov, A. Torralba, and J. Tenenbaum. Learning to share visual appearance for multiclass object detection. In *Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2011. 1

[25] R. Socher, M. Ganjoo, C. D. Manning, and A. Ng. Zero-shot learning through cross-modal transfer. In *Advances in Neural Information Processing Systems (NIPS)*, 2013. 1, 3, 7, 8

[26] J. Song, C. Shen, J. Lei, A.-X. Zeng, K. Ou, D. Tao, and M. Song. Selective zero-shot classification with augmented attributes. In *European Conference on Computer Vision (ECCV)*. Springer, 2018. 1, 3

[27] V. K. Verma and P. Rai. A simple exponential family framework for zero-shot learning. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases (ECML/PKDD)*. Springer, 2017. 1, 3, 8

[28] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The Caltech-UCSD Birds-200-2011 Dataset. Technical report, 2011. 2, 3

[29] B. Wang, L. Ma, W. Zhang, and W. Liu. Reconstruction network for video captioning. In *Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2018. 2, 3

[30] X. Wang, J. Wu, D. Zhang, Y. Su, and W. Y. Wang. Learning to compose topic-aware mixture of experts for zero-shot video captioning. 2018. 2, 3

[31] Y. Wu, L. Zhu, L. Jiang, and Y. Yang. Decoupled novel object captioner. In *2018 ACM Multimedia Conference on Multimedia Conference*. ACM, 2018. 2, 3

[32] 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, 2016. 2, 8

[33] Y. Xian, C. H. Lampert, B. Schiele, and Z. Akata. Zero-shot learning-a comprehensive evaluation of the good, the bad and the ugly. *transactions on Pattern Analysis and Machine Intelligence (tPAMI)*, 2018. 1, 8

[34] Y. Xian, B. Schiele, and Z. Akata. Zero-shot learning-the good, the bad and the ugly. In *Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2017. 4, 6, 7, 8

[35] Z. Zhang and V. Saligrama. Zero-shot learning via semantic similarity embedding. In *International Conference on Computer Vision (ICCV)*. IEEE, 2015. 1, 3, 8